

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»
Профиль подготовки: «Программное обеспечение средств вычислительной
техники и автоматизированных систем»

БАКАЛАВРСКАЯ РАБОТА

на тему

СОЗДАНИЕ САЙТА ОРГАНИЗАЦИИ С
ПРИМЕНЕНИЕМ
ГЕОИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Студент Широкова Ирина Дмитриевна
(подпись, дата) (ФИО полностью)

Руководитель Леонова Т. Ю.
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр Бычков А.С.
(подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 22.05.2026 г. № 14)

Заведующий кафедрой Митрохин М.А.
(подпись) (фамилия, инициалы)

Работа защищена с отметкой хорошо (протокол заседания ГЭК от
08 июня 2026 № 12)

Секретарь ГЭК Гурин Е.И.
(подпись) (фамилия, инициалы)

Пенза 2026

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
«07» апреля 2026 г.

**ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ**

1. Студент Широкова Ирина Дмитриевна гр. 22ВВП2 факультета вычислительной техники

2. Тема работы: «Создание сайта организации с применением геоинформационных технологий»

Тема утверждена приказом ПГУ № 347/о от "07" апреля 2026 г.

3. Руководитель бакалаврской работы к.т.н., доцент кафедры ВТ ПГУ Леонова Т. Ю.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Разработать веб-приложение для студии дизайна интерьеров с целью автоматизации процессов взаимодействия с клиентами, интерактивной визуализации портфолио и предварительного расчета стоимости услуг. Серверную часть реализовать на платформе ASP.NET Core с использованием языка C#. Для хранения информации применить реляционную СУБД MySQL. Для картографической функциональности задействовать библиотеку Leaflet и внешние геосервисы Nominatim и OSRM. Система должна обеспечивать интерактивное взаимодействие пользователя с картой, мгновенный пересчет стоимости в калькуляторе при изменении параметров, возможность управления проектами через защищенную панель администратора.

5. Перечень подлежащих разработке вопросов

Требуется провести анализ предметной области и изучить существующие решения в сфере веб-сайтов студий дизайна интерьеров. Обосновать выбор инструментов разработки и геоинформационных сервисов. Спроектировать клиент-серверную архитектуру приложения и разработать структуру реляционной базы данных. Спроектировать клиентскую часть, включающую интерактивную карту, калькулятор стоимости, галерею проектов и форму обратной связи. Разработать серверную часть, реализующую REST API, аутентификацию и CRUD-операции. Создать панель администратора. Разработать пользовательский интерфейс. Выполнить тестирование.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Изучение предметной области	45	07.04.26 – 13.04.26	<i>heel</i>
2	Анализ темы, аналогов, постановка задачи	24	13.04.26 – 17.04.26	<i>heel</i>
3	Выбор и настройка инструментов разработки	50	17.04.26 – 24.04.26	<i>heel</i>
4	Проектирование архитектуры приложения и структуры базы данных	30	24.04.26 – 28.04.26	<i>heel</i>
5	Разработка клиентской и серверной частей, панели администратора, интеграция геосервисов, тестирование	151	28.04.26 – 17.05.26	<i>heel</i>
6	Оформление пояснительной записки	24	17.05.26 – 21.05.26	<i>heel</i>

Дата выдачи задания "07" апреля 2026 г.

Руководитель бакалаврской работы *heel* 07.04.26 Леонова Т. Ю.
(подпись, дата)

Задание к исполнению принял студент *Широкова* 07.04.26 Широкова И.Д.
(подпись, дата)

Работу к защите допустить
Декан факультета *Л.Р.* 02.06.2025 Фионова Л.Р.
(подпись, дата)

Реферат

Бакалаврская работа содержит 189 листов, 33 рисунков, 3 таблицы, 8 источников, 4 приложения.

ВЕБ-ПРИЛОЖЕНИЕ, ДИЗАЙН ИНТЕРЬЕРОВ, REACT, JAVASCRIPT, ASP.NET CORE, C#, MYSQL, БАЗА ДАННЫХ, REST API, JWT, ИНТЕРАКТИВНАЯ КАРТА, LEAFLET, OSRM, АДМИНИСТРИРОВАНИЕ

Цель работы – разработать веб-сайт для студии дизайна интерьеров, обеспечивающий визуализацию портфолио с географической привязкой, автоматизированный расчет стоимости услуг и сбор заявок от клиентов.

Объектами разработки являются:

- клиентская часть веб-приложения, реализованная на JavaScript с использованием библиотеки React, включающая интерактивную карту проектов на основе Leaflet, галерею работ, калькулятор стоимости с интеграцией Nominatim и OSRM, а также форму обратной связи;

- серверная часть, построенная на платформе ASP.NET Core с использованием языка C#, предоставляющая REST API для управления проектами и заявками, а также механизм аутентификации администратора на основе JWT-токенов;

- база данных под MYSQL для хранения информации о проектах, заявках клиентов и учетных записях администраторов.

В конечном итоге создан веб-сайт, который удовлетворяет всем поставленным задачам. Разработанное решение включает карту реализованных объектов с возможностью фильтрации, галерею портфолио, модуль расчета стоимости услуг с учетом транспортных расходов и административную панель для управления проектами и отслеживания статусов поступивших обращений.

					ПГУ 1.090301.49.001 ПЗ			
Изм	Лист	№ докум.	Подпис	Дата	Создание сайта организации с применением геоинформационных технологий Пояснительная записка	Лит.	Лист	Листов
Разраб.		Широкова		25.05.26				
Провер.		Леонова		25.05.26			2	64
Н. Контр.		Бычков А.С.		25.05.26				
				25.05.26				
Утверд.		Митрохин				ФВТ, каф. ВТ гр. 22ВВП2		

Содержание

Перечень сокращений и обозначений	6
Введение	8
1 Обзор предметной области	9
1.1 Анализ существующих аналогичных решений	9
1.2 Постановка задачи	12
2 Выбор инструментов и среды разработки	13
2.1 Выбор инструментов разработки	13
2.1.1 Выбор языка и фреймворка для клиентской части	13
2.1.2 Выбор платформы и языка для серверной части	14
2.1.3 Выбор системы управления базами данных	14
2.1.4 Выбор внешних геоинформационных сервисов и библиотек	15
2.2 Выбор среды разработки	16
3 Проектирование системы	17
3.1 Архитектура клиент-серверного приложения	17
3.2 Проектирование структуры базы данных	18
3.3 Проектирование геоинформационного модуля	19
4 Реализация веб-приложения	20
4.1 Описание функционала и работы приложения	20
4.2 Реализация серверной части	26
4.2.1 Контроллер аутентификации	27
4.2.2 Контроллер управления проектами	28
4.2.3 Контроллер управления заявками	30
4.3 Реализация клиентской части	30

4.3.1 Компонентная архитектура интерфейса	31
4.3.2 Маршрутизация	32
4.3.3 Реализация калькулятора стоимости с логистикой	33
4.3.4 JWT-авторизация	35
4.3.5 Отправка запросов к серверу	36
4.3.6 Сжатие фотографии	37
4.3.7 Компонент публичной карты	38
4.3.8 Геокодирование в административной панели	39
5 Тестирование	43
5.1 Тестирование авторизации и аутентификации	43
5.2 Тестирование CRUD проектов и загрузки изображений	45
5.3 Тестирование работы геоинформационного модуля	49
5.4 Тестирование расчета стоимости дизайн-проекта и логистики	51
5.5 Тестирование формы обратной связи и просмотра заявок	53
6 Безопасность жизнедеятельности	56
6.1 Правила и техника безопасности при работе с ПК	56
6.2 Правила и техника безопасности при работе с мобильными устройствами	58
7 Оценка экономической эффективности	59
7.1 Расчет прямого экономического эффекта	59
7.2 Расчет косвенного экономического эффекта	61
7.3 Расчет итоговых показателей экономической эффективности	61
Заключение	63
Список использованных источников	64
Приложение А – UML-диаграммы	65

Приложение Б – Листинги программ	74
Приложение В – Презентация	181
Приложение Г – Справка о внедрении выпускной квалификационной работы ..	188

					ПГУ 1.090301.49.001 ПЗ	Лист
						5
Изм	Лист	№ докум.	Подпись	Дата		

Перечень сокращений и обозначений

В работе используются следующие обозначения и сокращения

БД – база данных

ИП – индивидуальный предприниматель

ОС – операционная система

ПК – персональный компьютер

СУБД – система управления базами данных

API – (англ. Application Programming Interface) – программный интерфейс приложения

ASP.NET – (англ. Active Server Pages .NET) – фреймворк для разработки веб-приложений

CORS – (англ. Cross-Origin Resource Sharing) – механизм совместного использования ресурсов разных источников

CRUD – (англ. Create, Read, Update, Delete) – операции создания, чтения, обновления и удаления данных

CSS – (англ. Cascading Style Sheets) – каскадные таблицы стилей

DOM – (англ. Document Object Model) – объектная модель документа

HTML – (англ. HyperText Markup Language) – язык гипертекстовой разметки

HTTP – (англ. HyperText Transfer Protocol) – протокол передачи гипертекста

HTTPS – (англ. HyperText Transfer Protocol Secure) – защищенный протокол передачи гипертекста

IDE – (англ. Integrated Development Environment) – интегрированная среда разработки

JSON – (англ. JavaScript Object Notation) – текстовый формат обмена данными

JWT – (англ. JSON Web Token) – веб-токен в формате JSON для передачи данных аутентификации

MVC – (англ. Model-View-Controller) – архитектурный шаблон разделения данных, логики и представления

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		6

OSM – (англ. OpenStreetMap) – проект по созданию свободной географической карты мира

OSRM – (англ. Open Source Routing Machine) – высокопроизводительный движок маршрутизации с открытым исходным кодом

REST – (англ. Representational State Transfer) – архитектурный стиль взаимодействия компонентов распределенного приложения

SPA – (англ. Single Page Application) – одностраничное веб-приложение

SQL – (англ. Structured Query Language) – язык структурированных запросов

UML – (англ. Unified Modeling Language) – унифицированный язык моделирования

URL – (англ. Uniform Resource Locator) – единый указатель ресурса

					ПГУ 1.090301.49.001 ПЗ	Лист
						7
Изм	Лист	№ докум.	Подпись	Дата		

Введение

На современном рынке услуг по дизайну интерьеров визуализация опыта компании играет ключевую роль в привлечении клиентов. Традиционные методы демонстрации портфолио в виде статичных фотогалерей или тематических подборок постепенно теряют свою эффективность. Современный пользователь, принимая решение о сотрудничестве, стремится получить максимум информации о компании, включая географию реализованных проектов. Заказчику важно понимать, имеет ли студия опыт работы с конкретным типом жилья. Использование геоинформационных технологий позволяет трансформировать портфолио в наглядную карту реализованных объектов. Такой подход не только повышает наглядность и прозрачность деятельности компании, но и формирует дополнительное доверие со стороны посетителя сайта. Автоматизация первичных расчетов стоимости услуг через онлайн-калькулятор позволяет отсеять «нецелевых» клиентов на предпроектной стадии и дать заинтересованному лицу инструмент для быстрой оценки бюджета.

Целью работы является разработка веб-сайта для студии дизайна интерьеров, который должен объединить в себе карту реализованных проектов, модуль расчета стоимости и административный функционал с возможностью управления содержимым портфолио.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ предметной области и существующих веб-решений;
- обосновать выбор технологического стека для разработки;
- спроектировать клиент-серверную архитектуру приложения;
- разработать клиентскую и серверную часть веб-сайта;
- реализовать панель администратора для управления проектами и обработки поступивших заявок;
- выполнить функциональное тестирование;
- оценить экономическую эффективность внедрения веб-сайта.

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		8

1 Обзор предметной области

1.1 Анализ существующих аналогичных решений

Для выявления сильных и слабых сторон существующих веб-решений, а также для обоснования выбора проектных решений, был проведен сравнительный анализ трех интернет-ресурсов компаний, работающих в сфере дизайна интерьеров.

1. Студия дизайна интерьера «Эдиз»

Особенности: данный сайт реализован в виде ленты, последовательно прокручиваемой вниз. Портфолио включает детальные проекты (от чертежей и 3D-визуализаций до фотографий готовых объектов), однако их количество строго ограничено (6 детальных проектов и 8 изображений в виде сетки). На сайте представлен перечень услуг, но полностью отсутствует информация о ценообразовании. Главная страница и организация портфолио студии представлены на рисунке 1.

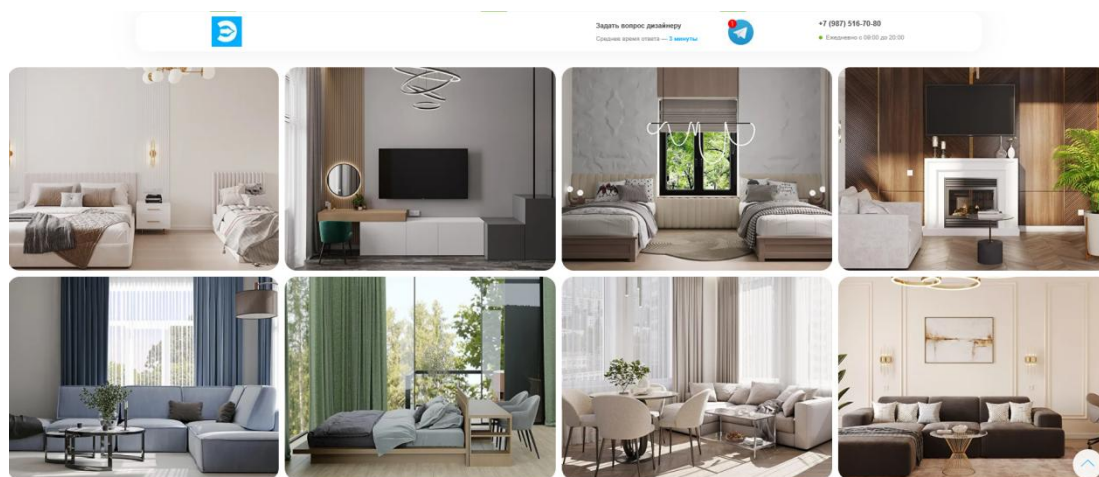


Рисунок 1 – Интерфейс сайта «Эдиз»

Недостатки: предварительную стоимость невозможно узнать без отправки персональных данных через форму обратной связи, малое количество представленных работ не дает полного представления о географии деятельности студии.

2. Архитектурное бюро «Арх.Бюро»

Особенности: структура портфолио разделена на четыре четкие категории (жилые интерьеры, коммерческие интерьеры, архитектура, реализованные проекты). Внешний вид структурированного каталога проектов бюро продемонстрирован на рисунке 2.

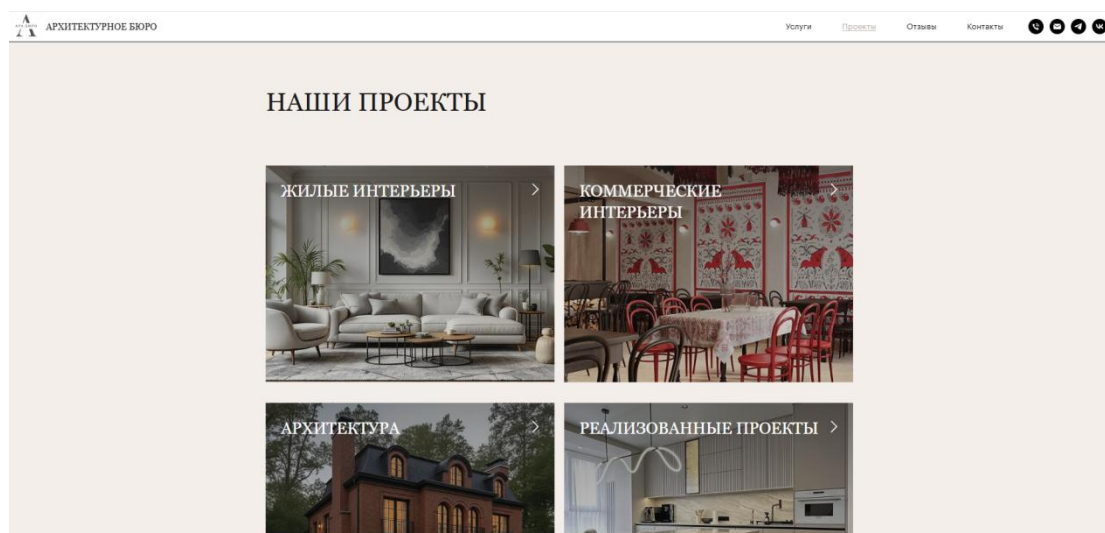


Рисунок 2 – Интерфейс сайта «Арх.Бюро»

Недостатки: несмотря на качественную структуру по типам объектов, портфолио представляет собой обычную фотогалерею. Пользователю сложно соотнести реализованные проекты с конкретными районами или типами застройки, что затрудняет оценку опыта компании. Финансовая информация представлена в виде статичного текстового прайс-листа, что исключает возможность интерактивной оценки бюджета и вынуждает пользователя производить ручные расчеты.

3. Дизайн-студия «Анфилада»

Особенности: имеет разветвленное навигационное меню, которое обеспечивает хорошую информативность и удобство перемещения по различным разделам сайта. Внешний вид навигации раздела с работами студии представлен на рисунке 3.

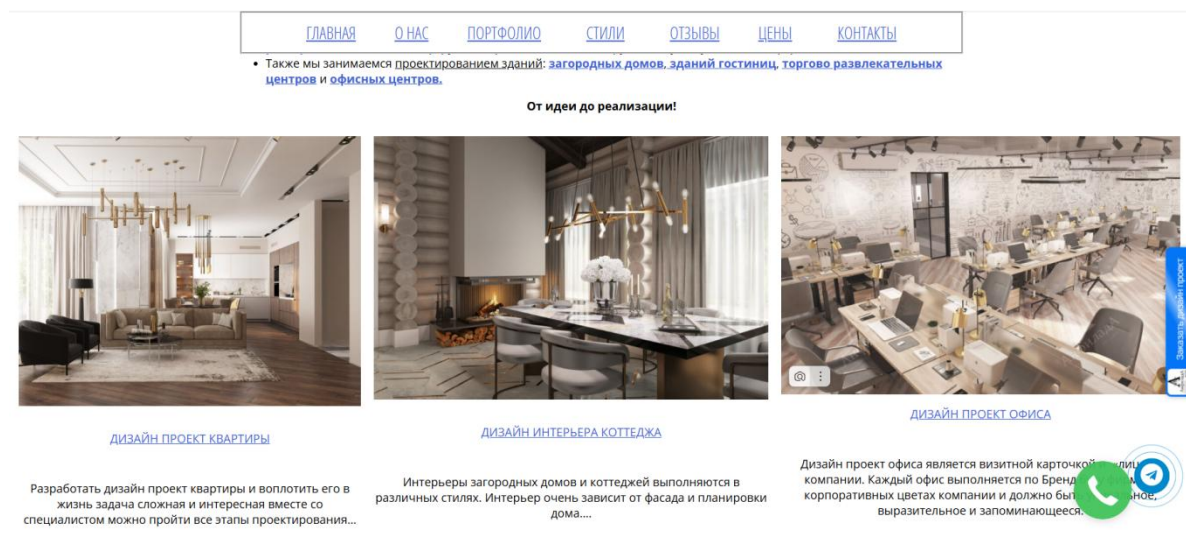


Рисунок 3 – Интерфейс сайта «Анфилада»

Недостатки: традиционный подход к визуализации портфолио ограничивает возможности пространственного анализа опыта компании. Кроме того, интерфейс ресурса не предлагает инструментов для вовлечения пользователя в процесс планирования.

Проведенный анализ выявил ряд общих недостатков в рассмотренных веб-решениях, снижающих их коммерческую эффективность и удобство для конечного пользователя. Для систематизации данных и обоснования выбора проектных решений результаты сравнительного анализа функциональных возможностей сведены в таблицу 1.

Таблица 1 – Сравнительный анализ сайтов-аналогов

Критерий сравнения	Студия «Эдиз»	Архитектурное бюро «Арх.Бюро»	Дизайн-студия «Анфилада»
Способ визуализации проектов	Ограниченная фотогалерея	Структурированный каталог	Каталог с навигацией
Формат информирования о стоимости услуг	Скрыт(предоставляется по запросу)	Текстовый прайс-лист	Текстовый прайс-лист
Автоматизация предварительной оценки	Нет	Нет	Нет

Перечисленные проблемы снижают эффективность привлечения клиентов и не позволяют пользователям самостоятельно оценивать стоимость услуг с учетом индивидуальных параметров. Предлагаемое решение призвано устранить выявленные недостатки, повысить наглядность портфолио и вовлеченность посетителей, а также предоставить студии эффективный инструмент для демонстрации своего опыта и автоматизации первичных расчетов.

1.2 Постановка задачи

В рамках выполнения работы требуется разработать информационную систему в виде веб-сайта для студии дизайна интерьеров. Система должна иметь клиент-серверную архитектуру и панель администратора.

В приложении должно быть реализовано:

- адаптивная верстка для корректного отображения на ПК, планшетах и мобильных устройствах;
- карта реализованных проектов;
- галерея проектов;
- калькулятор стоимости дизайн-проекта;
- автоматический расчет транспортных расходов на основе реального расстояния от офиса студии до объекта с использованием OSRM API;
- форма обратной связи;
- панель администратора, с возможностью изменять, удалять и добавлять проекты и просматривать список заявок клиентов.

Минимальные системные требования включают в себя:

- ОС Windows / Linux, .NET 8.0, MySQL 8.0 для серверной части
- любой современный браузер (Chrome, Firefox, Safari, Opera) с поддержкой JavaScript для клиентской части;
- наличие подключения к сети Интернет.

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		12

2 Выбор инструментов и среды разработки

2.1 Выбор инструментов разработки

Разработка современного веб-приложения требует обоснованного выбора технологического стека, от которого зависят такие характеристики системы, как скорость работы, масштабируемость, безопасность и удобство сопровождения кода. В рамках данной работы архитектура проекта разделена на три основные составляющие: клиентскую часть, серверную часть, систему управления базами данных, а также включает интеграцию с внешними геоинформационными сервисами.

2.1.1 Выбор языка и фреймворка для клиентской части

Клиентская часть веб-сайта студии дизайна интерьеров должна обеспечивать высокую интерактивность. Пользователь взаимодействует с интерактивной картой, калькулятором стоимости и формами обратной связи без перезагрузки страницы. Для достижения такого поведения приложение реализовано как одностраничное. В SPA маршрутизация и смена контента происходят на стороне клиента, что исключает полную перезагрузку страницы.

В качестве основного языка клиентской разработки выбран JavaScript, являющийся стандартом для веб-программирования. Для построения пользовательского интерфейса применяется React – декларативная JavaScript-библиотека для создания компонентных веб-приложений. Выбор React обоснован концепцией виртуального DOM, которая позволяет точно и мгновенно перерисовывать только изменившиеся элементы экрана. Это критически важно для работы модуля калькулятора, где пересчет стоимости должен происходить в реальном времени при изменении пользователем параметров помещения.

Для сборки и разработки фронтенд-части используется инструмент Vite. Он позволяет быстро запускать сервер для разработки, мгновенно обновляя страницу при внесении изменений в код.

2.1.2 Выбор платформы и языка для серверной части

Серверная часть приложения отвечает за бизнес-логику, обработку запросов от клиента, взаимодействие с базой данных и обеспечение безопасности административной панели.

Для реализации сервера выбран язык программирования C# и платформа ASP.NET Core. Данный выбор обусловлен сочетанием производительности, безопасности и удобства работы с данными. ASP.NET Core представляет собой высокопроизводительную кроссплатформенную среду, оптимизированную для обработки большого количества параллельных подключений, что важно для веб-приложений с несколькими одновременными пользователями. Платформа предоставляет встроенные механизмы аутентификации и авторизации, позволяющие надежно защитить панель администратора от несанкционированного доступа. Взаимодействие с базой данных упрощается за счет технологии объектно-реляционного отображения Entity Framework Core. Данная технология сводит к минимуму написание SQL-запросов вручную, автоматически генерируя их на основе объектной модели, и обеспечивает защиту от SQL-инъекций. Сам язык C# со строгой типизацией помогает выявлять ошибки на этапе написания кода. Развитые механизмы асинхронного программирования необходимы для эффективной обработки сетевых вызовов. C# имеет большую экосистему библиотек, богатую документацию и обширное сообщество разработчиков, что ускоряет процесс создания проекта.

2.1.3 Выбор системы управления базами данных

Для хранения структурированной информации о проектах, заявках пользователей и учетных записях администраторов требуется надежная реляционная система управления базами данных. В качестве такой СУБД была выбрана MySQL. Данное решение обусловлено тем, что MySQL хорошо подходит для хранения структурированных данных веб-сайтов и успешно справляется с соответствующими нагрузками. MySQL является одной из наиболее распространенных СУБД в веб-разработке, для нее существуют

стабильные и производительные драйверы, обеспечивающие совместимость с ASP.NET Core. Немаловажным фактором также выступает экономическая эффективность: MySQL Community Server распространяется бесплатно как решение с открытым исходным кодом и не требует дополнительных финансовых затрат при внедрении.

2.1.4 Выбор внешних геоинформационных сервисов и библиотек

В качестве основной картографической библиотеки выбран Leaflet – легковесная JavaScript-библиотека с открытым исходным кодом для создания интерактивных карт, интегрированная в React пакетом react-leaflet. Данное решение обусловлено несколькими факторами: Leaflet распространяется под свободной лицензией; библиотека имеет малый размер и не создает существенной нагрузки на клиентскую часть; Leaflet обладает гибкой системой кастомизации маркеров и всплывающих окон и совместим с большинством современных браузеров, хорошо документирован.

Для работы с пространственными данными задействованы два сервиса, базирующиеся на открытых данных OpenStreetMap:

Nominatim API используется для прямого и обратного геокодирования. Сервис отвечает за преобразование строкового адреса в географические координаты.

OSRM API применяется в модуле калькулятора для расчета транспортных расходов. Сервис выполняет построение оптимального автомобильного маршрута от офиса дизайн-студии до объекта заказчика, возвращая точное расстояние в километрах и геометрию маршрута для ее наглядной отрисовки поверх карты.

Оба сервиса являются общедоступными, не требуют регистрации, получения API-ключей и не вводят строгих квот на количество запросов при умеренном использовании. Точности и полноты данных OpenStreetMap достаточно для задач студии дизайна интерьеров. Основной географией работы является город и его окрестности, где OSM обладает хорошей детализацией

2.2 Выбор среды разработки

Для написания, отладки и сборки проекта будет использоваться следующее программное обеспечение:

- в качестве IDE для Backend выбран Microsoft Visual Studio 2022. Данная среда предоставляет мощные инструменты для работы с C#, включая отладчик с возможностью пошагового выполнения кода, встроенный менеджер пакетов NuGet для управления зависимостями, и визуальные конструкторы для работы с базами данных.

- в качестве редактор для Frontend выбран Visual Studio Code. Легковесный и расширяемый редактор кода, оптимизированный для работы с JavaScript, HTML и CSS. Обладает развитой экосистемой расширений и встроенным терминалом.

- в качестве СУБД выбран MySQL 8.0 и графическая утилита MySQL Workbench. MySQL Community Server служит для хранения структурированной информации о проектах, заявках и учетных записях администраторов. MySQL Workbench применяется для визуального проектирования схемы базы данных, создания индексов, выполнения отладочных запросов и анализа планов выполнения SQL-запросов.

3 Проектирование системы

3.1 Архитектура клиент-серверного приложения

Разрабатываемая информационная система построена по классической клиент-серверной архитектуре, разделенной на три основные части: пользовательский интерфейс (клиент), бизнес-логику (сервер) и базу данных. Такое разделение позволяет разрабатывать, тестировать и обновлять каждую часть независимо друг от друга, что делает работу всего веб-приложения более быстрой и стабильной.

Клиентская часть представляет собой одностраничное веб-приложение, написанное с помощью библиотеки React. Интерфейс собирается из отдельных блоков – компонентов. Переход между страницами происходит мгновенно, без перезагрузки сайта в браузере.

Серверная часть разработана на базе ASP.NET Core и работает как REST API. В ее основе лежит шаблон MVC, где контроллеры принимают запросы от клиентской части, а модели описывают структуру данных (заявки, проекты, администраторы).

Взаимодействие между клиентом и сервером происходит по протоколу HTTP, а информация передается в текстовом формате JSON. Так как клиент и сервер могут быть запущены на разных сетевых адресах или портах, на сервере настроена технология CORS. Она настроена так, чтобы сервер принимал запросы только от доверенного адреса сайта.

За безопасность и авторизацию отвечает стандарт JWT. Когда администратор вводит верный логин и пароль, сервер создает зашифрованный токен и отправляет его клиенту. Приложение сохраняет токен в памяти браузера (localStorage) и прикрепляет его к каждому следующему запросу в админ-панели. Сервер проверяет этот токен и если он верный команда выполняется, иначе доступ блокируется.

3.2 Проектирование структуры базы данных

Для хранения информации в разрабатываемой системе используется реляционная система управления базами данных MySQL. Взаимодействие серверного приложения с БД реализовано через технологию Entity Framework Core с применением подхода Code First. Структура таблиц генерируется автоматически на основе объектно-ориентированных классов языка C#. [1]. Контекст базы данных объединяет три основные сущности: проекты, заявки и администраторов.

Таблица «Проекты» предназначена для хранения информации о реализованных дизайн-проектах, которые выводятся в каталоге и на интерактивной карте. Для хранения коллекций данных, таких как пути к загруженным фотографиям (Images), в базе данных используется текстовый формат. На уровне контекста БД настроена автоматическая конвертация списков в формат JSON при сохранении и их обратная распаковка при чтении.

Таблица «Заявки» используется для накопления обращений от потенциальных клиентов, поступающих из модуля калькулятора или формы обратной связи.

Таблица «Администраторы» обеспечивает хранение учетных записей сотрудников, имеющих доступ к закрытой панели управления контентом.

Для описания объектно-ориентированной структуры серверной части приложения разработана диаграмма классов, которая представлена на рисунке 4. Она отражает основные программные компоненты: контроллеры, контекст базы данных и модели данных

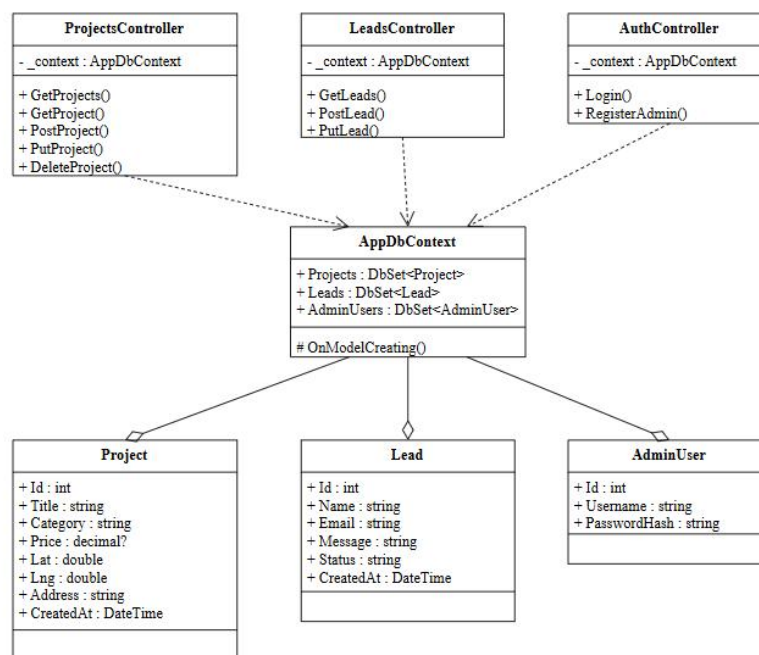


Рисунок 4 – UML-диаграмма классов

3.3 Проектирование геоинформационного модуля

Геоинформационный модуль является самостоятельной подсистемой в составе клиентской части и отвечает за визуализацию проектов на карте, фильтрацию маркеров, а также за получение данных от внешних геосервисов.

Для преобразования адресов в координаты и обратного получения адреса по координатам на карте используется открытый сервис Nominatim на базе данных OpenStreetMap. Запросы к нему отправляются по протоколу HTTPS как из панели администратора, так и из калькулятора.

Для расчета расстояния и построения маршрута задействован сервис OSRM. Он вызывается из модуля калькулятора и возвращает два основных параметра: протяженность маршрута в километрах и массив координат для отрисовки полилинии поверх карты. Для обеспечения высокой отказоустойчивости системы заложен резервный механизм вычисления расстояний с применением формулы гаверсина.

4 Реализация веб-приложения

4.1 Описание функционала и работы приложения

Система рассчитана на две роли: обычный пользователь и администратор.

Пользователь может:

- просматривать список реализованных проектов в виде галереи или на карте;
- фильтровать проекты по категориям (квартиры, спальни, кухни, гостиные, ванные);
- открывать детальную страницу проекта с фотографиями и описанием;
- рассчитывать стоимость дизайн-проекта в интерактивном калькуляторе, который учитывает площадь, тип объекта, выбранный пакет услуг, а также транспортные расходы (для тарифов с выездом на объект);
- оставлять заявку на консультацию, причем если заявка отправляется из калькулятора, поле сообщения автоматически заполняется деталями расчета.

Администратор после входа по логину и паролю получает доступ к панели управления, где может:

- добавлять, редактировать и удалять проекты;
- загружать несколько фотографий для каждого проекта;
- указывать для проекта точные координаты на карте;
- выбирать, где отображать проект: только в галерее, только на карте или везде;
- просматривать список заявок, поступивших с сайта, и менять их статус (новая / обработанная).

На рисунке 5 представлена UML-диаграмма прецедентов которая наглядно показывает, какие функции доступны каждой роли.



Рисунок 5 – UML-диаграмма прецедентов

Пользователь взаимодействует только с клиентской частью веб-сайта. Для того чтобы начать пользоваться функционалом, достаточно открыть сайт в любом современном браузере. Пользователю доступны страницы: главная, о нас, услуги, проекты, калькулятор, контакты.

При работе с калькулятором пользователь задает параметры: площадь помещения, тип объекта (новостройка или вторичное жилье), выбирает пакет услуг (планировочный, базовый, комплексный или VIP) и по желанию дополнительные услуги. При изменении любого параметра стоимость пересчитывается мгновенно, без перезагрузки страницы. Если выбран тариф «Комплексный» или «VIP», пользователь может включить опцию транспортного сопровождения. Для этого он вводит адрес объекта и нажимает кнопку «Рассчитать логистику». Клиент отправляет GET-запрос к Nominatim API для преобразования адреса в координаты, затем к OSRM API для построения маршрута и получения расстояния в километрах. На карте отображается полилиния маршрута, а транспортные расходы добавляются к итоговой

стоимости. При недоступности OSRM используется резервное вычисление по формуле гаверсинуса с коэффициентом 1,2. Этот процесс проиллюстрирован на рисунке 6.

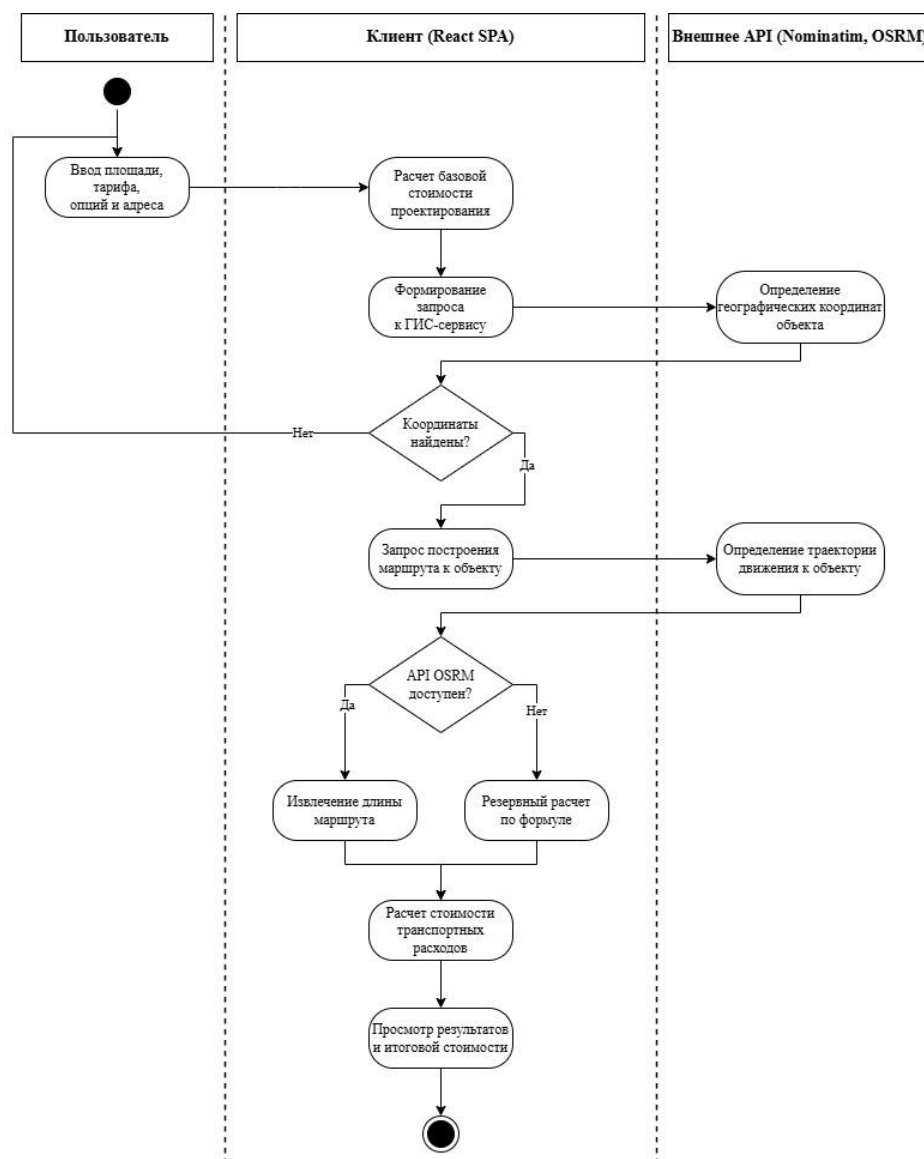


Рисунок 6 – UML-диаграмма деятельности для расчета стоимости проекта в калькуляторе

Если пользователь хочет оставить заявку через форму обратной связи, он заполняет поля «Имя», «Email» и «Сообщение». Клиентская часть перед отправкой проверяет, что поля не пустые и email имеет корректный формат. Данные отправляются на сервер через POST-запрос. На сервере добавляется

временная метка и статус «new». Заявка сохраняется в таблицу Leads. В случае успеха возвращает статус 200 OK, а клиенту показывается всплывающее сообщение «Успешно!» Данная последовательность действий отображена на рисунке 7.

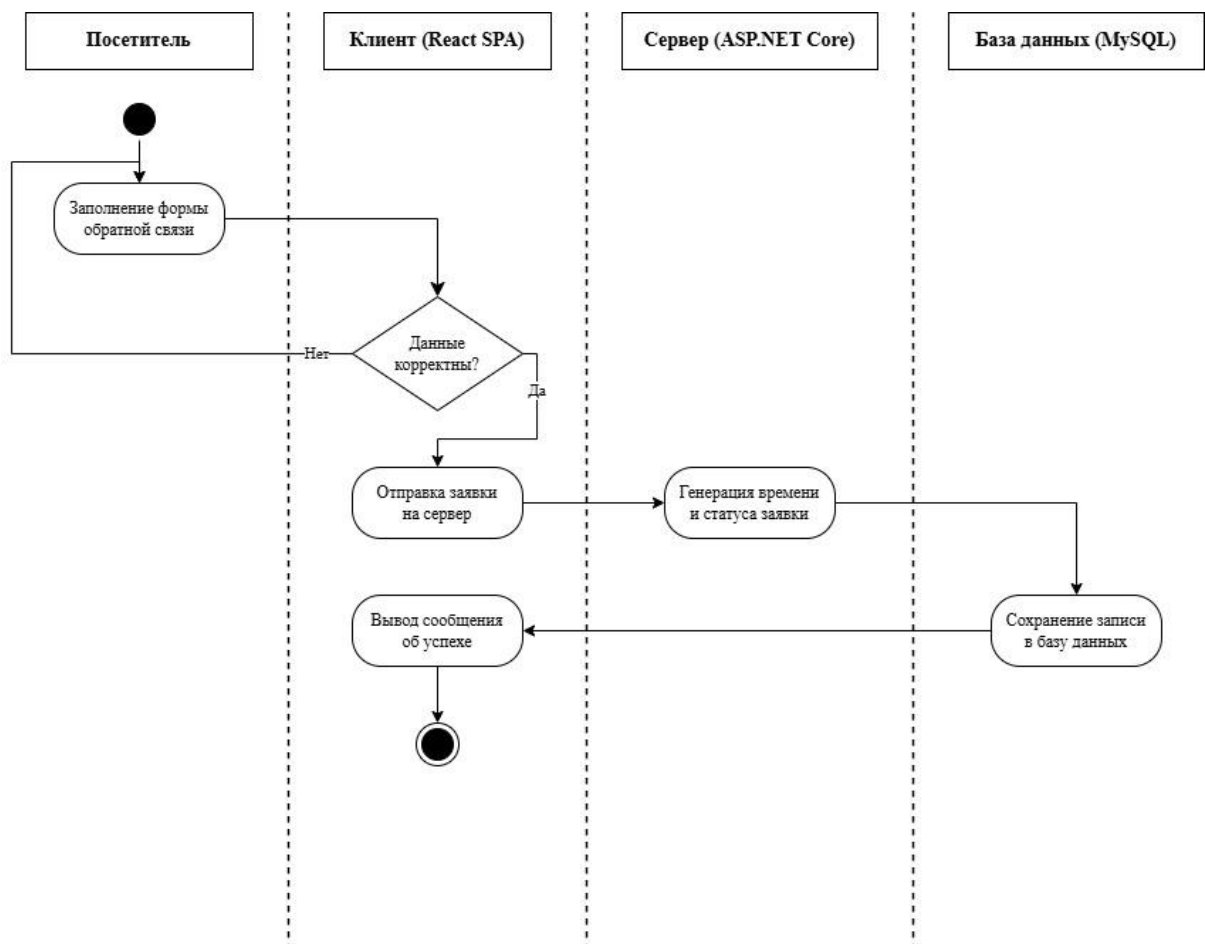


Рисунок 7 – UML-диаграмма деятельности для заполнения формы обратной связи

Для администратора работа начинается со страницы входа. Он вводит логин и пароль, клиент отправляет эти данные на сервер. Сервер сверяет их с записями в таблице администраторов, и если данные верны, генерирует JWT, который возвращается клиенту. Клиент сохраняет токен в локальном хранилище браузера и перенаправляет администратора на защищенную панель управления. Все последующие запросы к серверу (добавление, редактирование или удаление

проектов, просмотр заявок) клиент сопровождает этим токеном в заголовке Authorization.

На рисунке 8 приведена UML-диаграмма для процесса добавления нового проекта администратором. Сначала заполняются поля (название, описание, категория, адрес) и загружаются изображения (на клиенте автоматически сжимаются до 1200px и конвертируются в webp формат). После ввода адреса администратор нажимает кнопку «Найти на карте». Клиентское приложение отправляет GETзапрос к внешнему API Nominatim, которое возвращает координаты (широту и долготу). Для обратного геокодирования при клике по карте вручную также используется внешний API Nominatim, что позволяет автоматически подставлять адрес в текстовое поле формы. После нажатия кнопки «Опубликовать» отправляется POSTзапрос на сервер с данными в формате JSON и изображениями в Base64. Сервер декодирует изображение, сохраняет его в папку, генерируя уникальные имена. Затем запись о проекте добавляется в таблицу Projects базы данных. При успешном сохранении сервер возвращает клиенту статус 201 с данными созданного проекта. Клиент отображает уведомление «Успешно» и обновляет список проектов. Если обязательные поля не заполнены или изображения отсутствуют, сервер возвращает ошибку, а пользователю показывается соответствующее предупреждение.

Все выполняемые действия как пользователя, так и администратора подробно изображены на диаграммах последовательности на рисунках 9 и 10.

Физическое размещение компонентов системы показано на рисунке 11. Клиент взаимодействует через протокол HTTPS с веб-сервером, на котором запущено приложение ASP.NET Core. Сервер, в свою очередь, подключается к серверу MySQL. Для геоинформационных функций клиент напрямую обращается к внешним API: Nominatim для поиска адресов и OSRM для построения маршрутов. Эти сервисы не требуют авторизации и работают по протоколу HTTP/HTTPS.

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		24

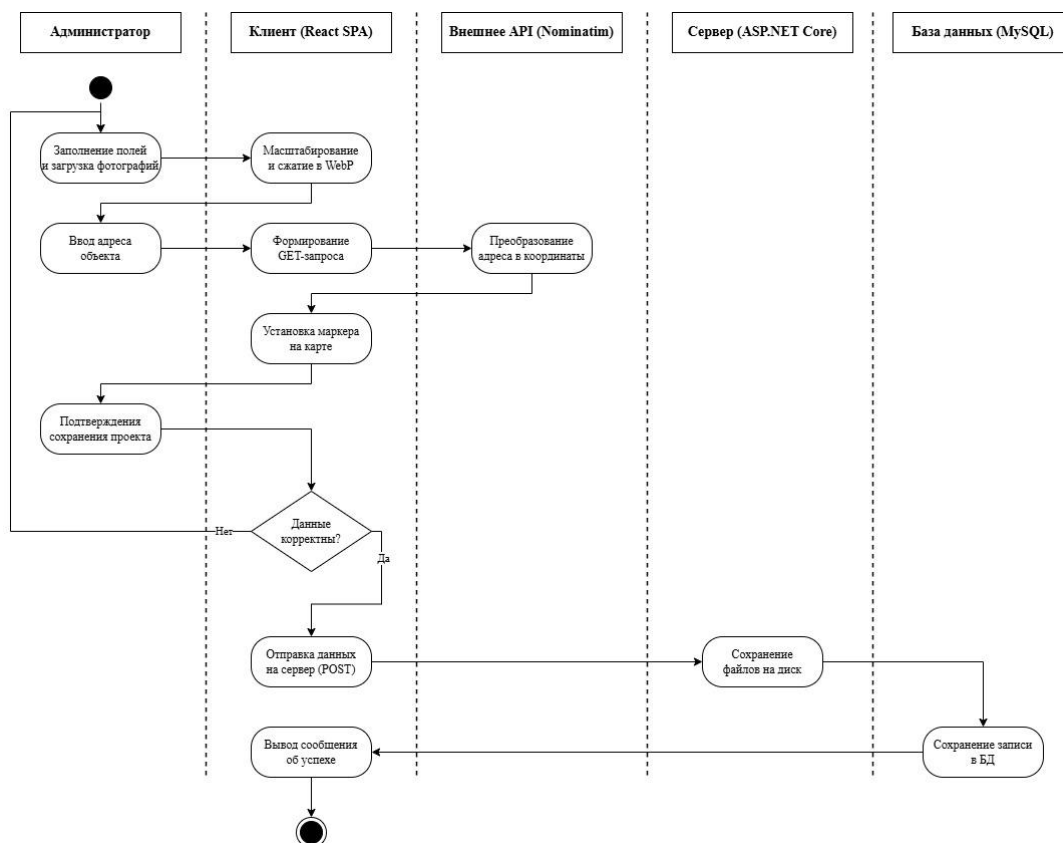


Рисунок 8 – UML-диаграмма деятельности для добавления проекта в БД

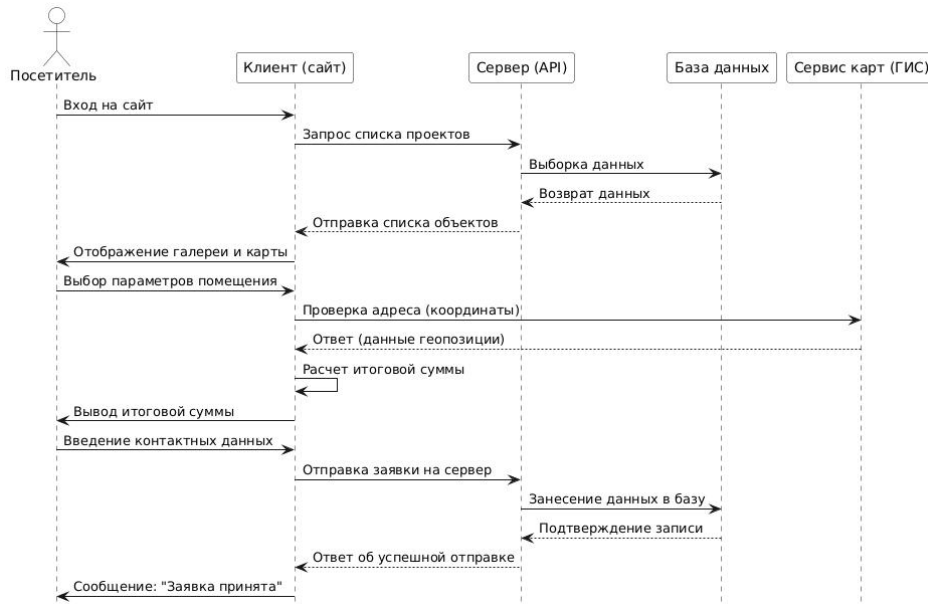


Рисунок 9 – UML-диаграмма последовательности для посетителя

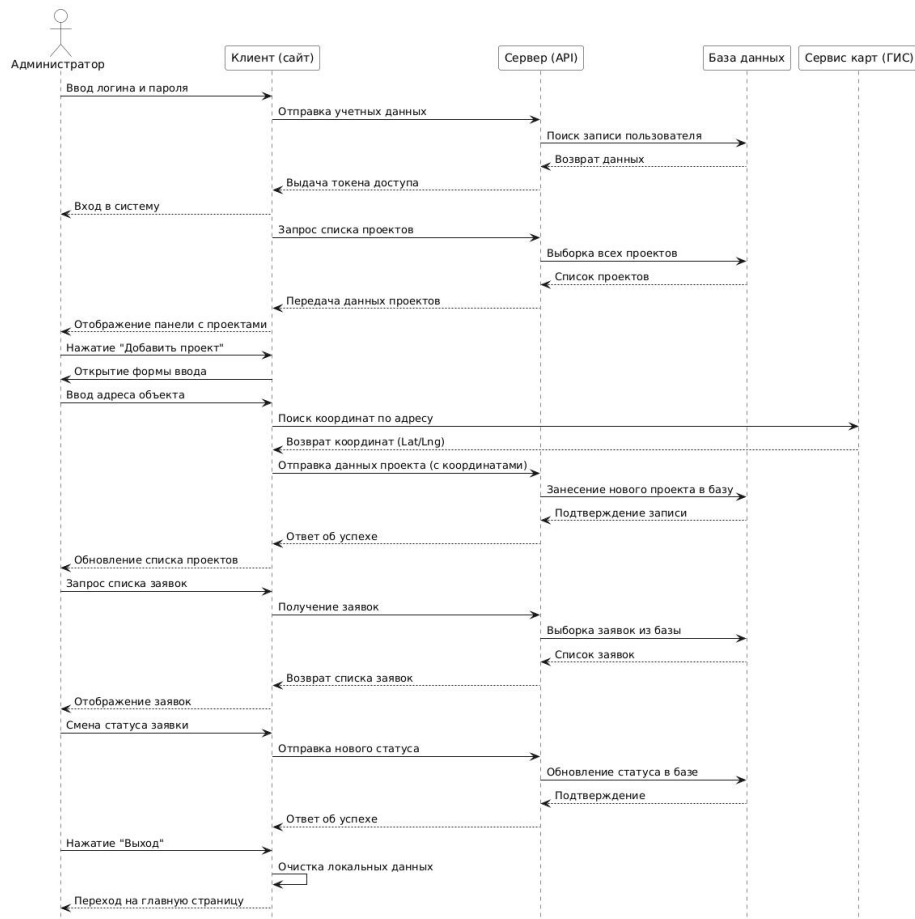


Рисунок 10 – UML-диаграмма последовательности для администратора

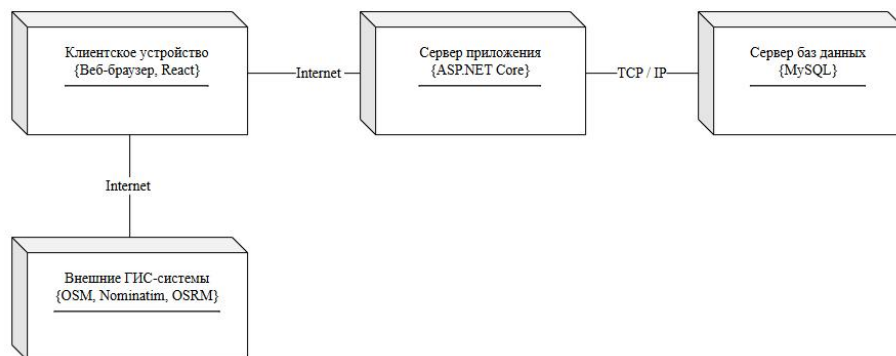


Рисунок 11 – UML-диаграмма развертывания

4.2 Реализация серверной части

Серверная часть веб-приложения написана на платформе ASP.NET Core. Главным конфигурационным файлом выступает Program.cs. В нем прописаны все базовые правила работы бэкенда: подключение к базе данных MySQL, запуск

системы безопасности с проверкой JWT и настройка правил CORS, которые разрешают серверу принимать запросы исключительно от React-клиента.

Взаимодействие серверного кода с базой данных реализовано через технологию Entity Framework Core. Для этого в проекте создан класс ApplicationDbContext, который служит связующим звеном между объектно-ориентированным кодом на C# и реляционными таблицами MySQL. Использование подхода Code First позволило описать всю структуру хранимых данных в виде обычных классов-моделей[1].

Логика обработки HTTP-запросов распределена по контроллерам – специализированным классам, наследующим ControllerBase. Каждый такой контроллер отвечает за конкретную предметную область [2].

4.2.1 Контроллер аутентификации

Контроллер AuthController отвечает за аутентификацию. При поступлении запроса на POST метод Login ищет пользователя в таблице AdminUsers по указанному имени, а затем проверяет пароль с помощью библиотеки BCrypt.

При совпадении хэшей сервер генерирует JWT-токен, подписанный алгоритмом HMACSHA256, содержащий утверждения об имени пользователя и его роли "Admin". Токен имеет срок жизни 12 часов [3]. Фрагмент кода, реализующий аутентификацию пользователя, представлен ниже.

```
[HttpPost("login")]
public IActionResult Login([FromBody] LoginRequest request)
{
    var admin = _context.AdminUsers.FirstOrDefault(a =>
a.Username == request.Username);

    if (admin == null
|| !BCrypt.Net.BCrypt.Verify(request.Password, admin.PasswordHash))
    {
        return Unauthorized(new { message = "Неверный логин
или пароль" });
    }
    var keyStr = _configuration["Jwt:Key"];
    var issuer = _configuration["Jwt:Issuer"];
    var audience = _configuration["Jwt:Audience"];
```

```

        var securityKey = new
SymmetricSecurityKey(Encoding.UTF8.GetBytes(keyStr!));
        var credentials = new SigningCredentials(securityKey,
SecurityAlgorithms.HmacSha256);

        var claims = new[]
        {
            new Claim(ClaimTypes.Name, admin.Username),
            new Claim(ClaimTypes.Role, "Admin")
        };

        var token = new JwtSecurityToken(
            issuer: "StoryHome",
            audience: "StoryHomeAdmin",
            claims: claims,
            expires: DateTime.Now.AddHours(12),
            signingCredentials: credentials);

        return Ok(new { token = new
JwtSecurityTokenHandler().WriteToken(token) });
    }

```

4.2.2 Контроллер управления проектами

Контроллер `ProjectsController` реализует полный набор CRUD-операций над сущностью «Проект» (таблица 2). Операции чтения доступны всем пользователям, а операции записи, изменения и удаления – только аутентифицированному администратору

Таблица 2 – Методы контроллера `ProjectsController`

Метод	Маршрут	Доступ
<code>GetProjects</code>	<code>api/projects</code>	Публичный
<code>GetProjects</code>	<code>api/projects/{id}</code>	Публичный
<code>PostProjects</code>	<code>api/projects</code>	Администратор
<code>PutProjects</code>	<code>api/projects{id}</code>	Администратор
<code>DeleteProjects</code>	<code>api/projects{id}</code>	Администратор

Обработка изображений происходит по следующему алгоритму: клиент присылает массив строк Base64; сервер извлекает данные, декодирует, генерирует уникальное имя через `Guid.NewGuid()` и сохраняет файл в каталог `wwwroot/images/projects/`. Формируется публичный URL, который возвращается

клиенту и сохраняется в базе. Фрагмент кода, реализующий сохранение изображений, представлен ниже.

```
private async Task<List<string>>
SaveImagesToFolderAsync(List<string> base64Images)
{
    var savedUrls = new List<string>();

    if (base64Images == null || base64Images.Count == 0)
    {
        return savedUrls;
    }

    var folderPath =
Path.Combine(Directory.GetCurrentDirectory(), "wwwroot", "images",
"projects");

    if (!Directory.Exists(folderPath))
    {
        Directory.CreateDirectory(folderPath);
    }

    foreach (var base64 in base64Images)
    {
        if (string.IsNullOrEmpty(base64)) continue;

        if (base64.StartsWith("http") ||
base64.StartsWith("/"))
        {
            savedUrls.Add(base64);
            continue;
        }

        try
        {
            var base64Data = base64.Contains(",") ?
base64.Split(',')[1] : base64;
            var bytes = Convert.FromBase64String(base64Data);

            var fileName = Guid.NewGuid().ToString() +
".jpg";
            var filePath = Path.Combine(folderPath,
fileName);

            await System.IO.File.WriteAllBytesAsync(filePath,
bytes);

            savedUrls.Add($"https://localhost:7002/images/projects/{fileName}");
        }
        catch (Exception ex)
```

					ПГУ 1.090301.49.001 ПЗ	Лист
						29
Изм	Лист	№ докум.	Подпись	Дата		

```

        {
            Console.WriteLine("Ошибка сохранения картинки: "
+ ex.Message);
        }
    }
    return savedUrls;
}

```

4.2.3 Контроллер управления заявками

Контроллер LeadsController предназначен для сбора и управления заявками. Метод PostLead добавляет временную метку и статус «new». Методы GetLeads и PutLead позволяют администратору просматривать все заявки и изменять их статус. Фрагмент кода, реализующий добавление новой заявки, представлен ниже.

```

[HttpPost]
public async Task<ActionResult<Lead>> PostLead(Lead lead)
{
    if (string.IsNullOrEmpty(lead.Date))
        lead.Date = DateTime.Now.ToString("dd.MM.yyyy, HH:mm");

    lead.Status = "new";

    _context.Leads.Add(lead);
    await _context.SaveChangesAsync();

    return Ok(lead);
}

```

4.3 Реализация клиентской части

Клиентская часть веб-приложения написана на базе библиотеки React в виде одностраничного приложения. Главной точкой входа в проект является файл main.jsx, который отвечает за первичную инициализацию React-приложения, монтирование корневого компонента в DOM-дерево браузера и подключение глобальных стилей Tailwind CSS через конфигурационный CSS-файл[4].

Для построения интерфейса используются функциональные компоненты и встроенные React-хуки, что позволяет эффективно управлять состоянием страниц и обрабатывать асинхронные запросы к серверному API[5].

4.3.1 Компонентная архитектура интерфейса

Компонент MainLayout включает три логических блока: верхний Navbar, динамическую область для отображения содержимого текущей страницы Outlet и нижний Footer.

Компонент Navbar реализует навигационную панель, содержащую логотип компании, основное меню с пунктами, которые определены в файле data.js, и адаптивное выезжающее боковое меню для мобильных устройств. В десктопной версии навигация осуществляется через горизонтальное меню на основе компонента Navigation, для мобильных устройств используется выдвижной компонент MobileNavigation, отображающийся поверх основного содержимого.

Компонент Footer содержит информацию о студии, ссылки на основные разделы сайта и контактные данные.

Компонент ProtectedRoute проверяет наличие действительного JWT-токена в локальном хранилище браузера и при отсутствии перенаправляет пользователя на страницу входа. Фрагмент кода, реализующий проверку JWT-токена и защиту от неавторизованного доступа, представлен ниже.

```
import { Navigate, useNavigate } from "react-router-dom";
import { useEffect, useState } from "react";

const ProtectedRoute = ({ children }) => {
  const navigate = useNavigate();
  const [isAuthorized, setIsAuthorized] = useState(true);

  useEffect(() => {

    const verifyToken = () => {
      const token = localStorage.getItem("adminToken");

      if (!token) {
        setIsAuthorized(false);
        navigate("/admin-login", { replace: true });
        return;
      }

      try {
        const payload = JSON.parse(atob(token.split('.')[1]));
        const isExpired = payload.exp * 1000 < Date.now();

        if (isExpired) {
```

```

        localStorage.removeItem("adminToken");
        setIsAuthorized(false);
        navigate("/admin-login", { replace: true });
    }
} catch (error) {
    localStorage.removeItem("adminToken");
    setIsAuthorized(false);
    navigate("/admin-login", { replace: true });
}
};

const intervalId = setInterval(verifyToken, 1000);

return () => clearInterval(intervalId);
}, [navigate]);

const token = localStorage.getItem("adminToken");
if (!token || !isAuthorized) {
    return <Navigate to="/admin-login" replace />;
}

return children;
};

export default ProtectedRoute;

```

4.3.2 Маршрутизация

Маршруты сконфигурированы в файле `Routes.jsx`. Публичные страницы сгруппированы внутри `MainLayout`, административные вынесены отдельно. Фрагмент кода, реализующий настройку маршрутизации, представлен ниже.

```

const router = createBrowserRouter([
  {
    path: "/",
    element: <MainLayout />,
    errorElement: <ErrorPage />,
    children: [
      { path: "/", element: <Home /> },
      { path: "/projects", element: <Projects /> },
      { path: "/projects/:id", element: <ProjectDetails /> },
      { path: "/services", element: <ServicesPage /> },
      { path: "/services/:id", element: <ServiceDetails /> },
      { path: "/contact", element: <Contact /> },
      { path: "/about", element: <About /> },
      { path: "/calculator", element: <Calculator /> },
    ],
  },
  { path: "/admin-login", element: <AdminLogin /> },
  {

```

```

    path: "/admin",
    element: (
      <ProtectedRoute>
        <AdminPage />
      </ProtectedRoute>
    ),
  },
]);

```

4.3.3 Реализация калькулятора стоимости с логистикой

Страница калькулятора (CalculatorPage.jsx) использует хуки `useState` и `useMemo` для расчета стоимости на основе площади, типа объекта, пакета услуг и дополнительных опций. Хук `useMemo` кэширует результат вычисления общей стоимости и пересчитывает его только при изменении одного из зависимых параметров. Фрагмент кода, реализующий динамический расчет итоговой стоимости дизайн-проекта, представлен ниже.

```

const totalCost = useMemo(() => {
  let pricePerSqM = PROJECT_TYPES[projectType].price +
    PROPERTY_TYPES[propertyType].price;
  if (options.smartHome) pricePerSqM +=
    OPTIONS.smartHome.pricePerSqM;
  if (options.ventilation) pricePerSqM +=
    OPTIONS.ventilation.pricePerSqM;
  let total = pricePerSqM * area;
  if (options.approval) total += OPTIONS.approval.fixedPrice;
  if (showTransport && distanceKm > 0) {
    const tripsCount = TRIPS_MAP[projectType] || 0;
    const transportCost = distanceKm * 2 * tripsCount *
    PRICE_PER_KM;
    total += transportCost;
  }
  return Math.round(total);
}, [area, propertyType, projectType, options, showTransport,
distanceKm]);

```

Для тарифов «Комплексный» и «VIP» появляется блок «Транспортное сопровождение». Пользователь вводит адрес объекта и нажимает кнопку «Рассчитать логистику». При нажатии выполняется асинхронная функция `handleCalculateRoute`. Сначала через `fetch` отправляется запрос к Nominatim API (<https://nominatim.openstreetmap.org/search?format=json&q=...>). Ответ содержит массив с координатами. Если адрес найден, извлекаются `lat` и `lon`. Далее выполняется запрос к OSRM по адресу <https://router.project->

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		33

osrm.org/route/v1/driving/{lng1},{lat1};{lng2},{lat2}?overview=full&geometries=geojson. В ответе приходит JSON с полем routes[0].distance (в метрах) и geometry.coordinates (массив точек для построения линии маршрута)[6]. Для расчета расстояния и отображения маршрута используется функция handleCalculateRoute. Фрагмент кода, реализующий запрос к OSRM API, представлен ниже.

```
const osrmRes = await fetch(
  `https://router.project-
osrm.org/route/v1/driving/${OFFICE_COORDS.lng},${OFFICE_COORDS.lat};
${clientLng},${clientLat}?overview=full&geometries=geojson`
);
const osrmData = await osrmRes.json();
if (osrmData.code === "Ok" && osrmData.routes &&
osrmData.routes.length > 0) {
  const route = osrmData.routes[0];
  const calculatedDistanceKm = route.distance / 1000;
  const lineCoords = route.geometry.coordinates.map(coord =>
[coord[1], coord[0]]);
  setDistanceKm(calculatedDistanceKm);
  setRouteCoords(lineCoords);
}
```

Если OSRM возвращает успешный ответ, из него извлекается дистанция в километрах и массив координат для отрисовки полилинии на карте.

Если OSRM вернул ошибку или не ответил, используется запасной вариант: расстояние рассчитывается с применением формулы гаверсина:

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

где a – промежуточный параметр;

φ_1, φ_2 – широта начальной и конечной точке;

$\Delta\varphi$ – разность широт;

$\Delta\lambda$ – разность долгот;

После нахождения промежуточного параметра a , вычисляется центральный угол c (угловое расстояние между двумя точками), на который опирается искомая дуга по формуле

$$c = 2 \cdot \operatorname{atan} 2(\sqrt{a}, \sqrt{1-a})$$

где c – угловое расстояние;

На заключительном этапе вычисляется итоговое расстояние по поверхности сферы по формуле

$$d = R \cdot c$$

где c – средний радиус Земли (принимается равным 6371).

К итоговому расстоянию добавляется 20% для учета извилистости дорог. Фрагмент кода, реализующий резервное вычисление расстояния, представлен ниже.

```
const applyFallback = () => {
  const R = 6371; /

  const dLat = (clientLat - OFFICE_COORDS.lat) * (Math.PI / 180);
  const dLon = (clientLng - OFFICE_COORDS.lng) * (Math.PI / 180);

  const a =
    Math.sin(dLat / 2) * Math.sin(dLat / 2) +
    Math.cos(OFFICE_COORDS.lat * (Math.PI / 180)) *
    Math.cos(clientLat * (Math.PI / 180)) *
    Math.sin(dLon / 2) * Math.sin(dLon / 2);

  const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
  const distanceStraight = R * c;

  setDistanceKm(distanceStraight * 1.2);
  setClientCoords({ lat: clientLat, lng: clientLng });

  setRouteCoords([
    [OFFICE_COORDS.lat, OFFICE_COORDS.lng],
    [clientLat, clientLng],
  ]);
  setIsAddressCalculated(true);
};
```

После завершения расчета клиент может отправить заявку, при этом сообщение в форме обратной связи будет автоматически предзаполнено деталями расчета.

4.3.4 JWT-авторизация

Метод `handleLogin` обрабатывает событие отправки формы авторизации. Выполняется POST-запрос к `/api/auth/login`. При успешном ответе полученный

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		35

токен сохраняется в localStorage под ключом adminToken, поля формы очищаются, и пользователь перенаправляется на защищенную страницу /admin. Фрагмент кода, реализующий отправку запроса и сохранение JWT в локальном хранилище браузера, представлен ниже.

```
const handleLogin = async (e) => {
  e.preventDefault();
  setError("");

  try {
    const response = await
fetch("https://localhost:7002/api/auth/login", {
  method: "POST",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify({ username, password }),
});

    if (response.ok) {
      const data = await response.json();
      localStorage.setItem("adminToken", data.token);
      setUsername("");
      setPassword("");
      navigate("/admin");
    } else {
      setError("Неверный логин или пароль");
      setPassword("");
    }
  } catch {
    setError("Ошибка соединения с сервером");
  }
};
```

4.3.5 Отправка запросов к серверу

При добавлении нового проекта или редактировании существующего клиентская часть отправляет HTTP-запросы к серверу с использованием встроенного интерфейса fetch. Сначала система извлекает из локального хранилища браузера ранее сохраненный JWT-токен, необходимый для аутентификации администратора. Затем динамически определяются переменные URLадреса и метода запроса: для добавления нового объекта применяется POST, а для редактирования существующей записи – PUT. Далее формируются заголовки запроса: явно указывается тип содержимого application/json, и с помощью синтаксиса spread-оператора условно добавляется заголовок

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		36

Authorization с токеном в формате Bearer, если токен был успешно получен. Полезная нагрузка (объект payload, содержащий текстовые данные проекта и закодированные изображения) сериализуется в JSON-строку и помещается в тело запроса (body)[3]. Фрагмент кода, реализующий отправку сформированного пакета данных, представлен ниже.

```
const token = localStorage.getItem("adminToken");
const response = await fetch(url, {
  method: method,
  headers: {
    "Content-Type": "application/json",
    ...(token && { Authorization: `Bearer ${token}` }) ,
  },
  body: JSON.stringify(payload),
});
```

4.3.6 Сжатие фотографии

Разработанный алгоритм выполняет перехват загружаемых файлов и их последовательную обработку. Сначала функция преобразует полученный объект `FileList` в стандартный массив, а для чтения бинарных данных каждого файла используется встроенный интерфейс `FileReader`, который асинхронно считывает файл и конвертирует его во временную строку формата Base64. Для манипуляций с размером изображения задействуется технология HTML5 Canvas. Создается виртуальный элемент, в который с помощью объекта `Image` загружается исходное изображение. Для стандартизации портфолио задается максимальная ширина изображения 1200 пикселей. Отмасштабированное изображение отрисовывается в контексте холста, после чего осуществляется его экспорт в строковый формат Base64 методом `toDataURL`. Изображение перекодировается в формат WEBP с коэффициентом качества 0,8 позволяя значительно уменьшить вес файла при сохранении высокого качества. Использование конструкции `await new Promise` гарантирует, что цикл не перейдет к следующему файлу до тех пор, пока текущий не будет полностью обработан и сжат. После завершения цикла преобразованные Base64-строки добавляются в массив `projectForm.images` через функцию обновления состояния `React`, делая их доступными для предварительного просмотра и последующей

отправки на сервер. Фрагмент кода, реализующий сжатие загружаемых изображений, представлен ниже.

```
const handleImageUpload = async (e) => {
  const files = Array.from(e.target.files);
  if (files.length === 0) return;
  const compressedImages = [];

  for (const file of files) {
    const compressedBase64 = await new Promise((resolve) => {
      const reader = new FileReader();
      reader.readAsDataURL(file);
      reader.onload = (event) => {
        const img = new Image();
        img.src = event.target.result;
        img.onload = () => {
          const canvas = document.createElement("canvas");
          const MAX_WIDTH = 1200;
          let width = img.width;
          let height = img.height;
          if (width > MAX_WIDTH) {
            height = Math.round((height * MAX_WIDTH) / width);
            width = MAX_WIDTH;
          }
          canvas.width = width;
          canvas.height = height;
          const ctx = canvas.getContext("2d");
          ctx.drawImage(img, 0, 0, width, height);
          resolve(canvas.toDataURL("image/webp", 0.8));
        };
      };
    });
    compressedImages.push(compressedBase64);
  }
  setProjectForm((prev) => ({
    ...prev,
    images: [...prev.images, ...compressedImages],
  }));
};
```

4.3.7 Компонент публичной карты

Для исключения дублирования кода, функционал интерактивной карты вынесен в переиспользуемый компонент BaseMap. Этот компонент выступает в роли универсальной обертки, которая применяется как в публичной части сайта для демонстрации портфолио, так и в калькуляторе для прокладки маршрута.

PublicMap автоматически настраивает карту так, чтобы на ней были видны все маркеры проектов. Тайловый слой использует OpenStreetMap. Тайловый слой подгружается по URL-шаблону, содержащему параметры зума и координат тайла в стандартной системе XYZ[7].

Маркеры проектов отображаются на карте с использованием компонента Marker. Для каждого проекта создается маркер в указанных координатах. Для каждого маркера предусмотрено всплывающее окно (Popup), отображающее краткую информацию о проекте при клике на маркер. Фрагмент кода, реализующий инициализацию карты, представлен ниже.

```
export const customIcon = new L.Icon({
  iconUrl: "https://unpkg.com/leaflet@1.9.4/dist/images/marker-
icon.png",
  iconRetinaUrl:
"https://unpkg.com/leaflet@1.9.4/dist/images/marker-icon-2x.png",
  shadowUrl: "https://unpkg.com/leaflet@1.9.4/dist/images/marker-
shadow.png",
  iconSize: [25, 41],
  iconAnchor: [12, 41],
  popupAnchor: [1, -34],
  shadowSize: [41, 41],
});

// Сам базовый компонент карты
const BaseMap = ({ center, zoom, className, children, ...props }) =>
{
  return (
    <MapContainer center={center} zoom={zoom} className={className}
    {...props}>
      <TileLayer
        attribution='&copy; <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>'
        url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
      />
      {children}
    </MapContainer>
  );
};

export default BaseMap;
```

4.3.8 Геокодирование в административной панели

Для удобства добавления новых объектов портфолио в административной панели реализован двусторонний механизм взаимодействия с картографическим

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		39

сервисом. Администратор имеет возможность как находить координаты по текстовому адресу (прямое геокодирование), так и автоматически получать адрес путем визуального выбора точки на интерактивной карте (обратное геокодирование).

При выполнении прямого геокодирования сначала блокируется стандартная перезагрузка формы с помощью `e.preventDefault()` и проверяется наличие введенного текста. Для корректной передачи кириллических символов и пробелов в URL-строке используется встроенный метод `encodeURIComponent`. Затем формируется GET-запрос к API Nominatim с параметром `format=json`. Из полученного массива извлекается первый, наиболее точный результат. Поскольку API возвращает координаты в строковом формате, применяется функция `parseFloat` для строгого приведения значений к числу с плавающей точкой. Далее полученные координаты передаются в функцию `setPosition`, которая обновляет состояние компонента, автоматически перемещая маркер в указанную точку и центрируя карту. [8]. Фрагмент кода, реализующий прямое геокодирование, представлен ниже.

```
const handleSearchAddress = async (e) => {
  e.preventDefault();
  if (!projectForm.address.trim()) return;

  try {
    const response = await fetch(
      `https://nominatim.openstreetmap.org/search?format=json&q=${encodeURIComponent(projectForm.address)}`
    );
    const data = await response.json();

    if (data && data.length > 0) {
      const lat = parseFloat(data[0].lat);
      const lon = parseFloat(data[0].lon);
      setPosition({ lat, lng: lon });
    } else {
      alert("Координаты не найдены. Укажите адрес точнее или кликните по карте вручную.");
    }
  } catch (error) {
    console.error("Ошибка при поиске адреса:", error);
  }
}
```

```
};
```

В основе компонента `LocationMarker` лежит специализированный хук `useMapEvents` из библиотеки `react-leaflet`, позволяющий перехватывать события карты в контексте React-приложения. При возникновении события `click` из объекта события извлекаются широта и долгота (`e.latlng`), после чего маркер немедленно перемещается в новую позицию вызовом функции `setPosition`. Далее формируется запрос к `/reverse API Nominatim` с обязательным параметром `addressdetails=1`, который требует от сервиса разбить итоговый адрес на атомарные составляющие (город, улица, номер дома) вместо возврата единой неструктурированной строки.[8] На клиентской стороне реализован алгоритм сборки адреса. Учитывая возможную разрозненность данных `OpenStreetMap`, последовательно проверяется наличие населенного пункта по трем ключам – `city`, `town` или `village`. Формируется базовая связка «Улица + Номер дома». Если эти данные отсутствуют в базе `OSM` для выбранной точки, берется первая смысловую часть из поля `display_name`. Сформированная строка с префиксом «г.» передается в родительский компонент через `callback-функцию onAddressFound` и автоматически отображается в текстовом поле формы. Завершается работа компонента отрисовкой графического маркера `<Marker>` в выбранной точке. Фрагмент кода, реализующий обратное геокодирование, представлен ниже.

```
const LocationMarker = ({ position, setPosition, onAddressFound })
=> {
  useMapEvents({
    async click(e) {
      const { lat, lng } = e.latlng;
      setPosition(e.latlng);

      try {
        const response = await fetch(
          `https://nominatim.openstreetmap.org/reverse?format=json&lat=${lat}&lon=${lng}&zoom=18&addressdetails=1`
        );
        const data = await response.json();

        if (data && data.address) {
          const road = data.address.road || "";
          const houseNumber = data.address.house_number || "";
```

```

        const city = data.address.city || data.address.town ||
data.address.village || "";

        let finalAddress = `${road} ${houseNumber}`.trim();
        if (!finalAddress) finalAddress =
data.display_name.split(",")[0];

        const prefix = city ? `г. ${city}, ` : "";
        onAddressFound(`${prefix}${finalAddress}`);
    }
} catch (error) {
    console.error(error);
}
},
});
return position === null ? null : <Marker position={position}
icon={customIcon} />;
};

```

					ПГУ 1.090301.49.001 ПЗ	Лист
						42
Изм	Лист	№ докум.	Подпись	Дата		

5 Тестирование

Для того чтобы качественно протестировать веб-приложение был сформирован план тестирования, который содержит в себе пять основных этапов. План тестирования показан в таблице 3.

Таблица 3 – План тестирования

Описание проверки	Ожидаемый результат
Авторизация и аутентификация	Успешный вход по валидным данным, доступ к панели только при наличии JWT
Тестирование CRUD проектов и загрузки изображений	Корректное выполнение операций создания, редактирования, удаления проектов, сжатие изображений
Работа геоинформационного модуля	Отображение карты с маркерами проектов, фильтрация по категориям, геокодирование адресов (прямое и обратное)
Расчет стоимости дизайн-проекта и логистики	Динамический пересчет стоимости при изменении параметров, получение расстояния и маршрута через OSRM, корректный расчет транспортных расходов
Форма обратной связи и хранение заявок	Валидация полей, успешная отправка заявки, отображение заявки в панели администратора

5.1 Тестирование авторизации и аутентификации

На первом этапе была проведена проверка входа в административную панель. При неверно введенном логине или пароле пользователь получает сообщение об ошибке. На рисунке 12 представлено окно входа, содержащее поля для ввода логина и пароля, а также кнопку «Войти», при неправильно введенном пароле.

Вход

Неверный логин или пароль

Логин

Irina

Пароль

Войти

Рисунок 12 – Сообщение об ошибке при вводе данных

Если данные были введены корректно, администратор получает доступ к панели управления, которая изображена на рисунке 13.

Администрирование

Управление проектами

Заявки клиентов

Выйти из панели

Список проектов + Добавить новый

Категории:

Все Квартира под ключ Гостиная Спальня Кухня Ванная Другое

С: DD-MM-YYYY По: DD-MM-YYYY Сбросить

Карта Пензы

Сдержанная роскошь 18.03.2024 Изменить Удалить

Квартира под ключ - Москва ул. Крестовый 2

Интерьер этого дома - история о балансе. Мы совместили тепло натурального дерева... Читать далее

Лаконичность и стиль 29.01.2024 Изменить Удалить

Квартира под ключ - Москва ул. Крестовый 2

Интерьер квартиры в ЖК «Символ» - это история о том, как создать пространство... Читать далее

Две истории одного детства 09.12.2023 Изменить Удалить

Другое - Саранск ул. Кирова 9

Мы обустраивали две детские комнаты для двух детей с разными характерами. Нам было... Читать далее

Рисунок 13 – Панель администратора

При входе генерируется JWT-токен и в локальном хранилище браузера (localStorage) появляется запись adminToken.(рисунок 14)

http://localhost:5173	
Key	Value
adminToken	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8v

Рисунок 14 – Запись adminToken

5.2 Тестирование CRUD проектов и загрузки изображений

На втором этапе проверяется управление проектами. Чтобы добавить новый проект необходимо перейти на вкладку «+ Добавить новый» и заполнить необходимые поля. После нажать «Опубликовать в портфолио». На рисунке 15 показана заполненная форма.

Название проекта: Тестовый проект

Категория: Гостиная

Бюджет (\$): 300000

Дата сдачи проекта: 14.05.2026

Где отображать проект?: Везде (Карта + Галерея)

Описание проекта: Описание проекта

Фотографии объекта: Выберите файлы. Выбрано файлов: 2

Физический адрес и координаты: Пенза рахманиова 45. Найти на карте

Рисунок 15 – Заполненная форма добавления проекта

При корректно введенных данных появляется окно об успехе, представленное на рисунке 16

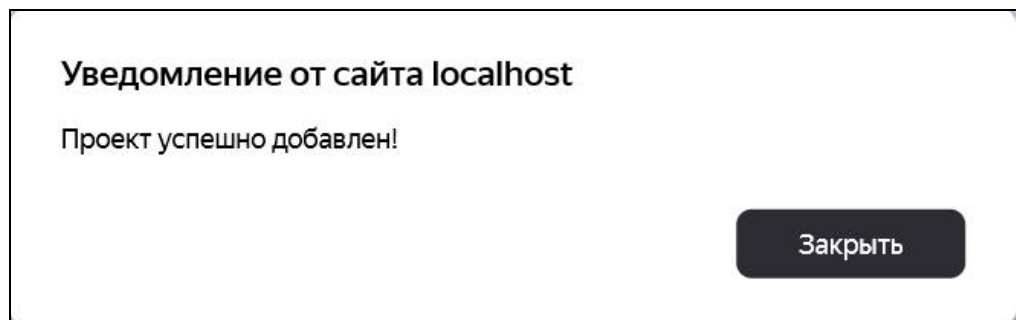


Рисунок 16 – Уведомление при успешной публикации

Если обязательное поле проекта перед отправкой не было заполнено, то после нажатия на «Опубликовать в портфолио» появится сообщение-напоминание, как на рисунке 17

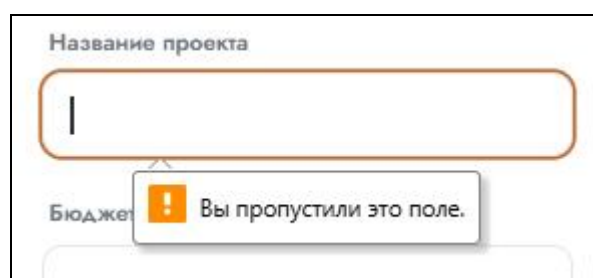


Рисунок 17 – Сообщение о пропущенном поле

Чтобы отредактировать проект необходимо в списке проектов нажать на кнопку «Изменить». Список проектов представлен на рисунке 18

Откроется форма редактирования с предзаполненными данными. После внесения изменений и нажатия «Обновить данные» появляется подтверждение обновления (рисунок 19).

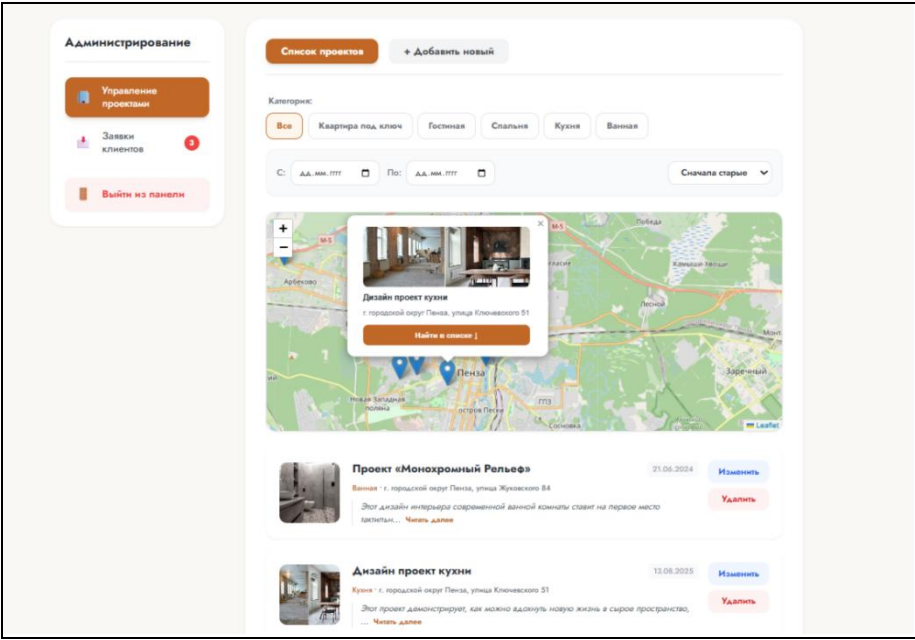


Рисунок 18 – Список проектов

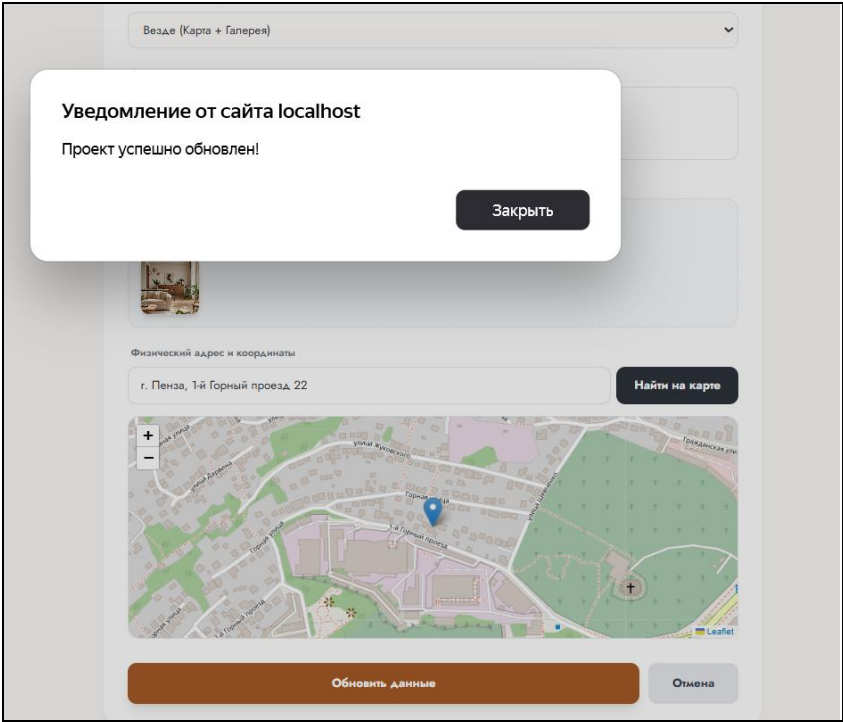


Рисунок 19 – Уведомление об успешном обновлении

Если администратор решает удалить проект, то в списке необходимо нажать кнопку «Удалить». Появится окно подтверждения с вопросом «Вы

уверены, что хотите удалить этот проект?» (рисунок 20). При подтверждении проект исчезает из списка и с карты.

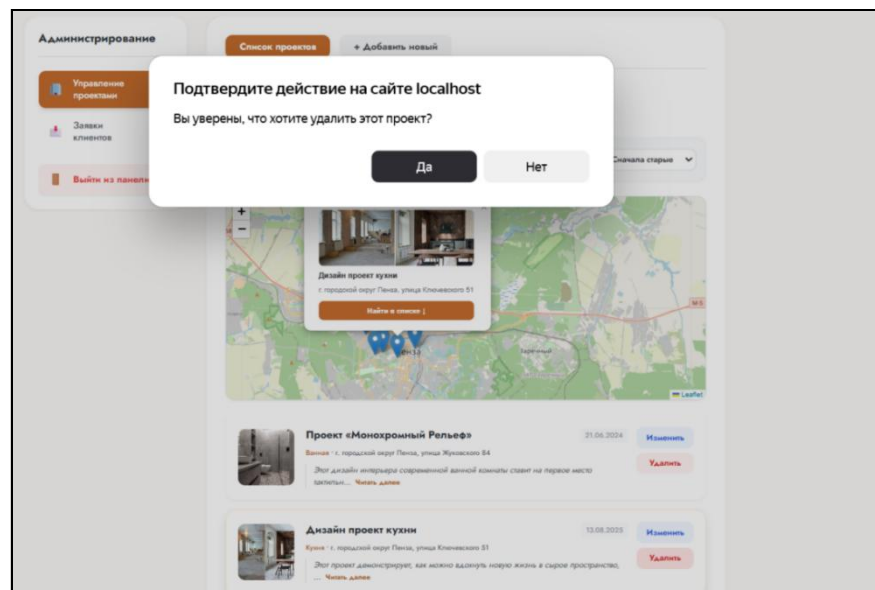


Рисунок 20 – Запрос подтверждения удаления

Для проверки механизма сжатия загружено изображение размером 2,21МБ (рисунок 21). В инструментах разработчика (вкладка Network) при отправке формы обнаружен POST-запрос к /api/projects. В заголовке запроса поле Content-Length имеет значение 548565 байт (примерно 549КБ) (рисунок 22).

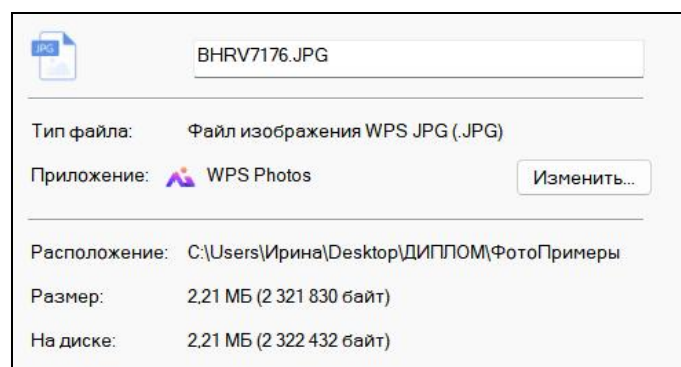
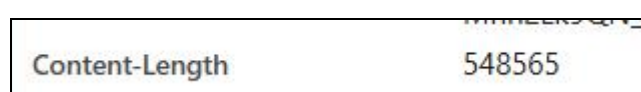


Рисунок 21 – Исходный размер изображения



5.3 Тестирование работы геоинформационного модуля

На третьем этапе тестируется геоинформационный модуль. Когда посетитель переходит на страницу с проектами и выбирает режим «Объекты на карте», карта отображается с центром на городе Пенза. На рисунке 23 на карте отображаются маркеры всех проектов.

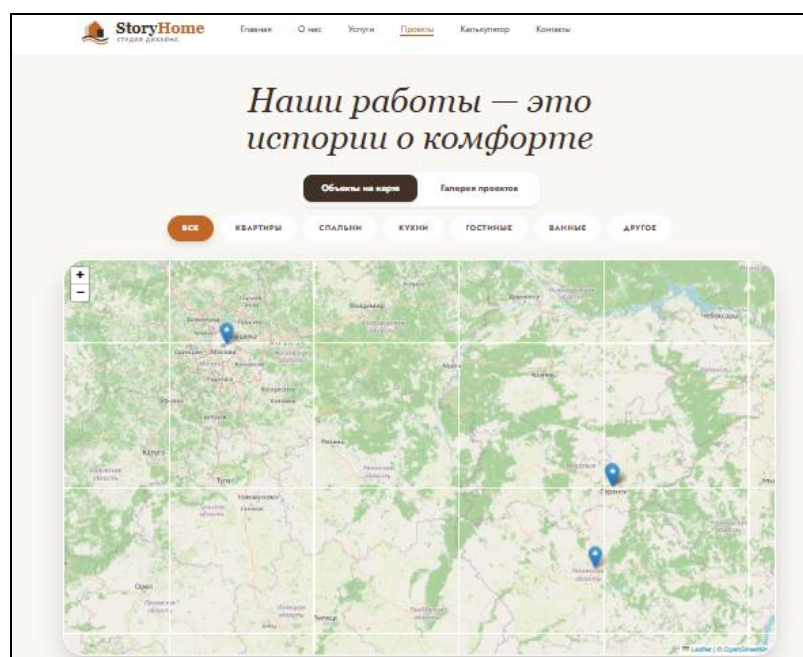


Рисунок 23 – Карта всех проектов

На той же странице если нажать на кнопку «Спальня» карта перерисовывается, отображая только маркеры спален (рисунок 24).

Когда пользователь кликает по любому маркеру открывается всплывающее окно, содержащее первое изображение проекта, его название и кнопку «Подробнее» (рисунок 25). При нажатии на кнопку выполняется переход на страницу детального просмотра проекта (рисунок 26).

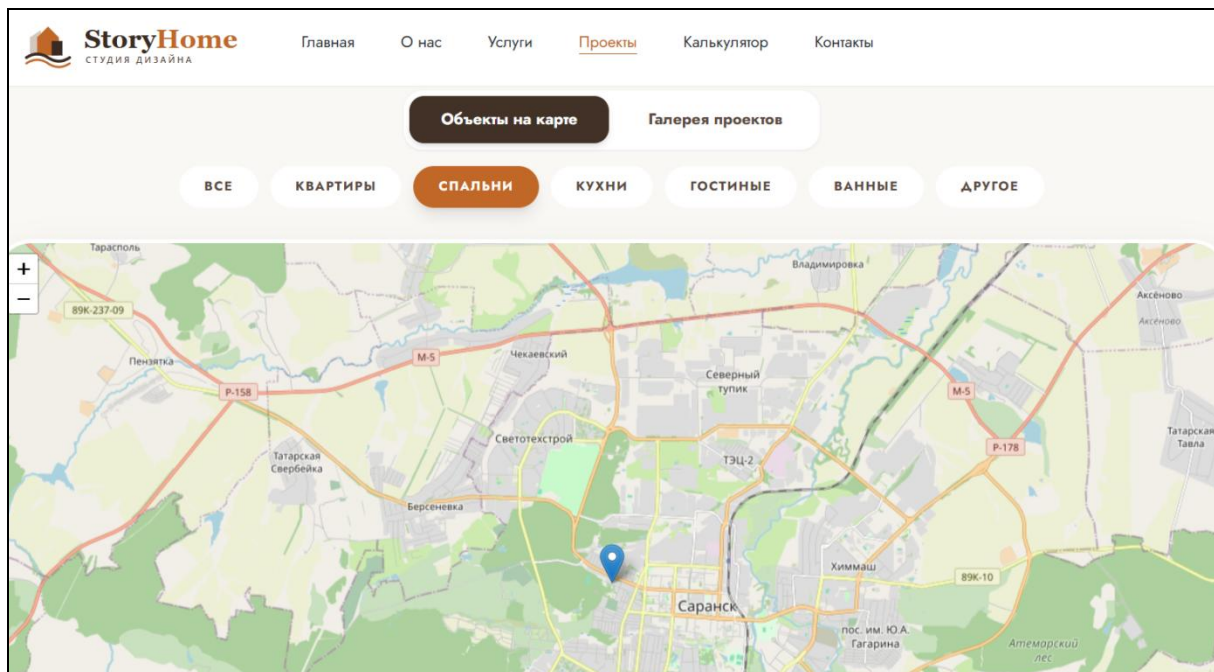


Рисунок 24 – Отображение выбранной категории проектов на карте

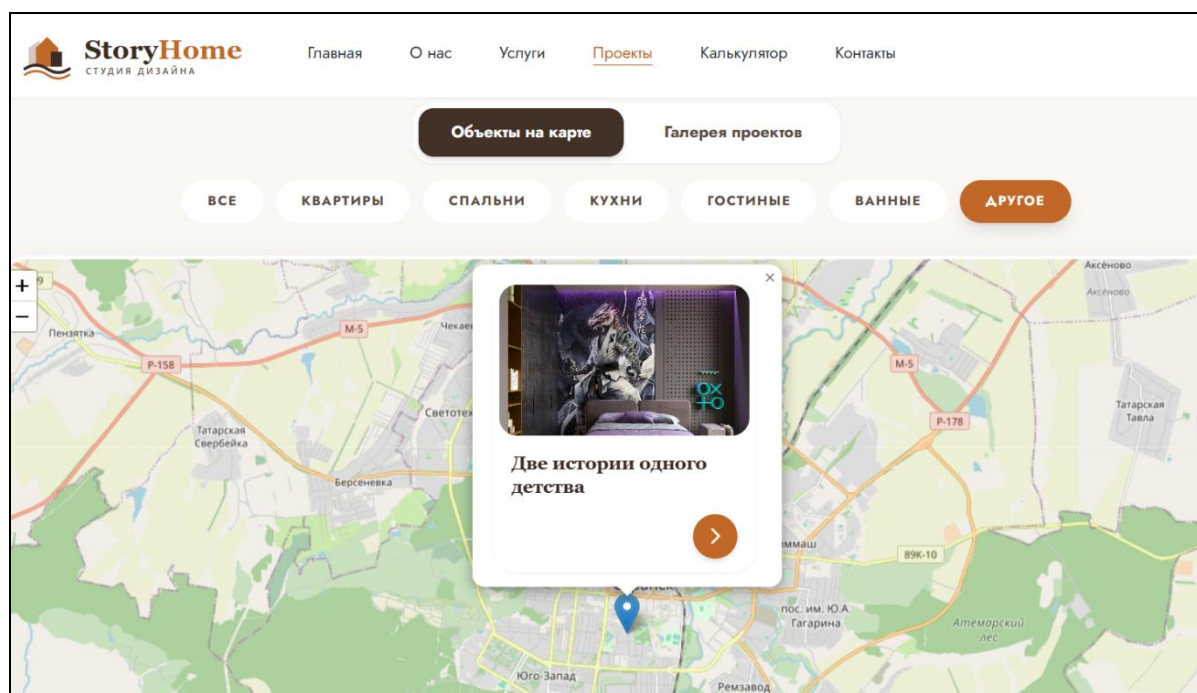


Рисунок 25 –Всплывающее окно проекта

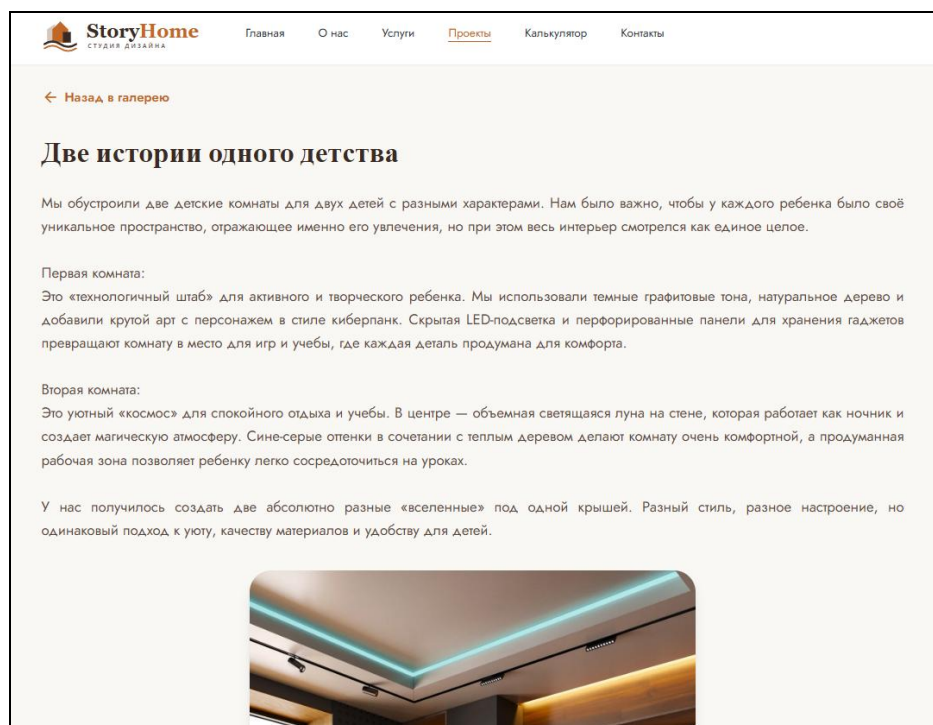


Рисунок 26 – Страница детального просмотра проекта

5.4 Тестирование расчета стоимости дизайн-проекта и логистики

На четвертом этапе тестируется работа калькулятора стоимости. Когда посетитель переходит на страницу калькулятора по умолчанию заданы площадь 50м, тип объекта «Новостройка», пакет «Базовый». Итоговая стоимость составляет 85000 руб. При изменении любых критериев или добавления необязательных услуг сумма мгновенно пересчитывается. (рисунок 27)

Для тарифов «Комплексный» и «VIP» пользователю доступна опция расчета логистики. При вводе адреса объекта и нажатии кнопки «Рассчитать логистику» на экране отображается расстояние и построенная линия маршрута. Транспортные расходы добавляются к итоговой стоимости (рисунок 28).

При недоступности OSRM активируется резервный алгоритм – расчет с использованием формулы гаверсина с поправочным коэффициентом 1,2 (рисунок 29).

Рисунок 27 – Страница калькулятора с рассчитанной стоимостью

Рисунок 28 –Добавление транспортных расходов

Рисунок 29 – Расчет маршрута по формуле

5.5 Тестирование формы обратной связи и просмотра заявок

Завершающий этап проверяет форму обратной связи. При переходе на страницу контактов из калькулятора по кнопке «Оставить заявку» в поле «Сообщение» автоматически подставляется текст, содержащий выбранные параметры и рассчитанную стоимость (рисунок 30).

Рисунок 30 – Страница формы обратной связи

При незаполнении какого либо поля, появляется сообщение с предупреждением, как на рисунке 31.

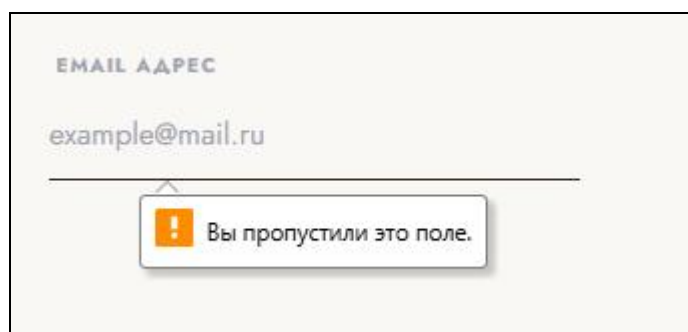


Рисунок 31 – Предупреждение при незаполненном поле

Если все данные заполнены, при нажатии кнопки «Отправить», появляется окно «Успешно!»(рисунок 32)

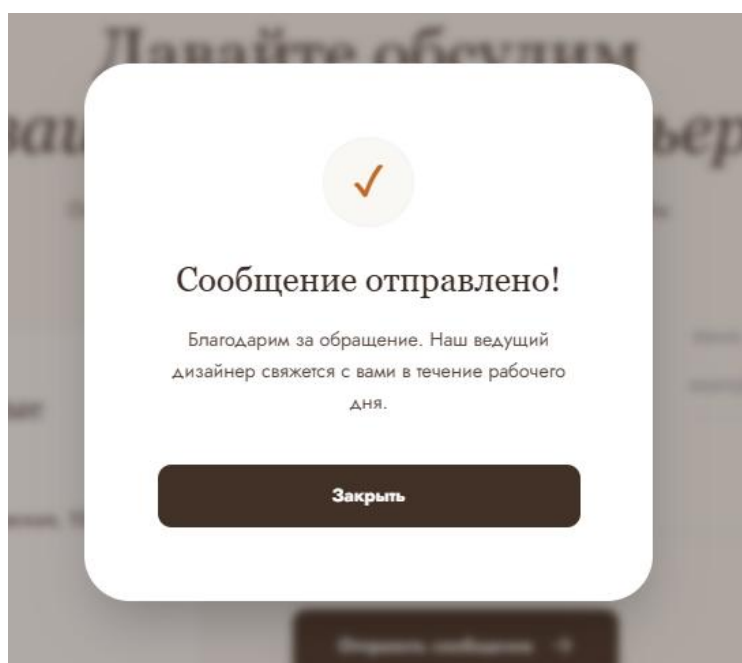


Рисунок 32 – Окно, уведомляющее об успешном отправлении заявки

В панели администратора в разделе «Заявки клиентов» отображается список заявок. Новые заявки подсвечиваются желтым цветом. Если администратор нажимает кнопку «Пометить как обработанную», кнопка

заменяется на надпись «Готово». На рисунке 33 показан список заявок до и после обработки.

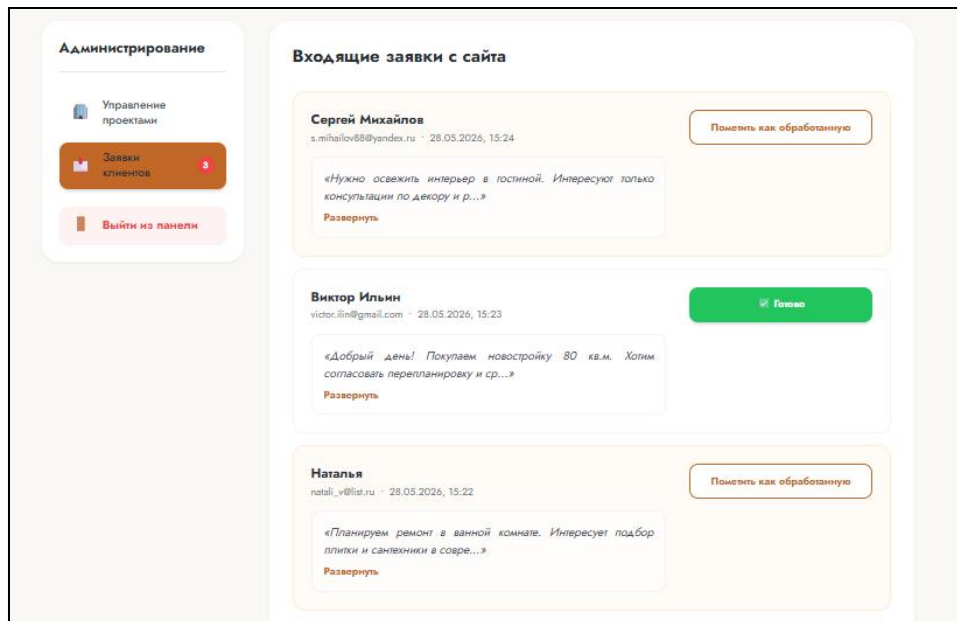


Рисунок 33 – Список всех заявок на панели администратора

Проведенное функциональное тестирование веб-приложения охватило все ключевые модули системы.

Все тестовые сценарии выполнены без ошибок. Система корректно обрабатывает как штатные, так и пограничные ситуации (неверные учетные данные, пустые обязательные поля, недоступность внешнего сервиса маршрутизации).

6 Безопасность жизнедеятельности

При разработке и последующей эксплуатации веб-сайта студии дизайна интерьеров необходимо учитывать требования охраны труда и техники безопасности, предъявляемые к работе с компьютерной техникой. Пользователями системы могут быть как сотрудники студии, работающие за персональными компьютерами, так и посетители сайта, использующие мобильные устройства. В данном разделе рассмотрены основные факторы, влияющие на безопасность и здоровье человека при взаимодействии с вычислительной техникой, а также меры по минимизации негативного воздействия.

6.1 Правила и техника безопасности при работе с ПК

Деятельность, связанная с эксплуатацией персональных компьютеров, сопровождается воздействием ряда физических и эргономических факторов, которые при отсутствии должного контроля могут привести к ухудшению здоровья пользователя. К наиболее значимым из них относятся зрительное утомление, статические нагрузки на мышцы шеи, спины и верхних конечностей, а также риск поражения электрическим током при нарушении правил подключения оборудования.

Рабочее место при использовании ПК должно быть организовано таким образом, чтобы обеспечить оптимальную позу пользователя. Стул должен иметь регулируемую высоту и спинку, позволяющую фиксировать поясничный изгиб. Ноги при работе должны полностью стоять на полу, а руки сгибаться в локтях под углом 90-100 градусов, предплечья свободно лежать на столе. Монитор устанавливается на расстоянии вытянутой руки, при этом верхняя граница экрана должна находиться на уровне глаз или чуть ниже. Такое положение снижает нагрузку на шейный отдел позвоночника и предотвращает развитие хронических болей в спине.

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		56

Искусственное освещение в помещении не должно создавать прямых бликов на экране монитора. Оптимальным считается расположение рабочего стола перпендикулярно оконным проемам, что минимизирует попадание солнечных лучей на дисплей. При недостатке естественного света используются настольные светильники с регулируемым направлением луча, при этом источник света не должен светить непосредственно в глаза. Для снижения зрительного утомления рекомендуется каждые 20-30 минут переводить взгляд на удаленные объекты на 10-15 секунд.

Питание системного блока, монитора и периферийных устройств осуществляется от сети переменного тока напряжением 220 В. Для предотвращения поражения электрическим током все элементы вычислительной техники должны подключаться к розеткам с заземляющим контактом. Запрещается использовать удлинители с поврежденной изоляцией, самостоятельно вскрывать блоки питания или эксплуатировать оборудование с видимыми дефектами кабелей. В случае обнаружения запаха гари, нехарактерного шума или искрения необходимо немедленно отключить питание и обратиться к обслуживающему персоналу.

Продолжительность непрерывной работы за компьютером не должна превышать 45-60 минут. В течение рабочего дня суммарное время регламентированных перерывов составляет от 30 до 90 минут в зависимости от интенсивности труда. Во время перерывов рекомендуется выполнять простые физические упражнения, что способствует восстановлению кровообращения и снижению мышечного напряжения. Соблюдение указанных норм позволяет существенно снизить риск развития профессиональных заболеваний, связанных с длительной работой за компьютером.

6.2 Правила и техника безопасности при работе с мобильными устройствами

Разработанный веб-сайт адаптирован для корректного отображения на экранах мобильных устройств, поэтому часть пользователей будет взаимодействовать с системой через смартфоны и планшеты. Несмотря на то, что мобильные устройства относятся к низковольтному оборудованию (напряжение питания не превышает 5 В) и не представляют угрозы поражения электрическим током, при их использовании следует соблюдать определенные меры предосторожности.

В первую очередь необходимо избегать механических повреждений устройства: падений, ударов, изгибов, а также попадания влаги и экстремальных температур. Использование смартфона с треснувшим экраном или поврежденным корпусом может привести к травмам рук и лица. Следует применять только сертифицированные зарядные устройства и кабели, соответствующие параметрам конкретной модели телефона, так как неоригинальные аксессуары способны вызвать перегрев аккумулятора и его возгорание.

Длительное использование мобильных устройств может оказывать нагрузку на зрение из-за небольшого размера экрана и высокой яркости подсветки. Рекомендуется делать перерывы каждые 20-30 минут, а также регулировать яркость дисплея в зависимости от внешнего освещения. Кроме того, чрезмерное увлечение смартфоном способно приводить к нарушению осанки, снижению концентрации внимания и нарушениям сна. Пользователям следует ограничивать время работы с мобильными устройствами перед сном и использовать функцию «ночной режим» для уменьшения воздействия синего света.

7 Оценка экономической эффективности

Внедрение разработанного клиент-серверного веб-приложения в деятельность студии «Дизайн Земли» (ИП Новокрещенова О.В.) направлено на достижение двух ключевых коммерческих целей: уменьшение операционных расходов путем автоматизации регулярных процедур вычисления стоимости услуг и рост объемов продаж за счет увеличения конверсии посетителей сайта в реальных заказчиков.

Для оценки финансовой обоснованности проекта используется подход, разделяющий прямой и косвенный экономический эффекты. Под прямым эффектом понимается экономия рабочего времени персонала вследствие сокращения трудоемкости при первичной обработке поступающих заявок. Косвенный эффект представляет собой дополнительную прибыль, образующуюся благодаря увеличению числа заключенных договоров после запуска интерактивной карты и онлайн-калькулятора.

Капитальные вложения (K) складываются из расходов на архитектурное проектирование, реализацию серверной части (REST API на платформе ASP.NET Core), клиентского интерфейса (React), подключения геоинформационных сервисов (OSRM, Nominatim) и создание панели администратора.

По результатам обзора рыночных предложений усредненная стоимость разработки программного комплекса, аналогичного по функциональности, оценивается в $K = 150\,000$ рублей. Эта величина принимается за основу при расчете стартовых вложений при анализе прибыльности проекта.

7.1 Расчет прямого экономического эффекта

До внедрения системы процедура расчета логистических затрат и формирования первичной сметы выполнялась менеджером студии в ручном режиме. Процесс включал сбор адресных данных, использование сторонних картографических сервисов для определения расстояния и применение

коэффициентов тарифа. Интеграция модуля автоматического расчета на базе внешних API позволила исключить данный этап из прямых обязанностей сотрудника.

Для количественной оценки изменения производительности труда применялся метод сравнения затрат рабочего времени. Средняя заработная плата специалиста по работе с клиентами (по данным регионального рынка труда г. Пензы на расчетный период) составляет $ЗП = 65\ 000$ руб. в месяц при нормативной продолжительности рабочего времени 160 часов. Часовая тарифная ставка ($C_ч$) равняется 406,25 руб./ч.

Затраты времени на ручную обработку одного запроса составляли $t_1 = 0,5$ ч. После внедрения системы задача менеджера сводится к обработке сформированной заявки в административной панели, что занимает $t_2 = 0,16$ ч. Абсолютное снижение трудоемкости одной операции составляет $\Delta t = 0,34$ ч. При среднем потоке в 20 обращений ежемесячно, высвобожденное рабочее время рассчитывается следующим образом:

$$T_{мес} = N \cdot \Delta t = 20 \cdot 0,34 = 6,8ч$$

где $T_{мес}$ – высвобожденное рабочее время;

N – количество обращений ежемесячно;

Δt – разница затрачиваемого времени.

Снижение затрат предприятия на оплату труда за один месяц вычисляется по формуле:

$$S_{мес} = T_{мес} \cdot C_ч = 6,8 \cdot 406,25 = 2762,5 \text{ руб.}$$

где $S_{мес}$ – снижение затрат на оплату труда за один месяц;

$T_{мес}$ – высвобожденное рабочее время;

$C_ч$ – часовая тарифная ставка.

В соответствии со статьей 425 Налогового кодекса Российской Федерации, работодатель производит обязательные страховые отчисления в размере 30% от

фонда оплаты труда. Следовательно, годовая экономия предприятия рассчитывается с учетом налогового коэффициента 1,3:

$$\mathcal{E}_n = S_{\text{мес}} \cdot 12 \cdot 1,3 = 2762,5 \cdot 12 \cdot 1,3 = 43095 \text{руб./год.}$$

где $\mathcal{E}_n$ – годовая экономия;

Высвобожденное рабочее время не ведет к сокращению штата или уменьшению оклада сотрудника, а перераспределяется на выполнение профильных задач, непосредственно связанных с генерацией прибыли (проведение углубленных консультаций, сопровождение текущих сделок).

7.2 Расчет косвенного экономического эффекта

Наличие на официальном сайте студии интерактивной карты реализованных проектов и прозрачной системы ценообразования снижает процент отказов потенциальных заказчиков. Повышение уровня доверия и улучшение пользовательского опыта прямо влияют на коэффициент конверсии.

До ввода приложения в эксплуатацию средний показатель успешных сделок составлял 4 договора в месяц. Ожидаемый прирост клиентской базы за счет нового функционала составляет 1 дополнительный договор каждые 2 месяца (6 проектов в год). При оценке средней чистой прибыли с одного базового дизайн-проекта на уровне 25 000 руб., годовой прирост прибыли составит:

$$\mathcal{E}_{np} = \Delta K \cdot P = 6 \cdot 25000 = 150000 \text{руб./год}$$

где $\mathcal{E}_{np}$ – годовой прирост прибыли;

ΔK – прирост клиентской базы в год;

P – средняя прибыль с одного дизайн-проекта.

7.3 Расчет итоговых показателей экономической эффективности

Суммарный годовой экономический эффект от внедрения разрабатываемой системы определяется по формуле:

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		61

$$\mathcal{E}_{год} = \mathcal{E}_n + \mathcal{E}_{np} = 43095 + 150000 = 193095 \text{ руб.}$$

где $\mathcal{E}_{год}$ – суммарная годовая прибыль.

Ключевым показателем является срок окупаемости вложений, рассчитанный по формуле:

$$T_{ок} = \frac{K}{\mathcal{E}_{год}} = \frac{150000}{193095} \approx 0,78 \text{ года}$$

где $T_{ок}$ – срок окупаемости;

K – начальные вложения.

Расчеты показали, что разработка и внедрение клиент-серверного веб-приложения для дизайн-студии является экономически обоснованным решением. Расчетный срок окупаемости системы составляет 0,78 года (около 9,3 месяцев). Проект полностью окупает затраты на свою разработку до истечения первого года эксплуатации. В дальнейшем программный продукт начнет приносить заказчику чистую экономическую выгоду в размере более 190 тысяч рублей ежегодно.

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		62

Заключение

В ходе выпускной квалификационной работы была достигнута поставленная цель: разработано и внедрено веб-приложение для студии дизайна интерьеров, которое объединяет функции интерактивной демонстрации портфолио, автоматизированного расчета стоимости услуг и администрирования контента.

Для достижения указанной цели были последовательно решены все запланированные задачи:

- проведен анализ предметной области и существующих веб-решений;
- обоснован технологический стек для разработки;
- спроектирована клиент-серверная архитектура приложения;
- разработана клиентская часть с интегрированными геоинформационными сервисами и серверная часть веб-сайта;
- реализована панель администратора для управления проектами и обработки поступивших заявок с использованием механизмов аутентификации;
- выполнено функциональное тестирование;
- оценена экономическая эффективность внедрения веб-сайта.

Разработанный веб-ресурс полностью соответствует функциональным требованиям заказчика. Проект представляет собой практический инструмент для повышения конкурентоспособности студии, обеспечивающий автоматизацию взаимодействия с клиентами и эффективное управление портфолио. Все поставленные задачи выполнены в полном объеме, а цель работы достигнута.

Список использованных источников

1. Entity Framework Core [электронный ресурс]. – URL: <https://learn.microsoft.com/ru-ru/ef/core/> (дата обращения: 24.04.2026).
2. ASP.NET Core [электронный ресурс]. – URL: <https://learn.microsoft.com/ru-ru/aspnet/core/> (дата обращения: 17.04.2026).
3. JWT – стандарт RFC 7519 [электронный ресурс]. – URL: <https://datatracker.ietf.org/doc/html/rfc7519> (дата обращения: 08.05.2026).
4. Tailwind CSS [электронный ресурс]. – URL: <https://tailwindcss.com/docs> (дата обращения 14.05.2026).
5. React [электронный ресурс]. – URL: <https://react.dev/> (дата обращения: 17.04.2026).
6. OSRM (Open Source Routing Machine) [электронный ресурс]. – URL: <http://project-osrm.org/docs/> (дата обращения: 05.05.2026).
7. Leaflet [электронный ресурс]. – URL: <https://leafletjs.com/reference.html> (дата обращения: 01.05.2026).
8. Nominatim [электронный ресурс]. – URL: <https://nominatim.org/release-docs/latest/api/Overview/> (дата обращения: 05.05.2026).

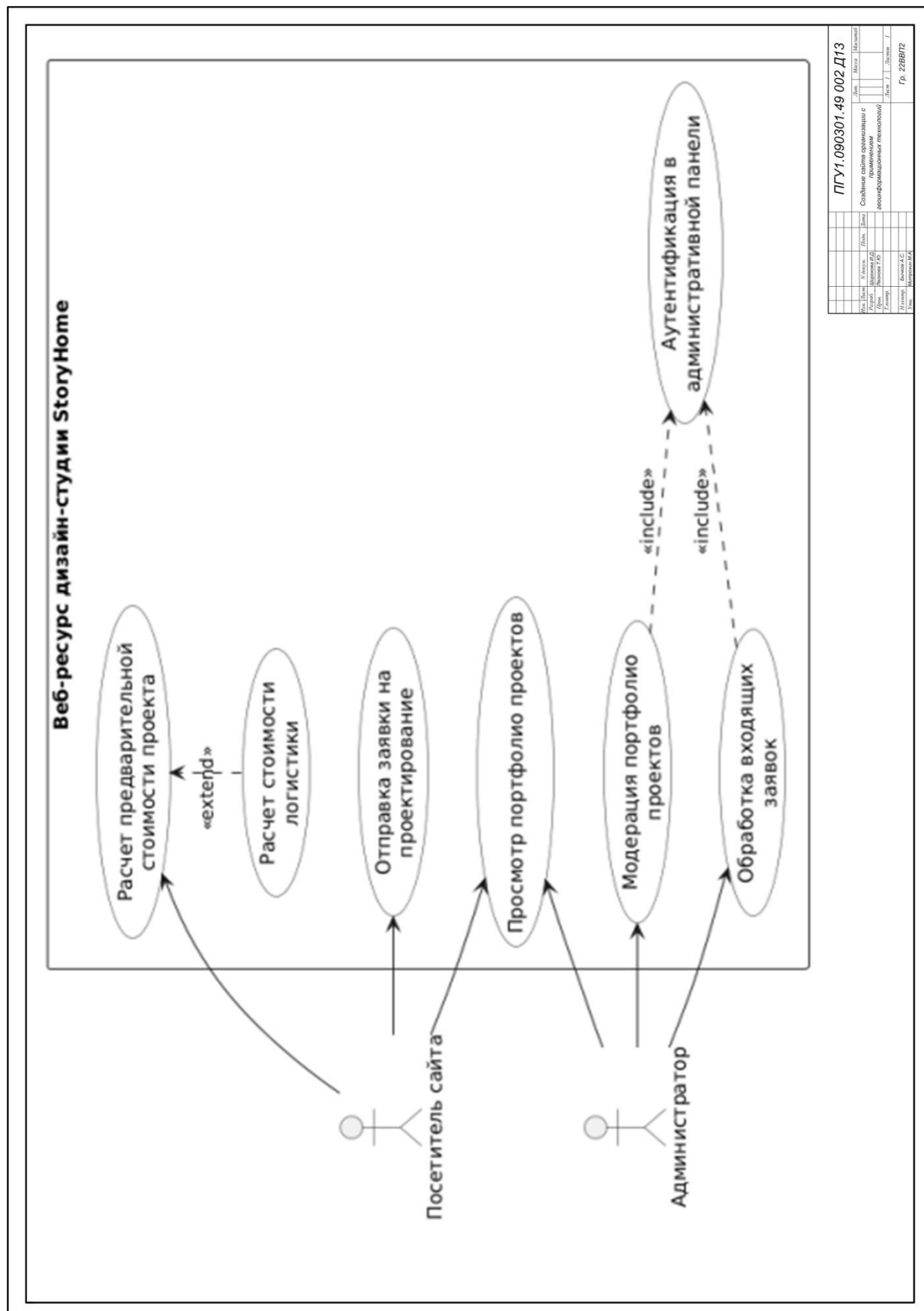
Приложение А

(обязательное)

Создание сайта организации с применением геоинформационных технологий

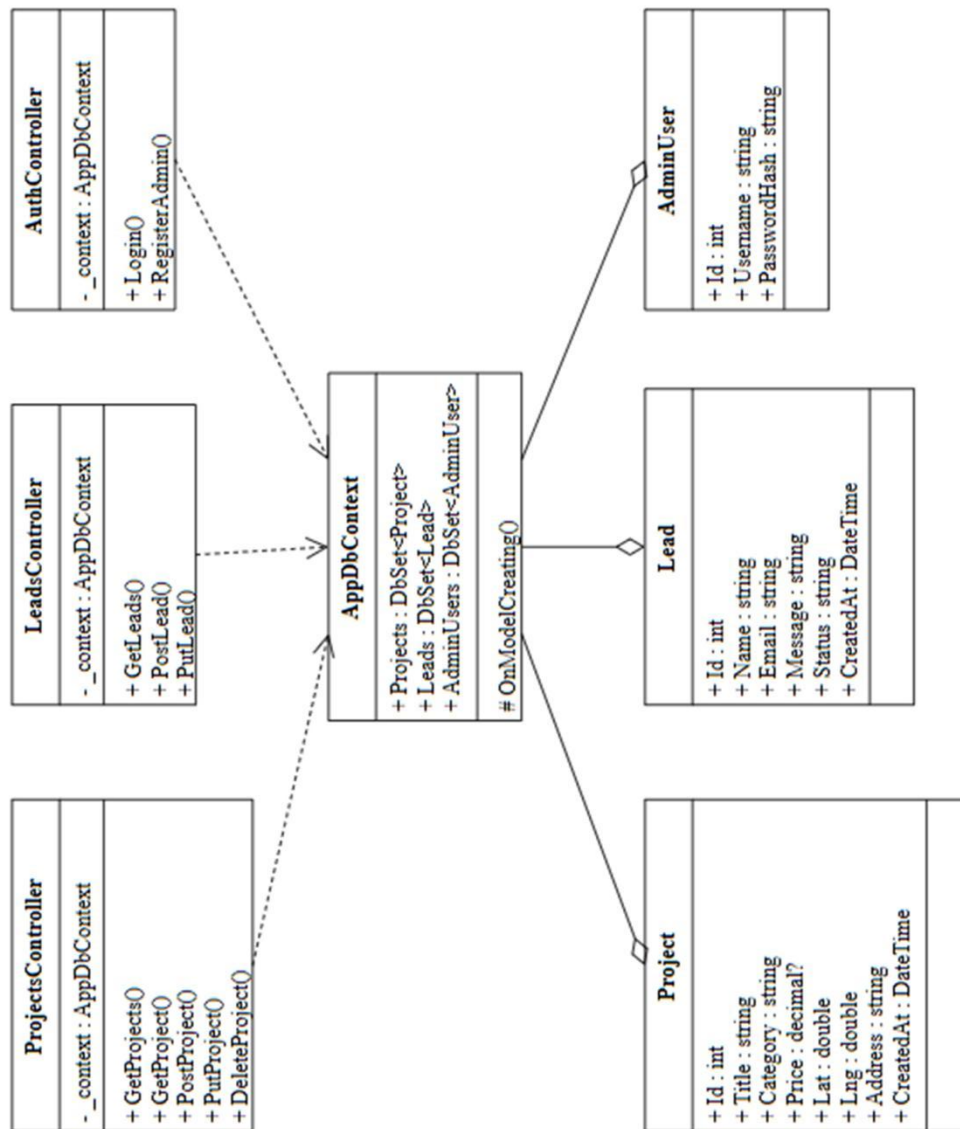
UML-диаграммы

					ПГУ 1.090301.49.001 ПЗ	Лист
						65
Изм	Лист	№ докум.	Подпись	Дата		

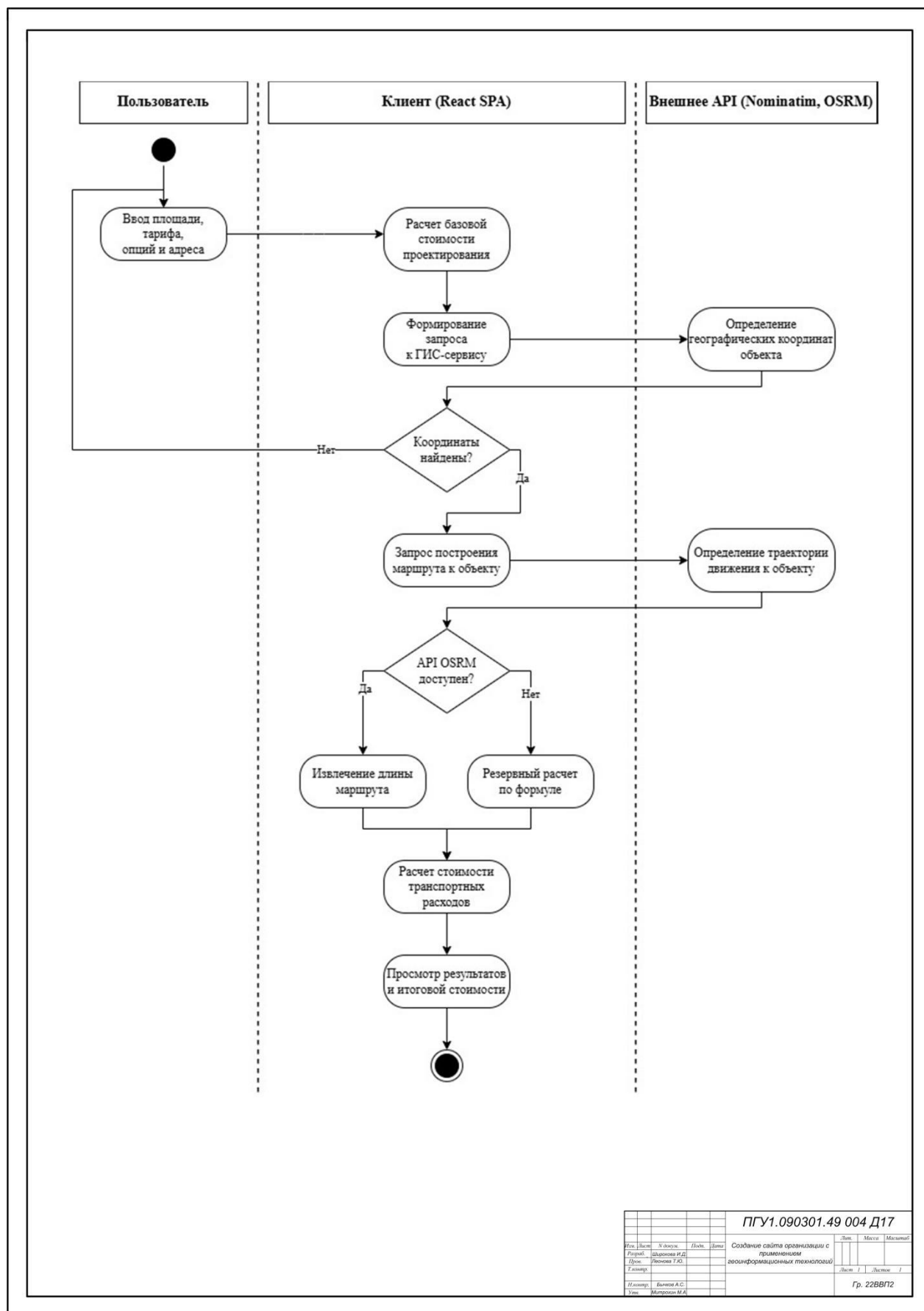


ПГУ 1.090301.49 002 Д13									
Дата	Лист	Итого	Лист	Итого	Лист	Итого	Лист	Итого	Лист
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9
10	10	10	10	10	10	10	10	10	10

UML-диаграмма классов

[illegible]

UML-диаграмма деятельности расчета в калькуляторе



ПГУ1.090301.49 004 Д17									
Изм.	Дет.	№ докум.	Подп.	Дата	Создание сайта организации с применением геоинформационных технологий				
Разраб.	Иванова И.И.				Дет.	Место	Масштаб		
Про.	Петрова Т.Ю.				Дет.	Дет.	Дет.	Дет.	Дет.
Т.эксп.									
Исполн.	Васильев А.С.				Гр. 22ВВП2				
Упр.	Михайлов И.И.								

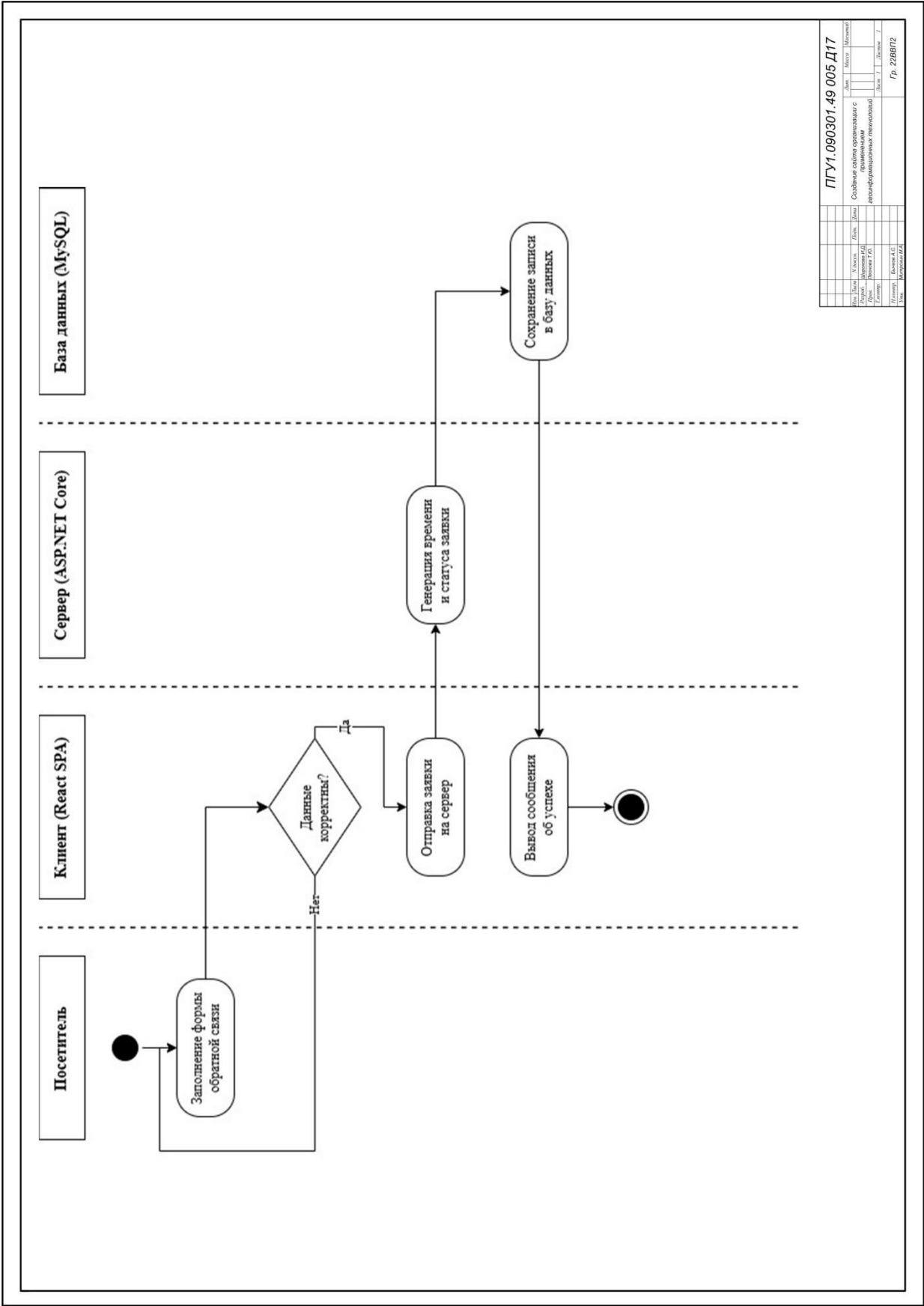
Изм	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.49.001 ПЗ

Лист

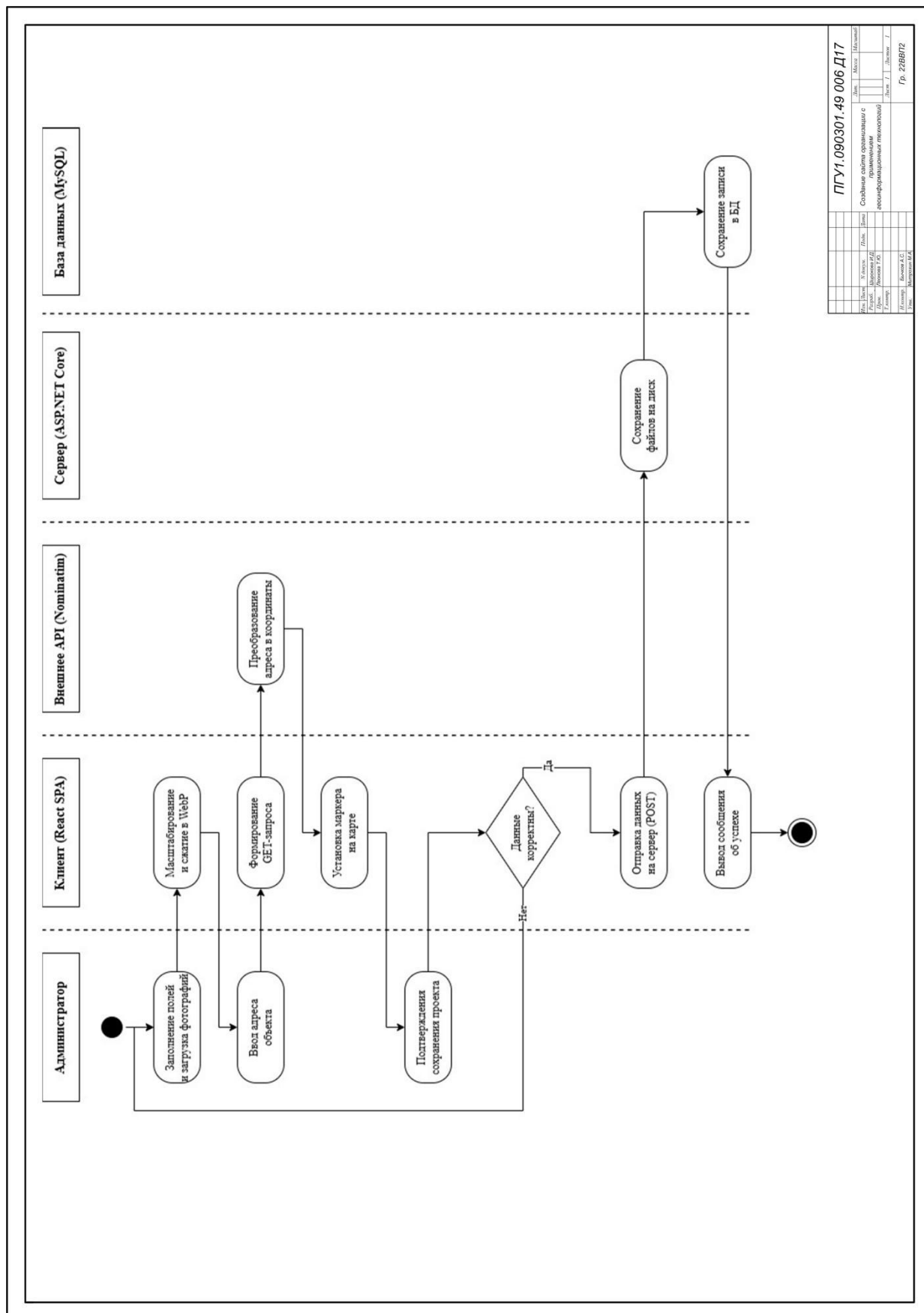
68

UML-диаграмма деятельности формы обратной связи



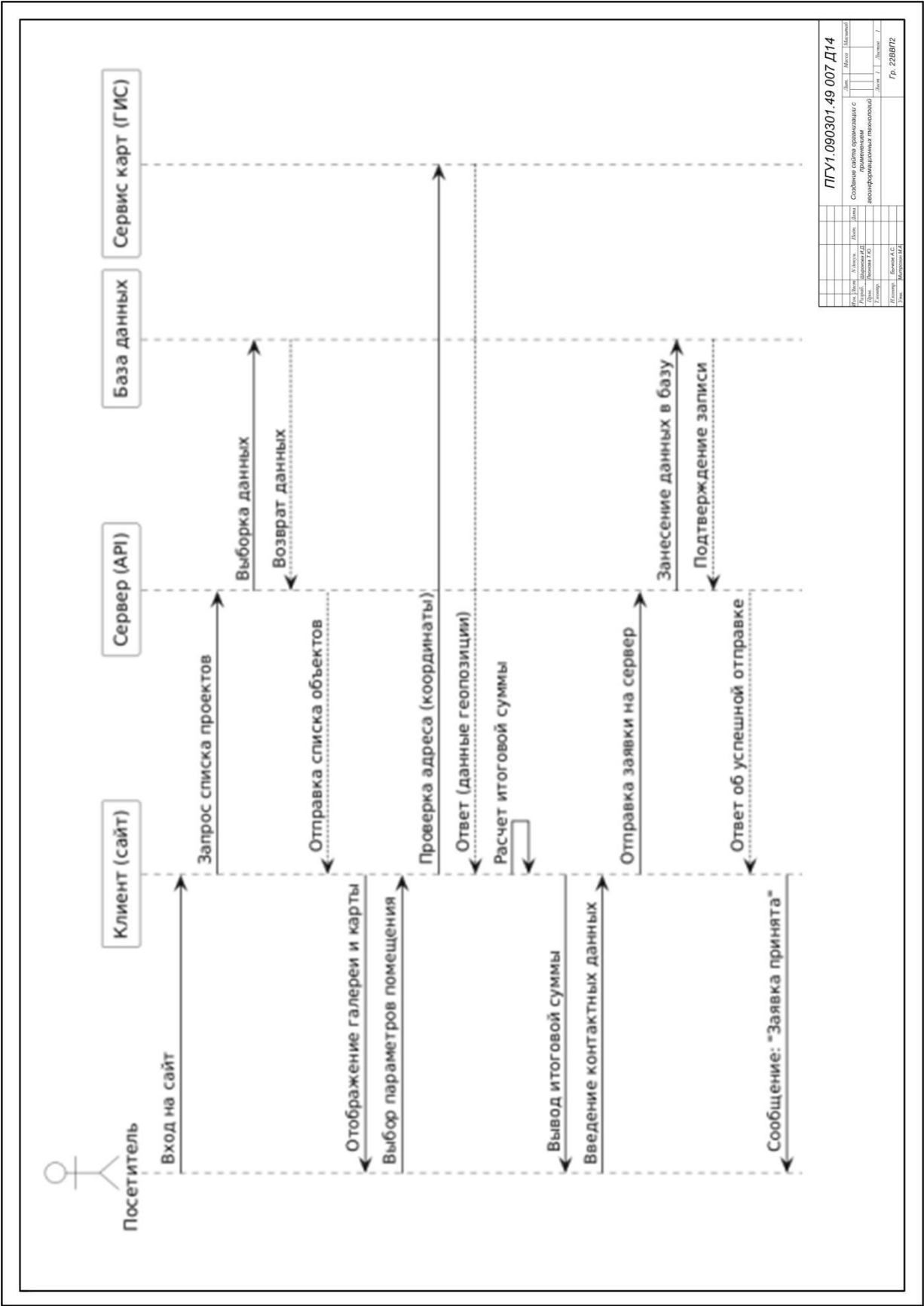
ПГУ 1.090301.49.005 Д17									
Содержание документа с грифом «Для служебного пользования»									
Имя	Фамилия	Имя	Фамилия	Имя	Фамилия	Имя	Фамилия	Имя	Фамилия
Должность	Должность	Должность	Должность	Должность	Должность	Должность	Должность	Должность	Должность
Подпись	Подпись	Подпись	Подпись	Подпись	Подпись	Подпись	Подпись	Подпись	Подпись
Дата	Дата	Дата	Дата	Дата	Дата	Дата	Дата	Дата	Дата
Гр. 228ВВ72									

UML-диаграмма деятельности для добавления проекта



ПГУ 1.090301.49.006 Д17									
Имя	Фамилия	Пол	Место	Создание сайта организации с геоинформационными технологиями					
Дан.	Дан.	Дан.	Дан.	Дан.	Дан.	Дан.	Дан.	Дан.	Дан.
Группа	Воспитатель	Воспитатель	Воспитатель	Воспитатель	Воспитатель	Воспитатель	Воспитатель	Воспитатель	Воспитатель
Гр. 22ВВП2									

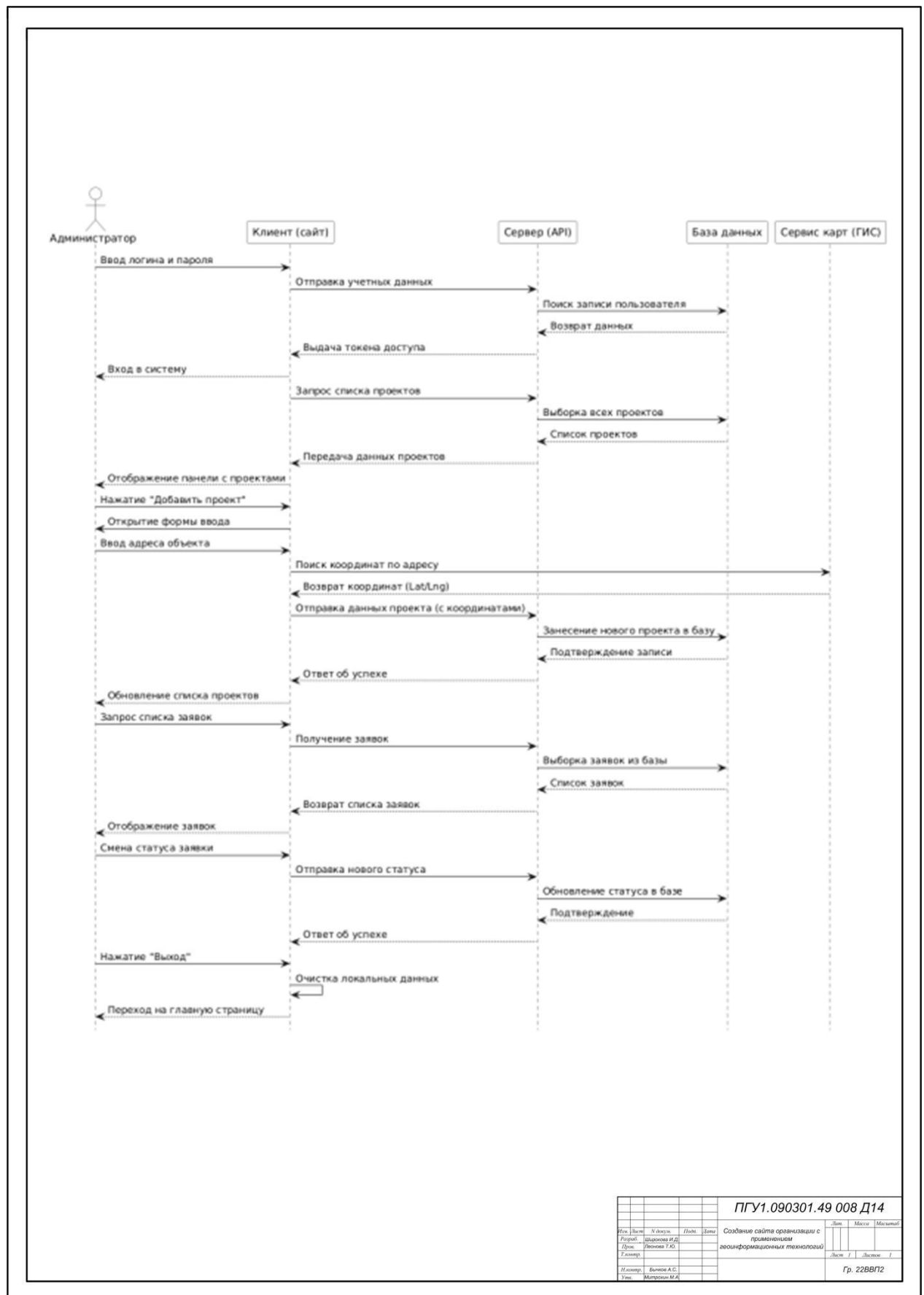
UML-диаграмма последовательности для посетителя



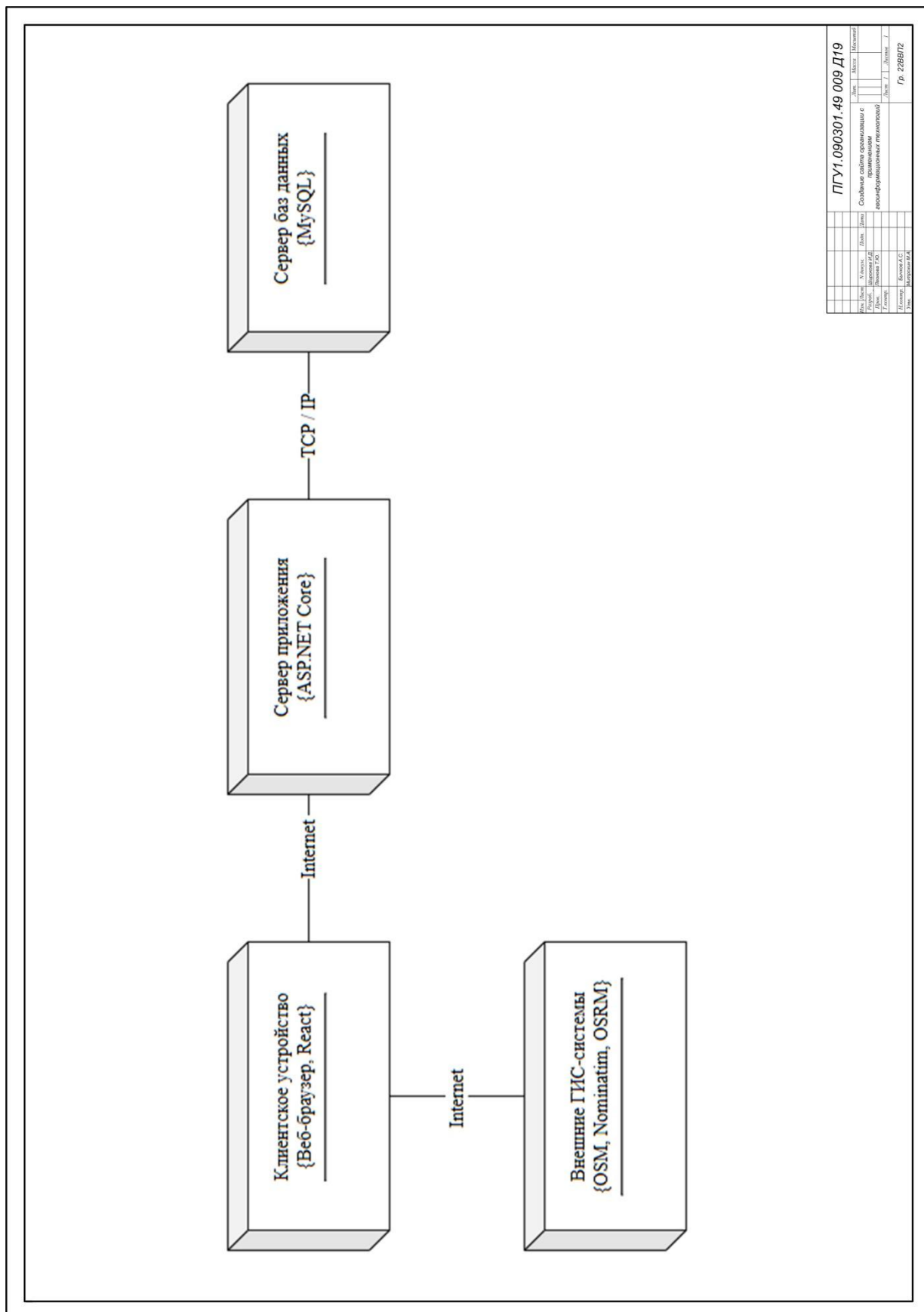
ПГУ 1.090301.49.007 Д14									
Содержит сайт, разработанный с применением геоинформационных технологий									
Имя	Фамилия	Имя	Фамилия	Имя	Фамилия	Имя	Фамилия	Имя	Фамилия
Должность	Должность	Должность	Должность	Должность	Должность	Должность	Должность	Должность	Должность
Подпись	Подпись	Подпись	Подпись	Подпись	Подпись	Подпись	Подпись	Подпись	Подпись
Дата	Дата	Дата	Дата	Дата	Дата	Дата	Дата	Дата	Дата
Гр. 228872									

Изм	Лист	№ докум.	Подпись	Дата

UML-диаграмма последовательности для администратора



UML-диаграмма развертывания



Приложение Б
(обязательное)

Создание сайта организации с применением геоинформационных технологий

Листинги программ

					ПГУ 1.090301.49.001 ПЗ	Лист
						74
Изм	Лист	№ докум.	Подпись	Дата		

Листинг для серверной части

Файл Program.cs

```
using DesignStudio.Server.Data;
using Microsoft.EntityFrameworkCore;
using Microsoft.AspNetCore.Http.Features;
using Microsoft.AspNetCore.Authentication.JwtBearer;
using Microsoft.IdentityModel.Tokens;
using System.Text;

var builder = WebApplication.CreateBuilder(args);

builder.WebHost.ConfigureKestrel(serverOptions =>
{
    serverOptions.Limits.MaxRequestBodySize = 104_857_600;
});

builder.Services.Configure<FormOptions>(options =>
{
    options.ValueLengthLimit = int.MaxValue;
    options.MultipartBodyLengthLimit = int.MaxValue;
    options.MemoryBufferThreshold = int.MaxValue;
});

var connectionString =
builder.Configuration.GetConnectionString("DefaultConnection");

builder.Services.AddDbContext<AppDbContext>(options =>
    options.UseMySQL(connectionString,
        ServerVersion.AutoDetect(connectionString)));

builder.Services.AddCors(options =>
{
    options.AddPolicy("AllowReactApp",
        policy =>
        {
            policy.WithOrigins("http://localhost:5173")
                .AllowAnyHeader()
                .AllowAnyMethod();
        });
});

var jwtKey = builder.Configuration["Jwt:Key"];
var jwtIssuer = builder.Configuration["Jwt:Issuer"];
var jwtAudience = builder.Configuration["Jwt:Audience"];
```

					ПГУ 1.090301.49.001 ПЗ	Лист
						75
Изм	Лист	№ докум.	Подпись	Дата		

```

builder.Services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
    .AddJwtBearer(options =>
    {
        options.TokenValidationParameters = new
TokenValidationParameters
        {
            ValidateIssuer = true,
            ValidateAudience = true,
            ValidateLifetime = true,
            ValidateIssuerSigningKey = true,
            ValidIssuer = jwtIssuer,
            ValidAudience = jwtAudience,
            IssuerSigningKey = new
SymmetricSecurityKey(Encoding.UTF8.GetBytes(jwtKey!))
        };
    });

builder.Services.AddControllers();
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();

if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseCors("AllowReactApp");

app.UseAuthentication();

app.UseAuthorization();

app.MapControllers();

```

					ПГУ 1.090301.49.001 ПЗ	Лист
						76
Изм	Лист	№ докум.	Подпись	Дата		

```
app.Run();
```

Файл AuthController.cs

```
using DesignStudio.Server.Data;
using DesignStudio.Server.Models;
using Microsoft.AspNetCore.Mvc;
using Microsoft.IdentityModel.Tokens;
using System.IdentityModel.Tokens.Jwt;
using System.Security.Claims;
using System.Text;

namespace DesignStudio.Server.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class AuthController : ControllerBase
    {
        private readonly AppDbContext _context;
        private readonly IConfiguration _configuration;
        public AuthController(AppDbContext context, IConfiguration
configuration)
        {
            _context = context;
            _configuration = configuration;
        }

        [HttpPost("login")]
        public IActionResult Login([FromBody] LoginRequest request)
        {
            var admin = _context.AdminUsers.FirstOrDefault(a =>
a.Username == request.Username);

            if (admin == null
|| !BCrypt.Net.BCrypt.Verify(request.Password, admin.PasswordHash))
            {
                return Unauthorized(new { message = "Неверный логин
или пароль" });
            }
            var keyStr = _configuration["Jwt:Key"];
            var issuer = _configuration["Jwt:Issuer"];
            var audience = _configuration["Jwt:Audience"];

            var securityKey = new
SymmetricSecurityKey(Encoding.UTF8.GetBytes(keyStr!));
            var credentials = new SigningCredentials(securityKey,
SecurityAlgorithms.HmacSha256);

            var claims = new[]
            {
```

					ПГУ 1.090301.49.001 ПЗ	Лист
						77
Изм	Лист	№ докум.	Подпись	Дата		

```

        new Claim(ClaimTypes.Name, admin.Username),
        new Claim(ClaimTypes.Role, "Admin")
    };

    var token = new JwtSecurityToken(
        issuer: "StoryHome",
        audience: "StoryHomeAdmin",
        claims: claims,
        expires: DateTime.Now.AddHours(12),
        signingCredentials: credentials);

    return Ok(new { token = new
JwtSecurityTokenHandler().WriteToken(token) });
}

[HttpPost("register-first-admin")]
public IActionResult RegisterAdmin([FromBody] LoginRequest
request)
{
    if (_context.AdminUsers.Any(a => a.Username ==
request.Username))
    {
        return BadRequest("Пользователь с таким логином уже
существует.");
    }

    string passwordHash =
BCrypt.Net.BCrypt.HashPassword(request.Password);

    var newAdmin = new AdminUser
    {
        Username = request.Username,
        PasswordHash = passwordHash
    };

    _context.AdminUsers.Add(newAdmin);
    _context.SaveChanges();

    return Ok(new { message = "Администратор успешно
создан!" });
}

public class LoginRequest
{
    public string Username { get; set; } = string.Empty;
    public string Password { get; set; } = string.Empty;
}

```

					ПГУ 1.090301.49.001 ПЗ	Лист
						78
Изм	Лист	№ докум.	Подпись	Дата		

Файл LeadsController.cs

```
using DesignStudio.Server.Data;
using DesignStudio.Server.Models;
using Microsoft.AspNetCore.Authorization; // ДОБАВЛЕНО
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using System;

namespace DesignStudio.Server.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class LeadsController : ControllerBase
    {
        private readonly ApplicationDbContext _context;
        public LeadsController(ApplicationDbContext context)
        {
            _context = context;
        }

        [HttpGet]
        [Authorize]
        public async Task<ActionResult<IEnumerable<Lead>>> GetLeads()
        {
            return await _context.Leads.OrderByDescending(l =>
1.Id).ToListAsync();
        }

        [HttpPost]

        public async Task<ActionResult<Lead>> PostLead(Lead lead)
        {
            lead.CreatedAt = DateTime.UtcNow;
            lead.Status = "new";

            _context.Leads.Add(lead);
            await _context.SaveChangesAsync();

            return Ok(lead);
        }

        [HttpPut("{id}")]
        [Authorize]
        public async Task<IActionResult> PutLead(int id, Lead lead)
        {
            if (id != lead.Id)
            {
                return BadRequest();
            }

            _context.Entry(lead).State = EntityState.Modified;
        }
    }
}
```

Изм	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.49.001 ПЗ

Лист

79

```

        await _context.SaveChangesAsync();

        return NoContent();
    }
}

```

Файл ProjectsController.cs

```

using DesignStudio.Server.Data;
using DesignStudio.Server.Models;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;

namespace DesignStudio.Server.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class ProjectsController : ControllerBase
    {
        private readonly AppDbContext _context;

        public ProjectsController(AppDbContext context)
        {
            _context = context;
        }

        [HttpGet]

        public async Task<ActionResult<IEnumerable<Project>>>
        GetProjects()
        {
            return await _context.Projects.OrderByDescending(p =>
p.Id).ToListAsync();
        }

        [HttpGet("{id}")]

        public async Task<ActionResult<Project>> GetProject(int id)
        {
            var project = await _context.Projects.FindAsync(id);

            if (project == null)
            {
                return NotFound("Проект не найден");
            }

            return project;
        }

        [HttpPost]
        [Authorize]

```

```

        public async Task<ActionResult<Project>> PostProject(Project
project)
        {
            if (project.Images == null)
            {
                project.Images = new List<string>();
            }

            project.Images = await
SaveImagesToFolderAsync(project.Images);

            if (project.CreatedAt == default)
            {
                project.CreatedAt = DateTime.UtcNow;
            }

            _context.Projects.Add(project);
            await _context.SaveChangesAsync();

            return Ok(project);
        }

        [HttpDelete("{id}")]
        [Authorize]
        public async Task<IActionResult> DeleteProject(int id)
        {
            var project = await _context.Projects.FindAsync(id);
            if (project == null)
            {
                return NotFound();
            }
            DeleteImagesFromFolder(project.Images);

            _context.Projects.Remove(project);
            await _context.SaveChangesAsync();

            return NoContent();
        }

        [HttpPut("{id}")]
        [Authorize]
        public async Task<IActionResult> PutProject(int id, Project
project)
        {
            if (id != project.Id)
            {
                return BadRequest("ID проекта не совпадает");
            }
            var existingProject = await
_context.Projects.AsNoTracking().FirstOrDefaultAsync(p => p.Id ==
id);

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		81

```

        if (existingProject == null)
        {
            return NotFound();
        }

        if (project.Images == null)
        {
            project.Images = new List<string>();
        }

        var imagesToDelete =
existingProject.Images.Except(project.Images).ToList();

        DeleteImagesFromFolder(imagesToDelete);

        project.Images = await
SaveImagesToFolderAsync(project.Images);
        _context.Entry(project).State = EntityState.Modified;

        try
        {
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!_context.Projects.Any(e => e.Id == id))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }

        return NoContent();
    }

    private async Task<List<string>>
SaveImagesToFolderAsync(List<string> base64Images)
    {
        var savedUrls = new List<string>();

        if (base64Images == null || base64Images.Count == 0)
        {
            return savedUrls;
        }
    }

```

					ПГУ 1.090301.49.001 ПЗ	Лист
						82
Изм	Лист	№ докум.	Подпись	Дата		

```

        var folderPath =
Path.Combine(Directory.GetCurrentDirectory(), "wwwroot", "images",
"projects");

        if (!Directory.Exists(folderPath))
        {
            Directory.CreateDirectory(folderPath);
        }

        foreach (var base64 in base64Images)
        {
            if (string.IsNullOrEmpty(base64)) continue;

            if (base64.StartsWith("http") ||
base64.StartsWith("/"))
            {
                savedUrls.Add(base64);
                continue;
            }

            try
            {
                var base64Data = base64.Contains(",") ?
base64.Split(', ')[1] : base64;
                var bytes = Convert.FromBase64String(base64Data);

                var fileName = Guid.NewGuid().ToString() +
".webp";
                var filePath = Path.Combine(folderPath,
fileName);

                await System.IO.File.WriteAllBytesAsync(filePath,
bytes);

                savedUrls.Add($"https://localhost:7002/images/projects/{fileName}");
            }
            catch (Exception ex)
            {
                Console.WriteLine("Ошибка сохранения картинки: "
+ ex.Message);
            }
        }
        return savedUrls;
    }

    private void DeleteImagesFromFolder(List<string> imageUrls)
    {
        if (imageUrls == null || !imageUrls.Any()) return;
    }

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		83

```

        var folderPath =
Path.Combine(Directory.GetCurrentDirectory(), "wwwroot", "images",
"projects");

        foreach (var url in imageUrls)
        {
            if (string.IsNullOrEmpty(url) ||
url.StartsWith("data:image")) continue;

            try
            {

                var uri = new Uri(url);
                var fileName = Path.GetFileName(uri.LocalPath);
                var filePath = Path.Combine(folderPath,
fileName);

                if (System.IO.File.Exists(filePath))
                {
                    System.IO.File.Delete(filePath);
                }
            }
            catch (Exception ex)
            {
                Console.WriteLine($"Ошибка удаления файла:
{ex.Message}");
            }
        }
    }
}

```

Файл AppDbContext.cs

```

using DesignStudio.Server.Models;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.ChangeTracking;
using System.Text.Json;

namespace DesignStudio.Server.Data
{
    public class AppDbContext : DbContext
    {
        public AppDbContext(DbContextOptions<AppDbContext> options) :
base(options)
        {
        }

        public DbSet<Project> Projects { get; set; }
        public DbSet<Lead> Leads { get; set; }
        public DbSet<AdminUser> AdminUsers { get; set; }
    }
}

```

```

        protected override void OnModelCreating(ModelBuilder
modelBuilder)
        {
            base.OnModelCreating(modelBuilder);

            var stringListComparer = new
ValueComparer<List<string>>(
                (c1, c2) => c1!.SequenceEqual(c2!),
                c => c.Aggregate(0, (a, v) => HashCode.Combine(a,
v.GetHashCode()))),
                c => c.ToList());

            modelBuilder.Entity<Project>()
                .Property(p => p.Images)
                .HasConversion(
                    v => JsonSerializer.Serialize(v,
(JsonSerializerOptions)null!),
                    v => string.IsNullOrEmpty(v) ? new
List<string>() : JsonSerializer.Deserialize<List<string>>(v,
(JsonSerializerOptions)null!)!
                )
                .Metadata.SetValueComparer(stringListComparer);
        }
    }
}

```

Файл AdminUser.cs

```

using System.ComponentModel.DataAnnotations;

namespace DesignStudio.Server.Models
{
    public class AdminUser
    {
        public int Id { get; set; }

        [Required]
        [MaxLength(50)]
        public string Username { get; set; } = string.Empty;

        [Required]
        public string PasswordHash { get; set; } = string.Empty;
    }
}

```

Файл Lead.cs

```

namespace DesignStudio.Server.Models

```

```

{
    public class Lead
    {
        public int Id { get; set; }
        public string Name { get; set; } = string.Empty;
        public string Email { get; set; } = string.Empty;
        public string Message { get; set; } = string.Empty;
        public string Status { get; set; } = "new";
        public DateTime CreatedAt { get; set; }
    }
}

```

Файл Project.cs

```

using Microsoft.EntityFrameworkCore.Metadata.Internal;
using System.ComponentModel.DataAnnotations.Schema;

namespace DesignStudio.Server.Models
{
    public class Project
    {
        public int Id { get; set; }
        public string Title { get; set; } = string.Empty;
        public string Category { get; set; } = string.Empty;
        [Column(TypeName = "decimal(18, 2)")]
        public decimal? Price { get; set; }
        public double Lat { get; set; }
        public double Lng { get; set; }
        public List<string> Images { get; set; } = new
List<string>();
        public string DisplayLocation { get; set; } = "both";

        public string? Description { get; set; }
        public string? Address { get; set; }
        public DateTime CreatedAt { get; set; } = DateTime.UtcNow;
    }
}

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		86

Листинг для клиентской части

Файл BaseMap.jsx

```
import { MapContainer, TileLayer } from "react-leaflet";
import L from "leaflet";

export const customIcon = new L.Icon({
  iconUrl: "https://unpkg.com/leaflet@1.9.4/dist/images/marker-
icon.png",
  iconRetinaUrl:
"https://unpkg.com/leaflet@1.9.4/dist/images/marker-icon-2x.png",
  shadowUrl: "https://unpkg.com/leaflet@1.9.4/dist/images/marker-
shadow.png",
  iconSize: [25, 41],
  iconAnchor: [12, 41],
  popupAnchor: [1, -34],
  shadowSize: [41, 41],
});

const BaseMap = ({ center, zoom, className, children, ...props }) =>
{
  return (
    <MapContainer center={center} zoom={zoom} className={className}
    {...props}>
      <TileLayer
        attribution='&copy; <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>'
        url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
      />
      {children}
    </MapContainer>
  );
};

export default BaseMap;
```

Файл data.jsx

```
export const menus = [
  { to: "/", label: "Главная" },
  { to: "/about", label: "О нас" },
  { to: "/services", label: "Услуги" },
  { to: "/projects", label: "Проекты" },
  { to: "/calculator", label: "Калькулятор" },
  { to: "/contact", label: "Контакты" },
];

export const reviewsData = [];

export const CLIENT_REVIEWS = [
  {
```

					ПГУ 1.090301.49.001 ПЗ	Лист
						87
Изм	Лист	№ докум.	Подпись	Дата		

```

    id: 1,
    rating: 5,
    text: "Заказывали проект «под ключ» для новой квартиры. Студия
взяла на себя абсолютно всё — от чертежей до приемки материалов. Мы
только согласовывали 3D-визуализации. Ремонт прошел вообще без
стресса!",
    authorName: "Олег и Анна",
    projectDetails: "КВАРТИРА 90 М², ПЕНЗА",
  },
  {
    id: 2,
    rating: 5,
    text: "Очень грамотный подход к планировке! У нас была
нестандартная геометрия кухни-гостиной, но дизайнеры предложили 3
варианта, о которых мы даже не думали. Теперь каждый сантиметр
используется с пользой.",
    authorName: "Екатерина В.",
    projectDetails: "ЕВРОТРЕШКА 65 М², ПЕНЗА",
  },
  {
    id: 3,
    rating: 5,
    text: "Для меня было критически важно уложиться в бюджет. Ребята
составили очень подробную смету, и мы ни на рубль не вышли за её
рамки. Никаких скрытых платежей, всё честно и прозрачно.
Рекомендую!",
    authorName: "Дмитрий К.",
    projectDetails: "ДОМ 120 М², ПЕНЗА",
  },
  {
    id: 4,
    rating: 5,
    text: "В маленькой студии удалось разместить полноценную кровать,
рабочее место и огромный шкаф, не перегрузив пространство. Визуально
квартира кажется в два раза больше. Отличная работа с эргономикой!",
    authorName: "Валерия",
    projectDetails: "СТУДИЯ 32 М², ПЕНЗА",
  },
  {
    id: 5,
    rating: 5,
    text: "Обратились за базовым проектом для строителей. Чертежи —
моё почтение! Бригадир сказал, что давно не видел такой подробной и
понятной документации. Ни одного вопроса во время ремонта не
возникло.",
    authorName: "Максим и Юлия",
    projectDetails: "НОВОСТРОЙКА 75 М², ПЕНЗА",
  },
  {
    id: 6,
    rating: 5,

```

```

    text: "Дизайнеры невероятно тонко чувствуют стиль. Я показала
    пару картинок из Pinterest, а они создали из этого цельную концепцию.
    3D-картинки не отличить от реальных фотографий после ремонта!",
    authorName: "София М.",
    projectDetails: "КВАРТИРА 110 М², ПЕНЗА",
  },
  {
    id: 7,
    rating: 5,
    text: "Делали ремонт дистанционно. Студия сама контролировала
    поставки, общалась с прорабом и присылала фотоотчеты. Я приехал
    только на финальную уборку. Это высочайший уровень сервиса.",
    authorName: "Артем",
    projectDetails: "АПАРТАМЕНТЫ 55 М², ПЕНЗА",
  },
  {
    id: 8,
    rating: 4,
    text: "Спасибо за уютную спальню! Идеально подобрали цвета и
    текстиль. Комната получилась именно такой, как я мечтала — светлой,
    теплой и располагающей к отдыху. Снизил балл только за небольшую
    задержку доставки мебели от поставщика.",
    authorName: "Алина С.",
    projectDetails: "СПАЛЬНЯ 18 М², ПЕНЗА",
  },
  {
    id: 9,
    rating: 5,
    text: "Заказывали дизайн ванной комнаты. Дизайнеры грамотно
    спрятали все коммуникации, сделали крутую подсветку ниш и подобрали
    идеальный керамогранит. Выглядит как в дорогом отеле.",
    authorName: "Семья Морозовых",
    projectDetails: "ТАУНХАУС 140 М², ПЕНЗЕНСКАЯ ОБЛ.",
  },
  {
    id: 10,
    rating: 5,
    text: "StoryHome полностью оправдывают свою философию — они
    делают интерьеры, в которых реально хочется жить. Возвращаться домой
    теперь сплошное удовольствие. Спасибо за ваш труд и внимание к
    мелочам!",
    authorName: "Елена П.",
    projectDetails: "КВАРТИРА 85 М², ПЕНЗА",
  },
];

```

```

export const CALCULATOR_PRICES = {
  PROJECT_TYPES: {
    plan: { name: "Планировочный", price: 1000 },
    base: { name: "Базовый", price: 1700 },
    complex: { name: "Комплексный", price: 3350 },
  },
};

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		89

```

    vip: { name: "VIP", price: 5000 },
  },
  PROPERTY_TYPES: {
    new: { name: "Новостройка", price: 0 },
    old: { name: "Вторичное жилье", price: 300 },
  },
  OPTIONS: {
    smartHome: { name: "Проект «Умный дом»", pricePerSqM: 800 },
    ventilation: { name: "Проект вентиляции", pricePerSqM: 600 },
    approval: { name: "Согласование планировки", fixedPrice: 50000 },
  },
};
end Behavioral;

```

Файл ProjectCard.jsx

```

import { IoIosArrowForward } from "react-icons/io";
import { Link } from "react-router-dom";

const ProjectCard = ({ e }) => {
  const coverImage = e?.images && e.images.length > 0 ? e.images[0] :
  null;

  return (
    <div className="flex flex-col min-w-0 h-full pb-2">

      <div className="w-full aspect-[4/3] overflow-hidden bg-gray-
100 rounded-2xl mb-4 shadow-sm">
        {coverImage ? (
          <img
            src={coverImage}
            alt={`Проект ${e?.title}`}
            className="h-full w-full object-cover hover:scale-105
transition-transform duration-700 cursor-pointer"
          />
        ) : (
          <div className="h-full w-full flex items-center justify-
center text-gray-400">
            Нет фото
          </div>
        )}
      </div>

      <div className="w-full px-1 mb-2">
        <h1
          className="text-[25px] md:text-[22px] lg:text-[25px] text-
[#292F36] leading-tight m-0 break-words"
          title={e?.title}
        >
          {e?.title}
        </h1>
      </div>
    </div>
  );
};

```

```

    </h1>
  </div>

  <div className="flex-grow"></div>

  <div className="w-full flex justify-end px-1 mt-4">
    <Link
      to={`\projects/${e?.id}`}
      onClick={() => sessionStorage.setItem("galleryScroll",
window.scrollToY)}
      className="group flex items-center justify-start bg-
[#C16828] text-white rounded-full h-[44px] w-[44px] hover:w-[145px]
transition-all duration-300 overflow-hidden shadow-md decoration-
none"
    >
      <div className="shrink-0 w-[44px] h-[44px] flex items-
center justify-center">
        <IoIosArrowForward className="text-2xl" />
      </div>
      <span className="whitespace-nowrap font-medium text-[15px]
opacity-0 group-hover:opacity-100 transition-opacity duration-300
pr-5">
        Подробнее
      </span>
    </Link>
  </div>

  </div>
);
};

```

```
export default ProjectCard;
```

Файл MenuLink.jsx

```

import { NavLink } from "react-router-dom";

const MenuLink = ({ to, label, onClick }) => (
  <li>
    <NavLink
      to={to}
      className="text-[16px] hover:text-themeColor"
      onClick={onClick}>
      {label}
    </NavLink>
  </li>
);

export default MenuLink;

```

					ПГУ 1.090301.49.001 ПЗ	Лист
						91
Изм	Лист	№ докум.	Подпись	Дата		

Файл MobileNavigation.jsx

```
import React from "react";
import { MdOutlineClose } from "react-icons/md";
import { menus } from "../../common/data";
import MenuLink from "../MenuLink";

const MobileNavigation = ({ setOpen }) => (
  <div
    className="fixed top-0 right-0 h-[100vh] w-[76%] flex-col bg-
[#f7cbab] z-[99] shadow-md"
    style={{ boxShadow: "-10px 0px 10px 1px #aaaaaa" }}
  >
    <MdOutlineClose
      className="text-3xl mt-1 absolute top-5 right-4 cursor-
pointer"
      onClick={() => setOpen(false)}
    />
    <ul className="py-20 px-10 flex flex-col gap-7 justify-center
item-center">
      {menus.map(({ to, label }) => (
        <MenuLink
          key={to}
          to={to}
          label={label}
          onClick={() => setOpen(false)}
        />
      ))}
    </ul>
  </div>
);

export default MobileNavigation;
```

Файл Navbar.jsx

```
import { useState } from "react";
import { MdOutlineMenu } from "react-icons/md";
import { Link } from "react-router-dom";
import MobileNavigation from "../MobileNavigation";
import Navigation from "../Navigation";
import Logo from "../../common/Logo";

const Navbar = () => {
  const [open, setOpen] = useState(false);

  return (
    <header className="sticky top-0 z-[100] bg-white shadow-sm w-
full">
      <nav
        aria-label="Main navigation"
      >
```

					ПГУ 1.090301.49.001 ПЗ	Лист
						92
Изм	Лист	№ докум.	Подпись	Дата		

```

        className="navbar max-w-[1200px] mx-auto bg-white flex
justify-normal gap-[60px] items-center py-3 px-4 lg:px-0"
    >
        <div className="flex justify-normal items-center w-full
lg:w-auto">
            <div className="flex justify-between items-center w-full">
                <Link to="/">
                    <Logo className="w-[180px] lg:w-[250px]" />
                </Link>
                <Navigation isMobile={false} setOpen={setOpen} />
            </div>
            <MdOutlineMenu
                className="flex lg:hidden text-3xl ml-auto cursor-
pointer"
                onClick={() => setOpen(!open)}
            />
        </div>
    </nav>

    {open && <MobileNavigation setOpen={setOpen} />}
</header>
);
};

```

export default Navbar;

Файл Navigation.jsx

```

import MenuLink from "../MenuLink";
import { menus } from "../../common/data";

const Navigation = ({ isMobile, setOpen }) => (
    <nav
        className={`hidden lg:flex ${
            isMobile ? "flex-col" : "justify-between items-center"
        }`}
        <ul className="flex justify-normal items-center gap-[47px] ml-
[34px] mb-1">
            {menus.map(({ to, label }) => (
                <MenuLink
                    key={to}
                    to={to}
                    label={label}
                    onClick={() => setOpen(false)}
                />
            ))}
        </ul>
    </nav>
);

export default Navigation;

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		93

Файл Footer.jsx

```
import { Link } from "react-router-dom";
import Logo from "../common/Logo";

const Footer = () => {
  return (
    <div className="border-t border-[#ECECEC] bg-white pt-10 pb-10">
      <div className="max-w-[1200px] mx-auto px-5 lg:px-0 flex flex-
col lg:flex-row justify-between items-center lg:items-start gap-12
lg:gap-8">

        <div className="flex flex-col items-center lg:items-start
text-center lg:text-left max-w-[320px]">
          <Link to="/">
            <Logo className="w-[180px] lg:w-[250px]" />
          </Link>
          <p className="text-[#5A4A42] text-[15px] mt-6 leading-
relaxed">
            Студия дизайна интерьера, создающая пространства, в
которых хочется
            жить, работать и творить.
          </p>
        </div>

        <div className="flex flex-col items-center w-full lg:w-auto
mt-2">
          <div className="flex flex-wrap justify-center gap-x-8 gap-
y-4 font-medium text-[16px] text-[#4D5053]">
            <Link to="/" className="hover:text-[#C16828] transition-
colors">
              Главная
            </Link>
            <Link to="/about" className="hover:text-[#C16828]
transition-colors">
              О нас
            </Link>
            <Link to="/services" className="hover:text-[#C16828]
transition-colors">
              Услуги
            </Link>
            <Link to="/projects" className="hover:text-[#C16828]
transition-colors">
              Проекты
            </Link>
            <Link to="/calculator" className="hover:text-[#C16828]
transition-colors">
              Калькулятор
            </Link>
          </div>
        </div>
      </div>
    </div>
  );
};
```

```

        <Link to="/contact" className="hover:text-[#C16828]
transition-colors">
            Контакты
        </Link>
    </div>
</div>

    <div className="flex flex-col items-center lg:items-end
text-center lg:text-right mt-2">
        <h4 className="font-bold text-[18px] mb-4 text-
[#3A2B24]">Связь с нами</h4>
        <div className="text-[15px] text-[#5A4A42] flex flex-col
gap-2">
            <p>г. Пенза, ул. Московская, 15</p>
            <p className="hover:text-[#C16828] transition-colors
cursor-pointer">hello@designstudio.ru</p>
            <p className="font-bold text-[#C16828] mt-1">+7 (999)
123-45-67</p>
        </div>
    </div>
</div>
);
};

```

export default Footer;

Файл MainLayout.jsx

```

import React, { useEffect } from "react";
import { Outlet, useLocation } from "react-router-dom";
import Navbar from "../Navbar/Navbar";
import Footer from "../Footer";

const MainLayout = () => {
    const { pathname } = useLocation();

    useEffect(() => {
        window.scrollTo(0, 0);
    }, [pathname]);

    return (
        <div>
            <Navbar />
            <Outlet />
            <Footer />
        </div>
    );
};

```

```
export default MainLayout;
```

Файл ProtectedRoute.jsx

```
import { Navigate, useNavigate } from "react-router-dom";
import { useEffect, useState } from "react";

const ProtectedRoute = ({ children }) => {
  const navigate = useNavigate();
  const [isAuthorized, setIsAuthorized] = useState(true);

  useEffect(() => {

    const verifyToken = () => {
      const token = localStorage.getItem("adminToken");

      if (!token) {
        setIsAuthorized(false);
        navigate("/admin-login", { replace: true });
        return;
      }

      try {
        const payload = JSON.parse(atob(token.split('.')[1]));
        const isExpired = payload.exp * 1000 < Date.now();

        if (isExpired) {
          localStorage.removeItem("adminToken");
          setIsAuthorized(false);
          navigate("/admin-login", { replace: true });
        }
      } catch (error) {
        localStorage.removeItem("adminToken");
        setIsAuthorized(false);
        navigate("/admin-login", { replace: true });
      }
    };

    const intervalId = setInterval(verifyToken, 1000);

    return () => clearInterval(intervalId);
  }, [navigate]);

  const token = localStorage.getItem("adminToken");
  if (!token || !isAuthorized) {
    return <Navigate to="/admin-login" replace />;
  }

  return children;
};
```

```
export default ProtectedRoute;
```

Файл ProtectedRoute.jsx

```
import { Link } from "react-router-dom";
import SEO from "../../components/SEO/SEO";
import ReviewCarousel from "../ReviewCarousel";
import aboutMainImg from "../../assets/1.jpg";
const About = () => {

  const principles = [
    {
      id: 1,
      title: "Прозрачность",
      desc: "Детальные сметы без скрытых платежей. Вы всегда знаете,
за что платите.",
    },
    {
      id: 2,
      title: "Эргономика",
      desc: "Красота не должна спорить с удобством. Мы продумываем
каждый сантиметр.",
    },
    {
      id: 3,
      title: "Точность сроков",
      desc: "Строгий график поставок и ремонтных работ, закрепленный
в договоре.",
    },
  ],

  return (
    <div className="bg-[#FAF8F5]">
      <SEO
        title="О нас - Design Studio"
        description="Информация о нашей дизайн-студии и нашем
подходе."
        keywords="о нас, дизайн студия, принципы работы"
        canonicalUrl="https://story-home-six.vercel.app/about"
      />

      <div className="max-w-[1300px] mx-auto px-5 pt-[50px] lg:pt-
[100px] pb-[100px]">
        <div className="flex flex-col lg:flex-row items-center gap-
12 lg:gap-20">
          <div className="w-full lg:w-1/2">
            <h3 className="text-[#C16828] font-bold tracking-widest
uppercase text-sm mb-4">
              О студии
            </h3>

```

```

        <h1 className="text-[40px] lg:text-[60px] font-serif
text-[#3A2B24] leading-[1.1] mb-8">
        Создаем пространства, <br className="hidden lg:block"
/>
        <span className="italic font-light">в которых хочется
жить</span>
    </h1>
    <p className="text-[18px] text-[#5A4A42] leading-relaxed
mb-6">
        Мы — команда профессионалов, влюбленных в дизайн и
архитектуру.
        Наша главная цель — создавать интерьеры, которые
идеально отражают
        характер своих владельцев, их привычки и образ жизни.
    </p>
    <p className="text-[18px] text-[#5A4A42] leading-
relaxed">
        Мы не гонимся за быстротечными трендами. Для нас важно
создать
        базу, которая останется актуальной десятилетиями,
будет легко
        адаптироваться под изменения в вашей жизни и радовать
безупречным
        качеством исполнения.
    </p>
</div>
<div className="w-full lg:w-1/2 relative">
    <img
        src={aboutMainImg}
        alt="Команда студии дизайна за работой"
        className="w-full h-[400px] lg:h-[550px] object-cover
rounded-[40px] shadow-2xl"
    />

    <div className="absolute -bottom-8 -left-8 w-40 h-40 bg-
[#E6D8C6] rounded-full -z-10 hidden lg:block"></div>
</div>
</div>

<div className="bg-[#423127] py-[100px] px-5 my-[50px]">
    <div className="max-w-[1000px] mx-auto text-center">
        <svg
            xmlns="http://www.w3.org/2000/svg"
            fill="currentColor"
            className="w-16 h-16 text-[#C16828] mx-auto mb-8
opacity-80"
            viewBox="0 0 975.036 975.036"
        >

```

```

        <path d="M925.036 57.197h-304c-27.6 0-50 22.4-50
50v304c0 27.601 22.4 50 50 50h145.5c-1.9 79.601-20.4 143.3-55.4
191.2-27.6 37.8-69.399 69.1-125.3 93.8-25.7 11.3-36.8 41.7-24.8
67.101136 76c11.6 24.399 40.3 35.1 65.1 24.399 66.2-28.6 122.101-
64.8 167.7-108.8 55.601-53.7 93.7-114.3 114.3-181.9 20.601-67.6
30.9-159.8 30.9-276.8v-239c0-27.599-22.401-50-50-50zM106.036
913.497c65.4-28.5 121-64.699 166.9-108.6 56.1-53.7 94.4-114.1 115-
181.2 20.6-67.1 30.899-159.6 30.899-277.5v-239c0-27.6-22.399-50-50-
50h-304c-27.6 0-50 22.4-50 50v304c0 27.601 22.4 50 50 50h145.5c-1.9
79.601-20.4 143.3-55.4 191.2-27.6 37.8-69.4 69.1-125.3 93.8-25.7
11.3-36.8 41.7-24.8 67.101135.9 75.8c11.601 24.399 40.501 35.2
65.301 24.4z"></path>
    </svg>
    <h2 className="text-[30px] lg:text-[45px] font-serif text-
white leading-tight mb-8">
        «Хороший дизайн — это тот, который становится незаметным,
потому что
        в нем всё работает именно так, как вы ожидали.»
    </h2>
    <p className="text-[#D4C8C1] text-lg uppercase tracking-
widest font-bold">
        — Философия нашей студии
    </p>
</div>
</div>

<div className="py-[100px] px-5 bg-white">
    <div className="max-w-[1200px] mx-auto">
        <div className="grid grid-cols-1 md:grid-cols-3 gap-12">
            {principles.map((item) => (
                <div key={item.id} className="border-t-2 border-
[#E6D8C6] pt-6">
                    <h3 className="text-[24px] font-bold text-[#3A2B24]
mb-4">
                        <span className="text-[#C16828] mr-
2">0{item.id}</span>
                        {item.title}
                    </h3>
                    <p className="text-[#5A4A42] leading-
relaxed">{item.desc}</p>
                </div>
            ))}
        </div>
    </div>
</div>

<div className="pt-[50px] pb-[100px]">
    <ReviewCarousel />
</div>

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		99

```

    <div className="py-[120px] px-5 bg-[#FAF8F5] border-t border-
gray-200">
      <div className="max-w-[800px] mx-auto text-center">
        <h2 className="text-[35px] lg:text-[45px] font-serif text-
[#3A2B24] leading-tight mb-6">
          Готовы обсудить ваш будущий интерьер?
        </h2>
        <p className="text-[18px] text-[#5A4A42] mb-10">
          Приглашаем вас на чашку кофе в наш офис или на онлайн-
встречу. Мы
          внимательно выслушаем ваши пожелания и предложим
оптимальный вектор
          работы.
        </p>
        <Link to="/contact">
          <button className="px-10 py-5 bg-[#423127] text-white
font-bold rounded-xl hover:bg-[#2C201A] transition-all shadow-xl">
            Записаться на консультацию
          </button>
        </Link>
      </div>
    </div>
  </div>
);
};

```

```
export default About;
```

Файл ReviewCard.jsx

```

import React from "react";

const ReviewCard = ({ review }) => {
  const { rating, text, authorName, projectDetails } = review;

  return (

    <div className="bg-white rounded-[30px] shadow-sm p-10 h-full
flex flex-col justify-between border border-gray-100">

      <div>

        <div className="flex items-center gap-0.5 text-[#C16828]
text-xl mb-6">
          {[...Array(5)].map((_, index) => (
            <span key={index} className={index < rating ? "opacity-
100" : "opacity-30"}>
              ★
            </span>
          ))}
        </div>
      </div>
    </div>
  );
}

```

```

    </div>

    <p className="font-medium text-[17px] text-[#4D5053]
leading-relaxed relative">
      <span className="text-4xl text-[#C16828] font-black
absolute -left-7 -top-2">"</span>
      {text}
      <span className="text-4xl text-[#C16828] font-black
absolute -right-7 bottom-0">"</span>
    </p>
  </div>

  <div>

    <hr className="border-t border-gray-100 my-8" />

    <div className="flex flex-col gap-1">
      <p className="font-bold text-[18px] text-[#292F36]">
        {authorName}
      </p>
      <p className="text-xs text-gray-400 font-medium uppercase
tracking-wider">
        {projectDetails}
      </p>
    </div>
  </div>
</div>
);
};

export default ReviewCard;

```

Файл ReviewCarousel.jsx

```

import React from "react";
import { CLIENT_REVIEWS } from "../../components/common/data";
import ReviewCard from "../ReviewCard";

import { Swiper, SwiperSlide } from "swiper/react";
import { Navigation } from "swiper/modules";
import "swiper/css";

const ReviewCarousel = () => {
  const stats = [
    { value: "65+", label: "ЗАВЕРШЁННЫХ ПРОЕКТОВ" },
    { value: "4.9/5", label: "СРЕДНЯЯ ОЦЕНКА" },
    { value: "2 года", label: "ОПЫТА РАБОТЫ" },
  ];

  return (

```

```

<div className="bg-[#faf8f5] py-20 px-4 md:px-0">

  <div className="max-w-[1300px] mx-auto relative">
    <div className="text-center mb-16">
      <h2 className="text-[35px] md:text-[50px] text-[#292F36]
font-serif font-bold leading-tight">
        ОТЗЫВЫ КЛИЕНТОВ
      </h2>
      <p className="text-[20px] text-[#4D5053] mt-4 max-w-[600px]
mx-auto">
        Истории успеха наших проектов
      </p>
    </div>

    <div className="relative w-full px-12 md:px-20">
      <Swiper
        modules={[Navigation]}
        spaceBetween={30}
        slidesPerView={1}
        loop={true}

        navigation={{
          prevEl: ".custom-prev-btn",
          nextEl: ".custom-next-btn",
        }}
        breakpoints={{
          640: { slidesPerView: 1.5, spaceBetween: 20 },
          768: { slidesPerView: 2, spaceBetween: 30 },
          1024: { slidesPerView: 3, spaceBetween: 30 },
        }}
        className="py-4"
      >
        {CLIENT_REVIEWS.map((review) => (
          <SwiperSlide key={review.id} className="h-auto">
            <ReviewCard review={review} />
          </SwiperSlide>
        ))}
      </Swiper>

      <button className="custom-prev-btn absolute left-0 top-1/2
-translate-y-1/2 z-10 w-[56px] h-[56px] rounded-full bg-[#f0e6d8]
hover:bg-[#e6d8c6] flex items-center justify-center transition-
colors shadow-sm cursor-pointer border-none outline-none">

        <svg
          xmlns="http://www.w3.org/2000/svg"
          fill="none"

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		102

```

        viewBox="0 0 24 24"
        stroke="#B08D5B"
        className="w-7 h-7"
      >
        <path
          strokeLinecap="round"
          strokeLinejoin="round"
          strokeWidth="1.2"
          d="M10.5 19.5L3 12m0 0l7.5-7.5M3 12h18"
        />
      </svg>
    </button>

```

```

    <button className="custom-next-btn absolute right-0 top-
1/2 -translate-y-1/2 z-10 w-[56px] h-[56px] rounded-full bg-[#f0e6d8]
hover:bg-[#e6d8c6] flex items-center justify-center transition-
colors shadow-sm cursor-pointer border-none outline-none">

```

```

    <svg
      xmlns="http://www.w3.org/2000/svg"
      fill="none"
      viewBox="0 0 24 24"
      stroke="#B08D5B"
      className="w-7 h-7"
    >
      <path
        strokeLinecap="round"
        strokeLinejoin="round"
        strokeWidth="1.2"
        d="M13.5 4.5L21 12m0 0l-7.5 7.5M21 12H3"
      />
    </svg>
  </button>
</div>

```

```

    <div className="grid grid-cols-1 md:grid-cols-3 gap-10
md:gap-4 text-center mt-20 border-t border-gray-100 pt-16">
      {stats.map((stat, index) => (
        <div key={index} className="flex flex-col items-center">
          <p className="text-[60px] md:text-[70px] font-bold
text-[#C16828] leading-none mb-3">
            {stat.value}
          </p>
          <p className="text-xs text-gray-400 font-medium
uppercase tracking-wider max-w-[180px]">
            {stat.label}
          </p>
        </div>
      ))}
    </div>

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		103

```

        </div>
      </div>
    </div>
  );
};

export default ReviewCarousel;

```

Файл AdminMapSelector.jsx

```

import { useEffect } from "react";
import { Marker, useMapEvents, useMap } from "react-leaflet";
import BaseMap, { customIcon } from
"../../../../../components/common/BaseMap";

const LocationMarker = ({ position, setPosition, onAddressFound })
=> {
  useMapEvents({
    async click(e) {
      const { lat, lng } = e.latlng;
      setPosition(e.latlng);

      try {
        const response = await fetch(
          `https://nominatim.openstreetmap.org/reverse?format=json&l
at=${lat}&lon=${lng}&zoom=18&addressdetails=1`
        );
        const data = await response.json();

        if (data && data.address) {
          const road = data.address.road || "";
          const houseNumber = data.address.house_number || "";
          let city = data.address.city || data.address.town ||
data.address.village || "";

          city = city.replace(/городской округ\s*/i, "").trim();

          let finalAddress = `${road} ${houseNumber}`.trim();

          if (!finalAddress) {
            finalAddress = data.display_name.split(",")[0];
          }

          const prefix = city ? `г. ${city}, ` : "";
          let resultAddress = `${prefix}${finalAddress}`;

          resultAddress = resultAddress.replace(/городской
округ\s*/i, "").trim();

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		104

```

        onAddressFound(resultAddress);
    }
    } catch (error) {
        console.error(error);
    }
    },
    });
    return position === null ? null : <Marker position={position}
    icon={customIcon} />;
};

const MapUpdater = ({ position }) => {
    const map = useMap();
    useEffect(() => {
        if (position) {
            map.flyTo([position.lat, position.lng], 16, { animate: true });
        } else {
            map.flyTo([53.195, 45.0183], 12, { animate: true });
        }
    }, [position, map]);
    return null;
};

const AdminMapSelector = ({ position, setPosition, onAddressFound })
=> {
    return (
        <BaseMap center={[53.195, 45.0183]} zoom={12} className="w-full
h-full">
            <LocationMarker position={position} setPosition={setPosition}
onAddressFound={onAddressFound} />
            <MapUpdater position={position} />
        </BaseMap>
    );
};

export default AdminMapSelector;

```

Файл ProjectForm.jsx

```

import { useState, useRef } from "react";
import AdminMapSelector from "../AdminMapSelector";

const getTodayDateString = () => {
    const today = new Date();
    return today.toISOString().split("T")[0];
};

const ProjectForm = ({ projectToEdit, onSuccess, onCancel }) => {
    const fileInputRef = useRef(null);
    const isEditing = !!projectToEdit;

    const [position, setPosition] = useState(

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		105

```

    projectToEdit && projectToEdit.lat && projectToEdit.lng
      ? { lat: projectToEdit.lat, lng: projectToEdit.lng }
      : null,
  );

const [projectForm, setProjectForm] = useState({
  title: projectToEdit?.title || "",
  description: projectToEdit?.description || "",
  category: projectToEdit?.category || "",
  price: projectToEdit?.price || "",
  address: projectToEdit?.address || "",
  images: projectToEdit?.images || [],
  displayLocation: projectToEdit?.displayLocation || "both",
  createdAt: projectToEdit?.createdAt
    ? projectToEdit.createdAt.split("T")[0]
    : getTodayDateString(),
});

const handleImageUpload = async (e) => {
  const files = Array.from(e.target.files);
  if (files.length === 0) return;
  const compressedImages = [];

  for (const file of files) {
    const compressedBase64 = await new Promise((resolve) => {
      const reader = new FileReader();
      reader.readAsDataURL(file);
      reader.onload = (event) => {
        const img = new Image();
        img.src = event.target.result;
        img.onload = () => {
          const canvas = document.createElement("canvas");
          const MAX_WIDTH = 1200;
          let width = img.width;
          let height = img.height;
          if (width > MAX_WIDTH) {
            height = Math.round((height * MAX_WIDTH) / width);
            width = MAX_WIDTH;
          }
          canvas.width = width;
          canvas.height = height;
          const ctx = canvas.getContext("2d");
          ctx.drawImage(img, 0, 0, width, height);
          resolve(canvas.toDataURL("image/webp", 0.8));
        };
      };
    });
    compressedImages.push(compressedBase64);
  }
  setProjectForm((prev) => ({
    ...prev,

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		106

```

        images: [...prev.images, ...compressedImages],
    }));
};

const removeImage = (indexToRemove) => {
    setProjectForm((prev) => ({
        ...prev,
        images: prev.images.filter((_, index) => index !==
indexToRemove),
    }));
};

const handleSearchAddress = async (e) => {
    e.preventDefault();
    if (!projectForm.address.trim()) return;

    try {
        const response = await fetch(
            `https://nominatim.openstreetmap.org/search?format=json&q=${
encodeURIComponent(projectForm.address)}`,
        );
        const data = await response.json();

        if (data && data.length > 0) {
            const lat = parseFloat(data[0].lat);
            const lon = parseFloat(data[0].lon);
            setPosition({ lat, lng: lon });
        } else {
            alert(
                "Координаты не найдены. Укажите адрес точнее или кликните
по карте вручную.",
            );
        }
    } catch (error) {
        console.error(error);
    }
};

const handleSaveProject = async (e) => {
    e.preventDefault();
    if (
        !position &&
        (projectForm.displayLocation === "both" ||
        projectForm.displayLocation === "map")
    ) {
        alert("Пожалуйста, выберите местоположение объекта на карте!");
        return;
    }

    let isoDate;
    try {

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		107

```

    isoDate = new Date(projectForm.createdAt).toISOString();
  } catch (err) {
    isoDate = new Date().toISOString();
  }

  const payload = {
    id: projectToEdit?.id || 0,
    title: projectForm.title,
    description: projectForm.description,
    category: projectForm.category,
    price: projectForm.price ? Number(projectForm.price) : null,
    address: projectForm.address,
    lat: position ? position.lat : 0,
    lng: position ? position.lng : 0,
    images: projectForm.images,
    displayLocation: projectForm.displayLocation,
    createdAt: isoDate,
  };

  const url = isEditing
    ? `https://localhost:7002/api/projects/${projectToEdit.id}`
    : "https://localhost:7002/api/projects";
  const method = isEditing ? "PUT" : "POST";

  try {
    const token = localStorage.getItem("adminToken");
    const response = await fetch(url, {
      method: method,
      headers: {
        "Content-Type": "application/json",
        ...(token && { Authorization: `Bearer ${token}` }),
      },
      body: JSON.stringify(payload),
    });

    if (response.ok) {
      onSuccess(
        isEditing ? "Проект успешно обновлен!" : "Проект успешно
добавлен!",
      );
    } else {
      alert("Ошибка при сохранении на сервер.");
    }
  } catch (error) {
    console.error(error);
  }
};

return (
  <div className="animate-fade-in w-full">
    <h2 className="text-2xl font-bold mb-6 text-[#292F36]">

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		108

```

        {isEditing ? "Редактировать проект" : "Добавить новый
объект"}
    </h2>
    <form
        onSubmit={handleSaveProject}
        className="grid grid-cols-1 gap-6 w-full"
    >
        <div className="flex flex-col md:flex-row gap-4 w-full">
            <div className="flex flex-col gap-2 w-full">
                <label className="text-sm font-semibold text-gray-500
ml-1">
                    Название проекта
                </label>
                <input
                    type="text"
                    required
                    className="w-full px-4 py-3 rounded-xl border border-
gray-200 focus:ring-2 focus:ring-[#C16828] outline-none"
                    value={projectForm.title}
                    onChange={ (e) =>
                        setProjectForm({ ...projectForm, title:
e.target.value })
                    }
                />
            </div>
            <div className="flex flex-col gap-2 w-full">
                <label className="text-sm font-semibold text-gray-500
ml-1">
                    Категория
                </label>
                <select
                    className="w-full px-4 py-3 rounded-xl border border-
gray-200 focus:ring-2 focus:ring-[#C16828] outline-none"
                    value={projectForm.category}
                    onChange={ (e) =>
                        setProjectForm({ ...projectForm, category:
e.target.value })
                    }
                    required
                >
                    <option value="">Выберите тип</option>
                    <option value="Turnkey">Квартира под ключ</option>
                    <option value="Living">Гостиная</option>
                    <option value="Bedroom">Спальня</option>
                    <option value="Kitchen">Кухня</option>
                    <option value="Bathroom">Ванная</option>
                </select>
            </div>
        </div>

        <div className="flex flex-col md:flex-row gap-4 w-full">

```

```

<div className="flex flex-col gap-2 w-full">
  <label className="text-sm font-semibold text-gray-500
ml-1">
    Бюджет ($)
  </label>
  <input
    type="number"
    className="w-full px-4 py-3 rounded-xl border border-
gray-200 focus:ring-2 focus:ring-[#C16828] outline-none"
    value={projectForm.price}
    onChange={ (e) =>
      setProjectForm({ ...projectForm, price:
e.target.value })
    }
  />
</div>
<div className="flex flex-col gap-2 w-full">
  <label className="text-sm font-semibold text-gray-500
ml-1">
    Дата сдачи проекта
  </label>
  <input
    type="date"
    max={getTodayDateString()}
    required
    className="w-full px-4 py-3 rounded-xl border border-
gray-200 focus:ring-2 focus:ring-[#C16828] outline-none"
    value={projectForm.createdAt}
    onChange={ (e) =>
      setProjectForm({ ...projectForm, createdAt:
e.target.value })
    }
  />
</div>
</div>

<div className="flex flex-col gap-2 w-full">
  <label className="text-sm font-semibold text-gray-500 ml-
1">
    Где отображать проект?
  </label>
  <select
    className="w-full px-4 py-3 rounded-xl border border-
gray-200 focus:ring-2 focus:ring-[#C16828] outline-none bg-orange-
50/30"
    value={projectForm.displayLocation}
    onChange={ (e) =>
      setProjectForm({
        ...projectForm,
        displayLocation: e.target.value,
      })
    }
  >

```

```

        }
        required
    >
        <option value="both">Везде (Карта + Галерея)</option>
        <option value="map">Только на Карте</option>
        <option value="gallery">Только в Галерее</option>
    </select>
</div>

<div className="flex flex-col gap-2 w-full">
    <label className="text-sm font-semibold text-gray-500 ml-
1">
        Описание проекта
    </label>
    <textarea
        rows="3"
        placeholder="Расскажите о проекте подробнее..."
        className="w-full px-4 py-3 rounded-xl border border-
gray-200 focus:ring-2 focus:ring-[#C16828] outline-none resize-none"
        value={projectForm.description}
        onChange={(e) =>
            setProjectForm({ ...projectForm, description:
e.target.value })
        }
    ></textarea>
</div>

<div className="flex flex-col gap-2 w-full">
    <label className="text-sm font-semibold text-gray-500 ml-
1">
        Фотографии объекта
    </label>
    <div className="flex flex-col gap-4 p-4 border-2 border-
dashed border-gray-200 rounded-2xl bg-gray-50">
        <div className="flex items-center gap-3">
            <button
                type="button"
                onClick={() => fileInputRef.current.click()}
                className="shrink-0 py-2 px-4 bg-[#292F36] text-
white rounded-full text-sm font-semibold hover:bg-black transition
shadow-sm"
            >
                Выберите файлы
            </button>
            <span className="text-sm text-gray-500 font-medium">
                {projectForm.images.length === 0
                    ? "Файлы не выбраны"
                    : `Выбрано файлов: ${projectForm.images.length}`}
            </span>
        </div>
    </div>
    <input

```

```

        type="file"
        accept="image/*"
        multiple
        ref={fileInputRef}
        onChange={handleImageUpload}
        className="hidden"
      />
      {projectForm.images.length > 0 && (
        <div className="flex flex-wrap gap-3 mt-2">
          {projectForm.images.map((img, index) => (
            <div key={index} className="relative group">
              <img
                src={img}
                alt="preview"
                className="h-20 w-20 object-cover rounded-xl
shadow-md border border-gray-200"
              />
              <button
                type="button"
                onClick={() => removeImage(index)}
                className="absolute -top-2 -right-2 bg-red-500
text-white rounded-full w-6 h-6 flex items-center justify-center
text-xs opacity-0 group-hover:opacity-100 transition-opacity shadow-
md"
              >
                ✕
              </button>
            </div>
          ))}
        </div>
      )}
    </div>
  </div>

  {projectForm.displayLocation !== "gallery" && (
    <div className="flex flex-col gap-2 transition-all w-
full">
      <label className="text-sm font-semibold text-gray-500
ml-1">
        Физический адрес и координаты
      </label>
      <div className="flex gap-2 mb-2 w-full">
        <input
          type="text"
          placeholder="Укажите точный адрес..."
          className="flex-1 min-w-0 px-4 py-3 rounded-xl
border border-gray-200 focus:ring-2 focus:ring-[#C16828] outline-
none"
          value={projectForm.address}
          onChange={(e) =>

```

```

        setProjectForm({ ...projectForm, address:
e.target.value })
    }
    />
    <button
        type="button"
        onClick={handleSearchAddress}
        className="shrink-0 px-6 bg-[#292F36] text-white
rounded-xl font-bold hover:bg-black transition"
    >
        Найти на карте
    </button>
</div>
<div className="h-[300px] w-full rounded-2xl overflow-
hidden border border-gray-200 shadow-inner z-0">
    <AdminMapSelector
        position={position}
        setPosition={setPosition}
        onAddressFound={(address) =>
address })
    >
    />
</div>
</div>
)}

<div className="flex gap-3 mt-2 w-full">
    <button
        type="submit"
        className="flex-1 bg-[#C16828] text-white py-4 rounded-
xl font-bold hover:bg-[#a35620] transition shadow-lg"
    >
        {isEditing ? "Обновить данные" : "Опубликовать в
портфолио"}
    </button>
    <button
        type="button"
        onClick={onCancel}
        className="shrink-0 px-8 bg-gray-200 text-gray-700
rounded-xl font-bold hover:bg-gray-300 transition"
    >
        Отмена
    </button>
</div>
</form>
</div>
);
};

export default ProjectForm;

```

Файл ProjectList.jsx

```
import { useState } from "react";
import { Marker, Popup } from "react-leaflet";
import BaseMap, { customIcon } from
"../../../../../components/common/BaseMap";

const ProjectList = ({ projects, onEdit, onDelete }) => {
  const [expandedProjects, setExpandedProjects] = useState([]);
  const [activeCategory, setActiveCategory] = useState("Bce");
  const [sortOrder, setSortOrder] = useState("newest");
  const [dateRange, setDateRange] = useState({ start: "", end: "" });
  const [highlightedId, setHighlightedId] = useState(null);

  const scrollToProject = (id) => {
    const element = document.getElementById(`project-card-${id}`);
    if (element) {
      element.scrollIntoView({ behavior: "smooth", block:
"center" });
      setHighlightedId(id);
      setTimeout(() => setHighlightedId(null), 2000);
    }
  };

  let displayedProjects =
    activeCategory === "Bce"
      ? projects
      : projects.filter((p) => p.category === activeCategory);

  if (dateRange.start) {
    displayedProjects = displayedProjects.filter(
      (p) => new Date(p.createdAt || 0) >= new Date(dateRange.start),
    );
  }
  if (dateRange.end) {
    const endDate = new Date(dateRange.end);
    endDate.setHours(23, 59, 59, 999);
    displayedProjects = displayedProjects.filter(
      (p) => new Date(p.createdAt || 0) <= endDate,
    );
  }

  displayedProjects.sort((a, b) => {
    const dateA = new Date(a.createdAt || 0);
    const dateB = new Date(b.createdAt || 0);
    return sortOrder === "newest" ? dateB - dateA : dateA - dateB;
  });

  const toggleExpandProject = (id) => {
    if (expandedProjects.includes(id)) {
      setExpandedProjects(expandedProjects.filter((projId) =>
projId !== id));
    }
  };
}
```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		114

```

    } else {
      setExpandedProjects([...expandedProjects, id]);
    }
  };

  return (
    <div className="animate-fade-in w-full">
      <div className="mb-4 w-full">
        <h3 className="text-sm font-semibold text-gray-500 mb-2 ml-1">
          Категория:
        </h3>
        <div className="flex flex-wrap gap-2 w-full">
          {["Все", "Turnkey", "Living", "Bedroom", "Kitchen", "Bathroom"].map(
            (category, index) => (
              <button
                key={index}
                onClick={() => setActiveCategory(category)}
                className={`px-4 py-2.5 rounded-xl text-sm font-bold transition-all border-2 ${
                  activeCategory === category
                    ? "border-[#C16828] bg-orange-50 text-[#C16828]"
                    : "border-gray-200 text-gray-500 hover:border-[#C16828] bg-white"
                }`}
              >
                {category === "Turnkey"
                  ? "Квартира под ключ"
                  : category === "Living"
                  ? "Гостиная"
                  : category === "Bedroom"
                  ? "Спальня"
                  : category === "Kitchen"
                  ? "Кухня"
                  : category === "Bathroom"
                  ? "Ванная"
                  : category}
              </button>
            ),
          )}
        </div>
      </div>

      <div className="mb-6 bg-gray-50 p-4 rounded-2xl border border-gray-100 flex flex-wrap gap-4 items-center justify-between w-full lg:w-auto">
        <div className="flex flex-wrap gap-3 items-center w-full">
          <div className="flex items-center gap-2">
            <span className="text-sm font-medium text-gray-500">C:</span>

```

```

        <input
            type="date"
            value={dateRange.start}
            onChange={ (e) =>
                setDateRange({ ...dateRange, start: e.target.value })
            }
            className="px-3 py-2 text-sm rounded-xl border border-
gray-200 outline-none focus:border-[#C16828]"
        />
    </div>
    <div className="flex items-center gap-2">
        <span className="text-sm font-medium text-gray-
500">По:</span>
        <input
            type="date"
            value={dateRange.end}
            onChange={ (e) =>
                setDateRange({ ...dateRange, end: e.target.value })
            }
            className="px-3 py-2 text-sm rounded-xl border border-
gray-200 outline-none focus:border-[#C16828]"
        />
    </div>
</div>
<select
    value={sortOrder}
    onChange={ (e) => setSortOrder(e.target.value)}
    className="w-full lg:w-auto px-4 py-2 text-sm rounded-xl
border border-gray-200 outline-none focus:border-[#C16828] font-
semibold bg-white"
>
    <option value="newest">Сначала новые</option>
    <option value="oldest">Сначала старые</option>
</select>
</div>

<div className="mb-6 w-full rounded-2xl overflow-hidden border
border-gray-200 shadow-sm h-[350px] relative z-0">
    <BaseMap
        center={[53.195, 45.0183]}
        zoom={12}
        className="w-full h-full z-0"
    >
        {displayedProjects
            .filter((p) => p.lat && p.lng && p.displayLocation !==
"gallery")
            .map((project) => (
                <Marker
                    key={project.id}
                    position={[project.lat, project.lng]}
                    icon={customIcon}

```

```

    >
    <Popup minWidth={200}>
      <div className="flex flex-col gap-2 p-1">
        {project.images && project.images.length > 0 &&
(
          <img
            src={project.images[0]}
            className="w-full h-24 object-cover rounded-
lg"
            alt="Preview"
          />
        )}
        <div className="font-bold text-[#292F36]
leading-tight">
          {project.title}
        </div>
        <div className="text-gray-500 text-xs mb-1">
          {project.address}
        </div>
        <button
          onClick={() => scrollToProject(project.id)}
          className="w-full py-2 bg-[#C16828] text-white
text-xs font-bold rounded-lg hover:bg-[#a35620] transition-colors
shadow-sm"
          >
          Найти в списке ↓
        </button>
      </div>
    </Popup>
  </Marker>
) ) }
</BaseMap>
</div>

<div className="flex flex-col gap-4 w-full">
  {displayedProjects.map((project) => {
    const isProjExpanded =
expandedProjects.includes(project.id);
    const isHighlighted = highlightedId === project.id;
    const titleText = project.title || "";
    const descText = project.description || "Описание
отсутствует";
    const isLongDesc = descText.length > 80;
    const displayDesc =
      isProjExpanded || !isLongDesc
        ? descText
        : descText.substring(0, 80) + "...";
    const createdAt = project.createdAt
      ? new Date(project.createdAt).toLocaleDateString("ru-RU")
      : "Неизвестно";

```

```

return (
  <div
    key={project.id}
    id={`project-card-${project.id}`}
    className={`flex flex-col md:flex-row items-start
justify-between p-5 rounded-2xl transition-all duration-500 gap-4 w-
full border-2 ${
      isHighlighted
        ? "border-[#C16828] shadow-xl scale-[1.02] bg-
orange-50"
        : "bg-white border-transparent hover:border-
orange-100 shadow-sm hover:shadow-lg"
      }`}
    >
    <div className="flex items-start gap-5 w-full md:flex-
1">
      <div className="shrink-0 w-24 h-24 bg-gray-100
rounded-xl overflow-hidden shadow-inner mt-1">
        {project.images && project.images.length > 0 ? (
          <img
            src={project.images[0]}
            alt="Project"
            className="w-full h-full object-cover"
          />
        ) : (
          <div className="flex items-center justify-center
h-full text-xs text-gray-400">
            Нет фото
          </div>
        )}
      </div>

      <div className="min-w-0 w-full flex flex-col gap-1">
        <div className="flex items-center justify-between
gap-4 w-full">
          <h4 className="font-bold text-lg text-[#292F36]
break-words">
            {titleText}
          </h4>
          <span className="shrink-0 text-xs text-gray-400
font-medium bg-gray-50 px-2 py-1 rounded-md">
            {createdDate}
          </span>
        </div>

        <p className="text-xs text-gray-500 font-medium
mt-1 w-full">
          <span className="text-[#C16828]">
            {project.category === "Turnkey"
              ? "Квартира под ключ"
              : project.category === "Living"

```

```

        ? "Гостиная"
        : project.category === "Bedroom"
        ? "Спальня"
        : project.category === "Kitchen"
        ? "Кухня"
        : project.category === "Bathroom"
        ? "Ванная"
        : project.category}
    </span>
    {project.displayLocation !== "gallery" &&
    ` • ${project.address || "Адрес не указан"}}`
  </p>

  <div className="text-sm text-gray-600 mt-2 italic
border-l-2 border-gray-200 pl-3 w-full break-words">
    {displayDesc}
    {isLongDesc && (
      <button
        onClick={() =>
toggleExpandProject(project.id)}
        className="ml-2 text-[#C16828] font-bold
text-xs hover:underline"
      >
        {isProjExpanded ? "Свернуть" : "Читать
далее"}
      </button>
    )}
  </div>
</div>
</div>

  <div className="shrink-0 flex flex-row md:flex-col
gap-2 w-full md:w-auto mt-2 md:mt-0">
    <button
      onClick={() => onEdit(project)}
      className="flex-1 md:flex-none px-4 py-2 bg-blue-
50 text-blue-600 rounded-xl font-bold hover:bg-blue-100 transition
text-sm text-center"
    >
      Изменить
    </button>
    <button
      onClick={() => onDelete(project.id)}
      className="flex-1 md:flex-none px-4 py-2 bg-red-50
text-red-600 rounded-xl font-bold hover:bg-red-100 transition text-
sm text-center"
    >
      Удалить
    </button>
  </div>
</div>

```

```

    );
  }}}
  {displayedProjects.length === 0 && (
    <div className="text-center bg-gray-50 rounded-2xl py-12
border border-gray-100 w-full">
      <p className="text-gray-400 font-medium">
        Проекты не найдены. Попробуйте изменить параметры
фильтра.
      </p>
    </div>
  )}
</div>
</div>
);
};

export default ProjectList;

```

Файл ProjectsTab.jsx

```

import { useState, useEffect } from "react";
import ProjectList from "../ProjectList";
import ProjectForm from "../ProjectForm";

const ProjectsTab = () => {
  const [projects, setProjects] = useState([]);
  const [viewMode, setViewMode] = useState("list");
  const [editingProject, setEditingProject] = useState(null);

  const fetchProjects = () => {
    fetch("https://localhost:7002/api/projects")
      .then((res) => res.json())
      .then((data) => setProjects(data))
      .catch((err) => console.error("Ошибка загрузки проектов:",
err));
  };

  useEffect(() => {
    fetchProjects();
  }, []);

  const handleDelete = async (id) => {
    if (window.confirm("Вы уверены, что хотите удалить этот
проект?")) {
      try {
        const token = localStorage.getItem("adminToken");
        const response = await fetch(
          `https://localhost:7002/api/projects/${id}`,
          {
            method: "DELETE",
            headers: { ...(token && { Authorization: `Bearer
${token}` }) },

```

```

        },
    );
    if (response.ok) {
        setProjects(projects.filter((p) => p.id !== id));
    } else {
        alert("Ошибка при удалении.");
    }
} catch (error) {
    console.error("Ошибка сети при удалении:", error);
}
}
};

const handleEdit = (project) => {
    setEditingProject(project);
    setViewMode("form");
    window.scrollTo({ top: 0, behavior: "smooth" });
};

const handleFormSuccess = (message) => {
    alert(message);
    fetchProjects();
    setEditingProject(null);
    setViewMode("list");
};

const handleFormCancel = () => {
    setEditingProject(null);
    setViewMode("list");
};

return (
    <div className="w-full bg-white p-8 rounded-[30px] shadow-sm
border border-gray-100 min-h-[600px] flex flex-col">
        <div className="flex gap-4 border-b border-gray-200 mb-8 pb-4
w-full">
            <button
                onClick={handleFormCancel}
                className={`px-6 py-2 rounded-xl font-bold transition-all
${
                    viewMode === "list"
                        ? "bg-[#C16828] text-white shadow-md"
                        : "bg-gray-100 text-[#4D5053] hover:bg-gray-200"
                }`>
                <div>
                    Список проектов
                </div>
            </button>
            <button
                onClick={() => {
                    setEditingProject(null);
                    setViewMode("form");
                }}
            </button>
        </div>
    </div>
);

```

```

    }}
    className={`px-6 py-2 rounded-xl font-bold transition-all
    ${
        viewMode === "form" && !editingProject
        ? "bg-[#C16828] text-white shadow-md"
        : "bg-gray-100 text-[#4D5053] hover:bg-gray-200"
    }}`
  >
    + Добавить новый
  </button>
</div>

<div className="w-full flex-grow flex">
  {viewMode === "list" && (
    <ProjectList
      projects={projects}
      onEdit={handleEdit}
      onDelete={handleDelete}
    />
  )}

  {viewMode === "form" && (
    <ProjectForm
      projectToEdit={editingProject}
      onSuccess={handleFormSuccess}
      onCancel={handleFormCancel}
    />
  )}
</div>
</div>
);
};

```

export default ProjectsTab;

Файл AdminLogin.jsx

```

import { useState } from "react";
import { useNavigate } from "react-router-dom";

const AdminLogin = () => {
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");
  const [error, setError] = useState("");
  const navigate = useNavigate();

  const handleLogin = async (e) => {
    e.preventDefault();
    setError("");

    try {

```

```

    const response = await
fetch("https://localhost:7002/api/auth/login", {
  method: "POST",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify({ username, password }),
});

if (response.ok) {
  const data = await response.json();
  localStorage.setItem("adminToken", data.token);

  setUsername("");
  setPassword("");

  navigate("/admin");
} else {
  setError("Неверный логин или пароль");
  setPassword("");
}
} catch (err) {
  setError("Ошибка соединения с сервером");
}
};

return (
  <div className="flex justify-center items-center h-screen bg-
gray-50">
    <div className="w-full max-w-md p-8 bg-white rounded-[20px]
shadow-xl min-h-[450px] flex flex-col">
      <h2 className="text-3xl font-bold text-center mb-8 text-
[#292F36]">
        Вход
      </h2>

      {error && <p className="text-red-500 text-center mb-
4">{error}</p>}

      <form onSubmit={handleLogin} className="flex flex-col flex-
1">
        <div className="space-y-6">
          <div>
            <label className="block text-[#4D5053] mb-2">Логин
</label>

            <input
              type="text"
              required
              value={username}
              onChange={(e) => setUsername(e.target.value)}

```

```

        className="w-full border-b-2 border-gray-300 py-2
outline-none focus:border-[#C16828] transition-colors bg-
transparent"
      />
    </div>
    <div>
      <label className="block text-[#4D5053] mb-
2">Пароль</label>
      <input
        type="password"
        required
        value={password}
        onChange={ (e) => setPassword(e.target.value)}
        className="w-full border-b-2 border-gray-300 py-2
outline-none focus:border-[#C16828] transition-colors bg-
transparent"
      />
    </div>
  </div>

  <div className="mt-auto pt-10 flex justify-center pb-2">
    <button
      type="submit"
      className="px-12 py-3 bg-[#C16828] text-white rounded-
[14px] text-lg font-semibold hover:bg-[#d6763e] hover:scale-105
transition-all duration-300 shadow-md"
    >
      Войти
    </button>
  </div>
</form>
</div>
</div>
);
};

export default AdminLogin;

```

Файл AdminPage.jsx

```

import React, { useState, useEffect } from "react";
import AdminSidebar from "../AdminSidebar";
import LeadsTab from "../LeadsTab";
import ProjectsTab from "../ProjectsTabComponents/ProjectsTab";

const AdminPage = () => {
  const [activeTab, setActiveTab] = useState("projects");
  const [leads, setLeads] = useState([]);

  useEffect(() => {

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		124

```

const fetchLeads = () => {
  const token = localStorage.getItem("adminToken");

  fetch("https://localhost:7002/api/leads", {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
      ...(token && { Authorization: `Bearer ${token}` })),
  },
  })
  .then((res) => {
    if (!res.ok) {
      throw new Error("Нет доступа или сервер вернул ошибку");
    }
    return res.json();
  })
  .then((data) => setLeads(data))
  .catch((err) => console.error("Ошибка загрузки заявок:",
err));
};

fetchLeads();

const intervalId = setInterval(fetchLeads, 15000);

return () => clearInterval(intervalId);
}, []);

const unreadCount = leads.filter((lead) => lead.status ===
"new").length;

return (
  <div className="min-h-screen bg-[#faf8f5] py-10 px-4">
    <div className="max-w-[1200px] w-full mx-auto flex flex-col
md:flex-row gap-8">

      <AdminSidebar
        activeTab={activeTab}
        setActiveTab={setActiveTab}
        unreadCount={unreadCount}
      />

      <div className="flex-1 min-w-0 flex flex-col gap-8">
        {activeTab === "projects" && <ProjectsTab />}

        {activeTab === "leads" && (
          <LeadsTab leads={leads} setLeads={setLeads} />
        )}
      </div>
    </div>
  )

```

```

        </div>
      </div>
    </div>
  );
};

export default AdminPage;

```

Файл AdminSidebar.jsx

```

import React from "react";
import { useNavigate } from "react-router-dom";

const AdminSidebar = ({ activeTab, setActiveTab, unreadCount }) => {
  const navigate = useNavigate();

  const handleLogout = () => {
    localStorage.removeItem("adminToken");
    navigate("/admin-login");
  };

  return (
    <div className="w-full md:w-[280px] shrink-0 flex flex-col gap-4 self-start">
      <div className="bg-white p-6 rounded-[25px] shadow-sm border border-gray-100 flex flex-col">
        <h2 className="text-xl font-bold mb-6 text-[#292F36] border-b pb-4">
          Администрирование
        </h2>

        <nav className="flex flex-col gap-2">
          <button
            onClick={() => setActiveTab("projects")}
            className={`w-full flex items-center justify-start gap-4 px-4 py-3 rounded-xl font-medium transition text-left ${
              activeTab === "projects"
                ? "bg-[#C16828] text-white shadow-md"
                : "text-gray-600 hover:bg-orange-50"
            }`}
          >
            <span className="text-xl shrink-0">🏠</span>
            <span className="flex-1 leading-tight">Управление проектами</span>
          </button>

          <button
            onClick={() => setActiveTab("leads")}

```

```

        className={`w-full flex items-center justify-start gap-4
px-4 py-3 rounded-xl font-medium transition text-left ${
        activeTab === "leads"
        ? "bg-[#C16828] text-white shadow-md"
        : "text-gray-600 hover:bg-orange-50"
        }}`
    >
        <span className="text-xl shrink-0">✉</span>
        <span className="flex-1 leading-tight">Заявки
клиентов</span>
        {unreadCount > 0 && (
            <span className="shrink-0 bg-red-500 text-white text-
xs font-bold w-6 h-6 flex items-center justify-center rounded-full
shadow-sm">
                {unreadCount}
            </span>
        )}
    </button>

    <button
        onClick={handleLogout}
        className="w-full flex items-center justify-start gap-4
px-4 py-3 mt-4 rounded-xl font-bold text-red-500 bg-red-50 hover:bg-
red-500 hover:text-white transition text-left shadow-sm"
    >
        <span className="text-xl shrink-0">📄</span>
        <span className="flex-1 leading-tight">Выйти из
панели</span>
    </button>

</nav>
</div>
</div>
);
};

export default AdminSidebar;

```

Файл LeadsTab.jsx

```

import React, { useState } from "react";

const LeadsTab = ({ leads, setLeads }) => {
    const [expandedLeads, setExpandedLeads] = useState([]);

    const toggleLeadStatus = async (lead) => {
        const newStatus = lead.status === "new" ? "contacted" : "new";
        const updatedLead = { ...lead, status: newStatus };

        try {
            const token = localStorage.getItem("adminToken");

```

```

    const response = await
fetch(`https://localhost:7002/api/leads/${lead.id}`, {
  method: "PUT",
  headers: {
    "Content-Type": "application/json",
    ...(token && { Authorization: `Bearer ${token}` })
  },
  body: JSON.stringify(updatedLead),
});

    if (response.ok) {
      setLeads(leads.map((l) => (l.id === lead.id ? updatedLead :
l)));
    } else {
      alert("Ошибка при обновлении статуса на сервере.");
    }
  } catch (error) {
    console.error("Ошибка сети:", error);
  }
};

const toggleExpandLead = (id) => {
  if (expandedLeads.includes(id)) {
    setExpandedLeads(expandedLeads.filter((leadId) => leadId !==
id));
  } else {
    setExpandedLeads([...expandedLeads, id]);
  }
};

return (
  <div className="w-full bg-white p-8 rounded-[30px] shadow-sm
border border-gray-100 min-h-[600px]">
    <h2 className="text-2xl font-bold mb-8 text-[#292F36]">
      Входящие заявки с сайта
    </h2>
    <div className="flex flex-col gap-4">
      {leads.map((lead) => {
        const isExpanded = expandedLeads.includes(lead.id);
        const messageText = lead.message || "";
        const isLongMessage = messageText.length > 80;
        const displayMessage =
          isExpanded || !isLongMessage
            ? messageText
            : messageText.substring(0, 80) + "...";

        const displayDate = lead.createdAt
          ? new Date(lead.createdAt).toLocaleString("ru-RU", {
              day: "2-digit",
              month: "2-digit",

```

```

        year: "numeric",
        hour: "2-digit",
        minute: "2-digit",
    })
    : "";

return (
    <div
        key={lead.id}
        className={`p-6 rounded-2xl border-2 transition w-full
${
            lead.status === "new"
            ? "border-orange-100 bg-orange-50/50"
            : "border-gray-100 bg-white"
        }`}
    >
        <div className="flex flex-col md:flex-row justify-between items-start gap-4">
            <div className="flex-1 min-w-0 w-full md:pr-4">
                <h3 className="font-bold text-lg text-[#292F36] truncate">
                    {lead.name}
                </h3>

                <div className="text-sm text-gray-500 mb-4">
                    <span className="block md:inline truncate">{lead.email}</span>
                    {displayDate && (
                        <span className="block md:inline truncate mt-1 md:mt-0 text-gray-400 md:text-gray-500">
                            <span className="hidden md:inline mx-2">•</span>
                        {displayDate}
                    </span>
                    )}
                </div>

                <div className="bg-white/50 p-4 rounded-xl border border-gray-100 text-gray-700">
                    <div className="italic whitespace-pre-wrap break-words text-justify">
                        «{displayMessage}»
                    </div>

                    {isLongMessage && (
                        <button
                            onClick={() => toggleExpandLead(lead.id)}
                            className="block mt-2 text-[#C16828] font-bold text-sm hover:underline"
                        >

```

```

        {isExpanded ? "Свернуть" : "Развернуть"}
      </button>
    )}
  </div>
</div>

<div className="shrink-0 w-full md:w-[250px] mt-4
md:mt-0">
  <button
    onClick={() => toggleLeadStatus(lead)}
    className={`w-full px-5 py-3 min-h-[48px]
rounded-xl font-bold text-sm transition border-2 ${
      lead.status === "new"
        ? "bg-white border-[#C16828] text-[#C16828]
hover:bg-[#C16828] hover:text-white"
        : "bg-green-500 border-transparent text-
white shadow-md"
      }`}
  >
    {lead.status === "new"
      ? "Пометить как обработанную"
      : "✓ Готово"}
  </button>
</div>
</div>
</div>
);
}}}
{leads.length === 0 && (
  <div className="text-center text-gray-500 italic py-10">
    Нет новых заявок
  </div>
)}
</div>
</div>
);
};

```

```
export default LeadsTab;
```

Файл Calculator.jsx

```

import { useState, useMemo, useEffect } from "react";
import SEO from "../../components/SEO/SEO";
import { Link } from "react-router-dom";
import { CALCULATOR_PRICES } from "../../components/common/data";

import { Marker, Polyline, useMap } from "react-leaflet";
import BaseMap, { customIcon } from
"../../components/common/BaseMap";

const OFFICE_COORDS = { lat: 53.195, lng: 45.0183 };

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		130

```

const TRIPS_MAP = { complex: 10, vip: 25 };
const PRICE_PER_KM = 30;

const MapAutoZoom = ({ bounds }) => {
  const map = useMap();
  useEffect(() => {

    if (bounds && bounds.length > 1) {
      try {
        map.fitBounds(bounds, { padding: [40, 40], animate: true });
      } catch (error) {
        console.warn("Ошибка зума карты:", error);
      }
    }
  }, [bounds, map]);
  return null;
};

const Calculator = () => {
  const [area, setArea] = useState(50);
  const [propertyType, setPropertyType] = useState("new");
  const [projectType, setProjectType] = useState("base");
  const [options, setOptions] = useState({
    smartHome: false,
    ventilation: false,
    approval: false,
  });

  const [address, setAddress] = useState("");
  const [distanceKm, setDistanceKm] = useState(0);
  const [isCalculatingRoute, setIsCalculatingRoute] =
useState(false);
  const [routeError, setRouteError] = useState("");
  const [isAddressCalculated, setIsAddressCalculated] =
useState(false);

  const [clientCoords, setClientCoords] = useState(null);
  const [routeCoords, setRouteCoords] = useState([]);

  const showTransport = projectType === "complex" || projectType ===
"vip";

  const totalCost = useMemo(() => {
    const { PROJECT_TYPES, PROPERTY_TYPES, OPTIONS } =
CALCULATOR_PRICES;

    let pricePerSqM =
      PROJECT_TYPES[projectType].price +
      PROPERTY_TYPES[propertyType].price;

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		131

```

    if (options.smartHome) pricePerSqM +=
OPTIONS.smartHome.pricePerSqM;
    if (options.ventilation) pricePerSqM +=
OPTIONS.ventilation.pricePerSqM;

    let total = pricePerSqM * area;
    if (options.approval) total += OPTIONS.approval.fixedPrice;

    if (showTransport && distanceKm > 0) {
        const tripsCount = TRIPS_MAP[projectType] || 0;
        const transportCost = distanceKm * 2 * tripsCount *
PRICE_PER_KM;
        total += transportCost;
    }

    return Math.round(total);
}, [area, propertyType, projectType, options, showTransport,
distanceKm]);

const currentTransportCost = useMemo(() => {
    if (!showTransport || distanceKm === 0) return 0;
    const tripsCount = TRIPS_MAP[projectType] || 0;
    return Math.round(distanceKm * 2 * tripsCount * PRICE_PER_KM);
}, [showTransport, distanceKm, projectType]);

const toggleOption = (optionName) => {
    setOptions((prev) => ({
        ...prev,
        [optionName]: !prev[optionName],
    }));
};

const handleCalculateRoute = async () => {
    if (!address.trim()) {
        setRouteError("Пожалуйста, введите адрес объекта.");
        return;
    }

    setIsCalculatingRoute(true);
    setRouteError("");
    setIsAddressCalculated(false);

    try {

        const geoRes = await fetch(
            `https://nominatim.openstreetmap.org/search?format=json&q=${
encodeURIComponent(address)}`,
        );
        const geoData = await geoRes.json();

        if (!geoData || geoData.length === 0) {

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		132

```

        setRouteError(
            "Адрес не найден. Попробуйте написать город и улицу
точнее.",
        );
        setIsCalculatingRoute(false);
        return;
    }

    const clientLat = parseFloat(geoData[0].lat);
    const clientLng = parseFloat(geoData[0].lon);

    const applyFallback = () => {
        const R = 6371;
        const dLat = (clientLat - OFFICE_COORDS.lat) * (Math.PI /
180);
        const dLon = (clientLng - OFFICE_COORDS.lng) * (Math.PI /
180);
        const a =
            Math.sin(dLat / 2) * Math.sin(dLat / 2) +
            Math.cos(OFFICE_COORDS.lat * (Math.PI / 180)) *
                Math.cos(clientLat * (Math.PI / 180)) *
                Math.sin(dLon / 2) *
                Math.sin(dLon / 2);
        const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
        const distanceStraight = R * c;

        setDistanceKm(distanceStraight * 1.2);
        setClientCoords({ lat: clientLat, lng: clientLng });
        setRouteCoords([
            [OFFICE_COORDS.lat, OFFICE_COORDS.lng],
            [clientLat, clientLng],
        ]);
        setIsAddressCalculated(true);
    };

    try {

        const osrmRes = await fetch(
            `https://router.project1-
osrm.org/route/v1/driving/${OFFICE_COORDS.lng},${OFFICE_COORDS.lat};
${clientLng},${clientLat}?overview=full&geometries=geojson`,
        );
        const osrmData = await osrmRes.json();

        if (
            osrmData.code === "Ok" &&
            osrmData.routes &&
            osrmData.routes.length > 0
        ) {
            const route = osrmData.routes[0];

```

```

        const calculatedDistanceKm = route.distance / 1000;
        const lineCoords = route.geometry.coordinates.map((coord)
=> [
            coord[1],
            coord[0],
        ]);

        setDistanceKm(calculatedDistanceKm);
        setClientCoords({ lat: clientLat, lng: clientLng });
        setRouteCoords(lineCoords);
        setIsAddressCalculated(true);
    } else {

        applyFallback();
    }
    } catch (osrmError) {
        console.warn("OSRM не отвечает, используем прямой расчет:",
osrmError);

        applyFallback();
    }
    } catch (err) {
        console.error(err);
        setRouteError("Ошибка сети при поиске адреса.");
    }

    setIsCalculatingRoute(false);
};

const handleAddressChange = (e) => {
    setAddress(e.target.value);
    setIsAddressCalculated(false);
    setDistanceKm(0);
    setRouteCoords([]);
    setClientCoords(null);
};

const { PROJECT_TYPES, PROPERTY_TYPES, OPTIONS } =
CALCULATOR_PRICES;

let messageToPass = `Здравствуйте! Меня интересует дизайн-
проект.\n\nМой предварительный расчет:\nПлощадь: ${area} м²\nОбъект:
${PROPERTY_TYPES[propertyType].name}\nТариф:
${PROJECT_TYPES[projectType].name}\nДоп. опции: ${options.smartHome ?
"Умный дом, " : ""}${options.ventilation ? "Вентиляция, " :
""}${options.approval ? "Согласование" : ""}`;

if (showTransport && isAddressCalculated) {
    messageToPass += ` \nАдрес: ${address} (Логистика:
${currentTransportCost.toLocaleString("ru-RU")} ₺)`;
}

```

					ПГУ 1.090301.49.001 ПЗ	Лист
						134
Изм	Лист	№ докум.	Подпись	Дата		

```

    messageToPass += ` \nПримерная стоимость:
    ${totalCost.toLocaleString("ru-RU")} ₺ \n \nПрошу связаться со мной.`;

    return (
      <div className="bg-[#FAF8F5] min-h-screen pb-[150px]">
        <SEO
          title="Калькулятор проекта - Design Studio"
          description="Рассчитайте примерную стоимость дизайн-проекта
вашей квартиры онлайн."
          canonicalUrl="https://story-home-six.vercel.app/calculator"
        />

        <div className="pt-[50px] lg:pt-[100px] pb-[80px] px-5 bg-
[#FAF8F5]">
          <div className="max-w-[1000px] mx-auto text-center">
            <h1 className="text-[40px] lg:text-[60px] font-serif text-
[#3A2B24] leading-[1.1] mb-6">
              Рассчитайте стоимость <br className="hidden sm:block" />
              <span className="italic font-light">вашего
проекта</span>
            </h1>
            <p className="text-[18px] text-[#5A4A42] leading-relaxed
max-w-[700px] mx-auto">
              Сформируйте предварительную смету, выбрав нужный пакет
проектирования, площадь и необходимые инженерные разделы.
            </p>
          </div>
        </div>

        <div className="max-w-[1300px] mx-auto px-5 lg:px-0">
          <div className="flex flex-col lg:flex-row gap-10">

            <div className="flex-1 bg-white p-8 lg:p-12 rounded-[40px]
shadow-sm border border-gray-100 h-fit">

              <div className="grid grid-cols-1 md:grid-cols-2 gap-10
mb-10">

                <div>
                  <label className="text-[20px] font-bold text-
[#3A2B24] block mb-4">
                    Площадь объекта (м²)
                  </label>
                  <input
                    type="number"
                    min="10"
                    max="1000"
                    value={area}
                    onChange={(e) => setArea(Number(e.target.value))}

```

```

        className="w-full px-6 py-4 rounded-xl border-2
border-gray-100 focus:border-[#423127] outline-none text-2xl font-
serif text-[#3A2B24] transition-colors bg-[#FAF8F5]"
      />
    </div>

    <div>
      <label className="text-[20px] font-bold text-
[#3A2B24] block mb-4">
        Тип объекта
      </label>
      <div className="flex flex-col gap-3">
        {Object.entries(PROPERTY_TYPES).map(([key, data])
=> (
          <button
            key={key}
            onClick={() => setPropertyType(key)}
            className={`py-3 px-5 rounded-xl border-2
font-bold transition-all text-left ${
              propertyType === key
                ? "border-[#423127] bg-[#423127] text-
white shadow-md"
                : "border-gray-100 bg-[#FAF8F5] text-
[#5A4A42] hover:border-[#423127]"
            }`}
          >
            {data.name}
          </button>
        ))}
      </div>
    </div>
  </div>

  <div className="mb-10 border-t border-gray-100 pt-10">
    <label className="text-[20px] font-bold text-[#3A2B24]
block mb-6">
      Уровень проектирования (Тариф)
    </label>
    <div className="grid grid-cols-1 md:grid-cols-2 gap-
4">
      {Object.entries(PROJECT_TYPES).map(([key, data]) =>
(
        <button
          key={key}
          onClick={() => setProjectType(key)}
          className={`py-5 px-5 rounded-xl border-2 flex
flex-col items-center gap-2 transition-all ${
            projectType === key
              ? "border-[#423127] bg-[#423127] text-white
shadow-md"

```

```

        : "border-gray-100 bg-[#FAF8F5] text-
[#5A4A42] hover:border-[#423127]"
        }` }
    >
    <span className="font-bold text-
[18px]">{data.name}</span>
    <span
        className={`text-sm ${projectType === key ?
"text-[#D4C8C1]" : "text-[#C16828]"}` }
    >
        or {data.price.toLocaleString()} ₸ / м²
    </span>
</button>
    )} }
</div>
</div>

<div className="border-t border-gray-100 pt-10">
    <label className="text-[20px] font-bold text-[#3A2B24]
block mb-6">
        Инженерные проекты и согласование
    </label>
    <div className="flex flex-col gap-4">
        {Object.entries(OPTIONS).map(([key, data]) => (
            <div
                key={key}
                onClick={() => toggleOption(key)}
                className={`flex items-center justify-between
cursor-pointer p-5 rounded-xl border-2 transition-all ${
                    options[key]
                        ? "border-[#423127] bg-white shadow-sm"
                        : "border-gray-100 bg-[#FAF8F5]"
                }
                hover:border-[#423127]"
            }` }
            >
                <div className="flex items-center gap-4">
                    <div
                        className={`w-6 h-6 rounded flex items-
center justify-center border-2 transition-colors ${
                            options[key]
                                ? "bg-[#C16828] border-[#C16828]"
                                : "border-gray-300 bg-white"
                        }` }
                    >
                        {options[key] && (
                            <svg
                                xmlns="http://www.w3.org/2000/svg"
                                className="h-4 w-4 text-white"
                                viewBox="0 0 20 20"

```

```

        fill="currentColor"
      >
        <path
          fillRule="evenodd"
          d="M16.707 5.293a1 1 0 010 1.414l-8
8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8 12.586l7.293-7.293a1 1
0 011.414 0z"
          clipRule="evenodd"
        />
      </svg>
    )}
  </div>

  <input
    type="checkbox"
    className="hidden"
    checked={options[key]}
    readOnly
  />

  <span className="text-[17px] font-bold text-
[#3A2B24] select-none">
    {data.name}
  </span>
</div>

  <span
    className={`text-sm font-bold select-none
${options[key] ? "text-[#C16828]" : "text-[#5A4A42]"}`}
  >
    {data.pricePerSqM
      ? `+ ${data.pricePerSqM} ₪/м²`
      : `фикс. ${data.fixedPrice.toLocaleString()}
₪`}
  </span>
</div>
)})}
</div>
</div>

{showTransport && (
  <div className="border-t border-gray-100 pt-10 mt-10
animate-fade-in">
    <label className="text-[20px] font-bold text-
[#3A2B24] block mb-3">
      Транспортное сопровождение проекта
    </label>
    <p className="text-[#5A4A42] mb-6 text-[15px]
leading-relaxed bg-[#FAF8F5] p-5 rounded-xl border border-gray-100">

```

```

        Для тарифа
<b>«{PROJECT_TYPES[projectType].name}»</b>{ " "}
        предусмотрено регулярное посещение объекта
(около{ " "}
        <b>{TRIPS_MAP[projectType]} выездов</b>). Введите
адрес, чтобы
        рассчитать стоимость логистики.
</p>

<div className="flex flex-col sm:flex-row gap-3 w-
full">
    <input
        type="text"
        placeholder="Ваш адрес, например: Москва, Ленина
10"
        value={address}
        onChange={handleAddressChange}
        className="flex-1 px-5 py-4 rounded-xl border-2
border-gray-100 bg-[#FAF8F5] focus:bg-white focus:border-[#423127]
outline-none text-[16px] text-[#3A2B24]"
    />
    <button
        type="button"
        onClick={handleCalculateRoute}
        disabled={isCalculatingRoute}
        className="shrink-0 px-8 py-4 bg-[#C16828] text-
white rounded-xl font-bold hover:bg-[#a35620] transition
disabled:opacity-50"
    >
        {isCalculatingRoute ? "Считаем..." : "Рассчитать
логистику"}
    </button>
</div>

{routeError && (
    <p className="text-red-500 text-sm mt-3 font-
medium">
        {routeError}
    </p>
)}

{isAddressCalculated &&
    distanceKm > 0 &&
    routeCoords.length > 0 && (
        <div className="mt-8 animate-fade-in">
            <div className="h-[300px] w-full mb-6 rounded-
2xl overflow-hidden shadow-md relative z-0">
                <BaseMap
                    center={[OFFICE_COORDS.lat,
OFFICE_COORDS.lng]}
                    zoom={12}

```

```

        scrollWheelZoom={false}
        className="w-full h-full z-0"
      >
        <Marker
          position={[OFFICE_COORDS.lat,
OFFICE_COORDS.lng]}
          icon={customIcon}
        />
        <Marker
          position={[clientCoords.lat,
clientCoords.lng]}
          icon={customIcon}
        />
        <Polyline
          positions={routeCoords}
          color="#C16828"
          weight={4}
          opacity={0.8}
        />
        <MapAutoZoom bounds={routeCoords} />
      </BaseMap>
    </div>

    <div className="bg-[#FAF8F5] p-6 rounded-2xl
border border-[#E6D8C6] flex flex-col sm:flex-row justify-between
items-center gap-4">
      <div className="flex flex-col gap-1 text-
[15px] text-[#5A4A42]">
        <p>
          📍 Расстояние:
        <b>{distanceKm.toFixed(1)} км</b>
        </p>
        <p>
          🚚 Выездов: <b>{TRIPS_MAP[projectType]}
раз</b>
        </p>
      </div>
      <div className="text-center sm:text-right
bg-white px-6 py-3 rounded-xl border border-gray-100 shadow-sm">
        <p className="text-xs text-gray-500
uppercase tracking-widest mb-1">
          Итого за логистику
        </p>
        <p className="text-[#C16828] font-bold
text-[22px]">
          +
          {currentTransportCost.toLocaleString("ru-RU")} ₺
        </p>
      </div>
    </div>
  </div>

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		140

```

    })
  </div>
})
</div>

```

```

    <div className="w-full lg:w-[450px] shrink-0 h-fit sticky
top-10 z-10">
      <div className="bg-[#423127] p-8 lg:p-10 rounded-[40px]
shadow-2xl flex flex-col text-white">
        <h3 className="text-[30px] font-serif border-b border-
[#5A4436] pb-5 mb-8 text-white">
          Ваша смета
        </h3>

        <div className="flex flex-col gap-5 mb-10 text-
[#D4C8C1] text-[18px]">
          <div className="flex justify-between items-end
border-b border-[#5A4436] border-dashed pb-2">
            <span>Площадь:</span>
            <span className="font-bold text-white text-
[20px]">
              {area} м²
            </span>
          </div>

          <div className="flex justify-between items-end
border-b border-[#5A4436] border-dashed pb-2">
            <span>Объект:</span>
            <span className="font-bold text-white text-right">
              {PROPERTY_TYPES[propertyType].name}
            </span>
          </div>

          <div className="flex justify-between items-end
border-b border-[#5A4436] border-dashed pb-2">
            <span>Тариф:</span>
            <span className="font-bold text-white text-right">
              {PROJECT_TYPES[projectType].name}
            </span>
          </div>

          {showTransport &&
            isAddressCalculated &&
            currentTransportCost > 0 && (
              <div className="flex justify-between items-end
border-b border-[#5A4436] border-dashed pb-2">
                <span>Логистика:</span>
                <span className="font-bold text-white text-
right">

```

```

        {currentTransportCost.toLocaleString("ru-
RU"))} ₺

        </span>
      </div>
    )}

    {(options.smartHome ||
      options.ventilation ||
      options.approval) && (
      <div className="mt-4 bg-[#2C201A] p-5 rounded-2xl
border border-[#5A4436]">
        <span className="block mb-3 text-sm uppercase
tracking-widest text-[#D4C8C1]">
          Доп. услуги:
        </span>
        <ul className="text-[16px] flex flex-col gap-2
text-white">
          {options.smartHome && (
            <li className="flex items-center gap-2">
              <span className="text-[#C16828]">✓</span>
            </li>
          )}
          {options.ventilation && (
            <li className="flex items-center gap-2">
              <span className="text-[#C16828]">✓</span>
            </li>
          )}
          {options.approval && (
            <li className="flex items-center gap-2">
              <span className="text-[#C16828]">✓</span>
            </li>
          )}
        </ul>
      </div>
    )}

    <div className="bg-white text-center p-6 rounded-2xl
mb-8">
      <p className="text-sm text-gray-500 uppercase
tracking-widest mb-1 font-bold">
        Итоговая стоимость
      </p>
      <p className="text-[40px] lg:text-[45px] font-serif
font-bold text-[#C16828] leading-none">
        {totalCost.toLocaleString("ru-RU")} ₺

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		142

```

        </p>
      </div>

      <p className="text-xs text-[#D4C8C1] text-center mb-8 opacity-70 leading-relaxed">
        * Расчет ориентировочный. Финальная стоимость
        фиксируется в
        договоре после выезда специалиста.
      </p>

      <div className="w-full">
        <Link
          to="/contact"
          state={{ prefilledMessage: messageToPass }}
          className="block w-full"
        >
          <button className="w-full py-5 bg-[#C16828] text-white font-bold text-lg rounded-xl hover:bg-white hover:text-[#423127] transition-colors shadow-lg">
            Оставить заявку
          </button>
        </Link>
      </div>
    </div>
  </div>
</div>
</div>
</div>
);
};

export default Calculator;

```

Файл Contact.jsx

```

import { useState, useRef, useEffect } from "react";
import { useLocation } from "react-router-dom";
import SEO from "../../components/SEO/SEO";

const Contact = () => {
  const location = useLocation();
  const prefilledMessage = location.state?.prefilledMessage || "";

  const [formData, setFormData] = useState({
    name: "",
    email: "",
    message: prefilledMessage,
  });

  const [isSent, setIsSent] = useState(false);
  const textareaRef = useRef(null);

```

					ПГУ 1.090301.49.001 ПЗ	Лист
						143
Изм	Лист	№ докум.	Подпись	Дата		

```

useEffect(() => {
  if (textareaRef.current) {
    textareaRef.current.style.height = "auto";
    textareaRef.current.style.height =
`${textareaRef.current.scrollHeight}px`;
  }
}, [formData.message]);

const handleSubmit = async (e) => {
  e.preventDefault();

  const dataToSend = {
    ...formData,
    status: "new",
  };

  try {
    const response = await
fetch("https://localhost:7002/api/leads", {
  method: "POST",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify(dataToSend),
});

    if (response.ok) {
      setIsSent(true);
      setFormData({ name: "", email: "", message: "" });
      setTimeout(() => setIsSent(false), 5000);
    } else {
      alert("Произошла ошибка при отправке. Попробуйте позже.");
    }
  } catch (error) {
    console.error("Ошибка сети:", error);
    alert("Не удалось связаться с сервером.");
  }
};

return (
  <div className="bg-[#FAF8F5] min-h-screen">
    <SEO
      title="Контакты - Design Studio"
      description="Свяжитесь с нами для заказа дизайн-проекта.
Наши контакты, адрес студии и форма обратной связи."
      keywords="контакты студии дизайна, заказать дизайн проект,
адрес студии дизайна"
      canonicalUrl="https://story-home-six.vercel.app/contact"
    />

    <div className="pt-[50px] lg:pt-[100px] pb-[60px] px-5 text-
center">

```

```

<div className="max-w-[1000px] mx-auto">

  <h1 className="text-[40px] lg:text-[60px] font-serif text-
[#3A2B24] leading-[1.1] mb-6">
    Давайте обсудим <br className="hidden sm:block"/>
    <span className="italic font-light">ваш будущий
интерьер</span>
  </h1>
  <p className="text-[18px] text-[#5A4A42] leading-relaxed
max-w-[600px] mx-auto">
    Оставьте заявку, и мы свяжемся с вами в ближайшее время,
чтобы договориться о встрече или онлайн-консультации.
  </p>
</div>
</div>

<section className="pb-[120px] px-5">
  <div className="max-w-[1200px] mx-auto">
    <div className="grid grid-cols-1 lg:grid-cols-12 gap-12
lg:gap-20">

      <div className="lg:col-span-5 flex flex-col justify-
center">

        <div className="bg-white p-10 lg:p-12 rounded-[40px]
shadow-sm border border-gray-100">

          <h2 className="text-[28px] font-serif text-[#3A2B24]
mb-10">Контактные данные</h2>

          <div className="space-y-8">
            <div className="flex items-start gap-5">
              <div className="w-12 h-12 bg-[#FAF8F5] rounded-
full flex items-center justify-center shrink-0 border border-gray-
100 text-[#C16828]">
                <svg xmlns="http://www.w3.org/2000/svg"
fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
stroke="currentColor" className="w-6 h-6">
                  <path strokeLinecap="round"
strokeLinejoin="round" d="M15 10.5a3 3 0 11-6 0 3 3 0 016 0z" />
                  <path strokeLinecap="round"
strokeLinejoin="round" d="M19.5 10.5c0 7.142-7.5 11.25-7.5 11.25S4.5
17.642 4.5 10.5a7.5 7.5 0 1115 0z" />
                </svg>
              </div>
              <div>
                <p className="text-sm text-gray-400 uppercase
tracking-widest font-bold mb-1">Наш офис</p>
                <p className="text-[18px] text-[#5A4A42] font-
medium">г. Пенза, ул. Московская, 15</p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </section>

```

```

        <div className="flex items-start gap-5">
            <div className="w-12 h-12 bg-[#FAF8F5] rounded-
full flex items-center justify-center shrink-0 border border-gray-
100 text-[#C16828]">
                <svg xmlns="http://www.w3.org/2000/svg"
fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
stroke="currentColor" className="w-6 h-6">
                    <path strokeLinecap="round"
strokeLinejoin="round" d="M2.25 6.75c0 8.284 6.716 15 15
15h2.25a2.25 2.25 0 02.25-2.25v-1.372c0-.516-.351-.966-.852-1.091l-
4.423-1.106c-.44-.11-.902.055-1.173.417l-.97 1.293c-.282.376-
.769.542-1.21.38a12.035 12.035 0 01-7.143-7.143c-.162-.441.004-
.928.38-1.211l1.293-.97c.363-.271.527-.734.417-1.173L6.963
3.102a1.125 1.125 0 00-1.091-.852H4.5A2.25 2.25 0 002.25 4.5v2.25z"
/>
                </svg>
            </div>
            <div>
                <p className="text-sm text-gray-400 uppercase
tracking-widest font-bold mb-1">Телефон</p>
                <p className="text-[18px] text-[#5A4A42] font-
medium">+7 (999) 123-45-67</p>
            </div>
        </div>

        <div className="flex items-start gap-5">
            <div className="w-12 h-12 bg-[#FAF8F5] rounded-
full flex items-center justify-center shrink-0 border border-gray-
100 text-[#C16828]">
                <svg xmlns="http://www.w3.org/2000/svg"
fill="none" viewBox="0 0 24 24" strokeWidth={1.5}
stroke="currentColor" className="w-6 h-6">
                    <path strokeLinecap="round"
strokeLinejoin="round" d="M21.75 6.75v10.5a2.25 2.25 0 01-2.25
2.25h-15a2.25 2.25 0 01-2.25-2.25V6.75m19.5 0A2.25 2.25 0 0019.5
4.5h-15a2.25 2.25 0 00-2.25 2.25m19.5 0v.243a2.25 2.25 0 01-1.07
1.916l-7.5 4.615a2.25 2.25 0 01-2.36 0L3.32 8.91a2.25 2.25 0 01-
1.07-1.916V6.75" />
                </svg>
            </div>
            <div>
                <p className="text-sm text-gray-400 uppercase
tracking-widest font-bold mb-1">E-mail</p>
                <p className="text-[18px] text-[#5A4A42] font-
medium">hello@designstudio.ru</p>
            </div>
        </div>
    </div>
</div>

```

```

        <div className="lg:col-span-7">
          <form onSubmit={handleSubmit} className="flex flex-col
space-y-8">
            <div className="grid grid-cols-1 md:grid-cols-2 gap-
8">
              <div className="flex flex-col gap-2">
                <label className="text-xs font-bold text-gray-
400 uppercase tracking-widest ml-1">Ваше имя</label>
                <input
                  type="text"
                  required
                  value={formData.name}
                  onChange={(e) => setFormData({ ...formData,
name: e.target.value })}
                  placeholder="Александр"
                  className="w-full bg-transparent border-b-2
border-gray-200 py-3 outline-none focus:border-[#423127] text-[18px]
transition-colors"
                />
              </div>
              <div className="flex flex-col gap-2">
                <label className="text-xs font-bold text-gray-
400 uppercase tracking-widest ml-1">Email адрес</label>
                <input
                  type="email"
                  required
                  value={formData.email}
                  onChange={(e) => setFormData({ ...formData,
email: e.target.value })}
                  placeholder="example@mail.ru"
                  className="w-full bg-transparent border-b-2
border-gray-200 py-3 outline-none focus:border-[#423127] text-[18px]
transition-colors"
                />
              </div>
            </div>

            <div className="flex flex-col gap-2">
              <label className="text-xs font-bold text-gray-400
uppercase tracking-widest ml-1">Сообщение</label>
              <textarea
                ref={textareaRef}
                rows="1"
                required
                value={formData.message}
                onChange={(e) => setFormData({ ...formData,
message: e.target.value })}
                placeholder="Расскажите о вашем проекте..."
              />
            </div>

```

```

        className="w-full bg-transparent border-b-2
border-gray-200 py-3 outline-none focus:border-[#423127] text-[18px]
transition-colors resize-none overflow-hidden min-h-[50px]"
    ></textarea>
</div>

<div className="pt-8">
    <button
        type="submit"
        className="group flex items-center gap-4 bg-
[#423127] text-white px-10 py-5 rounded-xl font-bold hover:bg-
[#2C201A] transition-all shadow-xl"
    >
        Отправить сообщение
        <svg xmlns="http://www.w3.org/2000/svg"
fill="none" viewBox="0 0 24 24" strokeWidth={2.5}
stroke="currentColor" className="w-5 h-5 group-hover:translate-x-2
transition-transform">
            <path strokeLinecap="round"
strokeLinejoin="round" d="M13.5 4.5L21 12m0 0l-7.5 7.5M21 12H3" />
        </svg>
    </button>
</div>
</form>
</div>

</div>
</div>
</section>

{isSent && (
    <div className="fixed inset-0 z-[100] flex items-center
justify-center px-5">
        <div className="absolute inset-0 bg-[#3A2B24]/40 backdrop-
blur-sm" onClick={() => setIsSent(false)}></div>
        <div className="relative bg-white px-8 lg:px-16 py-12
lg:py-16 rounded-[40px] shadow-2xl flex flex-col items-center text-
center max-w-[500px] animate-fade-in">
            <div className="w-20 h-20 bg-[#FAF8F5] text-[#C16828]
rounded-full flex items-center justify-center font-bold text-4xl mb-
6 border border-gray-100">
                ✓
            </div>
            <h4 className="text-[30px] font-serif text-[#3A2B24] mb-
4">Сообщение отправлено!</h4>
            <p className="text-[#5A4A42] text-[17px] mb-10 leading-
relaxed">
                Благодарим за обращение. Наш ведущий дизайнер свяжется
с вами в течение рабочего дня.
            </p>
            <button

```

```

        onClick={() => setIsSent(false)}
        className="w-full py-4 bg-[#423127] text-white font-
bold rounded-xl hover:bg-[#2C201A] transition-colors"
      >
        Закрыть
      </button>
    </div>
  </div>
)}
</div>
);
};

export default Contact;

```

Файл ErrorPage.jsx

```

import { MdArrowBack } from "react-icons/md";
import { Link } from "react-router-dom";
import ErrorImage from "../../assets/errorPage.png";
import SEO from "../../components/SEO/SEO";

const ErrorPage = () => {
  return (
    <div className="bg-[#FAF8F5] min-h-screen flex flex-col justify-
center items-center pb-[100px] px-5 pt-[80px] lg:pt-[120px]">
      <SEO
        title="Страница не найдена - Design Studio"
        description="К сожалению, запрашиваемая страница не
найдена."
      />

      <div className="max-w-[1200px] w-full mx-auto grid grid-cols-1
lg:grid-cols-2 gap-12 lg:gap-20 items-center">

        <div className="flex flex-col items-center lg:items-start
text-center lg:text-left order-2 lg:order-1">
          <h1 className="text-[120px] lg:text-[180px] font-serif
font-bold text-[#C16828] leading-none mb-4 drop-shadow-sm">
            404
          </h1>
          <h2 className="text-[30px] lg:text-[45px] font-serif text-
[#3A2B24] leading-tight mb-6">
            Кажется, вы <span className="italic font-
light">заблудились</span>
          </h2>
          <p className="text-[#5A4A42] text-[18px] lg:text-[20px]
mb-10 max-w-[500px] leading-relaxed">
            Страница, которую вы ищете, была удалена, перенесена или
никогда не существовала. Давайте вернемся к нашим прекрасным
интерьерам.
          </p>

```

```

        <Link to="/">
            <button className="px-10 py-5 bg-[#423127] text-white
font-bold rounded-xl hover:bg-[#C16828] transition-colors shadow-lg
flex items-center gap-3">
                <MdArrowBack className="text-2xl" />
                На главную страницу
            </button>
        </Link>
    </div>

    <div className="w-full h-[400px] lg:h-[600px] rounded-[40px]
overflow-hidden shadow-2xl order-1 lg:order-2">
        <img
            loading="lazy"
            src={ErrorImage}
            alt="Ошибка 404 - интерьер"
            className="h-full w-full object-cover"
        />
    </div>

</div>
</div>
);
};

export default ErrorPage;

```

Файл Home.jsx

```

import { Link } from "react-router-dom";
import SEO from "../../components/SEO/SEO";
import heroImg from "../../assets/7.jpg";
import blueprintImg from "../../assets/10.jpg";
import srv1 from "../../assets/3.jpg";
import srv2 from "../../assets/4.jpg";
import srv3 from "../../assets/5.jpg";
import srv4 from "../../assets/6.jpg";

const Home = () => {
    const features = [
        {
            id: 1,
            title: "Индивидуальный дизайн",
            desc: "Создан под ваш вкус и стиль жизни",
            icon: (
                <svg
                    xmlns="http://www.w3.org/2000/svg"
                    fill="none"
                    viewBox="0 0 24 24"
                    strokeWidth={1.5}
                    stroke="currentColor"

```

```

        className="w-8 h-8 text-[#C16828]"
      >
        <path
          strokeLinecap="round"
          strokeLinejoin="round"
          d="M16.5 10.5V6.75a4.5 4.5 0 10-9 0v3.75m-.75
11.25h10.5a2.25 2.25 0 002.25-2.25v-6.75a2.25 2.25 0 00-2.25-
2.25H6.75a2.25 2.25 0 00-2.25 2.25v6.75a2.25 2.25 0 002.25 2.25z"
        />
      </svg>
    ),
  },
  {
    id: 2,
    title: "Качественные материалы",
    desc: "Используем только лучшие бренды",
    icon: (
      <svg
        xmlns="http://www.w3.org/2000/svg"
        fill="none"
        viewBox="0 0 24 24"
        strokeWidth={1.5}
        stroke="currentColor"
        className="w-8 h-8 text-[#C16828]"
      >
        <path
          strokeLinecap="round"
          strokeLinejoin="round"
          d="M9 12.75L11.25 15 15 9.75m-3-7.036A11.959 11.959 0
013.598 6 11.99 11.99 0 003 9.749c0 5.592 3.824 10.29 9 11.623
5.176-1.332 9-6.03 9-11.622 0-1.31-.21-2.571-.598-3.751h-.152c-3.196
0-6.1-1.248-8.25-3.285z"
        />
      </svg>
    ),
  },
  {
    id: 3,
    title: "Команда экспертов",
    desc: "Опытные архитекторы и дизайнеры",
    icon: (
      <svg
        xmlns="http://www.w3.org/2000/svg"
        fill="none"
        viewBox="0 0 24 24"
        strokeWidth={1.5}
        stroke="currentColor"
        className="w-8 h-8 text-[#C16828]"
      >
        <path
          strokeLinecap="round"

```

```

        strokeLinejoin="round"
        d="M15 19.128a9.38 9.38 0 002.625.372 9.337 9.337 0
004.121-.952 4.125 4.125 0 00-7.533-2.493M15 19.128v-.003c0-1.113-
.285-2.16-.786-3.07M15 19.128v.106A12.318 12.318 0 018.624 21c-2.331
0-4.512-.645-6.374-1.766l-.001-.109a6.375 6.375 0 0111.964-3.07M12
6.375a3.375 3.375 0 11-6.75 0 3.375 3.375 0 016.75 0zm8.25
2.25a2.625 2.625 0 11-5.25 0 2.625 2.625 0 015.25 0z"
    />
</svg>
),
},
{
  id: 4,
  title: "Поддержка от А до Я",
  desc: "От концепта до финальной уборки",
  icon: (
    <svg
      xmlns="http://www.w3.org/2000/svg"
      fill="none"
      viewBox="0 0 24 24"
      strokeWidth={1.5}
      stroke="currentColor"
      className="w-8 h-8 text-[#C16828]"
    >
      <path
        strokeLinecap="round"
        strokeLinejoin="round"
        d="M9 12h3.75M9 15h3.75M9 18h3.75m3 .75H18a2.25 2.25 0
002.25-2.25V6.108c0-1.135-.845-2.098-1.976-2.192a48.424 48.424 0 00-
1.123-.08m-5.801 0c-.065.21-.1.433-.1.664
0 .414.336.75.75.75h4.5a.75.75 0 00.75-.75 2.25 2.25 0 00-.1-.664m-
5.8 0A2.251 2.251 0 0113.5 2.25H15c1.012 0 1.867.668 2.15 1.586m-5.8
0c-.376.023-.75.05-1.124.08C9.095 4.01 8.25 4.973 8.25 6.108V8.25m0
0H4.875c-.621 0-1.125.504-1.125 1.125v11.25c0 .621.504 1.125 1.125
1.125h9.75c.621 0 1.125-.504 1.125-1.125V9.375c0-.621-.504-1.125-
1.125-1.125H8.25zM6.75 12h.008v.008H6.75V12zm0
3h.008v.008H6.75V15zm0 3h.008v.008H6.75V18z"
      />
    </svg>
  ),
},
];

const services = [
  {
    id: 1,
    title: "Планировочный",
    desc: "Основа вашего комфорта: замеры, варианты перепланировки
и расстановка мебели.",
    price: "от 1 500 ₽ / м²",
    img: srv1,
  }

```

```

    },
    {
      id: 2,
      title: "Базовый проект",
      desc: "Планировка плюс вся необходимая рабочая документация для строителей.",
      price: "от 3 500 Р / м²",
      img: srv3,
    },
    {
      id: 3,
      title: "Комплексный",
      desc: "Полный дизайн-проект с 3D-визуализацией и подбором чистовых материалов.",
      price: "от 5 000 Р / м²",
      img: srv2,
    },
    {
      id: 4,
      title: "VIP под ключ",
      desc: "Проект, авторский надзор, полная комплектация объекта и управление стройкой.",
      price: "от 8 000 Р / м²",
      img: srv4,
    },
  ],

  return (
    <div className="bg-[#FAF8F5]">
      <SEO
        title="Студия Дизайна - Создаем идеальные интерьеры"
        description="Профессиональный дизайн интерьера квартир и домов. Разработка планировки, 3D визуализация, авторский надзор и ремонт под ключ."
        canonicalUrl="https://story-home-six.vercel.app/"
      />

      <div className="relative pt-[120px] lg:pt-[100px] pb-[50px] lg:pb-[250px] px-5 bg-gradient-to-br from-[#FAF8F5] to-[#f2ebe1]">
        <div className="max-w-[1300px] mx-auto flex flex-col lg:flex-row items-center gap-12 relative z-10">
          <div className="w-full lg:w-1/2 text-center lg:text-left">
            <h3 className="text-[#C16828] font-bold tracking-widest uppercase text-sm mb-4">
              Проектирование пространств
            </h3>

            <h1 className="text-[45px] lg:text-[70px] font-serif text-[#3A2B24] leading-[1.1] mb-6">
              Интерьеры, <br />

```

```

        <span className="italic font-
light">раскрывающие</span> <br />
        индивидуальность
    </h1>
    <p className="text-[18px] lg:text-[20px] text-[#5A4A42]
leading-relaxed mb-10 max-w-[500px] mx-auto lg:mx-0">
        Мы создаем вневременные интерьеры, которые отражают
вашу личность
        и поднимают повседневную жизнь на новый уровень
комфорта.
    </p>
    <div className="flex flex-col sm:flex-row items-center
gap-5 justify-center lg:justify-start">
        <Link to="/projects">

            <button className="px-8 py-4 bg-[#423127] text-white
font-bold rounded-xl hover:bg-[#2C201A] transition-all shadow-xl">
                Смотреть портфолио →
            </button>
        </Link>
        <Link to="/services">
            <button className="px-8 py-4 bg-white border border-
gray-200 text-[#3A2B24] font-bold rounded-xl hover:border-[#C16828]
hover:text-[#C16828] transition-all shadow-sm">
                Наши услуги
            </button>
        </Link>
    </div>
</div>

<div className="w-full lg:w-1/2 relative h-[450px] lg:h-
[650px]">
    <img
        src={heroImg}
        alt="Интерьер офиса студии дизайна StoryHome"
        className="w-full h-full object-cover rounded-[40px]
shadow-2xl border-[10px] border-white"
    />
</div>
</div>

<div className="absolute bottom-0 left-1/2 transform -
translate-x-1/2 translate-y-1/2 w-[95%] max-w-[1200px] bg-[#423127]
text-white rounded-[30px] p-8 lg:p-12 shadow-2xl z-20">
    <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-
cols-4 gap-8 lg:gap-4 divide-y md:divide-y-0 md:divide-x divide-
[#5A4436]">
        {features.map((item, index) => (
            <div
                key={item.id}

```

```

        className={`flex items-start gap-4 ${index !== 0 ?
"md:pl-6 lg:pl-8 pt-6 md:pt-0" : ""}`}
      >
        <div className="shrink-0">{item.icon}</div>
        <div>
          <h4 className="font-bold text-[18px] mb-
1">{item.title}</h4>
          <p className="text-[#D4C8C1] text-[14px] leading-
snug">
            {item.desc}
          </p>
        </div>
      </div>
    )}}
  </div>
</div>
</div>

<div className="pt-[250px] lg:pt-[200px] pb-[100px] px-5 bg-
white">
  <div className="max-w-[1200px] mx-auto flex flex-col
lg:flex-row items-center gap-16">
    <div className="w-full lg:w-5/12">
      <h2 className="text-[35px] lg:text-[50px] font-serif
text-[#3A2B24] leading-tight mb-6">
        Наша история
      </h2>
      <p className="text-[#5A4A42] text-[18px] leading-relaxed
mb-6">
        Мы создаем интерьеры, которые становятся доминантой
вашего
        пространства и служат десятилетиями.
      </p>
      <p className="text-[#5A4A42] text-[18px] leading-relaxed
mb-8">
        Каждый проект для нас — это вызов. От идеи до
финальной уборки, мы
        берем на себя все заботы, оставляя вам лишь радость
созидания.
      </p>
      <Link
        to="/about"
        className="text-[#C16828] font-bold flex items-center
gap-2 hover:gap-4 transition-all"
      >
        Подробнее о нас <span>→</span>
      </Link>
    </div>

    <div className="w-full lg:w-7/12 grid grid-cols-2 gap-8
lg:gap-12">

```

```

        <div className="bg-[#FAF8F5] p-8 rounded-[30px] text-
center border border-gray-100">
            <p className="text-[50px] font-bold text-[#3A2B24]
leading-none mb-2">
                2 года
            </p>
            <p className="text-sm text-gray-500 uppercase
tracking-widest">
                Опыта работы
            </p>
        </div>
        <div className="bg-[#FAF8F5] p-8 rounded-[30px] text-
center border border-gray-100">
            <p className="text-[50px] font-bold text-[#3A2B24]
leading-none mb-2">
                65+
            </p>
            <p className="text-sm text-gray-500 uppercase
tracking-widest">
                Проектов
            </p>
        </div>
        <div className="bg-[#FAF8F5] p-8 rounded-[30px] text-
center border border-gray-100">
            <p className="text-[50px] font-bold text-[#3A2B24]
leading-none mb-2">
                98%
            </p>
            <p className="text-sm text-gray-500 uppercase
tracking-widest">
                Довольных клиентов
            </p>
        </div>
        <div className="bg-[#FAF8F5] p-8 rounded-[30px] text-
center border border-gray-100">
            <p className="text-[50px] font-bold text-[#3A2B24]
leading-none mb-2">
                100%
            </p>
            <p className="text-sm text-gray-500 uppercase
tracking-widest">
                Контроль качества
            </p>
        </div>
    </div>
    </div>
    </div>

<div className="py-[120px] px-5 bg-[#FAF8F5]">
    <div className="max-w-[1300px] mx-auto">
        <div className="text-center mb-12">

```

```

        <h3 className="text-[#C16828] font-bold tracking-widest
uppercase text-sm mb-3">
        Индивидуальный подход
    </h3>
    <h2 className="text-[35px] lg:text-[50px] font-serif
text-[#3A2B24] leading-tight">
        Наши тарифы и услуги
    </h2>
</div>

<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-
cols-4 gap-6">
    {services.map((service) => (
        <div
            key={service.id}
            className="bg-white rounded-[30px] overflow-hidden
group border border-gray-100 hover:shadow-xl transition-all
duration-300"
        >
            <div className="h-[250px] overflow-hidden">
                <img
                    src={service.img}
                    alt={service.title}
                    className="w-full h-full object-cover group-
hover:scale-105 transition-transform duration-700"
                />
            </div>
            <div className="p-6">
                <h3 className="text-[20px] font-bold text-[#3A2B24]
mb-1">
                    {service.title}
                </h3>
                <p className="text-[#5A4A42] text-[15px] mb-6 min-
h-[60px]">
                    {service.desc}
                </p>
                <div className="flex justify-between items-center
border-t border-gray-100 pt-4">
                    <span className="text-[15px] font-bold text-
[#3A2B24]">
                        {service.price}
                    </span>
                    <Link
                        to="/services"
                        className="w-10 h-10 rounded-full bg-[#FAF8F5]
flex items-center justify-center group-hover:bg-[#C16828] group-
hover:text-white transition-colors"
                    >
                        <svg
                            xmlns="http://www.w3.org/2000/svg"
                            fill="none"

```

```

        viewBox="0 0 24 24"
        strokeWidth={2}
        stroke="currentColor"
        className="w-5 h-5"
      >
        <path
          strokeLinecap="round"
          strokeLinejoin="round"
          d="M13.5 4.5L21 12m0 0l-7.5 7.5M21 12H3"
        />
      </svg>
    </Link>
  </div>
</div>
</div>
))}
</div>
</div>
</div>

<div className="py-[120px] px-5 bg-white">
  <div className="max-w-[1300px] mx-auto bg-[#423127] rounded-
[40px] overflow-hidden flex flex-col lg:flex-row relative">
    <div className="w-full lg:w-1/2 p-10 lg:p-20 relative z-10
flex flex-col justify-center">
      <h2 className="text-[35px] lg:text-[45px] font-serif
text-white leading-tight mb-6">
        Индивидуальный проект вашей мечты
      </h2>
      <p className="text-[#D4C8C1] text-[18px] leading-relaxed
mb-10">
        Создадим уникальный интерьер по вашим размерам,
пожеланиям и
        стилю. Воспользуйтесь нашим умным калькулятором с
расчетом
        логистики по карте, чтобы узнать предварительный
бюджет. От идеи
        до монтажа – берем на себя всё.
      </p>
      <Link to="/calculator" className="w-fit">
        <button className="px-8 py-4 bg-[#C16828] text-white
font-bold rounded-xl hover:bg-white hover:text-[#3A2B24] transition-
colors flex items-center gap-3 shadow-lg">
          Запустить калькулятор
        <svg
          xmlns="http://www.w3.org/2000/svg"
          fill="none"
          viewBox="0 0 24 24"
          strokeWidth={2}
          stroke="currentColor"
          className="w-5 h-5"

```

```

        >
        <path
            strokeLinecap="round"
            strokeLinejoin="round"
            d="M17.25 8.25L21 12m0 0l-3.75 3.75M21 12H3"
        />
    </svg>
</button>
</Link>
</div>

<div className="w-full lg:w-1/2 min-h-[300px] lg:min-h-
full relative">

    <div className="absolute inset-0 bg-gradient-to-r from-
[#423127] to-transparent z-10 hidden lg:block"></div>
    <img
        src={blueprintImg}
        alt="Architecture blueprint"
        className="absolute inset-0 w-full h-full object-cover
opacity-60"
    />
    </div>
</div>
</div>
</div>
);
};

export default Home;

```

Файл ProjectDetails.jsx

```

import { useEffect, useState } from "react";
import { useParams, useNavigate, useLocation } from "react-router-
dom";
import { MdArrowBack } from "react-icons/md";
import SEO from "../../components/SEO/SEO";

import Slider from "react-slick";
import "slick-carousel/slick/slick.css";
import "slick-carousel/slick/slick-theme.css";

const ProjectDetails = () => {
    const { id } = useParams();
    const navigate = useNavigate();
    const location = useLocation();

    const previousViewMode = location.state?.viewMode || "gallery";

    const [project, setProject] = useState(null);
    const [isLoading, setIsLoading] = useState(true);

```

```

const [error, setError] = useState(null);

useEffect(() => {
  setIsLoading(true);
  fetch(`https://localhost:7002/api/projects/${id}`)
    .then((res) => {
      if (!res.ok) throw new Error("Проект не найден или ошибка сервера");
      return res.json();
    })
    .then((data) => {
      setProject(data);
      setIsLoading(false);
    })
    .catch((err) => {
      console.error("Ошибка загрузки деталей проекта:", err);
      setError(err.message);
      setIsLoading(false);
    });
}, [id]);

const sliderSettings = {
  dots: true,
  infinite: true,
  speed: 500,
  slidesToShow: 1,
  slidesToScroll: 1,
  autoplay: true,
  autoplaySpeed: 4000,
  arrows: false,
};

if (isLoading) {
  return (
    <div className="flex justify-center items-center h-screen bg-[#FAF8F5]">
      <p className="text-2xl text-[#C16828] font-semibold animate-pulse">Загрузка проекта...</p>
    </div>
  );
}

if (error || !project) {
  return (
    <div className="flex flex-col justify-center items-center h-[60vh] gap-4 bg-[#FAF8F5]">
      <p className="text-2xl text-red-500 font-semibold">{error || "Проект не найден"}</p>
      <button
        onClick={() => navigate('/projects', { state: { viewMode: previousViewMode } })}
      />
    </div>
  );
}

```

```

        className="flex items-center gap-2 text-[#C16828]
hover:text-black transition-colors duration-300 font-semibold text-
lg"
    >
        <MdArrowBack className="text-2xl" /> Вернуться назад
    </button>
</div>
);
}

const displayCategory =
    project.category === "Turnkey" ? "Квартира под ключ" :
    project.category === "Living" ? "Гостиная" :
    project.category === "Bedroom" ? "Спальня" :
    project.category === "Kitchen" ? "Кухня" :
    project.category === "Bathroom" ? "Ванная" : project.category;

const displayDate = project.createdAt ? new
Date(project.createdAt).toLocaleDateString("ru-RU") : "Не указана";

return (
    <div className="bg-[#FAF8F5] min-h-screen pb-[50px]">
        <SEO
            title={` ${project?.title || 'Детали проекта'} - Design
Studio`}
            description={project?.description?.substring(0, 150) ||
"Изучите детали нашего дизайн-проекта"}
            keywords={` ${project?.category || 'интерьер'}, дизайн
проект`}
            canonicalUrl={`https://story-home-
six.vercel.app/projects/${id}`}
        />

        <div className="pt-[80px] lg:pt-[50px] mx-auto max-w-[350px]
lg:max-w-[1200px] mb-[100px]">

            <button
                onClick={() => navigate('/projects', { state: { viewMode:
previousViewMode } })}
                className="flex items-center gap-2 text-[#C16828]
hover:text-[#3A2B24] transition-colors duration-300 font-semibold
text-lg mb-8"
            >
                <MdArrowBack className="text-2xl" />
                {previousViewMode === "map" ? "Вернуться к карте" : "Назад
в галерею"}
            </button>

            <div className="grid grid-cols-1 gap-10">
                <div>

```

```

        <h1 className="text-[35px] text-center lg:text-start
lg:text-[40px] font-bold text-[#3A2B24]">
            {project?.title}
        </h1>
        <p className="mt-4 text-justify text-[#5A4A42] text-
[20px] leading-relaxed">
            {project?.description || "Подробное описание проекта
отсутствует."}
        </p>
    </div>

    <div className="w-[350px] lg:w-[1200px] h-[341px] lg:h-
[561px] relative mb-12">
        {project?.images && project.images.length > 1 ? (
            <Slider {...sliderSettings} className="h-full w-full">
                {project.images.map((imageUrl, index) => (
                    <div key={index} className="h-[341px] lg:h-[561px]
outline-none cursor-grab active:cursor-grabbing">
                        <img
                            loading="lazy"
                            src={imageUrl}
                            alt={`${project?.title} - Фото ${index + 1}`}
                            className="h-full w-full object-cover rounded-
[50px] lg:rounded-[100px] shadow-lg"
                        />
                    </div>
                ))}
            </Slider>
        ) : (
            <img
                loading="lazy"
                src={project?.images?.[0] || "/placeholder.jpg"}
                alt={project?.title}
                className="h-full w-full object-cover rounded-[50px]
lg:rounded-[100px] shadow-lg"
            />
        )}
    </div>

    <div className="flex flex-col gap-6">

        <div className="flex flex-col md:flex-row md:items-
center justify-start">
            <h3 className="text-[22px] font-bold text-[#3A2B24]
min-w-[200px]">Категория:</h3>
            <p className="text-[20px] text-[#5A4A42] mt-1 md:mt-
0">{displayCategory}</p>
        </div>

        <div className="flex flex-col md:flex-row md:items-
center justify-start">

```

```

        <h3 className="text-[22px] font-bold text-[#3A2B24]
min-w-[200px]">Дата сдачи:</h3>
        <p className="text-[20px] text-[#5A4A42] mt-1 md:mt-
0">{displayDate}</p>
    </div>

    {project?.price ? (
        <div className="flex flex-col md:flex-row md:items-
center justify-start">
            <h3 className="text-[22px] font-bold text-[#3A2B24]
min-w-[200px]">Бюджет:</h3>
            <p className="text-[22px] font-bold text-[#C16828]
mt-1 md:mt-0">
                {project.price.toLocaleString('ru-RU')} ₸
            </p>
        </div>
    ) : null}

    {project?.address && (
        <div className="flex flex-col md:flex-row md:items-
center justify-start">
            <h3 className="text-[22px] font-bold text-[#3A2B24]
min-w-[200px]">Адрес:</h3>
            <p className="text-[20px] text-[#5A4A42] mt-1 md:mt-
0">{project.address}</p>
        </div>
    )}

    {!project?.address && project?.lat && (
        <div className="flex flex-col md:flex-row md:items-
center justify-start">
            <h3 className="text-[22px] font-bold text-[#3A2B24]
min-w-[200px]">Координаты:</h3>
            <p className="text-[20px] text-[#5A4A42] mt-1 md:mt-
0">{project.lat.toFixed(4)}, {project.lng.toFixed(4)}</p>
        </div>
    )}

</div>

</div>
</div>
</div>
);
};

export default ProjectDetails;

```

Файл Project.jsx

```

import { useEffect, useState, useLayoutEffect } from "react";
import { useLocation } from "react-router-dom";

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		163

```

import ProjectCard from "../../components/common/ProjectCard";
import SEO from "../../components/SEO/SEO";
import PublicMap from "../PublicMap";

const PROJECT_TABS = [
  { id: "Bce", label: "Все" },
  { id: "Turnkey", label: "Квартиры" },
  { id: "Bedroom", label: "Спальни" },
  { id: "Kitchen", label: "Кухни" },
  { id: "Living", label: "Гостиные" },
  { id: "Bathroom", label: "Ванные" },
];

const Projects = () => {
  const location = useLocation();
  const [viewMode, setViewMode] = useState(location.state?.viewMode || "map");
  const [projects, setProjects] = useState([]);
  const [galleryCategory, setGalleryCategory] = useState("Все");

  useEffect(() => {
    fetch("https://localhost:7002/api/projects")
      .then((res) => res.json())
      .then((data) => setProjects(data))
      .catch((err) => console.error("Ошибка загрузки проектов:", err));
  }, []);

  useEffect(() => {
    if (viewMode === "gallery" && projects.length > 0) {
      const savedScroll = sessionStorage.getItem("galleryScroll");
      if (savedScroll) {
        window.scrollTo({ top: parseInt(savedScroll), behavior: "instant" });
        sessionStorage.removeItem("galleryScroll");
      }
    }
  }, [viewMode, projects, galleryCategory]);

  const galleryProjects = projects.filter(
    (item) =>
      !item.displayLocation ||
      item.displayLocation === "both" ||
      item.displayLocation === "gallery",
  );

  const filteredGallery =
    galleryCategory === "Все"
      ? galleryProjects
      : galleryProjects.filter((item) => item.category === galleryCategory);

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		164

```

return (
  <div className="bg-[#FAF8F5] min-h-screen pb-[120px]">
    <SEO
      title="Наши проекты - Design Studio"
      description="Изучите наши дизайн-проекты в галерее или
посмотрите их расположение на интерактивной карте."
      canonicalUrl="https://story-home-six.vercel.app/projects"
    />

    <div className="pt-[50px] lg:pt-[100px] pb-[40px] px-5 text-
center">
      <div className="max-w-[1000px] mx-auto">

        <h1 className="text-[40px] lg:text-[60px] font-serif text-
[#3A2B24] leading-[1.1] mb-6">
          Наши работы — это <br className="hidden sm:block"/>
          <span className="italic font-light">истории о
комфорт</span>
        </h1>

        <div className="inline-flex gap-2 p-1.5 bg-white rounded-
[20px] shadow-sm border border-gray-100 mb-10">
          <button
            onClick={() => setViewMode("map")}
            className={`px-8 py-3 rounded-[15px] text-[16px] font-
bold transition-all duration-300 ${
              viewMode === "map"
                ? "bg-[#423127] text-white shadow-md"
                : "bg-transparent text-[#5A4A42] hover:text-
[#3A2B24]"
            }`>
            >
            Объекты на карте
          </button>
          <button
            onClick={() => {
              setViewMode("gallery");
              setGalleryCategory("Bce");
            }}
            className={`px-8 py-3 rounded-[15px] text-[16px] font-
bold transition-all duration-300 ${
              viewMode === "gallery"
                ? "bg-[#423127] text-white shadow-md"
                : "bg-transparent text-[#5A4A42] hover:text-
[#3A2B24]"
            }`>
            >
            Галерея проектов
          </button>
        </div>

```

```

    </div>
  </div>

  <div className="max-w-[1300px] mx-auto px-5">

    {viewMode === "gallery" && (
      <div className="animate-fade-in w-full flex flex-col
items-center">

        <div className="flex flex-wrap justify-center gap-3 mb-8
w-full">
          {PROJECT_TABS.map((tab) => (
            <button
              key={tab.id}
              onClick={() => setGalleryCategory(tab.id)}
              className={`px-6 py-2.5 rounded-full text-[13px]
font-bold uppercase tracking-widest transition-all border-2 ${
                galleryCategory === tab.id
                  ? "bg-[#C16828] border-[#C16828] text-white
shadow-lg"
                  : "bg-white border-white text-[#5A4A42]
hover:border-[#E6D8C6]"
              }`}
            >
              {tab.label}
            </button>
          ))}
        </div>

        {filteredGallery.length > 0 ? (
          <div className="w-full grid grid-cols-1 md:grid-cols-2
lg:grid-cols-3 gap-10">
            {filteredGallery.map((item, index) => (
              <ProjectCard e={item} key={item.id || index} />
            ))}
          </div>
        ) : (
          <p className="w-full text-center py-20 text-[#5A4A42]
italic opacity-60">
            Проекты в этой категории еще не добавлены.
          </p>
        )}
      </div>
    )}

    {viewMode === "map" && (
      <div className="animate-fade-in">
        <PublicMap projects={projects} />
      </div>
    )}
  </div>

```

```

    </div>
  );
};

export default Projects;

```

Файл PublicMap.jsx

```

import React, { useState } from "react";
import { Marker, Popup } from "react-leaflet";
import { Link } from "react-router-dom";
import { IoIosArrowForward } from "react-icons/io";
import BaseMap, { customIcon } from
"../../components/common/BaseMap";

const MAP_TABS = [
  { id: "Все", label: "Все" },
  { id: "Turnkey", label: "Квартиры" },
  { id: "Bedroom", label: "Спальни" },
  { id: "Kitchen", label: "Кухни" },
  { id: "Living", label: "Гостиные" },
  { id: "Bathroom", label: "Ванные" },
];

const PublicMap = ({ projects }) => {
  const [activeCategory, setActiveCategory] = useState("Все");

  const mapProjects = projects.filter(
    (item) =>
      (!item.displayLocation ||
        item.displayLocation === "both" ||
        item.displayLocation === "map") &&
        item.lat &&
        item.lng,
  );

  const filteredProjects =
    activeCategory === "Все"
      ? mapProjects
      : mapProjects.filter((p) => p.category === activeCategory);

  return (
    <div className="w-full flex flex-col items-center">
      <div className="flex flex-wrap justify-center gap-3 mb-8 w-
full">
        {MAP_TABS.map((tab) => (
          <button
            key={tab.id}
            onClick={() => setActiveCategory(tab.id)}
            className={`px-6 py-2.5 rounded-full text-[13px] font-
bold uppercase tracking-widest transition-all border-2 ${

```

```

        activeCategory === tab.id
        ? "bg-[#C16828] border-[#C16828] text-white shadow-
lg"
        : "bg-white border-white text-[#5A4A42]
hover:border-[#E6D8C6]"
    }` }
    >
    {tab.label}
  </button>
)}}
</div>

<div className="w-full h-[500px] lg:h-[700px] rounded-[40px]
overflow-hidden border-[4px] border-white shadow-2xl relative z-0">
  <BaseMap
    center={[53.195, 45.0183]}
    zoom={12}
    scrollWheelZoom={true}
    className="w-full h-full z-0"
  >
    {filteredProjects.map((project) => {
      const coverImage =
        project.images && project.images.length > 0
        ? project.images[0]
        : null;

      return (
        <Marker
          key={project.id}
          position={[project.lat, project.lng]}
          icon={customIcon}
        >
          <Popup className="custom-popup">

            <div className="w-[260px] p-1.5 font-sans bg-white
rounded-[24px] shadow-sm">
              {coverImage ? (
                <img
                  src={coverImage}
                  alt={project.title}

                  className="w-full h-[150px] object-cover
rounded-[18px]"
                />
              ) : (
                <div className="w-full h-[150px] bg-[#FAF8F5]
flex items-center justify-center rounded-[18px] text-gray-400 text-
xs italic">
                  Нет фото
                </div>
              )}
            </div>
          </Popup>
        </Marker>
      );
    })}
  </BaseMap>
</div>

```

```

        <div className="px-3 pb-2 pt-4 bg-white">
            <h3 className="font-serif font-bold text-[18px]
text-[#3A2B24] leading-tight mb-2 break-words">
                {project.title || "Проект дизайна"}
            </h3>

            <div className="flex items-center justify-end
mt-4">

                <Link
                    to={`\`/projects/${project.id}`}
                    state={{ viewMode: "map" }}
                    className="group flex items-center
justify-start bg-[#C16828] !text-white rounded-full h-[44px] w-[44px]
hover:w-[145px] transition-all duration-300 overflow-hidden shadow-
md no-underline"
                >
                    <div className="shrink-0 w-[44px] h-[44px]
flex items-center justify-center">
                        <IoIosArrowForward className="text-
2xl !text-white" />
                    </div>
                    <span className="whitespace-nowrap font-
medium text-[15px] !text-white opacity-0 group-hover:opacity-100
transition-opacity duration-300 pr-5">
                        Подробнее
                    </span>
                </Link>
            </div>
        </div>
    </div>
</Popup>
</Marker>
    );
    })}
</BaseMap>

    {filteredProjects.length === 0 && (
        <div className="absolute inset-0 z-[1000] flex items-
center justify-center bg-[#3A2B24]/10 backdrop-blur-[2px]">
            <div className="bg-white px-8 py-6 rounded-[25px]
shadow-2xl text-center max-w-[300px]">
                <p className="font-bold text-[#3A2B24] text-lg mb-2">
                    Нет объектов на карте
                </p>
                <p className="text-sm text-[#5A4A42]">
                    В этой категории пока нет проектов с указанным
адресом.
                </p>
            </div>
        )
    )
}

```

```

        </div>
      )}
    </div>
  </div>
);
};

export default PublicMap;

```

Файл ServicesPage.jsx

```

import { Link } from "react-router-dom";
import SEO from "../../components/SEO/SEO";
import srv1 from "../../assets/3.jpg";
import srv2 from "../../assets/4.jpg";
import srv3 from "../../assets/5.jpg";
import srv4 from "../../assets/6.jpg";

const ServicesPage = () => {
  const detailedServices = [
    {
      id: 1,
      title: "Планировочный",
      price: "от 1 500 ₽ / м²",
      desc: "Идеально для тех, кто делает ремонт сам, но хочет грамотно использовать пространство.",
      includes: [
        "Обмерный план",
        "3 варианта планировочного решения",
        "План демонтажа и монтажа стен",
        "План расстановки мебели"
      ],
      img: srv1
    },
    {
      id: 2,
      title: "Базовый проект",
      price: "от 3 500 ₽ / м²",
      desc: "Для строителей: вся необходимая рабочая документация без 3D-визуализаций.",
      includes: [
        "Всё из Планировочного тарифа",
        "Планы электрики, розеток и освещения",
        "Планы полов и потолков",
        "Развертки стен санузлов"
      ],
      img: srv3
    },
    {
      id: 3,
      title: "Комплексный",

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		170

```

    price: "от 5 000 Р / м²",
    desc: "Самый популярный тариф. Полный визуальный и технический
контроль над будущим интерьером.",
    includes: [
      "Всё из Базового проекта",
      "Фотореалистичная 3D-визуализация",
      "Подбор мебели, сантехники и света",
      "Спецификация чистовых материалов"
    ],
    img: srv2
  },
  {
    id: 4,
    title: "VIP под ключ",
    price: "от 8 000 Р / м²",
    desc: "Вы просто заезжаете в готовую квартиру. Мы берем на
себя абсолютно все заботы по стройке.",
    includes: [
      "Всё из Комплексного тарифа",
      "Авторский надзор (регулярные выезды)",
      "Закупка материалов и приемка на объекте",
      "Организация работы подрядчиков"
    ],
    img: srv4
  }
];

```

```

return (
  <div className="bg-[#FAF8F5]">
    <SEO
      title="Наши Услуги - Design Studio"
      description="Профессиональные услуги дизайна интерьера: от
консультации и 3D визуализации до комплексного ремонта под ключ."
      keywords="услуги дизайна интерьера, ремонт под ключ, 3d
визуализация"
      canonicalUrl="https://story-home-six.vercel.app/services"
    />

    <div className="pt-[50px] lg:pt-[100px] pb-[80px] px-5 bg-
[#FAF8F5]">
      <div className="max-w-[1000px] mx-auto text-center">

        <h1 className="text-[40px] lg:text-[60px] font-serif text-
[#3A2B24] leading-[1.1] mb-6">
          От голой идеи до <br className="hidden sm:block"/>
          <span className="italic font-light">финального
декора</span>
        </h1>

        <p className="text-[18px] text-[#5A4A42] leading-relaxed
max-w-[700px] mx-auto">

```

Мы предлагаем гибкие форматы сотрудничества. Вы можете заказать как точечную консультацию и планировку, так и полное делегирование ремонта нашей команде.

```
</p>
</div>
</div>

<div className="py-[60px] px-5">
  <div className="max-w-[1200px] mx-auto flex flex-col gap-12 lg:gap-20">
    {detailedServices.map((service, index) => (
      <div
        key={service.id}
        className={`flex flex-col lg:flex-row gap-10 lg:gap-16 items-center bg-white p-8 lg:p-12 rounded-[40px] shadow-sm border border-gray-100 ${
          index % 2 !== 0 ? 'lg:flex-row-reverse' : ''
        }`}
      >

        <div className="w-full lg:w-1/2 h-[300px] lg:h-[450px] overflow-hidden rounded-[30px]">
          <img
            src={service.img}
            alt={service.title}
            className="w-full h-full object-cover hover:scale-105 transition-transform duration-700"
          />
        </div>
        <div className="w-full lg:w-1/2 flex flex-col justify-center">
          <div className="flex flex-wrap items-end gap-4 mb-4">
            <h2 className="text-[30px] lg:text-[40px] font-serif text-[#3A2B24] leading-none">
              {service.title}
            </h2>
            <span className="text-[#C16828] text-[20px] font-bold pb-1 whitespace-nowrap">
              {service.price}
            </span>
          </div>

          <p className="text-[#5A4A42] text-[18px] leading-relaxed mb-8">
            {service.desc}
          </p>

          <div className="mb-10">
            <h4 className="text-[#3A2B24] font-bold mb-4 uppercase tracking-wider text-sm">Что входит в тариф:</h4>
```

```

        <ul className="flex flex-col gap-3">
          {service.includes.map((item, i) => (
            <li key={i} className="flex items-start gap-3
text-[#5A4A42]">
              <svg xmlns="http://www.w3.org/2000/svg"
fill="none" viewBox="0 0 24 24" strokeWidth={2}
stroke="currentColor" className="w-5 h-5 text-[#C16828] shrink-0 mt-
0.5">
                <path strokeLinecap="round"
strokeLinejoin="round" d="M4.5 12.75l6 9-13.5" />
              </svg>
              {item}
            </li>
          ))}
        </ul>
      </div>

      <Link to="/contact">
        <button className="w-full sm:w-auto px-10 py-4 bg-
transparent border-2 border-[#423127] text-[#423127] font-bold
rounded-xl hover:bg-[#423127] hover:text-white transition-all">
          Обсудить проект
        </button>
      </Link>
    </div>
  </div>
  </div>
  </div>

  <div className="py-[100px] px-5 bg-white border-t border-gray-
100 mt-10">
    <div className="max-w-[1200px] mx-auto text-center">
      <h2 className="text-[30px] lg:text-[45px] font-serif text-
[#3A2B24] mb-12">
        Включено в каждый проект
      </h2>
      <div className="grid grid-cols-1 md:grid-cols-3 gap-8">
        <div className="p-8 bg-[#FAF8F5] rounded-[30px]">
          <div className="w-14 h-14 bg-white rounded-full flex
items-center justify-center mx-auto mb-6 shadow-sm">
            <svg xmlns="http://www.w3.org/2000/svg" fill="none"
viewBox="0 0 24 24" strokeWidth={1.5} stroke="currentColor"
className="w-7 h-7 text-[#C16828]">
              <path strokeLinecap="round" strokeLinejoin="round"
d="M12 6v12m-3-2.818l1.879.659c1.171.879 3.07.879 4.242 0 1.172-.879
1.172-2.303 0-3.182C13.536 12.219 12.768 12 12 12c-.725 0-1.45-.22-
2.003-.659-1.106-.879-1.106-2.303 0-3.182s2.9-.879 4.006
0 1.415.33M21 12a9 9 0 11-18 0 9 9 0 0 118 0z" />
            </svg>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        <h4 className="text-[20px] font-bold text-[#3A2B24]
mb-3">Детальная смета</h4>
        <p className="text-[#5A4A42] text-[15px]">Фиксируем
стоимость работ и материалов. Вы защищены от внезапных переоплат.</p>
        </div>

        <div className="p-8 bg-[#FAF8F5] rounded-[30px]">
            <div className="w-14 h-14 bg-white rounded-full flex
items-center justify-center mx-auto mb-6 shadow-sm">
                <svg xmlns="http://www.w3.org/2000/svg" fill="none"
viewBox="0 0 24 24" strokeWidth={1.5} stroke="currentColor"
className="w-7 h-7 text-[#C16828]">
                    <path strokeLinecap="round" strokeLinejoin="round"
d="M6.75 3v2.25M17.25 3v2.25M3 18.75V7.5a2.25 2.25 0 012.25-
2.25h13.5A2.25 2.25 0 0121 7.5v11.25m-18 0A2.25 2.25 0 005.25
21h13.5A2.25 2.25 0 0021 18.75m-18 0v-7.5A2.25 2.25 0 015.25
9h13.5A2.25 2.25 0 0121 11.25v7.5" />
                </svg>
            </div>
            <h4 className="text-[20px] font-bold text-[#3A2B24]
mb-3">Соблюдение сроков</h4>
            <p className="text-[#5A4A42] text-[15px]">Четкий
график этапов работ, прописанный в официальном договоре.</p>
            </div>

            <div className="p-8 bg-[#FAF8F5] rounded-[30px]">
                <div className="w-14 h-14 bg-white rounded-full flex
items-center justify-center mx-auto mb-6 shadow-sm">
                    <svg xmlns="http://www.w3.org/2000/svg" fill="none"
viewBox="0 0 24 24" strokeWidth={1.5} stroke="currentColor"
className="w-7 h-7 text-[#C16828]">
                        <path strokeLinecap="round" strokeLinejoin="round"
d="M2.25 21h19.5m-18-18v18m10.5-18v18m6-13.5V21M6.75 6.75h.75m-.75
3h.75m-.75 3h.75m3-6h.75m-.75 3h.75m6.75 21v-3.375c0-
.621.504-1.125 1.125-1.125h2.25c.621 0 1.125.504 1.125 1.125V21M3
3h12m-.75 4.5H21m-3.75 3.75h.008v.008h-.008v-.008zm0 3h.008v.008h-
.008v-.008zm0 3h.008v.008h-.008v-.008z" />
                    </svg>
                </div>
                <h4 className="text-[20px] font-bold text-[#3A2B24]
mb-3">Проверенные подрядчики</h4>
                <p className="text-[#5A4A42] text-[15px]">Работаем
только с лучшими строительными бригадами и поставщиками мебели.</p>
                </div>
            </div>
        </div>
        <div className="py-[100px] px-5 bg-[#FAF8F5]">
            <div className="max-w-[1300px] mx-auto bg-[#423127] rounded-
[40px] p-10 lg:p-20 text-center flex flex-col items-center">

```

```

        <h2 className="text-[35px] lg:text-[45px] font-serif text-
white leading-tight mb-6 max-w-[800px]">
        Сложно выбрать подходящий тариф?
    </h2>
    <p className="text-[#D4C8C1] text-[18px] leading-relaxed
mb-10 max-w-[700px]">
        Перейдите в наш умный калькулятор. Ответьте на несколько
вопросов о вашем объекте, укажите адрес для расчета логистики, и
система сама подскажет примерный бюджет.
    </p>
    <Link to="/calculator">
        <button className="px-10 py-5 bg-[#C16828] text-white
font-bold rounded-xl hover:bg-white hover:text-[#3A2B24] transition-
all shadow-xl flex items-center gap-3">
            Рассчитать бюджет проекта
            <svg xmlns="http://www.w3.org/2000/svg" fill="none"
viewBox="0 0 24 24" strokeWidth={2} stroke="currentColor"
className="w-5 h-5">
                <path strokeLinecap="round" strokeLinejoin="round"
d="M17.25 8.25L21 12m0 0l-3.75 3.75M21 12H3" />
            </svg>
        </button>
    </Link>
</div>
</div>
);
};

export default ServicesPage;

```

Файл Routes.jsx

```

import { createBrowserRouter } from "react-router-dom";
import MainLayout from "../components/Layouts/MainLayout";
import ErrorPage from "../pages/ErrorPage/ErrorPage";
import About from "../pages/AboutPage/About";
import Home from "../pages/HomePage/Home";
import Contact from "../pages/ContactPage/Contact";
import AdminPage from "../pages/AdminPage/AdminPage";
import Projects from "../pages/ProjectsPage/Projects";
import ServicesPage from "../pages/ServicesPage/ServicesPage";
import ProjectDetails from
"../pages/ProjectDetailsPage/ProjectDetails";
import Calculator from "../pages/CalculatorPage/Calculator";
import AdminLogin from "../pages/AdminPage/AdminLogin";
import ProtectedRoute from "../components/ProtectedRoute";

const router = createBrowserRouter([
    {

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		175

```

    path: "/",
    element: <MainLayout />,
    errorElement: <ErrorPage />,
    children: [
      { path: "/", element: <Home /> },
      { path: "/projects", element: <Projects /> },
      { path: "/projects/:id", element: <ProjectDetails /> },
      { path: "/services", element: <ServicesPage /> },
      { path: "/contact", element: <Contact /> },
      { path: "/about", element: <About /> },
      { path: "/calculator", element: <Calculator /> },
    ],
  },
},

{
  path: "/admin-login",
  element: <AdminLogin />,
},

{
  path: "/admin",
  element: (
    <ProtectedRoute>
      <AdminPage />
    </ProtectedRoute>
  ),
},
]);

export default router;

```

Файл App.jsx

```

import { RouterProvider } from "react-router-dom";
import "./App.css";
import router from "./Routes/Routes";

const App = () => {
  return <RouterProvider router={router} />;
};

export default App;

```

Файл index.css

```

@import
url("https://fonts.googleapis.com/css2?family=DM+Serif+Display:ital@0;1&family=Jost:wght@200;300;400;500;600;700;800;900&display=swap");
@tailwind base;
@tailwind components;

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		176

```

@tailwind utilities;

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

h1,
.heading-text {
  font-family: "DM Serif Display", serif;
}

body {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Jost", sans-serif;
  color: #292f36;
}

.i::before {
  content: "";
  position: absolute;
  width: 0px;
  height: 0px;
  opacity: 20%;
  background: white;
  top: 0;
  left: 0;
  bottom: 0;
  right: 0;
  margin: auto;
}

.i:hover:before {
  animation: anim-in 0.7s forwards ease-out;
}

@keyframes anim-in {
  100% {
    opacity: 0%;
    border-radius: 0;
    width: 600px;
    height: 600px;
  }
  0% {
    width: 0px;
    height: 0px;
    border-radius: 100%;
    opacity: 20%;
  }
}

```

					ПГУ 1.090301.49.001 ПЗ	Лист
						177
Изм	Лист	№ докум.	Подпись	Дата		

```

    }
}
button {
    transition: background 0.7s ease;
}

.arrow1 {
    animation: slide1 1s ease-in-out infinite;
    margin-left: 9px;
}

@keyframes slide1 {
    0%,
    100% {
        transform: translate(0, 0);
    }

    50% {
        transform: translate(10px, 0);
    }
}

.slick-slide > div {
    margin: 0 10px;
}

.slick-dots li button:before {
    font-size: 16px !important;
    color: #c16828 !important;
    opacity: 0.3 !important;
    transition: all 0.3s ease;
}

.slick-dots li.slick-active button:before {
    opacity: 1 !important;
    transform: scale(1.3);
}

.slick-dots {
    bottom: -40px !important;
}

.box {
    background: white;
    border-radius: 20px;
    display: grid;
    grid-template-columns: 64px 1fr;
    position: relative;
}

.studio-button2 {
    position: absolute;

```

					ПГУ 1.090301.49.001 ПЗ	Лист
						178
Изм	Лист	№ докум.	Подпись	Дата		

```

bottom: -25px;
right: 5px;
display: flex;
cursor: pointer;
align-items: center;
background: #c16828;
color: white;
padding: 8px 13px;
border-radius: 50px;
box-shadow: 0 4px 14px rgba(0, 0, 0, 0.25);
transition: 0.35s ease all;
overflow: hidden;
max-width: 50px;
height: 50px;
}

.studio-button {
position: absolute;
bottom: 16px;
right: 5px;
display: flex;
cursor: pointer;
align-items: center;
background: #c16828;
color: white;
padding: 8px 13px;
border-radius: 50px;
box-shadow: 0 4px 14px rgba(0, 0, 0, 0.25);
transition: 0.35s ease all;
overflow: hidden;
max-width: 50px;
height: 50px;
}

.studio-button-icon {
position: relative;
top: 1px;
}

.studio-button-label {
white-space: nowrap;
padding: 0 8px;
opacity: 0;
transform: translateX(10px);
transition: 0.25s ease all;
}

.box:hover .studio-button {
max-width: 100%;
}

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		179

```

.box:hover .studio-button-label {
  opacity: 1;
  transform: translateX(0);
  transition:
    0.25s 0.1s ease-in opacity,
    0.15s 0.1s cubic-bezier(0.175, 0.885, 0.32, 1.275) transform;
}

.box:hover .studio-button2 {
  max-width: 500px;
}

.scroll-to-top {
  background-color: #c16828 !important;
  right: 40px;
  bottom: 40px;
  position: fixed;
  z-index: 2;
  cursor: pointer;
  border-radius: 100% !important;
  width: 50px !important;
  display: flex;
  justify-items: center;
  height: 50px !important;
  box-shadow: 0 9px 25px 0 rgba(132, 128, 177, 0.28);
  border: none;
  justify-content: center;
  align-items: center;
}

.active {
  color: #c16828;
  border-bottom: 1px solid #c16828;
}

input:focus,
textarea:focus {
  outline: none;
}

```

Файл main.jsx

```

import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App.jsx";
import "./index.css";
import "leaflet/dist/leaflet.css";
import ScrollToTop from "react-scroll-to-top";
ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <App />
    <ScrollToTop smooth color="#f2f0f5" />
  </React.StrictMode>,
);

```

					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		180

Приложение В
(обязательное)

Создание сайта организации с применением геоинформационных технологий

Презентация

					ПГУ 1.090301.49.001 ПЗ	Лист
						181
Изм	Лист	№ докум.	Подпись	Дата		

Создание сайта организации с применением геоинформационных технологий

Автор работы:

студентка гр.22ВВП2 Широкова Ирина Дмитриевна

Научный руководитель:

к.т.н., доцент кафедры ВТ Леонова Татьяна Юрьевна

Пензенский государственный университет

Цель и задачи

Цель работы – разработать веб-сайт для студии дизайна интерьеров, объединяющий карту проектов, калькулятор стоимости и административную панель.

Для достижения цели были решены следующие задачи:

- анализ существующих решений и выявление их недостатков;
- обоснование выбора технологического стека;
- проектирование клиент-серверной архитектуры и базы данных;
- разработка клиентской части на React с картой Leaflet и калькулятором;
- разработка серверной части на ASP.NET Core с JWT-аутентификацией;
- создание панели администратора;
- тестирование и оценка экономической эффективности.

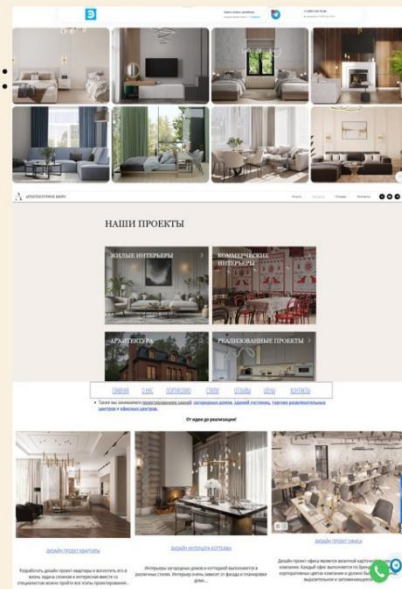
					ПГУ 1.090301.49.001 ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		182

Обзор аналогов

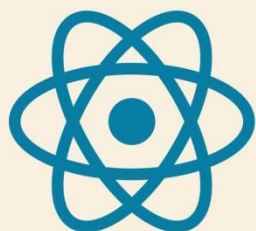
Были проанализированы сайты трёх студий:
«Эдиз», «Арх.Бюро», «Анфилада».

Выявлены общие недостатки:

- статичное представление финансовой информации (текстовые прайс-листы);
- отсутствие географической привязки проектов;
- низкий уровень интерактивности, нет возможности самостоятельного расчёта стоимости.



Выбор стека технологий



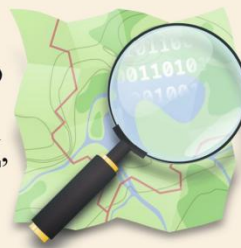
— обеспечивает мгновенную перерисовку интерфейса без перезагрузки страницы.



— высокопроизводительный кроссплатформенный фреймворк с поддержкой JWT и интеграцией с MySQL через Entity Framework Core.



— бесплатная, надёжная и широко распространённая реляционная СУБД, совместимая с ASP.NET Core.



— бесплатная экосистема для карт, геокодирования и маршрутизации без регистрации и API-ключей.

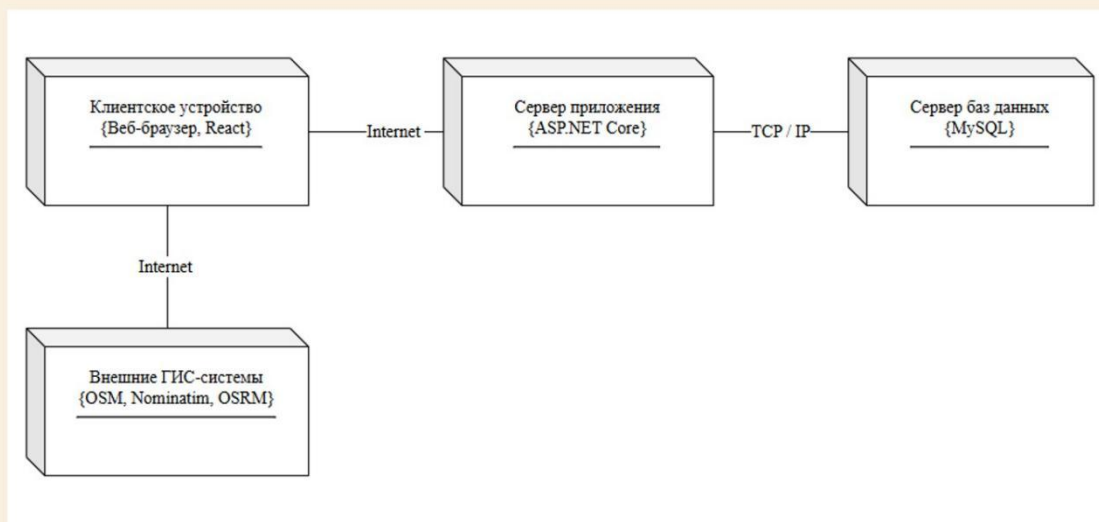
Изм	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.49.001 ПЗ

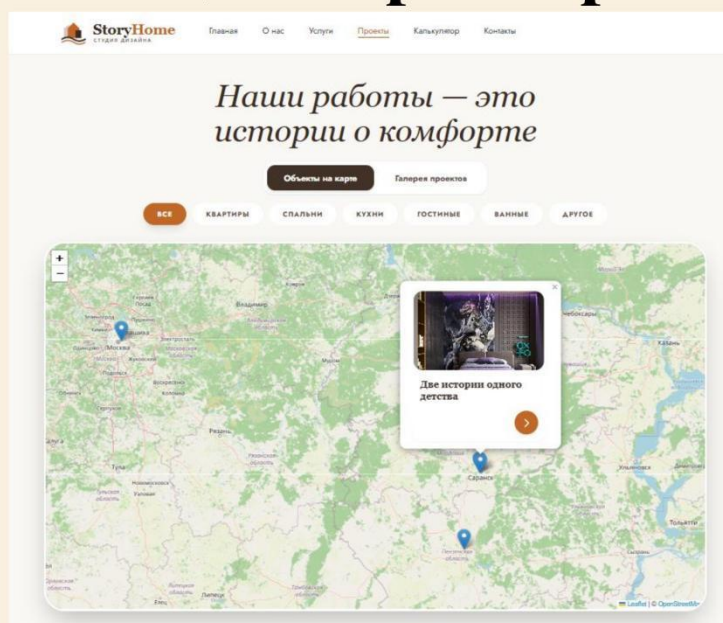
Лист

183

Архитектура системы



Реализация карты проектов



Изм	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.49.001 ПЗ

Лист

184

Реализация модуля калькулятора

Ваша смета

Площадь: 50 м²

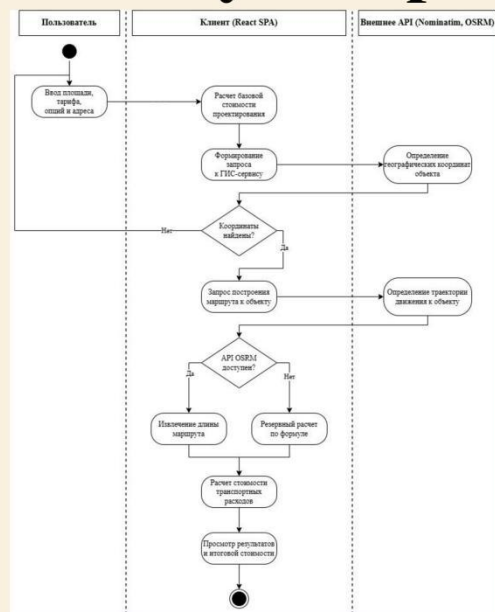
Объект: Новаястройка

Тип: VIP

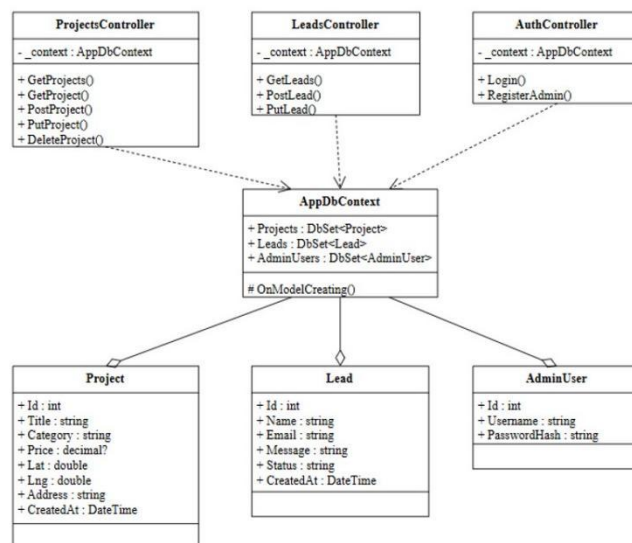
Политика: 3 989 ₽

ИТОГОВАЯ СТОИМОСТЬ
253 989 ₽

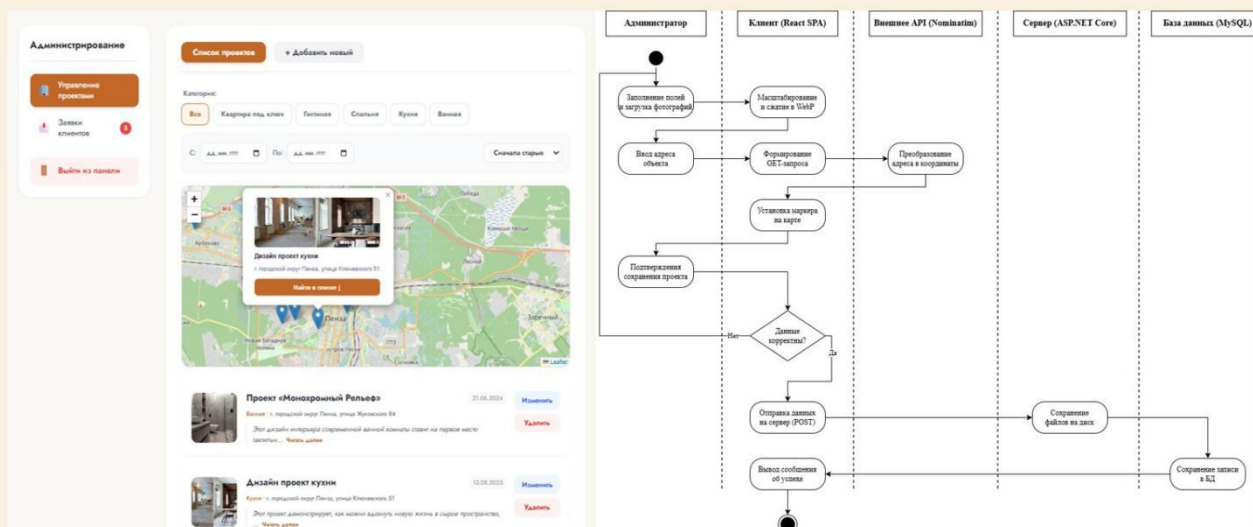
Оставить заявку



Реализация серверной части



Реализация панели администратора



Экономическая эффективность

Капитальные вложения (рыночная стоимость аналога) – 150 000 руб.

Прямая экономия фонда оплаты труда за счёт автоматизации – 43 095 руб./год.

Косвенный эффект (прирост чистой прибыли от дополнительных клиентов) – 150 000 руб./год.

Суммарный годовой эффект – 193 095 руб.

Срок окупаемости – 0,78 года (около 9,3 месяцев).

Спасибо за внимание!

					ПГУ 1.090301.49.001 ПЗ	Лист
						187
Изм	Лист	№ докум.	Подпись	Дата		

Приложение Г
(обязательное)

Создание сайта организации с применением геоинформационных технологий

Справка о внедрении выпускной квалификационной работы

					ПГУ 1.090301.49.001 ПЗ	Лист
						188
Изм	Лист	№ докум.	Подпись	Дата		

Справка о внедрении выпускной квалификационной работы

СПРАВКА

о внедрении результатов выпускной квалификационной работы
Широковой Ирины Дмитриевны
в практическую деятельность ИП Новокрещенова О.В. "Дизайн Земли"

В процессе выполнения выпускной квалификационной работы на тему: «Создание сайта организации с применением геоинформационных технологий» Широковой И.Д. разработан программный комплекс, предназначенный для официального сайта компании. Полученное клиент-серверное веб-приложение внедрено и используется в производственной деятельности ИП Новокрещенова О.В. "Дизайн Земли".

Внедрение данной системы позволило решить следующие задачи:

1. Обеспечить интерактивную визуализацию портфолио на картографической основе;
2. Автоматизировать сбор контактных данных посетителей;
3. Реализовать онлайн-калькуляцию логистических и проектных затрат.

Внедрение разработанной системы позволит получить существенный финансово-экономический эффект. Суммарный ожидаемый экономический эффект за первый год эксплуатации оценивается в 193 095 рублей. Указанный показатель достигнут благодаря следующим факторам:

1. Повышение производительности труда и снижение издержек.

Автоматизация процессов первичного расчета смет и логистических затрат позволила сократить время обработки входящих запросов на 67% (с 30 до 10 минут). Стоимость высвобожденного рабочего времени менеджера с учетом отчислений во внебюджетные фонды обеспечила экономический эффект в размере 43 095 рублей в год.

2. Рост коммерческих показателей.

Внедрение интерактивного картографического модуля портфолио и прозрачной онлайн-калькуляции повысило вовлеченность аудитории и увеличило конверсию посетителей сайта в реальные сделки. За счет привлечения дополнительных клиентов прогнозируемый годовой прирост чистой прибыли предприятия составляет 150 000 рублей.

Результаты, полученные Широковой И.Д., отвечают современным требованиям веб-разработки, демонстрируют экономическую целесообразность и успешно применяются в коммерческой деятельности ИП Новокрещенова О.В. для организации и оптимизации рабочего процесса.

Индивидуальный предприниматель
Новокрещенова О.В.
"Дизайн Земли"



Изм	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.49.001 ПЗ

Лист

189