

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

Кафедра

вычислительной техники

«Вычислительная техника»

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

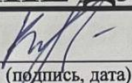
Профиль подготовки: «Программное обеспечение средств вычислительной
техники и автоматизированных систем»

БАКАЛАВРСКАЯ РАБОТА

на тему

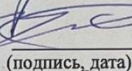
WEB-ПРИЛОЖЕНИЕ ДЛЯ САЛОНА КРАСОТЫ

Студент


(подпись, дата)

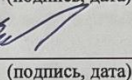
30.05.2025 Кирсанова Дарья Михайловна
(ФИО полностью)

Руководитель


(подпись, дата)

30.05.2025 Зинкин С.А.
(фамилия, инициалы)

Нормоконтролёр


(подпись, дата)

30.05.2025 Бычков А.С.
(фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 28.05.2025 г. № 16)

Заведующий кафедрой


(подпись)

М.А. Митрохин
(фамилия, инициалы)

Работа защищена с отметкой хорошо (протокол заседания ГЭК от

6 июня 2025 г. № 14)

Секретарь ГЭК


(подпись)

Е.И. Гурин
(фамилия, инициалы)

Пенза 2025

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

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
« 06 » 05 2025 г.

**ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ**

1. Студент Кирсанова Дарья Михайловна гр. 21ВВП2 факультета ФВТ
2. Тема работы Web-приложение для салона красоты

Тема утверждена приказом ПГУ ПГУ № 601/о от 06.05.2025 г.

3. Руководитель бакалаврской работы д.т.н., профессор кафедры ВТ ПГУ Зинкин С.А.,

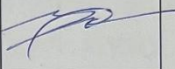
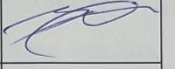
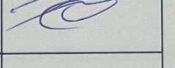
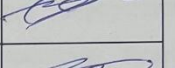
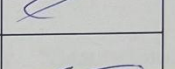
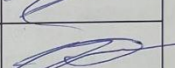
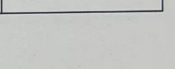
4. Задание на работу (назначение разработки, исходные данные и т.п.) Разработка прототипа веб-приложения "BeautySalon" для автоматизации процесса онлайн-записи клиентов на услуги салона красоты и предоставления информации об услугах и специалистах. Функционал приложения – предоставление информации клиентам: клиенты должны иметь доступ к актуальной информации об услугах, их стоимости, длительности, а также о специалистах салона; запись на услугу: клиент выбирает услугу, мастера, удобную дату и время; управление записями: клиент должен иметь возможность просматривать свои предстоящие и прошедшие записи, а также отменять их; регистрация и аутентификация: для доступа к персонализированным функциям клиенту необходима регистрация и вход в систему.

Серверная часть приложения должна обрабатывать HTTP-запросы, реализовывать бизнес-логику, взаимодействовать с базой данных. СУБД – MS SQL, а все приложение должно быть реализовано на языке JavaScript.

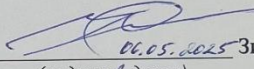
5. Перечень подлежащих разработке вопросов:

Произвести анализ предметной области: выявить достоинства и недостатки существующих веб-приложений в сфере онлайн-записи, сделать выводы, поставить итоговую задачу. Изучить работу с выбранной средой разработки. Подключить БД и программу администрирования СУБД. Выбрать и реализовать наиболее подходящую архитектуру, разработать и протестировать итоговое веб-приложение.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Изучение предметной области	32	06.05.2025 – 08.05.2025	
2	Анализ требований	32	09.05.2025 – 11.05.2025	
3	Создание макета графического интерфейса	30	12.05.2025 – 14.05.2025	
4	Выбор технологий, применяемых при проектировании и разработке	40	15.05.2025 – 18.05.2025	
5	Реализация графического интерфейса	66	19.05.2025 – 22.05.2025	
6	Разработка основной части программы	70	23.05.2025 – 27.05.2025	
7	Оформление пояснительной записки	54	28.05.2025 – 30.05.2025	

Дата выдачи задания "06" мая 2025 г.

Руководитель бакалаврской работы  06.05.2025 Зинкин С.А.
(подпись, дата)

Задание к исполнению принял студент  06.05.2025 Кирсанова Д.М.
(подпись, дата)

Работу к защите допустить
Декан факультета  04.06.2025 Фионова Л.Р.
(подпись, дата)

Реферат

Бакалаврская работа содержит 107 листов, 20 рисунков, 5 таблиц, 12 источников и 3 приложения.

WEB-ПРИЛОЖЕНИЕ, ОБРАБОТКА СОБЫТИЙ, ЛОГИРОВАНИЕ, БАЗА ДАННЫХ, ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ, JAVASCRIPT, MYSQL

Цель работы – разработка прототипа веб-приложения "BeautySalon" для автоматизации процесса онлайн-записи клиентов на услуги салона красоты и предоставления информации об услугах и специалистах.

Для достижения поставленной цели было разработано web-приложение, отвечающее следующим поставленным задачам: предоставление информации клиентам об услугах, их стоимости, длительности, а также о специалистах, запись на услугу, управление записями, регистрация и аутентификация.

В результате проведенной работы разработан проект "BeautySalon" предоставляющий базовую, но функциональную платформу для онлайн-записи для салона красоты. Он демонстрирует использование PHP для серверной логики, MySQL для хранения данных и JavaScript для улучшения пользовательского опыта на клиенте.

					ПГУ 1.090301.32.001 ПЗ		
Изм	Лист	№ докум.	Подпись	Дата			
Разраб.		Кирсанова Д.М.		30.05.25	Литера	Лист	Листов
Пров.		Зинкин С.А.		30.05.25		4	107
Н. Контр.		Бычков А.С.		30.05.25	ФВТ, каф. ВТ гр. 21ВВП2		
Утв.		Митрохин М.А.		30.05.25			

Web-приложение для салона красоты
Пояснительная страница

Содержание

Перечень сокращений и обозначений.....	7
Введение.....	8
1 Web-приложение для салона красоты.....	9
1.1 Анализ предметной области	9
1.1.1 Актуальность разработки.....	9
1.1.2 Анализ существующих аналогов	9
1.1.3 Вывод	14
1.1.4 Постановка задачи	15
1.2 Выбор технологий и средств разработки приложения	17
1.2.1 Выбор языка программирования	18
1.2.2 HTML	19
1.2.3 GSS	19
1.2.4 MySQL.....	19
1.2.5 Font Awesome	20
1.2.6. XAMPP.....	20
1.2.7. Выбор среды разработки	21
1.3 Проектирование информационной системы	22
1.3.1 Архитектура клиент-серверного приложения «BeautySalon».....	22
1.3.2 Проектирование базы данных	24
1.4 Разработка приложения	27
1.4.1 Реализация БД	27
1.4.2 Описание программы	30
1.4.3 Руководство пользователя	40
2 Охрана труда.....	45
2.1 Организация рабочего пространства	45
2.2 Освещение	46
2.3 Режим работы и отдыха	46
Заключение	48

Список использованных источников	49
Приложение А – UML-диаграммы	50
Приложение Б – Листинг программы	55
Приложение В – Презентация.....	103

					ПГУ 1.090301.32.001 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

Перечень сокращений и обозначений

В настоящей работе применяют следующие сокращения и обозначения

БД – база данных

СУБД – система управления базами данных

API (Application Programming Interface) – описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой

CRM система (Customer relationship management) – программа, которая помогает управлять бизнес-процессами

CSS (Cascading Style Sheets) – формальный язык декодирования и описания внешнего вида документа

DOM — это объектная модель документа, которую браузер создает в памяти компьютера на основании HTML-кода, полученного им от сервера

FK (Foreign key) – внешний ключ, понятие теории реляционных баз данных, относящееся к ограничениям целостности базы данных

HTML (HyperText Markup Language) – это язык разметки, используемый для создания веб-страниц

ID (Identifier) – уникальный номер объекта

IDE (Integrated development environment) – интегрированная среда разработки

JSON (JavaScript Object Notation) – текстовый формат обмена данными

PHP (Hypertext Preprocessor) — скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений

PK (Primary key) – первичный ключ, понятие теории реляционных баз данных, относящееся к ограничениям целостности базы данных

SQL (Structured Query Language) – язык структурированных запросов

SVG (Scalable Vector Graphics — масштабируемая векторная графика) — язык разметки масштабируемой векторной графики

UML (Unified Modeling Language) – унифицированный язык моделирования

UNIQUE – уникальное значение

Введение

В современном мире, где время является одним из самых ценных ресурсов, сфера услуг красоты переживает бурный рост. Салоны красоты стремятся предложить своим клиентам не только широкий спектр процедур, но и высокий уровень сервиса, удобство и индивидуальный подход. В этой связи, автоматизация бизнес-процессов и оптимизация взаимодействия с клиентами становятся ключевыми факторами успеха для предприятий индустрии красоты.

В последние годы наблюдается стремительный рост популярности онлайн-сервисов для записи на прием, просмотра каталога услуг и получения информации о салоне. Клиенты предпочитают иметь возможность самостоятельно выбирать удобное время, ознакомиться с ценами и отзывами, а также оперативно получать уведомления о предстоящих визитах. Владельцы салонов, в свою очередь, заинтересованы в эффективном управлении расписанием, клиентской базой, складским учетом и анализе эффективности работы.

Данная выпускная квалификационная работа посвящена разработке веб-приложения для салона красоты, которое позволит автоматизировать основные бизнес-процессы, улучшить качество обслуживания клиентов и повысить эффективность управления салоном. Целью работы является создание функционального, удобного и интуитивно понятного инструмента, отвечающего современным требованиям индустрии красоты.

1 Web-приложение для салона красоты

1.1 Анализ предметной области

1.1.1 Актуальность разработки

Разработка веб-приложения для онлайн записи в салон красоты сегодня становится всё более актуальной из-за растущих ожиданий клиентов к удобству и скорости обслуживания. Современный пользователь хочет иметь возможность записаться на услугу в любое время и из любого места, без необходимости звонить или лично посещать салон. Онлайн-запись значительно упрощает процесс бронирования, сокращая время ожидания и снижая нагрузку на сотрудников салона.

Кроме того, такое приложение помогает салонам красоты эффективно управлять расписанием мастеров и ресурсами, минимизируя риски двойного бронирования и простоев. Автоматизация процесса записи увеличивает общую производительность бизнеса, улучшает клиентский сервис и способствует удержанию постоянных клиентов за счёт удобства и прозрачности взаимодействия.

Наконец, в условиях высокой конкуренции на рынке услуг красоты наличие современного веб-приложения для онлайн-записи становится важным конкурентным преимуществом. Это не только улучшает имидж салона как технологичного и ориентированного на клиента, но и открывает дополнительные возможности для маркетинга и коммуникации, например, рассылки напоминаний, акций и персонализированных предложений.

1.1.2 Анализ существующих аналогов

В настоящее время на рынке представлены разные платформы для онлайн записи клиентов, каждая имеет свои особенности и недочеты. В данном пункте рассмотрены актуальные сервисы по схожей тематике:

- DIKIDI;
- salon1c.ru;
- yclients.

Рассмотрим возможности сервиса DIKIDI.

DIKIDI — самое популярное в России приложение для онлайн записи. Платформа для автоматизации бизнеса и онлайн-записи для предприятий сферы услуг, преимущественно салонов красоты.

Интерфейс сервиса DIKIDI представлен на рисунке 1.

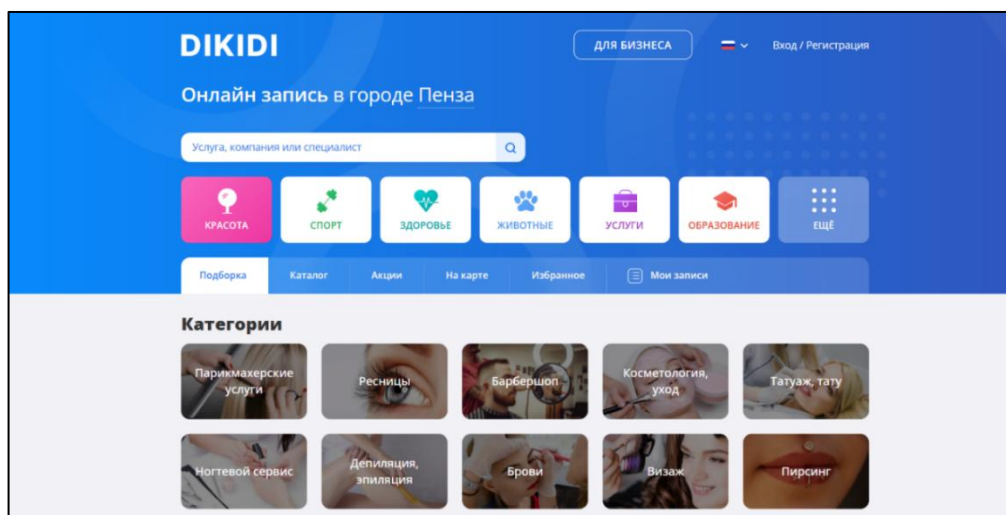


Рисунок 1 - Официальный сайт DIKIDI

Основными функциями DIKIDI являются:

- онлайн-запись, при которой клиент может через сайт, мобильное приложение или соцсети забронировать услугу в удобное для него время, а также выбрать конкретного мастера;
- управление расписанием, чтобы автоматически распределить нагрузку между сотрудниками с учётом перерывов, отпусков и рабочих часов;
- CRM и аналитика, благодаря которым можно вести клиентскую базу, анализировать загруженность, выручку и популярность услуги и сегментировать клиентов для персонализированных рассылок;
- контроль доходов и расходов, с возможностью формировать чеки и отчёты для налоговой;
- SMS и email-уведомления о записи, акциях и отменах.

Преимущества сервиса DIKIDI:

- автоматизация рутинных задач;

- программа делает анализ клиентской базы, на основе которого можно принимать решения;
- возможность выбора понравившегося мастера и удобного времени, при онлайн-записи;
- наличие смс-уведомлений, напоминающих о записи, для клиентов;
- понятный интерфейс (рисунок 2).

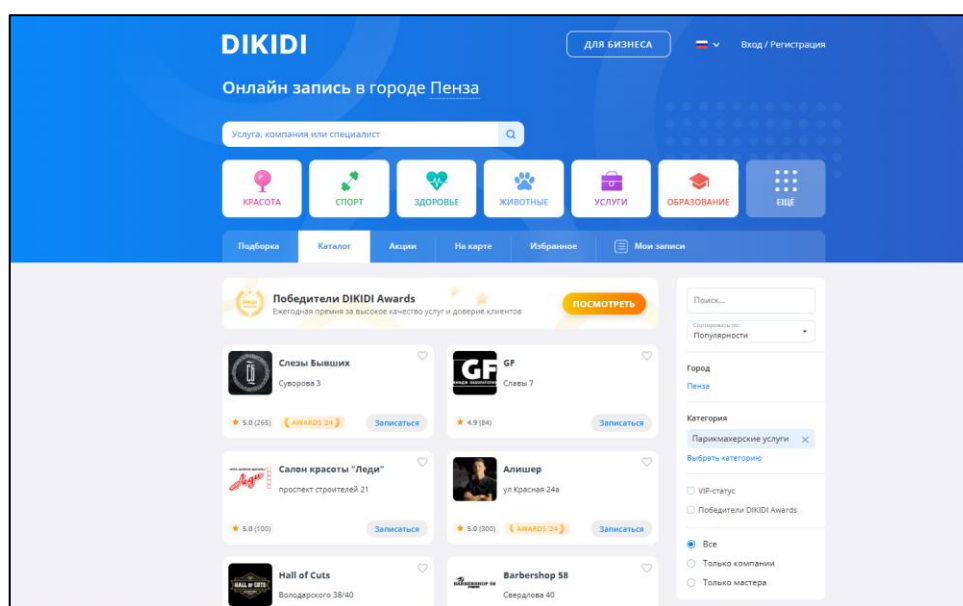


Рисунок 2 – Результат поиска «Парикмахерские услуги» в городе Пенза

Недостатки в работе сервиса DIKIDI:

- проблемы с производительностью;
- ограниченные функции при бесплатной версии;
- возможные проблемы с конфиденциальность данных.

Сайт salon1c.ru.

salon1c.ru — это CRM-система для управления как сетями, так и небольшими салонами, парикмахерскими, соляриями, массажными салонами и SPA.

Интерфейс сервиса представлен на рисунке 3.

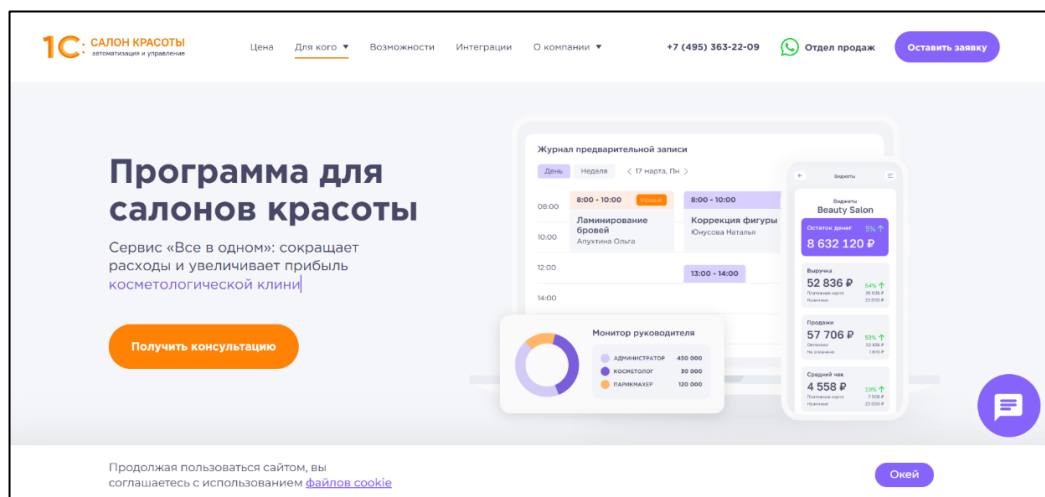


Рисунок 3 – Официальный сайт salon1c.ru

Основной функционал salon1c.ru:

- ведение клиентской базы, сегментация;
- SMS и email уведомления о записях, акциях и персональных предложениях;
- бронирование услуг с предпочтением времени и мастера;
- финансовый учёт и аналитика, для контроля доходов/расходов и формирования чеков и накладных;
- управление персоналом (учёт рабочего времени, отпусков и больничных; начисление зарплаты с учётом премий и штрафов; планирование смен и распределение нагрузки между сотрудниками);
- складской учёт.

Плюсы использования сайта salon1c.ru:

- широкий выбор услуг для бизнеса и частных лиц;
- удобный интерфейс и навигация;
- наличие полезных материалов по использованию программы;
- интеграции с СБП (рисунок 4).

Минусы использования salon1c.ru:

- отсутствие мобильной версии;
- ограниченный функционал;
- возможны задержки в обновлении информации о ценах и акциях.

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.32.001 ПЗ

Лист

12

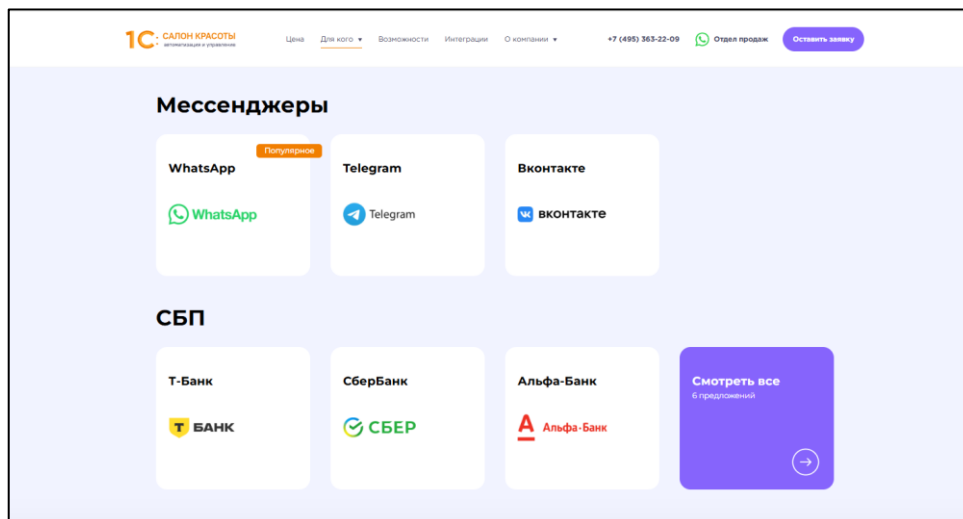


Рисунок 3 – Интеграции сайта «salon1c.ru»

Сервис YCLIENTS.

YCLIENTS — это приложение для онлайн записи и учёта клиентов.

На рисунке 4 показан интерфейс одного из сообществ.

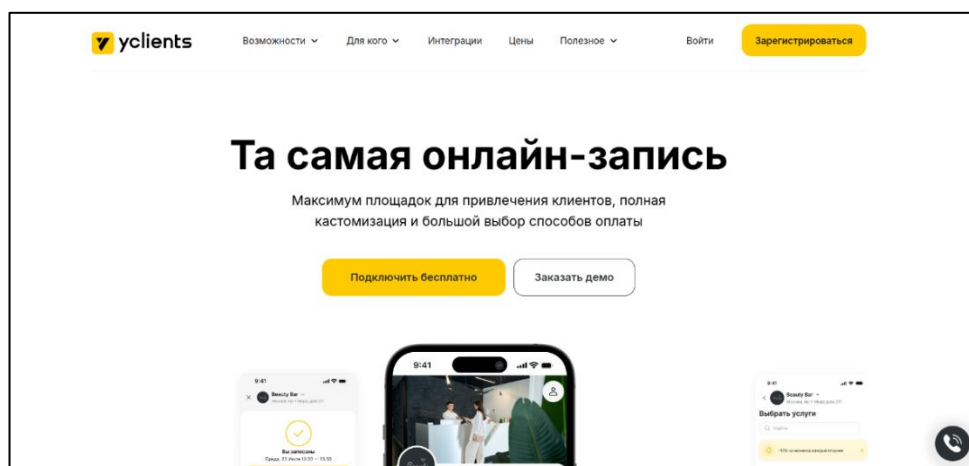


Рисунок 4 – Официальный сайт YCLIENTS

Задачи сервиса YCLIENTS:

- автоматическое распределение клиентов между сотрудниками по нагрузке или специализации;
- ведение базы клиентов с историей посещений и предпочтений (рисунок 5);
- финансовый учёт и отчётность;
- маркетинг и лояльность.

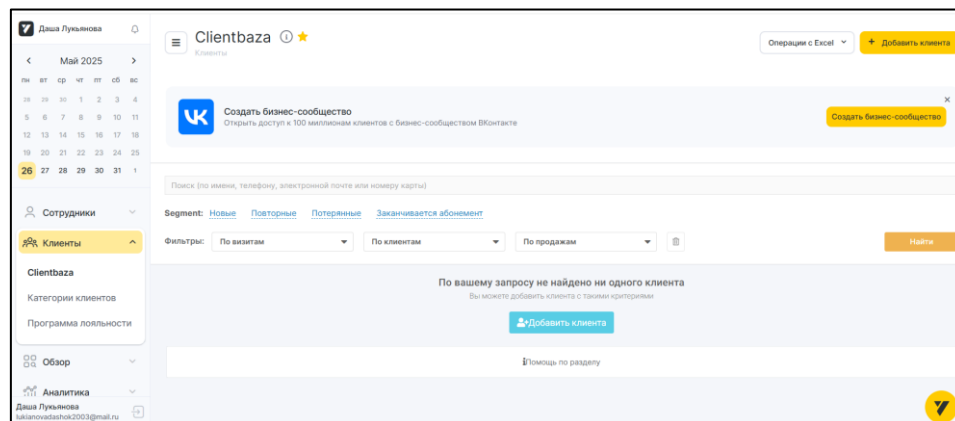


Рисунок 5 – Формирование клиентской базы в приложении YCLIENTS

Преимущества YCLIENTS:

- удобная система онлайн-записи для клиентов;
- широкий функционал для управления салонами, магазинами и другими бизнесами;
- AI-рекомендации: автоматический подбор услуг на основе истории клиента;
- возможность работы с несколькими филиалами в одной системе;
- поддержка работы в многопользовательском режиме;
- возможность интеграции с другими сервисами и crm;
- интуитивно понятный интерфейс.

Недостатки YCLIENTS:

- некоторые функции доступны только в платных тарифах;
- возможны проблемы с технической поддержкой;
- ограниченное количество интеграций в бесплатной версии
- пользователи сообщают о сложностях в миграции данных при переходе на другой сервис.

1.1.3 Вывод

Результаты анализа существующих аналогов можно представить в таблице 1, показывающие минусы и плюсы рассматриваемых сервисов.

Исходя из полученных результатов, было решено разработать клиент-серверное приложение «BeautySalon», которое будет предоставлять

пользователям удобный интерфейс, все основные функции, необходимые для комфортного использования, также обеспечить возможность авторизованным пользователям производить онлайн-запись на услуги в салоне красоты.

Таблица 1 – Достоинства и недостатки сервисов для онлайн-записи.

Функционал	«DIKIDI»	«salon1c.ru»	«YCLIENTS»
Онлайн-запись	+	+	+
Мобильное приложение	+	-	-
Понятный интерфейс	+	+	+
Безопасность данных	-	-	-
Онлайн-оплата	-	+	+
Бесплатный сервис	-	-	-
Смс-уведомления	+	+	+

1.1.4 Постановка задачи

Клиент-серверное приложение «BeautySalon» должно иметь следующий функционал:

Для администрации/сотрудников (рисунок 6):

- управление списком услуг и их стоимостью;
- управление графиком работы мастеров;
- просмотр и управление записями клиентов;
- ведение базы клиентов.

Для клиентов (рисунок 7):

- просмотр списка услуг и их описания;
- просмотр информации о мастерах салона;
- онлайн-запись на услуги к выбранному мастеру на удобное время;
- просмотр своих предстоящих и прошедших записей;
- регистрация и управление своим профилем.

Сервис «BeautySalon» должен соответствовать следующим техническим требованиям:

- удобный и интуитивно понятный интерфейс для работы с программой;
- возможность масштабирования интерфейса в процессе работы;
- наличие возможности сохранения данных локально, пока отсутствует подключения к серверу;
- операционная система: Windows 7 и более поздние версии.

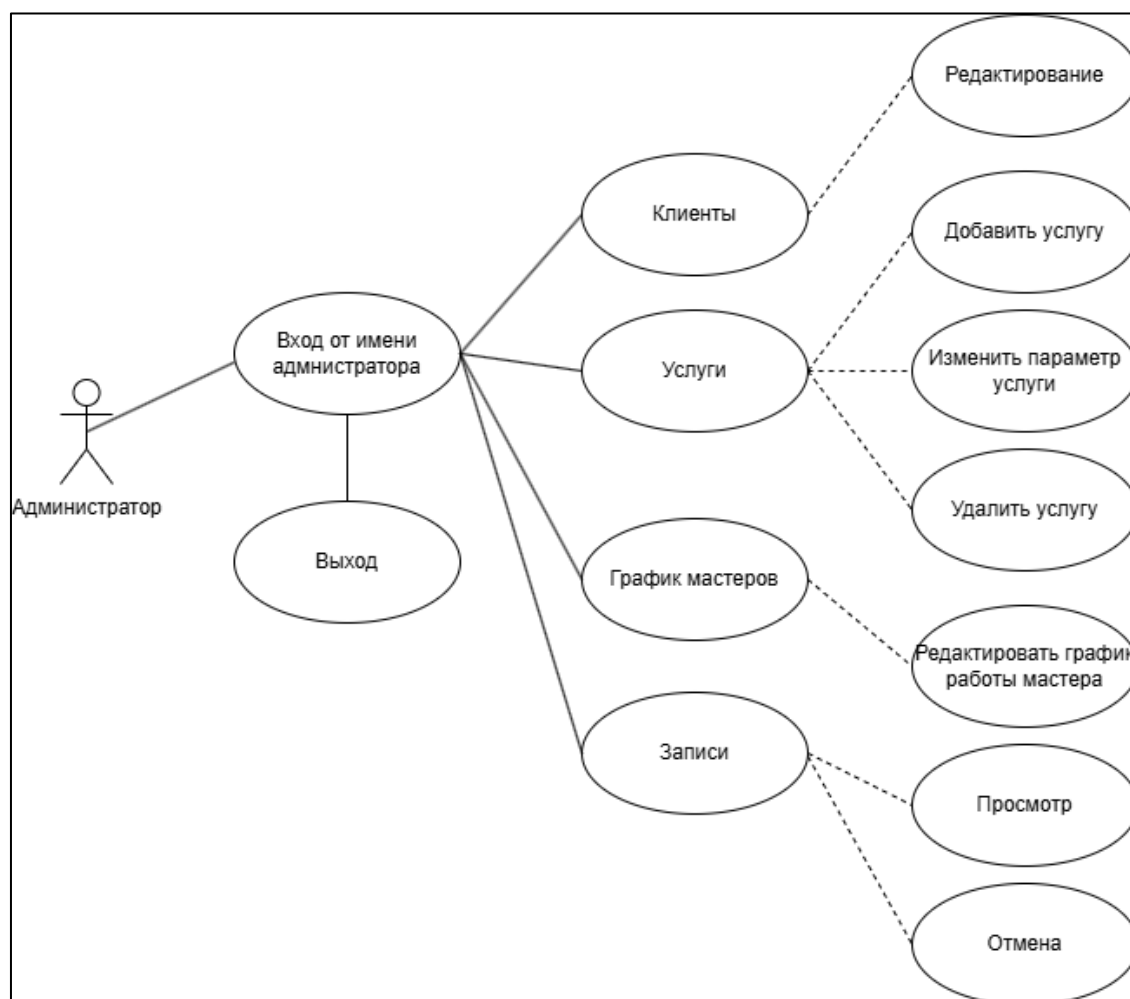


Рисунок 6 – Диаграмма вариантов использования приложения пользователя с учетной записью администратора

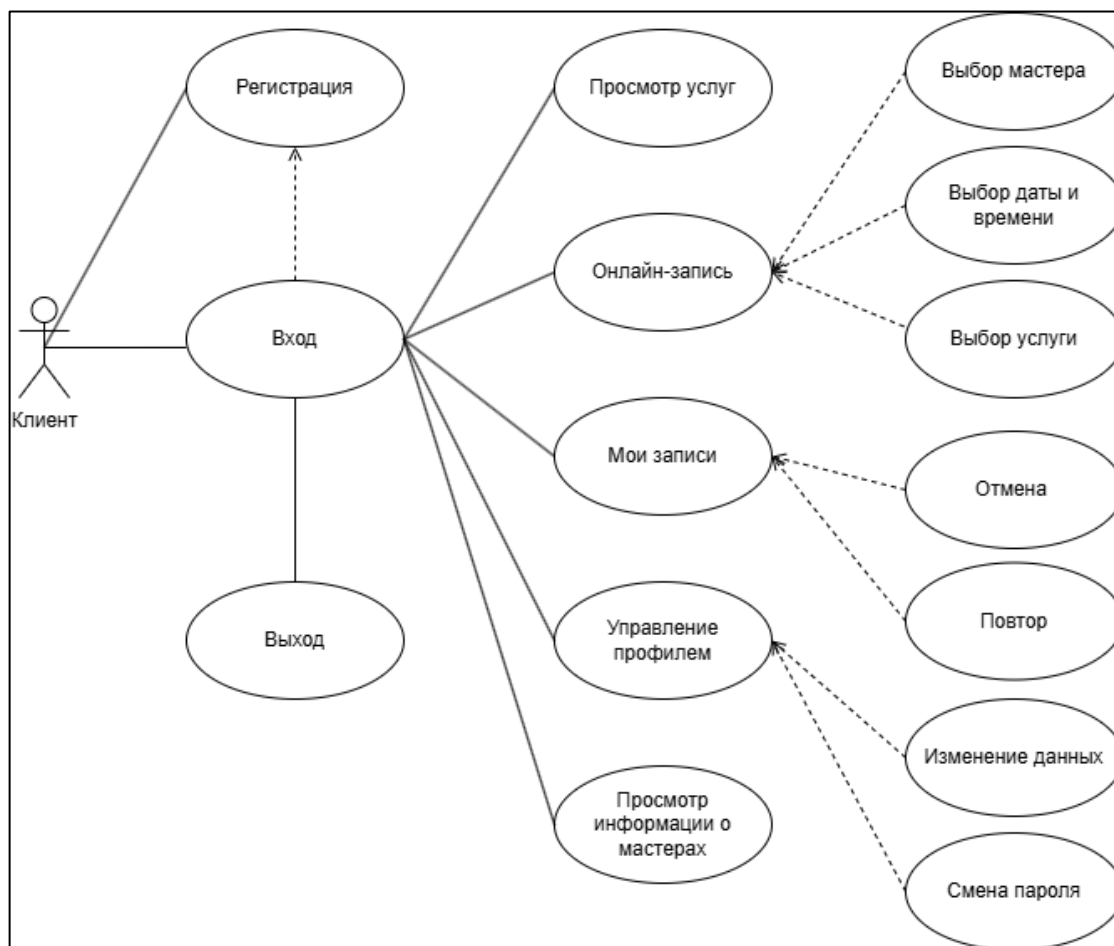


Рисунок 7 –Диаграмма вариантов использования приложения клиентом

1.2 Выбор технологий и средств разработки приложения

В данном под разделе будут описаны технологии и средства разработки, использованные при разработке программы «BeautySalon».

Ниже приведен список выбранных технологий и средств разработки:

- PHP;
- MySQL;
- HTML;
- CSS;
- JavaScript;
- Font Awesome;
- XAMPP.

1.2.1 Выбор языка программирования

PHP (Hypertext Preprocessor) — это серверный язык программирования, созданный специально для веб-разработки. Он появился в 1995 году и быстро стал одним из самых популярных инструментов для создания динамических сайтов. PHP легко встраивается в HTML-код, что позволяет генерировать контент «на лету», работать с формами, базами данных (например, MySQL) и управлять сессиями пользователей.

Синтаксис PHP напоминает C и Java, что делает его интуитивно понятным для разработчиков. Язык интерпретируемый, а его код выполняется на стороне сервера, что обеспечивает безопасность клиентской части. PHP поддерживает объектно-ориентированное программирование, трейты, анонимные функции и другие современные парадигмы [1,2].

JavaScript — это высокоуровневый, интерпретируемый язык программирования, который изначально создавался для добавления интерактивности на веб-страницы. Запускаясь в браузере, он позволяет «оживлять» статичный HTML-контент: обрабатывать клики, анимации, отправлять запросы к серверам без перезагрузки страницы и динамически менять структуру DOM. Со временем JavaScript вышел за рамки браузеров благодаря среде выполнения Node.js, что сделало его универсальным языком для серверной разработки, мобильных приложений (React Native) и даже десктопных программ (Electron). Его синтаксис, напоминающий C и Java, сочетает гибкость динамической типизации с поддержкой современных парадигм: объектно-ориентированного, функционального и асинхронного программирования.

Ключевые особенности JavaScript включают событийно-ориентированную модель, работу с асинхронными операциями через промисы (Promises) и async/await, а также прототипное наследование.

Сегодня JavaScript — один из самых популярных языков в мире, что подтверждается его использованием в гигантах вроде Google, Facebook и Netflix [3].

1.2.2 HTML

HTML (HyperText Markup Language) — это стандартный язык разметки, используемый для создания структуры веб-страниц. Он определяет элементы страницы, такие как заголовки, абзацы, изображения, ссылки и формы, с помощью тегов, которые браузер интерпретирует для отображения контента. HTML не является языком программирования — он отвечает за семантику и организацию содержимого, а для стилизации и интерактивности подключаются CSS и JavaScript.

Сегодня HTML остается фундаментом веб-разработки. Несмотря на появление фреймворков и инструментов вроде React или Vue.js, они всё равно компилируются в HTML для рендеринга в браузере [4].

1.2.3 CSS

CSS (Cascading Style Sheets) — это язык стилей, определяющий внешний вид веб-страниц, созданных с помощью HTML. Он позволяет управлять макетом, цветами, шрифтами, анимацией и адаптацией контента под разные устройства. CSS отделяет визуальное оформление от структуры документа, что делает код чище и упрощает поддержку.

Современный CSS предлагает мощные инструменты для сложных макетов: Flexbox для гибкого выравнивания элементов, Grid для двумерной сетки, а также медиазапросы для адаптивного дизайна под разные экраны. Дополнительные возможности включают анимации, переменные, фильтры и эффекты.

CSS-модули и методологии помогают организовать стили в больших проектах, избегая конфликтов. Несмотря на конкуренцию с CSS-in-JS (Styled Components), чистый CSS остаётся основой веб-дизайна, а его новые стандарты открывают доступ к низкоуровневым API, позволяя разработчикам создавать уникальные визуальные решения без ограничений браузерных движков [5].

1.2.4 MySQL

В качестве СУБД была выбрана MySQL — это реляционная система управления базами данных (СУБД) с открытым исходным кодом, разработанная для эффективного хранения, организации и обработки структурированных

данных. Она широко используется в веб-разработке благодаря своей надежности, простоте интеграции с такими языками, как PHP, Python и JavaScript. MySQL поддерживает стандартный язык запросов SQL, что позволяет создавать, модифицировать таблицы, выполнять сложные выборки и управлять отношениями между данными через внешние ключи. Её архитектура идеально подходит для проектов любого масштаба — от небольших сайтов до высоконагруженных приложений вроде социальных сетей или интернет-магазинов.

Преимущества MySQL:

- доступна бесплатно для использования, что делает ее доступной для широкого круга разработчиков;
- предлагает простой и интуитивно понятный язык запросов SQL для работы с данными;
- обладает хорошей производительностью и способна эффективно обрабатывать большие объемы данных;
- легко масштабируется как вертикально (добавление ресурсов на одном сервере), так и горизонтально (добавление дополнительных серверов) [6,7].

1.2.5. Font Awesome

Font Awesome — это популярная библиотека векторных иконок, предназначенная для упрощения интеграции графических элементов в веб-проекты. Иконки создаются в формате SVG и шрифтов, что позволяет масштабировать их без потери качества, а также стилизовать с помощью CSS: менять цвет, размер, добавлять тени, градиенты и анимации. Библиотека включает тысячи иконок, охватывающих темы от социальных сетей и интерфейсных элементов до тематических изображений. Font Awesome поддерживает все современные браузеры и устройства, что делает её универсальным инструментом для адаптивного дизайна.

1.2.6. XAMPP

ХАМРР — это бесплатный кроссплатформенный веб-сервер с открытым исходным кодом, предназначенный для локальной разработки и тестирования веб-приложений. В его состав входят ключевые компоненты: сервер Apache (для обработки HTTP-запросов), СУБД MySQL (или MariaDB), интерпретаторы языков PHP и Perl, а также инструменты phpMyAdmin (для управления базами данных) и FileZilla FTP. Пакет работает на Windows, macOS и Linux, что делает его универсальным решением для создания локальной среды без необходимости настройки каждого компонента отдельно.

Главное преимущество ХАМРР — простота установки и использования. ХАМРР предоставляет панель управления для запуска/остановки служб, настройки портов и доступа к логам [8].

1.2.7 Выбор среды разработки

PhpStorm — это мощная интегрированная среда разработки (IDE) от JetBrains, созданная специально для программистов на PHP. Она предоставляет продвинутые инструменты для написания, отладки и поддержки кода: умное автодополнение, анализ кода в реальном времени, интеграцию с базами данных (MySQL, PostgreSQL). Среда автоматически обнаруживает ошибки синтаксиса, предлагает рефакторинг и оптимизирует работу с шаблонами. PhpStorm поддерживает HTML, CSS, JavaScript и TypeScript, что делает её универсальным решением для веб-разработки. Кроссплатформенность (Windows, macOS, Linux) и интеграция с Docker, Vagrant и системами контроля версий позволяют работать в едином пространстве, не переключаясь между приложениями.

PhpStorm выделяется встроенным отладчиком с поддержкой Xdebug и профилированием, что ускоряет поиск и устранение ошибок. Инструменты для тестирования помогают писать и запускать тесты, а интеграция с REST-клиентом упрощает работу с API. Настраиваемый интерфейс, темы оформления и горячие клавиши повышают продуктивность, а плагины расширяют функционал.

– уровень данных (База данных): Сервер MySQL, хранящий всю информацию приложения.

Клиент (браузер) запрашивает страницу. Веб-сервер Apache [9] обрабатывает запрос и передает его PHP-интерпретатору. PHP-скрипт выполняется: подключает необходимые файлы, устанавливает соединение с базой данных MySQL, выполняет SQL-запросы для получения данных, обрабатывает пользовательский ввод, взаимодействует с сессиями для аутентификации пользователей, генерирует HTML-код страницы, вставляя в него полученные из БД данные. Сгенерированный HTML-код отправляется обратно в браузер клиента. Браузер отображает HTML, применяет CSS-стили и выполняет JavaScript-код. JavaScript на стороне клиента может выполнять AJAX-запросы к PHP-скриптам для получения данных без перезагрузки страницы. Эти PHP-скрипты также обращаются к БД и возвращают данные в формате JSON [10]. На рисунке 10 представлено взаимодействие компонентов приложения в виде диаграммы последовательности.

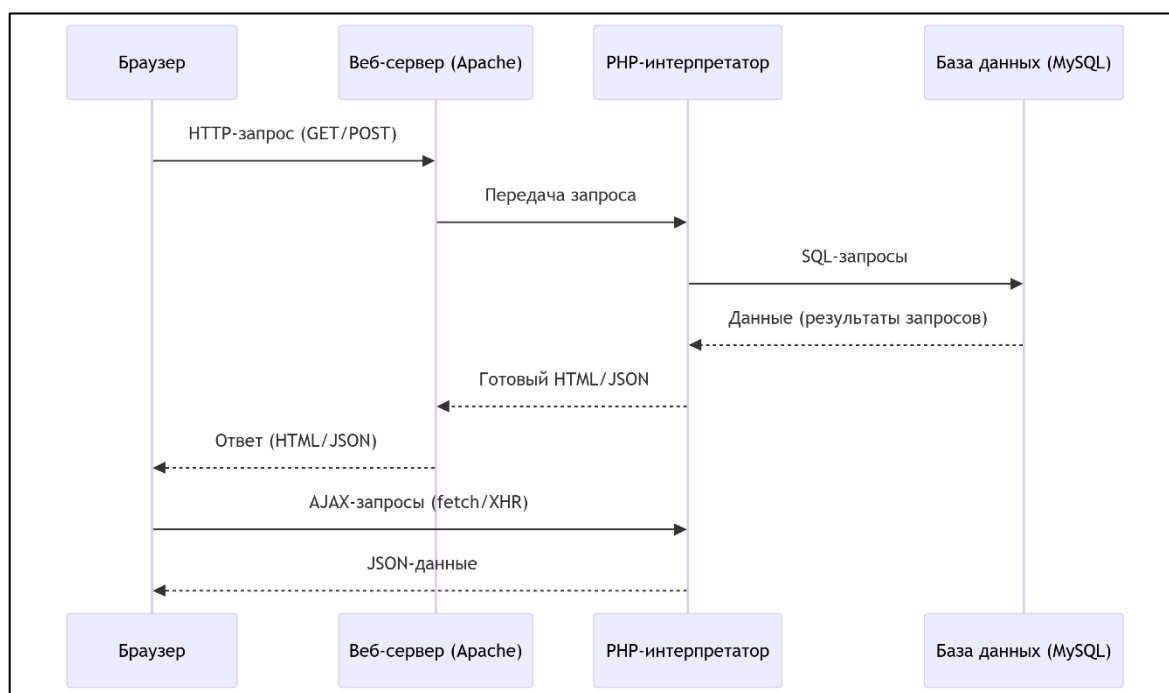


Рисунок 10 – Взаимодействие компонентов приложения

1.3.2 Проектирование базы данных

База данных beautysalon спроектирована для хранения информации о пользователях, услугах, сотрудниках, записях и графиках работы.

Основные таблицы:

Таблица 2 – Атрибуты таблицы users (информация о пользователях (клиенты, администраторы))

Имя столбца	Тип данных	Описание	Роль атрибута	Пример
user_id (UNIQUE)	INT	Идентификатор учетной записи	РК	1
first_name	VARCHAR(50)	Имя клиента		Иван
last_name	VARCHAR(50)	Фамилия клиента		Иванов
email (UNIQUE)	VARCHAR(100)	Логин пользователя		Ivanov000@mail.ru
phone_number	VARCHAR(20)	Номер телефона		+79001234567
Имя столбца	Тип данных	Описание	Роль атрибута	Пример
password_hash	VARCHAR(255)	Хэш пароля		\$2b\$12\$5KLhT7W1sC9Z7U9f3qo0.eQ6s0d4YV1mR8uOj3bN1sLxVcZz4vD7a
role	ENUM	Роль		Клиент
created_at	TIMESTAMP	Время создания записи о клиенте.		2024-10-10 00:00

Таблица 3 – Атрибуты таблицы services (каталог услуг)

Имя столбца	Тип данных	Описание	Роль атрибута	Пример
id	INT(11)	Идентификатор услуги	РК	4
name	VARCHAR(255)	Название услуги		Педикюр аппаратный

description	TEXT	Описание услуги		Комплексный уход за стопами с использованием аппарата, обработка кутикулы и ногтей
duration	VARCHAR(50)	Длительность услуги		75
price	DECIMAL(10,2)	Цена услуги		2200.00
category	VARCHAR(100)	Категория услуги		Ногтевой сервис
image_filename	VARCHAR(255)	Фото услуги		images/services/service_hair_cut_female.jpg
created_at	TIMESTAMP	Время создания записи		2025-05-28 17:01:15
updated_at	TIMESTAMP	Время создания записи		2025-05-28 17:01:15

Таблица 4 – Атрибуты таблицы staff (данные о сотрудниках салона)

Имя столбца	Тип данных	Описание	Роль атрибута	Пример
staff_id	INT(11)	Идентификатор сотрудника	PK	2
first_name	VARCHAR(50)	Имя сотрудника		Елена
last_name	VARCHAR(50)	Фамилия сотрудника		Смирнова
specialization	VARCHAR(100)	Специализация		Ведущий Стилист-Парикмахер
profile_image_url	VARCHAR(255)	Фото сотрудника		https://placeholder.co/500x500/1E88E5/FFFFFF?text=Елена+С.&font=lor
bio	TEXT	Слоган		Люблю сложные окрашивания и креативные стрижки

created_at	TIMESTAMP	Время создания записи		2025-05-28 17:01:15
updated_at	TIMESTAMP	Время создания записи		2025-05-28 17:01:15

Таблица 5 – Атрибуты таблицы appointments (записи клиентов на услуги)

Имя столбца	Тип данных	Описание	Роль атрибута	Пример
appointment_id	INT(11)	Идентификатор записи	PK	21
client_id	INT(11)	Идентификатор клиента	FK	3
staff_id	INT(11)	Идентификатор сотрудника	FK	1
service_id	INT(11)	Идентификатор услуги	FK	5
appointment_datetime	DATETIME	Дата записи на услугу		1
Имя столбца	Тип данных	Описание	Роль атрибута	Пример
status	ENUM	Статус записи		Забронировано
notes	TEXT	Заметки для записи		Нет
created_at	TIMESTAMP	Время создания записи о записи.		2025-02-10 11:00

Связи между таблицами реализованы с помощью внешних ключей для обеспечения целостности данных (рисунок 11).

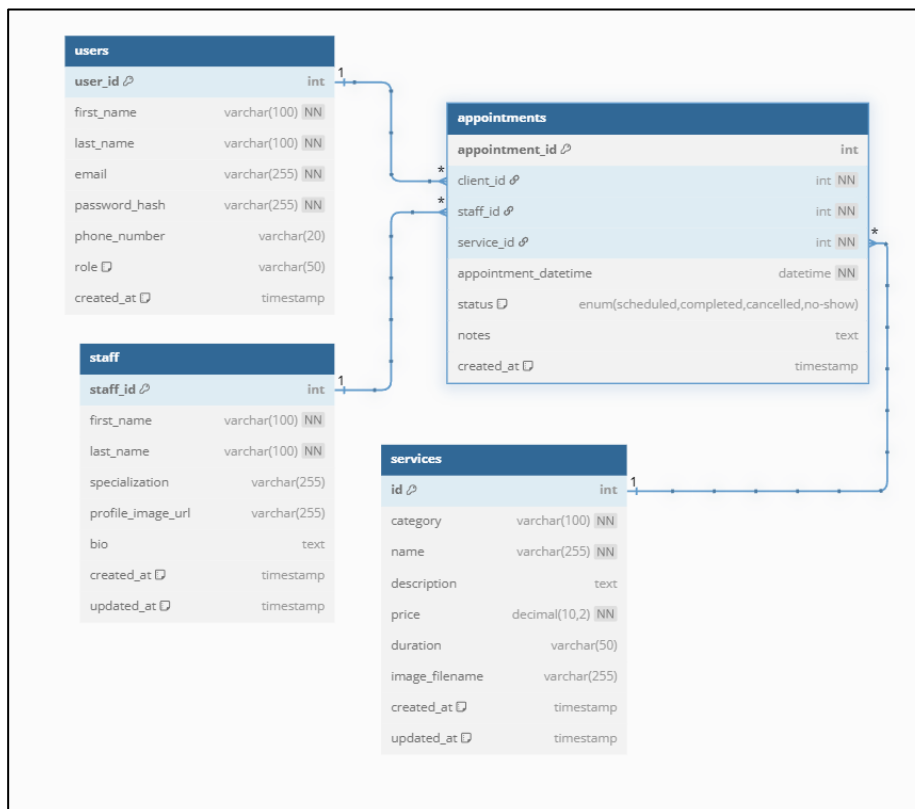


Рисунок 11 – Физическая модель БД

1.4 Разработка приложения

1.4.1 Реализация БД

Перед началом реализации БД её необходимо создать с помощью SQL-запроса:

```
CREATE DATABASE IF NOT EXISTS beautysalon;
```

Затем создадим внутри БД необходимые таблицы. В данном разделе будет описан процесс создания и работа с основными таблицами БД.

Для создания таблицы, которая будет хранить информацию о записях клиентов на услуги, напомним следующий запрос:

```
CREATE TABLE `appointments` (
  `appointment_id` int(11) NOT NULL,
  `client_id` int(11) NOT NULL,
  `staff_id` int(11) NOT NULL,
  `service_id` int(11) NOT NULL,
  `appointment_datetime` datetime NOT NULL,
  `status` enum('scheduled','completed','cancelled','no-show')
  DEFAULT 'scheduled',
  `notes` text DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()
```

```

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

```

Для создания таблицы, которая будет хранить информацию о услугах, напишем следующий запрос:

```

CREATE TABLE `services` (
  `id` int(11) NOT NULL,
  `category` varchar(100) NOT NULL,
  `name` varchar(255) NOT NULL,
  `description` text DEFAULT NULL,
  `price` decimal(10,2) NOT NULL,
  `duration` varchar(50) DEFAULT NULL,
  `image_filename` varchar(255) DEFAULT NULL,
  `badge_text` varchar(50) DEFAULT NULL,
  `badge_color` varchar(7) DEFAULT NULL,
  `is_featured` tinyint(1) DEFAULT 0,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NOT NULL DEFAULT current_timestamp()
ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

```

Для создания таблицы, которая будет хранить сведения о сотрудниках, напишем следующий запрос:

```

CREATE TABLE `staff` (
  `staff_id` int(11) NOT NULL,
  `first_name` varchar(100) NOT NULL,
  `last_name` varchar(100) NOT NULL,
  `specialization` varchar(255) DEFAULT NULL,
  `profile_image_url` varchar(255) DEFAULT NULL,
  `bio` text DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NOT NULL DEFAULT current_timestamp()
ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

```

Для создания таблицы, которая будет хранить информацию о клиентах, напишем следующий запрос:

```

CREATE TABLE `users` (
  `user_id` int AUTO_INCREMENT PRIMARY KEY,
  `first_name` varchar(100) NOT NULL,
  `last_name` varchar(100) NOT NULL,
  `email` varchar(255) NOT NULL,
  `password_hash` varchar(255) NOT NULL,
  `phone_number` varchar(20) DEFAULT NULL,
  `role` varchar(50) NOT NULL DEFAULT 'client',
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

```

Для добавления данных в таблицу использовался оператор INSERT.

Например, следующий запрос добавляет в таблицу staff информацию о мастерах:

```
INSERT INTO `staff` (`staff_id`, `first_name`, `last_name`,
`specialization`, `profile_image_url`, `bio`, `created_at`,
`updated_at`) VALUES
(7, 'Анна', 'Соколова', 'Топ-стилист, Арт-директор',
'specialist_anna_sokolova.jpg', '12+ лет опыта. Победитель
международных конкурсов. Креативные трансформации и сложные
окрашивания.', '2025-05-28 18:08:53', '2025-05-28 18:08:53'),
(8, 'Ольга', 'Максимова', 'Ведущий мастер нейл-арта',
'specialist_olga_maksimova.jpg', 'Эксперт по сложным дизайнам,
аппаратному маникюру и медицинскому педикюру. Более 9 лет в
индустрии.', '2025-05-28 18:08:53', '2025-05-28 18:08:53'),
(9, 'Елена', 'Ковалева', 'Врач дермато-косметолог',
'specialist_elena_kovaleva.jpg', 'Высшее медицинское образование.
Специалист по инъекционным методикам, аппаратной косметологии и
лечению кожи.', '2025-05-28 18:08:53', '2025-05-28 18:08:53');
```

В данном запросе мы заполняем таблицу services информацией об услугах:

```
INSERT INTO `services` (`id`, `category`, `name`,
`description`, `price`, `duration`, `image_filename`, `badge_text`,
`badge_color`, `is_featured`, `created_at`, `updated_at`) VALUES
(1, 'Парикмахерское Искусство', 'Авторская Женская Стрижка',
'Индивидуальный подбор формы и стиля, мытье головы премиум-шампунем
и стайлинг.', 2500.00, '90 мин',
'images/services/service_hair_cut_female.jpg', 'Popular', '#3B82F6',
1, '2025-05-28 17:01:15', '2025-05-28 17:01:15'),
(2, 'Парикмахерское Искусство', 'Сложное Окрашивание
(Airtouch/Balayage)', 'Техники Airtouch, балаяж, шатуш.
Использование щадящих красителей.', 7000.00, '3-5 часов',
'images/services/service_hair_coloring_complex.jpg', 'Хит!',
'#EF4444', 1, '2025-05-28 17:01:15', '2025-05-28 17:01:15'),
(3, 'Парикмахерское Искусство', 'Вечерняя или Свадебная
Прическа', 'Создание элегантного образа для любого торжественного
случая.', 4000.00, '2-3 часа',
'images/services/service_hair_styling_event.jpg', NULL, NULL, 0,
'2025-05-28 17:01:15', '2025-05-28 17:01:15'),
(4, 'Ногтевой Сервис', 'Премиум Маникюр с Покрытием Гель-Лак',
'Аппаратный или комбинированный маникюр, выравнивание ногтевой
пластины, покрытие гель-лаком Luxio/Beautix.', 2200.00, '120 мин',
'images/services/service_nails_manicure_gel.jpg', 'New!', '#22C55E',
1, '2025-05-28 17:01:15', '2025-05-28 17:01:15'),
(5, 'Ногтевой Сервис', 'SPA-Педикюр "Блаженство"', 'Полный
комплекс ухода за стопами, обработка пальцев и пяток, массаж,
покрытие лаком или гель-лаком.', 3000.00, '90 мин',
'images/services/service_nails_pedicure_spa.jpg', NULL, NULL, 0,
'2025-05-28 17:01:15', '2025-05-28 17:01:15'),
(6, 'Косметология', 'Атравматичная Чистка Лица Holy Land',
'Глубокое очищение без повреждения кожи, подходит для всех типов,
включая чувствительную.', 4500.00, '90 мин',
```

```
'images/services/service_cosmo_facial_holy_land.jpg', 'Рекомендуем',
'#F59E0B', 1, '2025-05-28 17:01:15', '2025-05-28 17:01:15'),
    (7, 'Косметология', 'Биоревитализация Juvederm Hydrate',
'Инъекционное омоложение, глубокое увлажнение и восстановление
упругости кожи гиалуроновой кислотой.', 8500.00, '60 мин',
'images/services/service_cosmo_biorevitalization.jpg', NULL, NULL,
0, '2025-05-28 17:01:15', '2025-05-28 17:01:15');
```

1.4.2 Описание программы

Описание клиентской части.

Файл `animations.css` реализует анимацию и эффекты появления для элементов страницы. Разберем некоторые части:

Анимация (пульсация) иконок:

```
@keyframes pulseIcon {
    0% { transform: scale(1); }
    50% { transform: scale(1.15) rotate(5deg); }
    100% { transform: scale(1) rotate(0deg); }
}
```

Эффект плавного появления секций:

```
.fade-in-section {
    opacity: 0;
    transform: translateY(30px);
    transition: opacity 0.7s ease-out, transform 0.7s ease-
out;
}
.fade-in-section.is-visible {
    opacity: 1;
    transform: translateY(0);
}
```

Последовательная анимация элементов:

```
.hero-content h1 {
    animation: fadeInDown 1s 0.2s ease-out forwards;
    opacity: 0;
}
.hero-content p {
    animation: fadeInDown 1s 0.5s ease-out forwards;
    opacity: 0;
}
.hero-content .btn {
    animation: fadeInDown 1s 0.8s ease-out forwards;
    opacity: 0;
}
```

Анимация `fadeInDown` (появление сверху):

```
@keyframes fadeInDown {
    from {
        opacity: 0;
        transform: translateY(-40px);
    }
    to {
```

```

        opacity: 1;
        transform: translateY(0);
    }
}

```

Файл main.js представляет из себя код, выполняющий несколько ключевых функций для веб-сайта салона красоты:

- автоматически добавляет класс active к текущей активной ссылке в навигации, сравнивая URL страницы с URL ссылок;
- обрабатывает клики по якорным ссылкам (начинающимся с #), обеспечивая плавную прокрутку к соответствующим разделам;
- использует IntersectionObserver для добавления класса is-visible к элементам с классом fade-in-section при их появлении в области видимости (эффект постепенного появления);
- управляет модальным окном (#staff-schedule-modal);
- обрабатывает ошибки (например, отсутствие данных или проблемы с сетью).

Вспомогательная функции: getBaseUrlForJs() – определяет базовый URL проекта для корректных AJAX-запросов:

```

function getBaseUrlForJs() {
    const logoLink = document.querySelector('a.logo');
    if (logoLink && logoLink.href) {
        let baseUrl = logoLink.href;
        if (baseUrl.endsWith('index.php')) {
            baseUrl = baseUrl.substring(0,
baseUrl.lastIndexOf('/') + 1);
        } else if (!baseUrl.endsWith('/')) {
            baseUrl = baseUrl.substring(0,
baseUrl.lastIndexOf('/') + 1);
        }
        return baseUrl;
    }
    const pathParts = window.location.pathname.split('/');
    const projectFolderName = 'BeautySalon';
    const projectFolderIndex = pathParts.indexOf(projectFolderName);
    if (projectFolderIndex !== -1) {
        return window.location.origin +
pathParts.slice(0, projectFolderIndex + 1).join('/') +
'/';
    }
}

```

```

    }
    return window.location.origin + '/';
}

```

Файл booking.js представляет из себя код, создающий интерактивную форму, которая помогает клиентам выбрать удобное время для записи, учитывая занятость мастеров и длительность услуги.

Формирование корректного базового URL для AJAX-запроса.

```

function getBaseUrlForBookingJs() {
    const logoLink = document.querySelector('a.logo');
    if (logoLink && logoLink.href) {
        let baseUrl = logoLink.href;
        if (baseUrl.endsWith('index.php')) {
            baseUrl = baseUrl.substring(0,
baseUrl.lastIndexOf('/') + 1);
        } else if (!baseUrl.endsWith('/')) {
            baseUrl = baseUrl.substring(0,
baseUrl.lastIndexOf('/') + 1);
        }
        return baseUrl.endsWith('/') ? baseUrl : baseUrl +
'/';
    }

    const pathParts = window.location.pathname.split('/');
    const projectFolderName = 'BeautySalon'; // Make sure
this matches your project folder
    const projectFolderIndex =
pathParts.indexOf(projectFolderName);
    if (projectFolderIndex !== -1) {
        return window.location.origin + pathParts.slice(0,
projectFolderIndex + 1).join('/') + '/';
    }
    return window.location.origin + '/'; // Fallback
}

```

При изменении мастера, времени или услуги вызывается функция:

```

function fetchAvailableSlots() {
    if (!staffSelect || !dateInput || !serviceSelect ||
!availableSlotsContainer || !timeInput) {
        return;
    }

    const staffId = staffSelect.value;
    const selectedDate = dateInput.value;
    const selectedServiceOption =
serviceSelect.options[serviceSelect.selectedIndex];
    const serviceDuration = selectedServiceOption ?
selectedServiceOption.getAttribute('data-duration') : null;

    availableSlotsContainer.innerHTML = '';
}

```

```

        timeInput.value = '';
        const activeSlotButton = document.querySelector('.slot-button.active');
        if (activeSlotButton) {
            activeSlotButton.classList.remove('active');
        }

        if (!staffId || !selectedDate || !serviceDuration) {
            availableSlotsContainer.innerHTML = '<p class="text-muted-color" style="font-size:0.9em; margin-top:5px;"><small>Выберите мастера, дату и услугу для просмотра доступных временных слотов.</small></p>';
            return;
        }

```

Описание серверной части.

PHP-скрипт register.php реализует регистрацию новых пользователей с проверкой уникальности email и хэшированием паролей:

PHP-скрипт login.php обеспечивает вход зарегистрированных пользователей с проверкой учетных данных:

```

<?php
$page_title = "Вход";
// Убедимся, что сессия стартует до любого вывода HTML
if (session_status() == PHP_SESSION_NONE) {
    session_start();
}
include 'includes/header.php'; // $base_url должен быть
определен здесь или ранее
require_once 'includes/db_connect.php';

$errors = [];

// Если $base_url не определен в header.php, определите его
здесь
if (!isset($base_url)) {
    $script_name = $_SERVER['SCRIPT_NAME'];
    $project_folder_name = 'BeautySalon';
    if (strpos($script_name, '/' . $project_folder_name . '/')
    !== false) {
        $base_url = '/' . $project_folder_name . '/';
    } else {
        $base_url = '/';
    }
}

// Если пользователь уже вошел, перенаправляем на страницу "Мои
записи"
if (isset($_SESSION['user_id'])) {

```

```

        header("Location: " . htmlspecialchars($base_url) .
"my_appointments.php");
        exit();
    }

    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $email = trim($_POST['email']);
        $password = $_POST['password'];

        if (empty($email)) {
            $errors[] = "Email обязателен для заполнения.";
        } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
            $errors[] = "Некорректный формат email.";
        }

        if (empty($password)) {
            $errors[] = "Пароль обязателен для заполнения.";
        }

        if (empty($errors)) {
            $stmt = $conn->prepare("SELECT user_id, first_name,
password_hash, role FROM users WHERE email = ?");
            if ($stmt) {
                $stmt->bind_param("s", $email);
                $stmt->execute();
                $result = $stmt->get_result();

                if ($result->num_rows == 1) {
                    $user = $result->fetch_assoc();
                    if (password_verify($password,
$user['password_hash'])) {
                        // Успешный вход
                        $_SESSION['user_id'] = $user['user_id'];
                        $_SESSION['user_first_name'] =
$user['first_name'];
                        $_SESSION['user_role'] = $user['role'];

                        // Перенаправление после успешного входа
                        // Если было прервано бронирование, вернем
на страницу бронирования
                        if (isset($_SESSION['redirect_to_booking'])
&& $_SESSION['redirect_to_booking'] === true) {
                            unset($_SESSION['redirect_to_booking']);
                            $booking_params_query = '';
                            if(isset($_SESSION['booking_params'])
&& !empty($_SESSION['booking_params'])) {
                                $booking_params_query = '?' .
http_build_query($_SESSION['booking_params']);
                                unset($_SESSION['booking_params']);
                            }

```

```

        header("Location: " .
htmlspecialchars($base_url) . "booking.php"
$booking_params_query);
        exit();
    } else {
        header("Location: " .
htmlspecialchars($base_url) . "my_appointments.php");
        exit();
    }
    } else {
        $errors[] = "Неверный email или пароль.";
    }
    } else {
        $errors[] = "Неверный email или пароль.";
    }
    $stmt->close();
} else {
    $errors[] = "Ошибка подготовки запроса: " . $conn-
>error;
}
}
$conn->close();
}
?>

<div class="container page-container auth-container">
    <h1 class="page-title">Вход в Аккаунт</h1>

    <?php if (isset($_GET['message'])): ?>
        <div class="alert alert-success"><?php echo
htmlspecialchars(urldecode($_GET['message'])); ?></div>
    <?php endif; ?>

    <?php if (!empty($errors)): ?>
        <div class="alert alert-danger">
            <ul>
                <?php foreach ($errors as $error): ?>
                    <li><?php echo htmlspecialchars($error);
?></li>
                <?php endforeach; ?>
            </ul>
        </div>
    <?php endif; ?>

    <form action="<?php echo htmlspecialchars($base_url) .
'login.php'); ?>" method="POST" class="auth-form">
        <div class="form-group">
            <label for="email">Email</label>
            <input type="email" id="email" name="email"
value="<?php echo isset($_POST['email']) ?
htmlspecialchars($_POST['email']) : ''; ?>" required>
        </div>
        <div class="form-group">

```

```

        <label for="password">Пароль</label>
        <input          type="password"          id="password"
name="password" required>
    </div>
    <button          type="submit"          class="btn          btn-primary
futuristic-button">Войти</button>
</form>
    <p class="auth-link">Нет аккаунта? <a href="<?php echo
htmlspecialchars($base_url          .          'register.php');
?>">Зарегистрироваться</a></p>
</div>
<?php include 'includes/footer.php'; ?>

```

PHP-скрипт `logout.php` позволяет пользователям выходить из системы, а также управляет сессиями PHP для отслеживания аутентифицированных пользователей:

```

<?php
// Строго в начале файла, до любого вывода HTML
if (session_status() == PHP_SESSION_NONE) {
    session_start();
}

// Определение $base_url, если он не определен глобально
// Эта логика должна быть такой же, как и в других ваших файлах
(header.php, index.php и т.д.)
if (!isset($base_url)) {
    $script_name = $_SERVER['SCRIPT_NAME'];
    $project_folder_name = 'BeautySalon'; // Убедитесь, что это
имя вашей папки проекта

    if (strpos($script_name, '/' . $project_folder_name . '/')
!= false) {
        $base_url = '/' . $project_folder_name . '/';
    }
    // Проверка, если скрипт в корневой папке проекта, которая
является корнем домена
    // Это более сложный случай, и может потребоваться более
точное определение
    // Для простоты, если не в подпапке BeautySalon, считаем
корнем '/'
    else {
        $base_url = '/';
    }
}

// 1. Очистить все переменные сессии
$_SESSION = array();

// 2. Если используется идентификатор сессии в cookie, удалить
cookie сессии
// Это важно для полного уничтожения сессии.

```

```

if (ini_get("session.use_cookies")) {
    $params = session_get_cookie_params();
    setcookie(session_name(), '', time() - 42000,
        $params["path"], $params["domain"],
        $params["secure"], $params["httponly"]
    );
}

// 3. Уничтожить сессию
// Это удаляет все данные, связанные с текущей сессией.
session_destroy();

// 4. Перенаправить на главную страницу или страницу входа
// Использование htmlspecialchars для $base_url здесь не
обязательно, т.к. это URL,
// но не повредит для единообразия.
header("Location: " . htmlspecialchars($base_url) .
    "index.php");
exit(); // Важно вызвать exit() после header("Location:") для
прекращения выполнения скрипта.
?>

```

Index.php – главная страница с общей информацией, популярными услугами и призывами к действию.

```

<?php
$page_title = "Главная";
include 'includes/header.php';
require_once 'includes/db_connect.php';
?>

<section class="hero-section">
    <div class="hero-overlay"></div>
    <div class="container hero-content">
        <h1 class="animated-text">Добро пожаловать в
        BeautySalon</h1>
        <p class="animated-text-delay">Ваш оазис красоты и
        релаксации в самом сердце города.</p>
        <a href="<?php echo $base_url; ?>booking.php"
        class="btn btn-primary btn-lg futuristic-button">Записаться
        онлайн</a>
    </div>
</section>

<section class="about-us-short">
    <div class="container">
        <h2>О нас</h2>
        <p>Мы предлагаем широкий спектр услуг для вашей красоты
        и здоровья. Наши мастера – профессионалы своего дела,
        использующие только лучшую косметику и современные технологии.
        Погрузитесь в атмосферу заботы и комфорта.</p>
    </div>
</section>

```

```

<section class="featured-services">
    <div class="container">
        <h2>Популярные Услуги</h2>
        <div class="services-grid">
            <?php
                $sql = "SELECT service_id, service_name,
description, price, category FROM services ORDER BY RAND()
LIMIT 3";
                $result = $conn->query($sql);
                if ($result && $result->num_rows > 0) {
                    while($row = $result->fetch_assoc()) {
                        echo "<div class='service-card-featured'>";
                        echo "    <div class='service-icon'><i
class='fas fa-spa'></i></div>"; // Generic icon
                        echo "        <h3>" .
htmlspecialchars($row['service_name']) . "</h3>";
                        echo "        <p>" .
htmlspecialchars(substr($row['description'], 0, 100)) .
"...</p>";
                        echo "        <p class='price'>Цена: " .
htmlspecialchars($row['price']) . " руб.</p>";
                        echo "        <a href='" . $base_url .
"booking.php?service_id=" . $row['service_id'] . "' class='btn
btn-secondary'>Записаться</a>";
                        echo "    </div>";
                    }
                } else {
                    echo "    <p>Популярные услуги скоро
появятся.</p>";
                }
            <?>
        </div>
    </div>
</section>

<section class="cta-section">
    <div class="container">
        <h2>Готовы преобразиться?</h2>
        <p>Запишитесь на процедуру уже сегодня и почувствуйте
разницу!</p>
        <a href="<?php echo $base_url; ?>services.php"
class="btn btn-outline-light futuristic-button">Все Услуги</a>
    </div>
</section>
<?php include 'includes/footer.php'; ?>

```

Services.php – динамически формируемый каталог услуг с фильтрацией или группировкой по категориям:

```

<?php
$page_title = "Наши Услуги";
include 'includes/header.php';

```

```

require_once 'includes/db_connect.php';
?>

<div class="container page-container">
    <h1 class="page-title">Наши Услуги</h1>
    <p class="page-subtitle">Откройте для себя полный спектр
процедур для вашей красоты и благополучия.</p>

    <div class="services-list">
        <?php
            $sql = "SELECT service_id, service_name, description,
duration_minutes, price, category FROM services ORDER BY
category, service_name";
            $result = $conn->query($sql);

            if ($result && $result->num_rows > 0) {
                $current_category = null;
                while($row = $result->fetch_assoc()) {
                    if ($current_category !== $row['category']) {
                        if ($current_category !== null) {
                            echo "</div>";
                        }
                        $current_category = $row['category'];
                        echo "    <h2 class='category-title'" .
htmlspecialchars($current_category) . "</h2>";
                        echo "    <div class='category-services-
grid'" .
                            echo "<div class='service-card'" .
                                echo "        <h3>" .
htmlspecialchars($row['service_name']) . "</h3>";
                                echo "        <p class='description'" .
htmlspecialchars($row['description']) . "</p>";
                                echo "        <p><strong>Продолжительность:</strong> " .
htmlspecialchars($row['duration_minutes']) . " мин.</p>";
                                echo "        <p class='price'" .
htmlspecialchars($row['price']) . " руб.</p>";
                                echo "        <a href='" . $base_url .
"booking.php?service_id=" . $row['service_id'] . "' class='btn
btn-primary futuristic-button">Записаться</a>";
                                echo "    </div>";
                            }
                        if ($current_category !== null) {
                            echo "</div>";
                        }
                    } else {
                        echo "    <p>Информация об услугах в данный момент
недоступна. Пожалуйста, зайдите позже.</p>";
                    }
                }
                $conn->close();
            }
        ?>
    </div>
</div>

```

```
<?php include 'includes/footer.php'; ?>
```

Team.php – страница с профилями сотрудников салона, извлекаемыми из базы данных:

Contact.php – отображение контактной информации и формы обратной связи.

Booking.php – основная форма записи, где пользователь выбирает услугу, мастера, дату и время. Требуется аутентификация:

Get_available_slots.php - PHP-скрипт, обрабатывающий AJAX-запросы от booking.php. На основе выбранного мастера, даты и длительности услуги, он запрашивает из базы данных рабочие часы мастера и существующие записи, вычисляет доступные временные слоты и возвращает их в формате JSON:

Get_staff_schedule.php - PHP-скрипт, обрабатывающий AJAX-запросы для отображения полного расписания мастеров в модальном окне:

My_appointments.php – отображает список предстоящих и прошедших записей для аутентифицированного пользователя; предоставляет возможность отмены предстоящих записей (с учетом возможных ограничений по времени):

1.4.3 Руководство пользователя

Чтобы начать работу с программой, необходимо пройти авторизацию в системе (рисунок 12). Для этого в поля «Логин» и «Пароль» введем учетные данные пользователя, после чего нажмем кнопку «Войти».

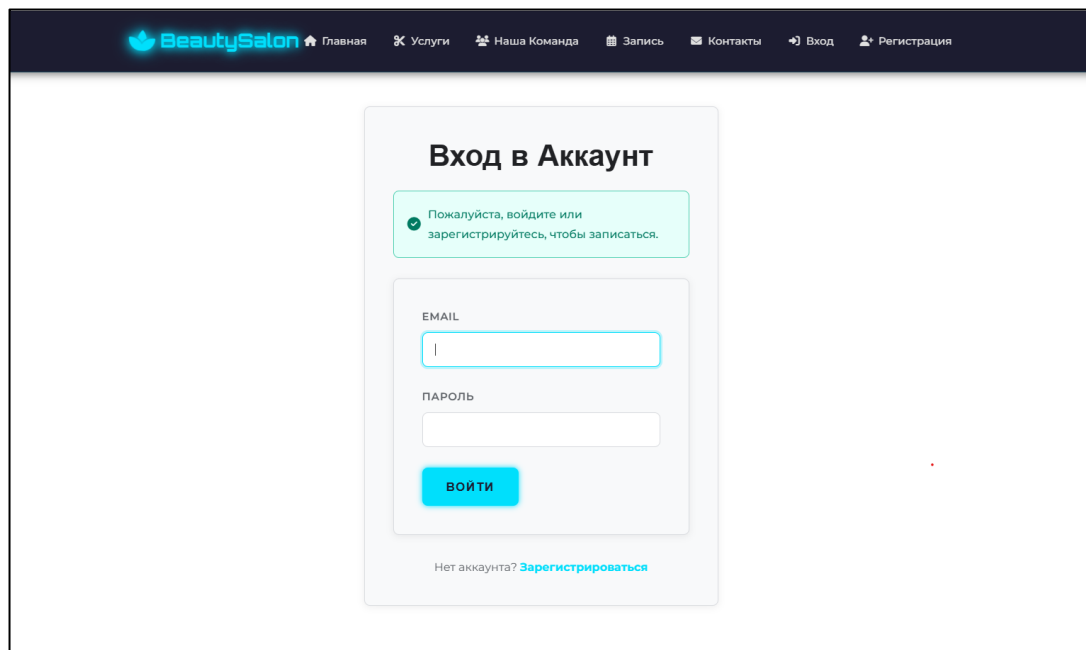


Рисунок 12 – Окно авторизации в системе

При попытке входа с хотя бы одним незаполненным полем на экране появится предупреждающее уведомление (рисунок 13).

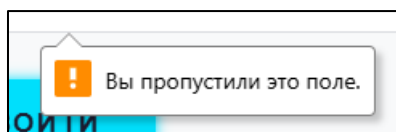


Рисунок 13 – Попытка входа с незаполненными полями

Если ввести неверный логин или пароль, появится уведомление сообщающее об этом (рисунок 14).

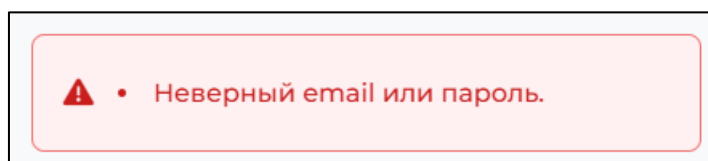


Рисунок 14 – Сообщение об ошибке

Если авторизация успешно пройдена, сайт выведет нас на главное меню программы (рисунок 15). Чтобы выполнить нужное действие, необходимо щелкнуть по той или иной кнопке действия.

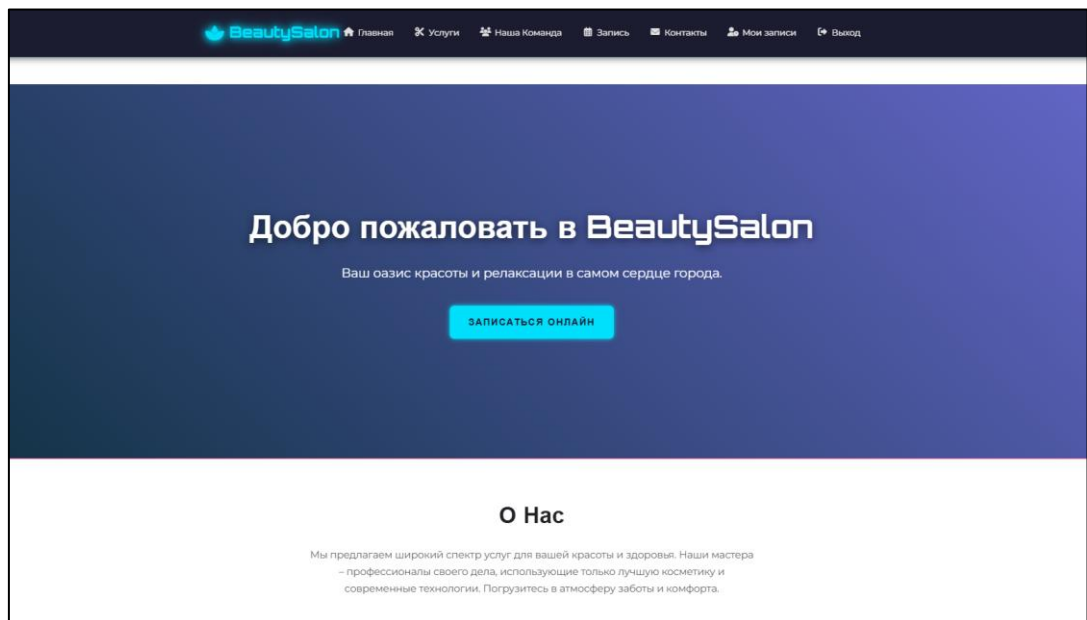


Рисунок 15 – Успешная авторизация

Рассмотрим основные функции и возможности программы. Чтобы внести сведения о новом посетителе заведения, нажмите кнопку «Регистрация». В появившемся окне (рисунок 16) заполните все поля и нажмите кнопку «Зарегистрироваться».

Рисунок 16 – Регистрация нового клиента

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.32.001 ПЗ

Лист

42

Рассмотрим возможности раздела «Услуги» (рисунок 17). В этом разделе отображается информация услугах, предоставляемых в салоне.

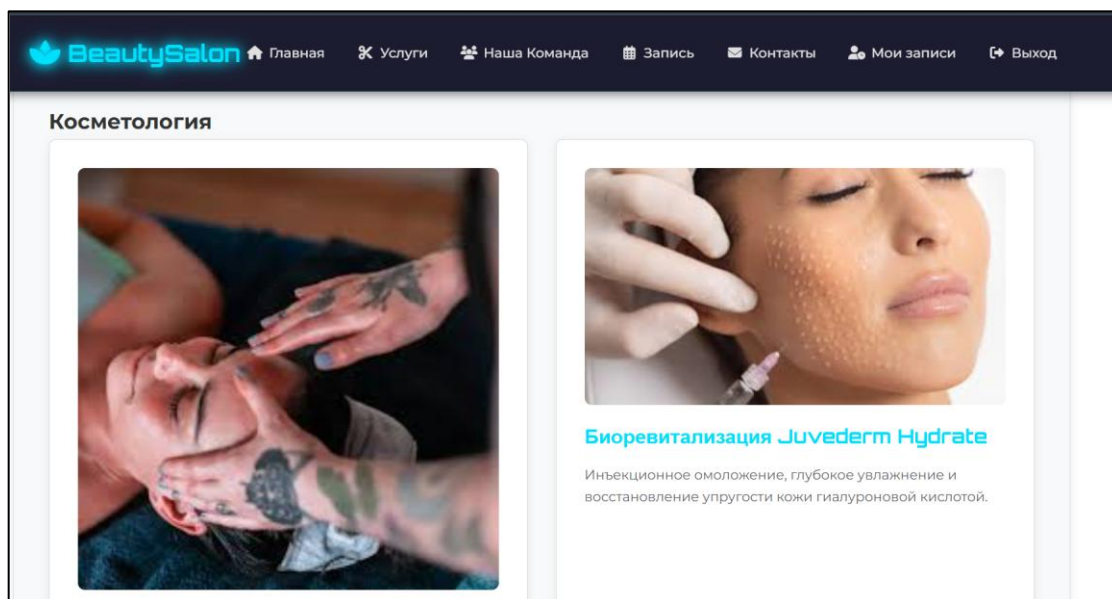


Рисунок 17 – Услуги

Рассмотрим возможности раздела «Наша Команда» (рисунок 18). В нем собрана вся информация о сотрудниках, а также есть возможность записи к понравившемуся мастеру (рисунок 18).

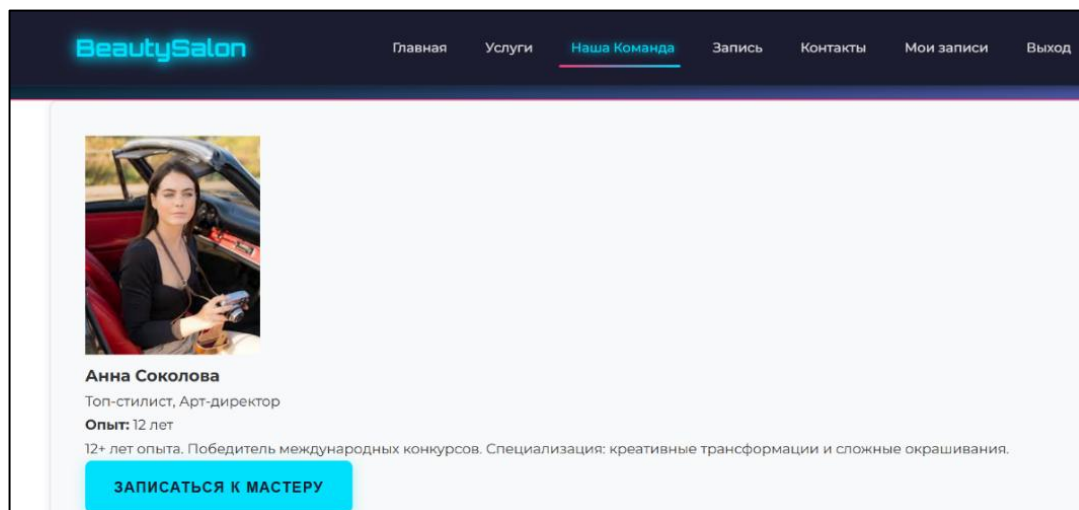


Рисунок 18 – Сотрудники салона

Также мы можем произвести запись на услугу (рисунок 19). После чего в разделе «Мои записи», появятся наши записи (рисунок 20).

BeautySalon

Главная

Услуги

Наша Команда

Запись

Контакты

Мои записи

Выход

Запись на услугу

УСЛУГА:

Авторская Женская Стрижка (90 мин)

МАСТЕР:

Анна Соколова

ДАТА:

04.06.2025

ВРЕМЯ:

11:00

ПРИМЕЧАНИЯ:

ЗАПИСАТЬСЯ

Рисунок 19 – Запись

BeautySalon

Главная

Услуги

Наша Команда

Запись

Контакты

Мои записи

Выход

Мои Записи

Здесь вы можете просмотреть историю ваших записей и управлять предстоящими визитами.

Вечерняя или Свадебная Прическа

Мастер: Ольга Максимова

Дата и время: 04.06.2025 12:00

Статус: ЗАПЛАНИРОВАНО

ОТМЕНИТЬ ЗАПИСЬ

Авторская Женская Стрижка

Мастер: Ольга Максимова

Дата и время: 30.05.2025 11:11

Статус: ЗАПЛАНИРОВАНО

Отмена возможна не позднее чем за 24 часа.

Рисунок 20 – Мои записи

2 Охрана труда

Работа за компьютером требует соблюдения мер безопасности для сохранения здоровья, особенно для разработчиков и других специалистов, проводящих за экраном большую часть дня. Игнорирование этих правил может привести к проблемам со зрением, нарушениям осанки, развитию туннельного синдрома и другим заболеваниям. Рассмотрим ключевые аспекты безопасной организации рабочего процесса.

2.1 Организация рабочего пространства

Правильно оборудованное рабочее место — основа профилактики профессиональных заболеваний.

Требования к помещению:

- расположение: Предпочтительны помещения с окнами на север или северо-восток;

- площадь: Не менее 4,5 м² на одно рабочее место для обеспечения свободного движения и комфорта;

- освещение: Обязательно сочетание естественного и искусственного света.

Для защиты от бликов используйте жалюзи или плотные шторы;

- вентиляция: Регулярное проветривание для снижения концентрации вредных веществ (например, формальдегида) и поддержания уровня кислорода;

- шум: Уровень не должен превышать 50 дБ — это минимизирует стресс и повышает концентрацию;

- микроклимат:

- 1) температура: 20–24°C (оптимально — 22°C);

- 2) влажность: 40–60% (максимум — 75%);

- 3) скорость движения воздуха: до 0,1 м/с.

Электробезопасность:

- заземление рабочего места для нейтрализации электромагнитного излучения от монитора и периферии;

- использование современных устройств с низким уровнем излучения.

2.2 Освещение

Неправильное освещение вызывает усталость глаз, головные боли и снижает эффективность работы.

Рекомендации:

- естественный свет: Рабочий стол располагайте так, чтобы свет из окна падал слева;
- искусственное освещение:
 - 1) общая освещенность: 300–500 лк;
 - 2) равномерное распределение света. Избегайте бликов на экране;
 - 3) используйте лампы теплого спектра (4000–6500 К), близкого к дневному свету;
 - 4) люминесцентные лампы должны быть оснащены рассеивателями.

2.3 Режим работы и отдыха

Регулярные перерывы и физическая активность — залог сохранения здоровья.

Физические требования:

- правильная поза: спина прямая, экран на уровне глаз, расстояние до монитора — 50–70 см;
- каждые 60–90 минут делайте перерыв на 10–15 минут: разминка для глаз, шеи и спины.

Психологические аспекты:

- используйте техники тайм-менеджмента (например, Pomodoro) для снижения стресса;
- избегайте переутомления: нормируйте рабочие часы.

Организационные меры:

- планируйте задачи так, чтобы чередовать работу за компьютером с другими видами активности;

– отдыхайте в выходные дни без использования гаджетов.

Соблюдение норм безопасности при работе за компьютером не только сохраняет здоровье, но и повышает продуктивность. Организуйте эргономичное пространство, следите за освещением, делайте перерывы и используйте специализированные программы. Помните: профилактика всегда проще, чем лечение последствий [11].

					ПГУ 1.090301.32.001 ПЗ	Лист
						47
Изм.	Лист	№ докум.	Подпись	Дата		

Заключение

В ходе выполнения данной выпускной квалификационной работы был разработан прототип веб-приложения "BeautySalon", реализующий основной функционал онлайн-записи в салон красоты. Приложение включает модули регистрации и аутентификации пользователей, каталог услуг, страницу команды, систему бронирования с динамической проверкой доступности времени и личный кабинет клиента.

Проект демонстрирует применение технологий PHP, MySQL, HTML, CSS и JavaScript для создания интерактивного веб-приложения. Реализованный дизайн с белым фоном и неоновыми акцентами в хедере придает приложению современный вид.

Проект имеет значительный потенциал для дальнейшего развития, включая создание полноценной административной панели, интеграцию платежных систем, разработку системы уведомлений, расширение функционала личных кабинетов для мастеров и внедрение системы отзывов. Дальнейшая работа также может быть направлена на повышение безопасности и оптимизацию производительности.

Опыт, полученный в ходе решения поставленных задач и разработке клиент серверного приложения, однозначно поможет в дальнейшем трудоустройстве и развитии себя.

Список использованных источников

1. PHP: Руководство по PHP [Электронный ресурс]. – Режим доступа: <https://www.php.net/manual/ru/> (дата обращений 28.04.2025).
2. PDO: PHP Data Objects // php.net. — URL: <https://www.php.net/manual/ru/book.pdo.php> (дата обращений 07.05.2025).
3. JavaScript // MDN Web Docs. — URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата обращений 07.05.2025).
4. HTML // World Wide Web Consortium (W3C). — URL: <https://www.w3.org/TR/html5/> (дата обращений 12.05.2025).
5. CSS // MDN Web Docs. — URL: <https://developer.mozilla.org/ru/docs/Web/CSS> (дата обращений 16.05.2025).
6. MySQL :: Reference Manual // Oracle. — URL: <https://dev.mysql.com/doc/> (дата обращений 20.05.2025).
7. MySQL [Электронный ресурс]. – Википедия: свободная энциклопедия. – Режим доступа: <https://ru.wikipedia.org/wiki/MySQL> (дата обращений 20.05.2025).
8. Локальная разработка с XAMPP [Электронный ресурс]. – Хабр. – Режим доступа: <https://habr.com/ru/post/335928/> (дата обращений 25.05.2025).
9. Apache Friends. Официальная документация XAMPP [Электронный ресурс]. – Режим доступа: <https://www.apachefriends.org/ru/index.html> (дата обращений 25.05.2025).
10. JSON // ECMA International. — URL: <https://www.ecma-international.org/publications-and-standards/standards/ecma-404/> (дата обращения: 28.05.2025)
11. СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы»

Приложение А

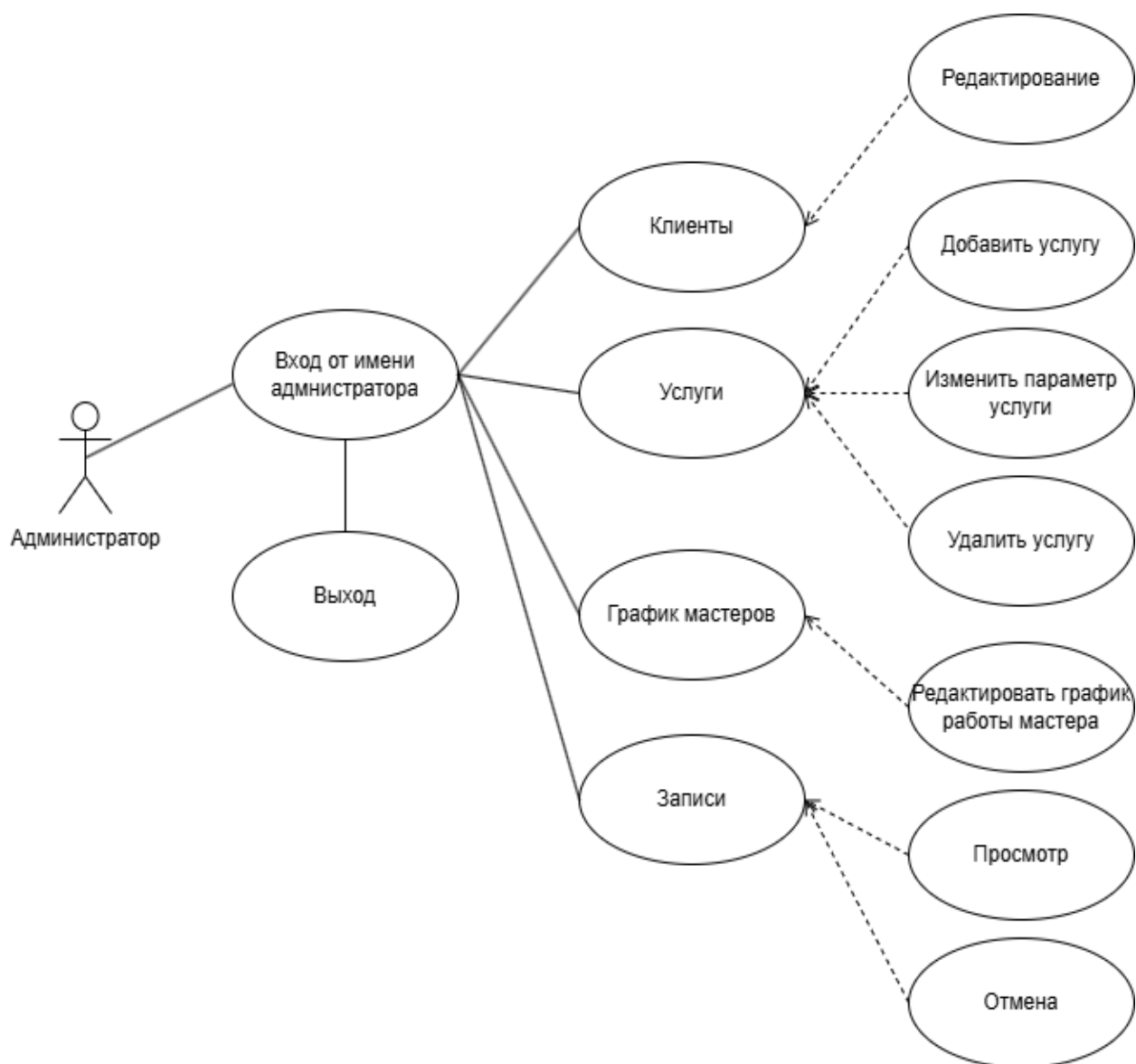
(Обязательное)

Web-приложение для онлайн записи для салонов красоты

UML-диаграммы

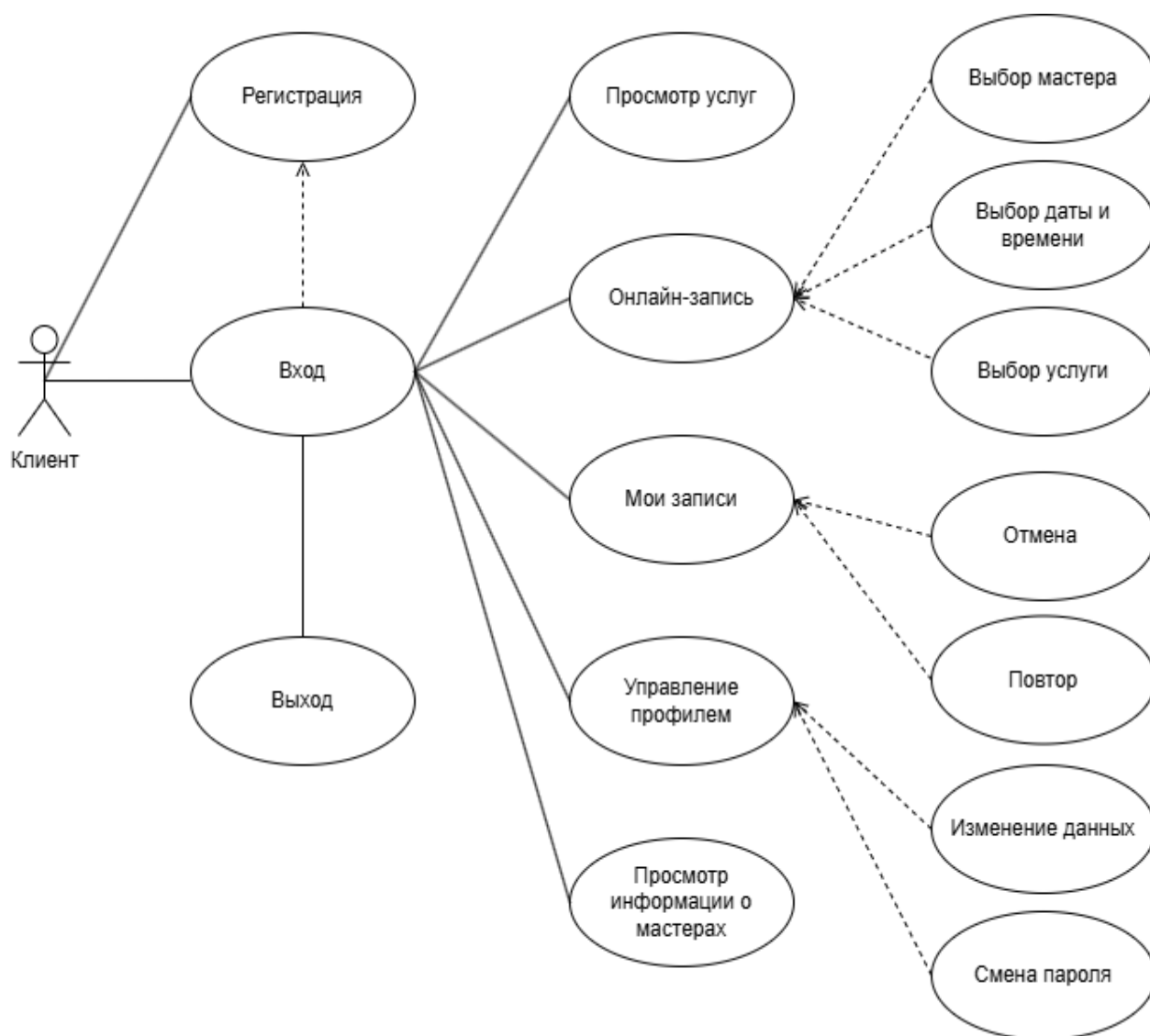
					ПГУ 1.090301.32.001 ПЗ	Лист
						50
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А.1 – Диаграмма прецедентов

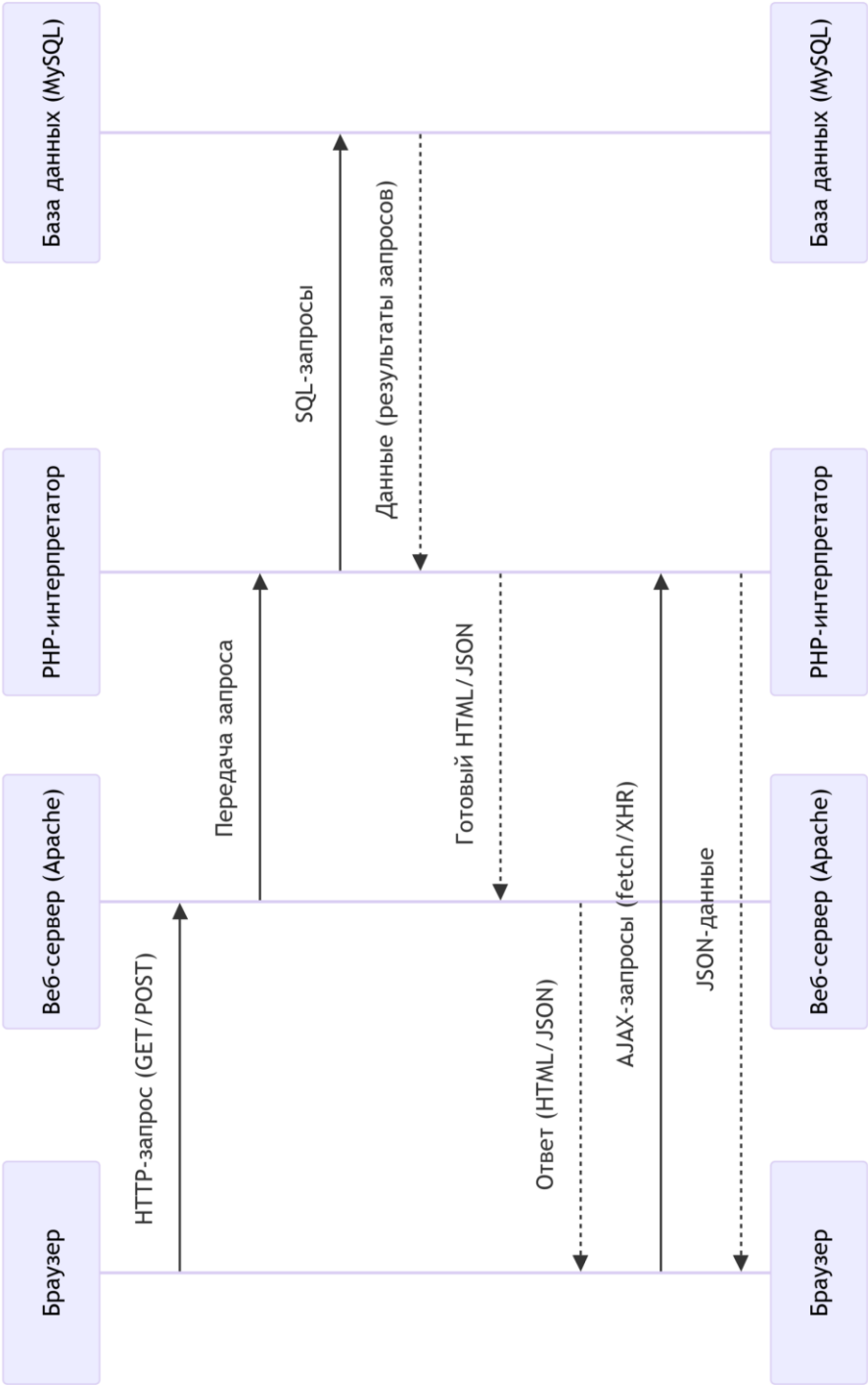


ПГУ 1.090301.32.002 Д13									
Web-Приложение для сайта					Лист 1				
УМ. Диаграмма прецедентов					Лист 1				
Создан					Лист 1				
Исполнен					Лист 1				
Проверен					Лист 1				
Утвержден					Лист 1				
Создан					Лист 1				
Исполнен					Лист 1				
Проверен					Лист 1				
Утвержден					Лист 1				

Изм.	Лист	№ докум.	Подпись	Дата

[illegible]

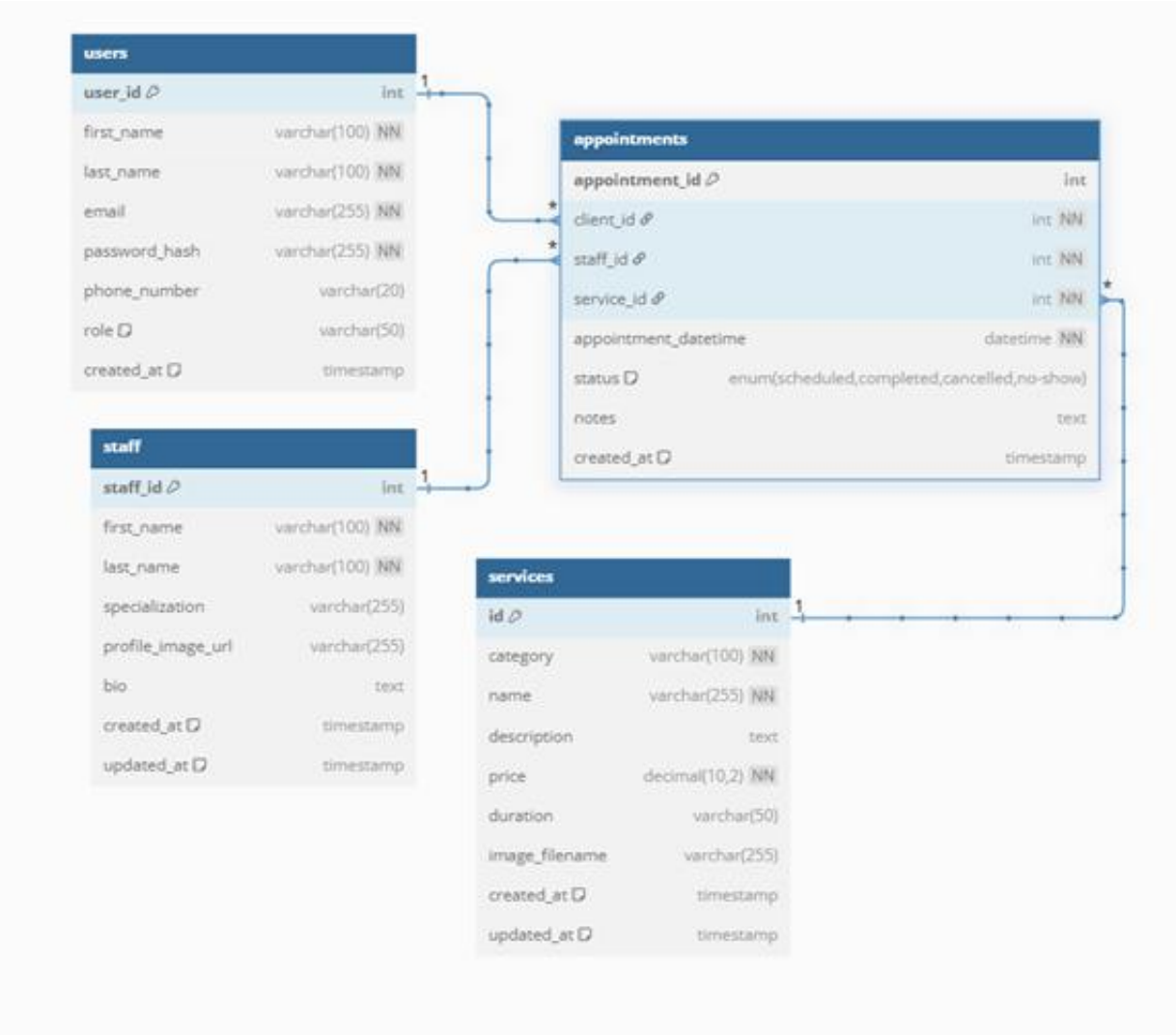
Приложение А.3 – Диаграмма последовательностей



ПГУ 1.090301.32.004 Д14									
Web-Приложение для салона красоты									
UML Диаграмма последовательностей									
Имя	Фамилия	Имя	Фамилия	Имя	Фамилия	Имя	Фамилия	Имя	Фамилия
Михайлов	Александр	Михайлов	Александр	Михайлов	Александр	Михайлов	Александр	Михайлов	Александр
Имя	Фамилия	Имя	Фамилия	Имя	Фамилия	Имя	Фамилия	Имя	Фамилия
Михайлов	Александр	Михайлов	Александр	Михайлов	Александр	Михайлов	Александр	Михайлов	Александр
Имя	Фамилия	Имя	Фамилия	Имя	Фамилия	Имя	Фамилия	Имя	Фамилия
Михайлов	Александр	Михайлов	Александр	Михайлов	Александр	Михайлов	Александр	Михайлов	Александр

Приложение А.4 – Диаграмма классов

ПУУ 1.090301.32.005 Д11									
Имя	Фамилия	Имя	Фамилия	Имя	Фамилия	Имя	Фамилия	Имя	Фамилия
Михайлов	Иван	Петров	Сергей	Сидоров	Александр	Трофимов	Владимир	Харченко	Евгений
Web-Приложение для салона красоты									
ИМ. Диаграмма классов									



Приложение Б

(Обязательное)

Web-приложение для онлайн записи для салона красоты

Листинги программы

					ПГУ 1.090301.32.001 ПЗ	Лист
						55
Изм.	Лист	№ докум.	Подпись	Дата		

Файл main.js:

```
document.addEventListener('DOMContentLoaded', function() {
  try {
    const currentLocation = window.location.href;
    const navLinks = document.querySelectorAll('.main-nav a');
    let activeLinkFound = false;
    navLinks.forEach(link => {
      link.classList.remove('active');
      const linkHref = link.href.replace(/\/$/ ,
'').split(/[#]/)[0];
      const currentHref = currentLocation.replace(/\/$/ ,
'').split(/[#]/)[0];
      if (linkHref === currentHref) {
        link.classList.add('active');
        activeLinkFound = true;
      }
    });
    if (!activeLinkFound) {
      const logoElement = document.querySelector('a.logo');
      const baseProjectUrl = logoElement ?
logoElement.href.replace(/index\.php$/, '').replace(/\/$/ , '') :
(window.location.origin + '/BeautySalon').replace(/\/$/ , '');
      const currentCleanUrl = currentLocation.replace(/\/$/ ,
'').split(/[#]/)[0];
      if (currentCleanUrl === baseProjectUrl ||
currentCleanUrl === baseProjectUrl + '/index.php') {
        navLinks.forEach(link => {
          const linkHrefAttribute =
link.getAttribute('href');
          if (linkHrefAttribute &&
(linkHrefAttribute.includes('index.php') || linkHrefAttribute ===
'./' || link.href.replace(/\/$/ , '') === baseProjectUrl) ) {
            link.classList.add('active');
          }
        });
      }
    }
  } catch (e) {
    console.error("Error setting active navigation link:", e);
  }
  document.querySelectorAll('a[href^="#"]').forEach(anchor => {
    anchor.addEventListener('click', function (e) {
      const hrefAttribute = this.getAttribute('href');
      // Only prevent default for actual in-page anchors, not
      modal triggers handled separately
      if (hrefAttribute && hrefAttribute.length > 1 &&
hrefAttribute !== '#staff-schedule-modal') {
        try {
          const targetElement =
document.querySelector(hrefAttribute);
          if (targetElement) {
```

```

        e.preventDefault();
        targetElement.scrollIntoView({
            behavior: 'smooth'
        });
    }
    } catch (error) {
    }
}
});
});
const sectionsToFade = document.querySelectorAll('.fade-in-section');
if (typeof IntersectionObserver !== 'undefined') {
    const observerOptions = {
        root: null,
        rootMargin: '0px',
        threshold: 0.1
    };
    const intersectionObserver = new
IntersectionObserver((entries, observer) => {
        entries.forEach(entry => {
            if (entry.isIntersecting) {
                entry.target.classList.add('is-visible');
                observer.unobserve(entry.target);
            }
        });
    }, observerOptions);
    sectionsToFade.forEach(section => {
        intersectionObserver.observe(section);
    });
} else {
    sectionsToFade.forEach(section => section.classList.add('is-visible'));
}
const modal = document.getElementById("staff-schedule-modal");
const openModalLinks =
document.querySelectorAll("a[href='#staff-schedule-modal'],
#openScheduleModal, #openScheduleModalDirect"); // More specific
selectors
const closeModalButton = modal ? modal.querySelector(".close-button") : null;
function openScheduleModal(event) {
    if(event) event.preventDefault(); // Prevent default anchor
behavior
    if (modal) {
        modal.style.display = "block";
        modal.classList.add("show");
        document.body.style.overflow = 'hidden';
        fetchStaffSchedule();
    }
}
function closeScheduleModal() {
    if (modal) {

```

```

        modal.classList.remove("show");
        setTimeout(() => {
            modal.style.display = "none";
            document.body.style.overflow = '';
        }, 300);
    }
}
openModalLinks.forEach(link => {
    if(link) {
        link.addEventListener("click", openScheduleModal); //
Pass the event to the handler
    }
});
if(closeModalButton) {
    closeModalButton.addEventListener("click",
closeScheduleModal);
}
window.addEventListener("click", function(event) {
    if (modal && event.target == modal) {
        closeScheduleModal();
    }
});
window.addEventListener("keydown", function(event) {
    if (event.key === "Escape" && modal && modal.style.display
=== "block") {
        closeScheduleModal();
    }
});
function getBaseUrlForJs() {
    const logoLink = document.querySelector('a.logo');
    if (logoLink && logoLink.href) {
        let baseUrl = logoLink.href;
        if (baseUrl.endsWith('index.php')) {
            baseUrl = baseUrl.substring(0,
baseUrl.lastIndexOf('/') + 1);
        } else if (!baseUrl.endsWith('/')) {
            baseUrl = baseUrl.substring(0,
baseUrl.lastIndexOf('/') + 1);
        }
        return baseUrl;
    }
    const pathParts = window.location.pathname.split('/');
    const projectFolderName = 'BeautySalon';
    const projectFolderIndex =
pathParts.indexOf(projectFolderName);
    if (projectFolderIndex !== -1) {
        return window.location.origin + pathParts.slice(0,
projectFolderIndex + 1).join('/') + '/';
    }
    return window.location.origin + '/';
}
function fetchStaffSchedule() {

```

```

    const scheduleContent = document.getElementById('schedule-
content');
    if (!scheduleContent) return;
    scheduleContent.innerHTML = '<p style="text-align:center;
padding: 20px;">Зарпузка расписания <i class="fas fa-spinner fa-
spin"></i></p>';
    const baseUrlForJs = getBaseUrlForJs();
    const scheduleUrl = `${baseUrlForJs}get_staff_schedule.php`;
    fetch(scheduleUrl)
        .then(response => {
            if (!response.ok) {
                throw new Error(`Сетевой ответ был не в порядке:
${response.status} ${response.statusText}`);
            }
            return response.json();
        })
        .then(responseData => {
            if (responseData.error) {
                scheduleContent.innerHTML = `<div class="alert
alert-danger">${responseData.error}</div>`;
                return;
            }
            const data = responseData.data;
            let html =
'<table><thead><tr><th>Мастер</th><th>Специализация</th><th>День
недели</th><th>Начало</th><th>Конец</th></tr></thead><tbody>';
            if (data && data.length > 0) {
                const staffGrouped = data.reduce((acc, item) =>
{
                    const key = item.staff_id;
                    if (!acc[key]) {
                        acc[key] = {
                            name: `${item.first_name}
${item.last_name}`,
                            specialization: item.specialization
                            || '-',
                            schedule: []
                        };
                    }
                    if(item.day_of_week && item.start_time &&
item.end_time) {
                        acc[key].schedule.push({
                            day: translateDay(item.day_of_week),
                            start: item.start_time,
                            end: item.end_time
                        });
                    }
                    return acc;
                }, {});
                let hasVisibleSchedule = false;
                for (const staffKey in staffGrouped) {
                    const staff = staffGrouped[staffKey];
                    if(staff.schedule.length > 0) {

```

```

        hasVisibleSchedule = true;
        staff.schedule.sort((a,b) =>
dayOrder(a.day) - dayOrder(b.day) ||
a.start.localeCompare(b.start));
        staff.schedule.forEach((daySchedule,
index) => {
            html += `<tr>`;
            if (index === 0) {
                html += `<td class="staff-name"
rowspan="${staff.schedule.length}">${staff.name}</td>
<td
rowspan="${staff.schedule.length}">${staff.specialization}</td>`;
            }
            html += `<td>${daySchedule.day}</td>

<td>${daySchedule.start}</td>

<td>${daySchedule.end}</td>
</tr>`;
        });
    } else {
        html += `<tr>
<td class="staff-
name">${staff.name}</td>

<td>${staff.specialization}</td>

<td colspan="3" class="not-
available" style="text-align:center;">Нет доступного расписания</td>
</tr>`;
        hasVisibleSchedule = true;
    }
}
if (!hasVisibleSchedule &&
Object.keys(staffGrouped).length === 0) {
    html += `<tr><td colspan="5" style="text-
align:center;">На данный момент нет информации о мастерах или их
расписании.</td></tr>`;
} else if (!hasVisibleSchedule &&
Object.keys(staffGrouped).length > 0) {
    html += `<tr><td colspan="5" style="text-
align:center;">На данный момент нет доступного расписания
мастеров.</td></tr>`;
}
} else {
    html += `<tr><td colspan="5" style="text-
align:center;">Расписание не найдено или временно
недоступно.</td></tr>`;
}
html += `</tbody></table>`;
scheduleContent.innerHTML = html;
})
.catch(error => {

```

```

        scheduleContent.innerHTML = `

### Файл booking.js:



```

document.addEventListener('DOMContentLoaded', function() {
 const staffSelect = document.getElementById('staff_id');
 const dateInput = document.getElementById('appointment_date');
 const timeInput = document.getElementById('appointment_time');
 const serviceSelect = document.getElementById('service_id');
 const availableSlotsContainer =
document.getElementById('available-slots-container');
 function getBaseUrlForBookingJs() {
 const logoLink = document.querySelector('a.logo');
 if (logoLink && logoLink.href) {
 let baseUrl = logoLink.href;
 if (baseUrl.endsWith('index.php')) {
 baseUrl = baseUrl.substring(0,
baseUrl.lastIndexOf('/') + 1);
 } else if (!baseUrl.endsWith('/')) {
 baseUrl = baseUrl.substring(0,
baseUrl.lastIndexOf('/') + 1);
 }
 return baseUrl.endsWith('/') ? baseUrl : baseUrl + '/';
 }
 const pathParts = window.location.pathname.split('/');
 const projectFolderName = 'BeautySalon'; // Make sure this
matches your project folder

```



|      |      |          |         |      |
|------|------|----------|---------|------|
|      |      |          |         |      |
| Изм. | Лист | № докум. | Подпись | Дата |



ПГУ 1.090301.32.001 ПЗ



Лист



61


```

```

        const projectFolderIndex =
pathParts.indexOf(projectFolderName);
        if (projectFolderIndex !== -1) {
            return window.location.origin + pathParts.slice(0,
projectFolderIndex + 1).join('/') + '/';
        }
        return window.location.origin + '/'; // Fallback
    }
    function fetchAvailableSlots() {
        if (!staffSelect || !dateInput || !serviceSelect ||
!availableSlotsContainer || !timeInput) {
            return;
        }
        const staffId = staffSelect.value;
        const selectedDate = dateInput.value;
        const selectedServiceOption =
serviceSelect.options[serviceSelect.selectedIndex];
        const serviceDuration = selectedServiceOption ?
selectedServiceOption.getAttribute('data-duration') : null;
        availableSlotsContainer.innerHTML = '';
        timeInput.value = '';
        const activeSlotButton = document.querySelector('.slot-
button.active');
        if (activeSlotButton) {
            activeSlotButton.classList.remove('active');
        }
        if (!staffId || !selectedDate || !serviceDuration) {
            availableSlotsContainer.innerHTML = '<p class="text-
muted-color" style="font-size:0.9em; margin-
top:5px;"><small>Выберите мастера, дату и услугу для просмотра
доступных временных слотов.</small></p>';
            return;
        }
        availableSlotsContainer.innerHTML = '<p style="padding:10px
0; color: var(--text-color-muted);"><i class="fas fa-spinner fa-
spin"></i> Загрузка доступных слотов...</p>';
        const baseUrlForJs = getBaseUrlForBookingJs();
        const slotsUrl =
`${baseUrlForJs}get_available_slots.php?staff_id=${staffId}&date=${s
electedDate}&duration=${serviceDuration}`;
        fetch(slotsUrl)
            .then(response => {
                if (!response.ok) {
                    throw new Error(`Сетевой ответ был не в порядке:
${response.status} ${response.statusText}`);
                }
                return response.json();
            })
            .then(data => {
                availableSlotsContainer.innerHTML = '';
                if (data.error) {

```

Изм.	Лист	№ докум.	Подпись	Дата

```

        availableSlotsContainer.innerHTML = `<p
class="alert alert-danger" style="font-size:0.9em; padding:
0.8rem;">${data.error}</p>`;
    } else if (data.slots && data.slots.length > 0) {
        let slotsHtml = '<p style="margin-bottom:
0.7rem; font-weight: 600; color: var(--text-color-main); font-
size:0.95em;">Доступные слоты:</p><div class="slots-grid">';
        data.slots.forEach(slot => {
            slotsHtml += `<button type="button"
class="btn btn-sm slot-button" data-
time="${slot}">${slot}</button>`;
        });
        slotsHtml += '</div>';
        availableSlotsContainer.innerHTML = slotsHtml;
        document.querySelectorAll('.slot-
button').forEach(button => {
            button.addEventListener('click', function()
{
                timeInput.value =
this.getAttribute('data-time');
                document.querySelectorAll('.slot-
button.active').forEach(activeBtn =>
activeBtn.classList.remove('active'));
                this.classList.add('active');
            });
        });
    } else {
        availableSlotsContainer.innerHTML = '<p
style="color: var(--text-color-muted); font-size:0.9em;
padding:10px; background-color: var(--surface-bg-color); border-
radius:var(--border-radius); border:1px solid var(--border-color-
light);">На выбранную дату у данного мастера нет доступных слотов
для выбранной услуги. Пожалуйста, выберите другую дату или
мастера.</p>';
    }
})
.catch(error => {
    availableSlotsContainer.innerHTML = `<p class="alert
alert-danger" style="font-size:0.9em; padding: 0.8rem;">Ошибка при
загрузке слотов: ${error.message}. Проверьте консоль
разработчика.</p>`;
    console.error('Ошибка при загрузке доступных
слотов:', error);
});
}
if (staffSelect) staffSelect.addEventListener('change',
fetchAvailableSlots);
if (dateInput) dateInput.addEventListener('change',
fetchAvailableSlots);
if (serviceSelect) serviceSelect.addEventListener('change',
fetchAvailableSlots);
if (dateInput) {
    const today = new Date();

```

```

        const yyyy = today.getFullYear();
        const mm = String(today.getMonth() + 1).padStart(2, '0');
        const dd = String(today.getDate()).padStart(2, '0');
        dateInput.setAttribute('min', `${yyyy}-${mm}-${dd}`);
    }
    if (staffSelect && staffSelect.value && dateInput &&
dateInput.value && serviceSelect && serviceSelect.value &&
serviceSelect.options[serviceSelect.selectedIndex]?.getAttribute('da
ta-duration')) {
        fetchAvailableSlots();
    }
});

```

Файл style.css:

```

document.addEventListener('DOMContentLoaded', function() {
    const staffSelect = document.getElementById('staff_id');
    const dateInput = document.getElementById('appointment_date');
    const timeInput = document.getElementById('appointment_time');
    const serviceSelect = document.getElementById('service_id');
    const availableSlotsContainer =
document.getElementById('available-slots-container');

    function getBaseUrlForBookingJs() {
        const logoLink = document.querySelector('a.logo');
        if (logoLink && logoLink.href) {
            let baseUrl = logoLink.href;
            if (baseUrl.endsWith('index.php')) {
                baseUrl = baseUrl.substring(0,
baseUrl.lastIndexOf('/') + 1);
            } else if (!baseUrl.endsWith('/')) {
                baseUrl = baseUrl.substring(0,
baseUrl.lastIndexOf('/') + 1);
            }
            return baseUrl.endsWith('/') ? baseUrl : baseUrl + '/';
        }
    }

    const pathParts = window.location.pathname.split('/');
    const projectFolderName = 'BeautySalon'; // Make sure this
matches your project folder
    const projectFolderIndex =
pathParts.indexOf(projectFolderName);
    if (projectFolderIndex !== -1) {
        return window.location.origin + pathParts.slice(0,
projectFolderIndex + 1).join('/') + '/';
    }
    return window.location.origin + '/'; // Fallback
}

    function fetchAvailableSlots() {
        if (!staffSelect || !dateInput || !serviceSelect ||
!availableSlotsContainer || !timeInput) {

```

```

        return;
    }

    const staffId = staffSelect.value;
    const selectedDate = dateInput.value;
    const selectedServiceOption =
serviceSelect.options[serviceSelect.selectedIndex];
    const serviceDuration = selectedServiceOption ?
selectedServiceOption.getAttribute('data-duration') : null;

    availableSlotsContainer.innerHTML = '';
    timeInput.value = '';
    const activeSlotButton = document.querySelector('.slot-
button.active');
    if (activeSlotButton) {
        activeSlotButton.classList.remove('active');
    }

    if (!staffId || !selectedDate || !serviceDuration) {
        availableSlotsContainer.innerHTML = '<p class="text-
muted-color" style="font-size:0.9em; margin-
top:5px;"><small>Выберите мастера, дату и услугу для просмотра
доступных временных слотов.</small></p>';
        return;
    }

    availableSlotsContainer.innerHTML = '<p style="padding:10px
0; color: var(--text-color-muted);"><i class="fas fa-spinner fa-
spin"></i> Загрузка доступных слотов...</p>';

    const baseUrlForJs = getBaseUrlForBookingJs();
    const slotsUrl =
`${baseUrlForJs}get_available_slots.php?staff_id=${staffId}&date=${s
electedDate}&duration=${serviceDuration}`;

    fetch(slotsUrl)
        .then(response => {
            if (!response.ok) {
                throw new Error(`Сетевой ответ был не в порядке:
${response.status} ${response.statusText}`);
            }
            return response.json();
        })
        .then(data => {
            availableSlotsContainer.innerHTML = '';
            if (data.error) {
                availableSlotsContainer.innerHTML = `<p
class="alert alert-danger" style="font-size:0.9em; padding:
0.8rem;">${data.error}</p>`;
            } else if (data.slots && data.slots.length > 0) {
                let slotsHtml = '<p style="margin-bottom:
0.7rem; font-weight: 600; color: var(--text-color-main); font-
size:0.95em;">Доступные слоты:</p><div class="slots-grid">';

```

```

        data.slots.forEach(slot => {
            slotsHtml += `<button type="button"
class="btn btn-sm slot-button" data-
time="\${slot}">\${slot}</button>`;
        });
        slotsHtml += '</div>';
        availableSlotsContainer.innerHTML = slotsHtml;

        document.querySelectorAll('.slot-
button').forEach(button => {
            button.addEventListener('click', function()
            {
                timeInput.value =
this.getAttribute('data-time');
                document.querySelectorAll('.slot-
button.active').forEach(activeBtn =>
activeBtn.classList.remove('active'));
                this.classList.add('active');
            });
        });

        } else {
            availableSlotsContainer.innerHTML = '<p
style="color: var(--text-color-muted); font-size:0.9em;
padding:10px; background-color: var(--surface-bg-color); border-
radius:var(--border-radius); border:1px solid var(--border-color-
light);">На выбранную дату у данного мастера нет доступных слотов
для выбранной услуги. Пожалуйста, выберите другую дату или
мастера.</p>';
        }
    })
    .catch(error => {
        availableSlotsContainer.innerHTML = `<p class="alert
alert-danger" style="font-size:0.9em; padding: 0.8rem;">Ошибка при
загрузке слотов: \${error.message}. Проверьте консоль
разработчика.</p>`;
        console.error('Ошибка при загрузке доступных
слотов:', error);
    });
}

if (staffSelect) staffSelect.addEventListener('change',
fetchAvailableSlots);
if (dateInput) dateInput.addEventListener('change',
fetchAvailableSlots);
if (serviceSelect) serviceSelect.addEventListener('change',
fetchAvailableSlots);

if (dateInput) {
    const today = new Date();
    const yyyy = today.getFullYear();
    const mm = String(today.getMonth() + 1).padStart(2, '0');
    const dd = String(today.getDate()).padStart(2, '0');

```

```

        dateInput.setAttribute('min', `${yyyy}-${mm}-${dd}`);
    }

    if (staffSelect && staffSelect.value && dateInput &&
dateInput.value && serviceSelect && serviceSelect.value &&
serviceSelect.options[serviceSelect.selectedIndex]?.getAttribute('da
ta-duration')) {
        fetchAvailableSlots();
    }
});

```

Файл animations.css:

```

@keyframes pulseIcon {
    0% { transform: scale(1); }
    50% { transform: scale(1.15) rotate(5deg); }
    100% { transform: scale(1) rotate(0deg); }
}

.fade-in-section {
    opacity: 0;
    transform: translateY(30px);
    transition: opacity 0.7s ease-out, transform 0.7s ease-out;
}

.fade-in-section.is-visible {
    opacity: 1;
    transform: translateY(0);
}

.hero-content h1 {
    animation: fadeInDown 1s 0.2s ease-out forwards;
    opacity: 0;
}

.hero-content p {
    animation: fadeInUp 1s 0.5s ease-out forwards;
    opacity: 0;
}

.hero-content .btn {
    animation: fadeInUp 1s 0.8s ease-out forwards;
    opacity: 0;
}

@keyframes fadeInDown {
    from {
        opacity: 0;
        transform: translateY(-40px);
    }
    to {
        opacity: 1;
        transform: translateY(0);
    }
}

```

```

@keyframes fadeInUp {
  from {
    opacity: 0;
    transform: translateY(40px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}

@keyframes slideInModal {
  from {
    opacity: 0;
    transform: translateY(-60px) scale(0.95);
  }
  to {
    opacity: 1;
    transform: translateY(0) scale(1);
  }
}

.modal.show .modal-content {
  animation: slideInModal 0.5s ease-out forwards;
}

.neon-glow-text {
  animation: neonGlow 1.5s infinite alternate ease-in-out;
}

@keyframes neonGlow {
  from {
    text-shadow: 0 0 5px var(--primary-color),
      0 0 10px var(--primary-color),
      0 0 15px var(--primary-color),
      0 0 20px var(--secondary-color),
      0 0 25px var(--secondary-color),
      0 0 30px var(--secondary-color),
      0 0 35px var(--secondary-color);
  }
  to {
    text-shadow: 0 0 3px var(--primary-color),
      0 0 5px var(--primary-color),
      0 0 8px var(--primary-color),
      0 0 10px var(--secondary-color),
      0 0 12px var(--secondary-color),
      0 0 15px var(--secondary-color),
      0 0 18px var(--secondary-color);
  }
}

.fa-spinner.fa-spin {
  animation: fa-spin 1s infinite linear;
}

```

```

}

@keyframes fa-spin {
  0% {
    transform: rotate(0deg);
  }
  100% {
    transform: rotate(360deg);
  }
}

.btn::before {
  content: '';
  position: absolute;
  top: 0;
  left: -100%;
  width: 50%;
  height: 100%;
  background: linear-gradient(90deg, transparent,
rgba(255,255,255,0.2), transparent);
  transition: left 0.6s ease;
  transform: skewX(-20deg);
}
.btn:hover::before {
  left: 120%;
}

```

Файл booking.php:

```

<?php
// booking.php

ini_set('display_errors', 1);
error_reporting(E_ALL);

session_start();
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php?message=" . urlencode("Пожалуйста,
войдите, чтобы записаться."));
    exit;
}

$page_title = "Запись онлайн";
include 'includes/header.php';
require_once 'includes/db_connect.php';

$error = '';
$success = '';

// Получаем список услуг

```

```

$services = $conn->query("SELECT id, name, duration FROM services
ORDER BY name");

// Получаем список мастеров из таблицы staff
$staffList = $conn->query("
    SELECT staff_id, CONCAT(first_name, ' ', last_name) AS
staff_name
    FROM staff
    ORDER BY last_name, first_name
");

// Обработка отправки формы
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $client_id = $_SESSION['user_id'];
    $service_id = intval($_POST['service_id']);
    $staff_id = intval($_POST['staff_id']);
    $date = $_POST['appointment_date'] ?? '';
    $time = $_POST['appointment_time'] ?? '';
    $notes = trim($_POST['notes'] ?? '');

    if (!$service_id || !$staff_id || !$date || !$time) {
        $error = "Заполните все обязательные поля.";
    } else {
        $appointment_datetime = $date . ' ' . $time . ':00';

        $stmt = $conn->prepare("
            INSERT INTO appointments
                (client_id, staff_id, service_id,
appointment_datetime, notes)
            VALUES (?, ?, ?, ?, ?)
        ");
        $stmt->bind_param(
            "iiiss",
            $client_id,
            $staff_id,
            $service_id,
            $appointment_datetime,
            $notes
        );

        if ($stmt->execute()) {
            $success = "Вы успешно записаны на " .
htmlspecialchars($date) . " в " . htmlspecialchars($time) . ".";
        } else {
            $error = "Ошибка при записи: " . $stmt->error;
        }
        $stmt->close();
    }
}
?>

<div class="container page-container">
    <h1 class="page-title">Запись на услугу</h1>

```

```

<?php if ($error): ?>
    <div class="alert alert-danger"><?php echo $error; ?></div>
<?php endif; ?>

<?php if ($success): ?>
    <div class="alert alert-success"><?php echo $success;
?></div>
<?php endif; ?>

<form method="post" class="booking-form">
    <div class="form-group">
        <label for="service_id">Услуга:</label>
        <select id="service_id" name="service_id" required>
            <option value="">-- выберите услугу --</option>
            <?php while ($s = $services->fetch_assoc()): ?>
                <option value="<?php echo $s['id']; ?>"
                    <?php echo htmlspecialchars("{ $s['name'] }
({ $s['duration'] })); ?>
                </option>
            <?php endwhile; ?>
        </select>
    </div>

    <div class="form-group">
        <label for="staff_id">Мастер:</label>
        <select id="staff_id" name="staff_id" required>
            <option value="">-- выберите мастера --</option>
            <?php while ($m = $staffList->fetch_assoc()): ?>
                <option value="<?php echo $m['staff_id']; ?>"
                    <?php echo htmlspecialchars($m['staff_name']); ?>
                </option>
            <?php endwhile; ?>
        </select>
    </div>

    <div class="form-group">
        <label for="appointment_date">Дата:</label>
        <input
            type="date"
            id="appointment_date"
            name="appointment_date"
            required
            min="<?php echo date('Y-m-d'); ?>"
        >
    </div>

    <div class="form-group">
        <label for="appointment_time">Время:</label>
        <input
            type="time"
            id="appointment_time"
            name="appointment_time" required>
    </div>

```

```

        <div class="form-group">
            <label for="notes">Примечания:</label>
            <textarea id="notes" name="notes"></textarea>
        </div>

        <button type="submit" class="btn btn-primary futuristic-
button">Записаться</button>
    </form>
</div>

<?php
$conn->close();
include 'includes/footer.php';
?>

```

Файл contact.php:

```

<?php
$page_title = "Контакты";
// Убедимся, что сессия стартует до любого вывода HTML, если она нужна
(например, для $base_url из сессии)
if (session_status() == PHP_SESSION_NONE) {
    session_start();
}
include 'includes/header.php'; // $base_url должен быть определен
здесь или ранее

$message_sent = false;
$error_message = '';
$form_data = ['name' => '', 'email' => '', 'subject' => '', 'message'
=> ''];

// Если $base_url не определен в header.php, определите его здесь
if (!isset($base_url)) {
    $script_name = $_SERVER['SCRIPT_NAME'];
    $project_folder_name = 'BeautySalon';
    if (strpos($script_name, '/' . $project_folder_name . '/') !==
false) {
        $base_url = '/' . $project_folder_name . '/';
    } else {
        $base_url = '/';
    }
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $form_data['name'] = trim($_POST['name']);
    $form_data['email'] = trim($_POST['email']);
    $form_data['subject'] = trim($_POST['subject']);
    $form_data['message'] = trim($_POST['message']);
}

```

```

        if (empty($form_data['name']) || empty($form_data['email']) ||
empty($form_data['subject']) || empty($form_data['message'])) {
            $error_message = "Пожалуйста, заполните все поля.";
        } elseif (!filter_var($form_data['email'], FILTER_VALIDATE_EMAIL))
{
            $error_message = "Пожалуйста, введите корректный email.";
        } else {
            // Логика отправки письма (закомментирована для примера)
            /*
            $to = "your_salon_email@example.com"; // ЗАМЕНИТЕ НА ВАШ EMAIL
            $email_subject = "Новое сообщение с сайта BeautySalon: " .
htmlspecialchars($form_data['subject']);
            $email_body = "Имя: " . htmlspecialchars($form_data['name']) .
"\n";
            $email_body .= "Email: " .
htmlspecialchars($form_data['email']) . "\n";
            $email_body .= "Сообщение:\n" .
htmlspecialchars($form_data['message']) . "\n";
            $headers = "From: no-reply@beautysalon.example.com\r\n"; //
Или используйте email отправителя
            $headers .= "Reply-To: " .
htmlspecialchars($form_data['email']) . "\r\n";
            $headers .= "Content-Type: text/plain; charset=UTF-8\r\n";

            if (mail($to, $email_subject, $email_body, $headers)) {
                $message_sent = true;
                // Очистка данных формы после успешной отправки
                $form_data = ['name' => '', 'email' => '', 'subject' =>
'', 'message' => ''];
            } else {
                $error_message = "Ошибка при отправке сообщения.
Пожалуйста, попробуйте позже.";
            }
            */
            // Для демонстрации без реальной отправки письма:
            $message_sent = true;
            $form_data = ['name' => '', 'email' => '', 'subject' => '',
'message' => '']; // Очищаем форму
        }
    }
}
?>

```

```

<div class="container page-container">
    <h1 class="page-title">Свяжитесь с Нами</h1>
    <p class="page-subtitle">Мы всегда рады ответить на ваши вопросы и
помочь с выбором услуг.</p>

    <div class="contact-grid">
        <div class="contact-form-container">
            <h2>Отправьте нам сообщение</h2>
            <?php if ($message_sent): ?>

```

```

        <div class="alert alert-success">Спасибо! Ваше
сообщение успешно отправлено. Мы свяжемся с вами в ближайшее
время.</div>
        <?php endif; ?>
        <?php if (!empty($error_message)): ?>
            <div class="alert alert-danger"><?php echo
htmlspecialchars($error_message); ?></div>
            <?php endif; ?>

        <form action="<?php echo htmlspecialchars($base_url .
'contact.php'); ?>" method="POST" class="contact-form">
            <div class="form-group">
                <label for="name">Ваше Имя</label>
                <input type="text" id="name" name="name"
value="<?php echo htmlspecialchars($form_data['name']); ?>" required>
            </div>
            <div class="form-group">
                <label for="email">Ваш Email</label>
                <input type="email" id="email" name="email"
value="<?php echo htmlspecialchars($form_data['email']); ?>" required>
            </div>
            <div class="form-group">
                <label for="subject">Тема</label>
                <input type="text" id="subject" name="subject"
value="<?php echo htmlspecialchars($form_data['subject']); ?>"
required>
            </div>
            <div class="form-group">
                <label for="message">Сообщение</label>
                <textarea id="message" name="message" rows="5"
required><?php echo htmlspecialchars($form_data['message']);
?></textarea>
            </div>
            <button type="submit" class="btn btn-primary
futuristic-button">Отправить</button>
        </form>
    </div>

    <div class="contact-details-container">
        <h2>Наши Контакты</h2>
        <div class="contact-info-group">
            <p><i class="fas fa-map-marker-alt fa-fw"></i>
<strong>Адрес:</strong><br><span>г. Москва, ул. Примерная, д. 1, офис
101</span></p>
            <p><i class="fas fa-phone fa-fw"></i>
<strong>Телефон:</strong><br><span>+7 (495) 123-45-67</span></p>
            <p><i class="fas fa-envelope fa-fw"></i>
<strong>Email:</strong><br><span>info@beautysalon.example.com</span></p>
            <p><i class="far fa-clock fa-fw"></i> <strong>Часы
работы:</strong><br><span>Пн-Сб: 9:00 - 21:00, Вс: 10:00 -
19:00</span></p>
        </div>
    </div>

```

```

        <div class="map-placeholder">
            <!--
                ВАЖНО: Замените URL-адрес в атрибуте src на код для
                встраивания вашей реальной карты Google.
                Вы можете сгенерировать его на Google Картах:
                1. Найдите нужное место.
                2. Нажмите "Поделиться".
                3. Выберите "Встраивание карт".
                4. Скопируйте HTML-код и извлеките из него URL для
                атрибута src.

                Пример ниже - это просто демонстрационный URL для
                Москвы.
            -->
            <iframe

src="https://maps.google.com/maps?q=%D0%9C%D0%BE%D1%81%D0%BA%D0%B2%D0%
B0&t=&z=13&ie=UTF8&iwloc=&output=embed"
                width="100%"
                height="300"
                style="border:0; border-radius: 8px;"
                allowfullscreen=""
                loading="lazy"
                referrerpolicy="no-referrer-when-downgrade"
                title="Карта расположения салона">
            </iframe>
        </div>
    </div>
</div>
</div>
</div>

<?php include 'includes/footer.php'; ?>

```

Файл get_available_slots.php:

```

<?php
header('Content-Type: application/json; charset=UTF-8');
require_once 'includes/db_connect.php';

$response = ['error' => null, 'slots' => []];

if (!$conn || (is_object($conn) && property_exists($conn,
'connect_error') && $conn->connect_error)) {
    $response['error'] = 'Ошибка подключения к БД: ' . ($conn &&
is_object($conn) && property_exists($conn, 'connect_error') ?
htmlspecialchars($conn->connect_error) : "Объект \$conn не создан");
    echo json_encode($response);
    exit;
}

$staff_id = filter_input(INPUT_GET, 'staff_id', FILTER_VALIDATE_INT);
$date_str = filter_input(INPUT_GET, 'date', FILTER_SANITIZE_STRING);

```

```

$duration_minutes_input = filter_input(INPUT_GET, 'duration',
FILTER_VALIDATE_INT);

if (!$staff_id || !$date_str || $duration_minutes_input === false ||
$duration_minutes_input <= 0) {
    $response['error'] = 'Отсутствуют или некорректны необходимые
параметры: ID мастера, дата или длительность услуги.';
    echo json_encode($response);
    exit;
}
$duration_minutes = (int)$duration_minutes_input;

try {
    $selected_date_obj = new DateTime($date_str);
    $now = new DateTime(); // Current date and time
    // Prevent booking on past dates. Allow booking for today.
    if ($selected_date_obj < $now->setTime(0,0,0) &&
$selected_date_obj->format('Y-m-d') !== $now->format('Y-m-d')) {
        $response['error'] = 'Нельзя выбрать прошедшую дату.';
        echo json_encode($response);
        exit;
    }
} catch (Exception $e) {
    $response['error'] = 'Неверный формат даты: ' .
htmlspecialchars($date_str);
    echo json_encode($response);
    exit;
}

$day_of_week = $selected_date_obj->format('l'); // e.g., 'Monday',
'Tuesday'

// 1. Get staff working hours for the selected day
$stmt_avail = $conn->prepare("SELECT start_time, end_time FROM
availability
                                WHERE staff_id = ? AND day_of_week = ?
AND is_available = TRUE");
if(!$stmt_avail){
    $response['error'] = 'Ошибка подготовки запроса для получения
доступности мастера: ' . htmlspecialchars($conn->error);
    echo json_encode($response); exit;
}
$stmt_avail->bind_param("is", $staff_id, $day_of_week);
$stmt_avail->execute();
$result_avail = $stmt_avail->get_result();

if ($result_avail->num_rows == 0) {
    $response['error'] = 'Мастер не работает в выбранный день (' .
htmlspecialchars($day_of_week) . ').';
    $stmt_avail->close();
    if (isset($conn) && is_object($conn) && method_exists($conn,
'close')) { $conn->close(); }
    echo json_encode($response);
}

```

```

        exit;
    }

    $work_hours = $result_avail->fetch_assoc();
    $stmt_avail->close();

    $work_start_time_str = $selected_date_obj->format('Y-m-d') . ' ' .
    $work_hours['start_time'];
    $work_end_time_str = $selected_date_obj->format('Y-m-d') . ' ' .
    $work_hours['end_time'];

    try {
        $current_slot_start = new DateTime($work_start_time_str);
        $work_end_dt = new DateTime($work_end_time_str);
    } catch (Exception $e) {
        $response['error'] = 'Ошибка при обработке рабочего времени
мастера: ' . $e->getMessage();
        echo json_encode($response); exit;
    }

    // 2. Get existing appointments for the staff on the selected day
    $existing_appointments = [];
    $stmt_app = $conn->prepare(
        "SELECT TIME_FORMAT(appointment_datetime, '%H:%i:%s') as
appointment_start_time,
        (SELECT duration_minutes FROM services s WHERE
s.service_id = a.service_id) as booked_duration
        FROM appointments a
        WHERE staff_id = ? AND DATE(appointment_datetime) = ? AND status
= 'scheduled'"
    );

    if(!$stmt_app){
        $response['error'] = 'Ошибка подготовки запроса для получения
существующих записей: ' . htmlspecialchars($conn->error);
        echo json_encode($response); exit;
    }
    $selected_date_sql = $selected_date_obj->format('Y-m-d');
    $stmt_app->bind_param("is", $staff_id, $selected_date_sql);
    $stmt_app->execute();
    $result_app = $stmt_app->get_result();

    while ($row_app = $result_app->fetch_assoc()) {
        try {
            $app_start_str = $selected_date_sql . ' ' .
$хrow_app['appointment_start_time'];
            $app_start = new DateTime($app_start_str);
            $app_end = clone $app_start;
            $app_end->add(new DateInterval('PT' .
(int)$row_app['booked_duration'] . 'M'));
            $existing_appointments[] = ['start' => $app_start, 'end' =>
$app_end];
        }
    }
}

```

```

    } catch (Exception $e) { /* skip invalid appointment data if any
*/ }
}
$stmt_app->close();

// 3. Generate potential slots and check against existing appointments
$slot_interval = 15; // Check every 15 minutes, can be adjusted (e.g.,
30)
$now_for_slots = new DateTime(); // Current time for comparison

while (true) {
    $potential_slot_end = clone $current_slot_start;
    $potential_slot_end->add(new DateInterval('PT' . $duration_minutes
. 'M'));

    // If the end of the potential slot is beyond the work end time,
stop
    if ($potential_slot_end > $work_end_dt) {
        break;
    }

    // If the selected date is today, ensure the slot is not in the
past
    if ($selected_date_obj->format('Y-m-d') === $now_for_slots-
>format('Y-m-d') && $current_slot_start < $now_for_slots) {
        $current_slot_start->add(new DateInterval('PT' .
$slot_interval . 'M'));
        continue; // Skip past slots for today
    }

    $is_slot_available = true;
    foreach ($existing_appointments as $booked_app) {
        // Check for overlap: (SlotStart < AppEnd) and (SlotEnd >
AppStart)
        if (($current_slot_start < $booked_app['end']) &&
($potential_slot_end > $booked_app['start'])) {
            $is_slot_available = false;
            break;
        }
    }

    if ($is_slot_available) {
        $response['slots'][] = $current_slot_start->format('H:i');
    }

    // Move to next potential slot start
    $current_slot_start->add(new DateInterval('PT' . $slot_interval .
'M'));
    // If next slot start is at or after work end, break
    if ($current_slot_start >= $work_end_dt) break;
}

```

```

if (empty($response['slots']) && $response['error'] === null) { //
Only set this error if no other error occurred
    $response['error'] = 'На выбранную дату нет свободных слотов для
услуги такой продолжительности у данного мастера.';
}

if (isset($conn) && is_object($conn) && method_exists($conn, 'close'))
{ $conn->close(); }
echo json_encode($response);
?>

```

Файл get_staff_schedule.php:

```

<?php
header('Content-Type: application/json; charset=UTF-8');
require_once 'includes/db_connect.php';

$response = ['error' => null, 'data' => []];

if (!$conn || (is_object($conn) && property_exists($conn,
'connect_error') && $conn->connect_error)) {
    $response['error'] = 'Ошибка подключения к БД: ' . ($conn &&
is_object($conn) && property_exists($conn, 'connect_error') ?
htmlspecialchars($conn->connect_error) : "Объект \$conn не создан");
    echo json_encode($response);
    exit;
}

// Основной SQL-запрос для получения всех сотрудников и их доступных
слотов времени
// LEFT JOIN используется, чтобы включить всех сотрудников, даже если
у них нет записей в 'availability'
$sql = "SELECT s.staff_id, s.first_name, s.last_name,
s.specialization,
           a.day_of_week,
           IF(a.start_time IS NOT NULL, TIME_FORMAT(a.start_time,
'%H:%i'), NULL) AS start_time,
           IF(a.end_time IS NOT NULL, TIME_FORMAT(a.end_time,
'%H:%i'), NULL) AS end_time
FROM staff s
LEFT JOIN availability a ON s.staff_id = a.staff_id AND
a.is_available = TRUE
ORDER BY s.last_name, s.first_name,
           FIELD(a.day_of_week, 'Monday', 'Tuesday',
'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'),
           a.start_time";

$result = $conn->query($sql);

if ($result) {
    $staff_schedule_data = [];
    while ($row = $result->fetch_assoc()) {

```

```

        // Каждая строка из этого запроса представляет либо сотрудника
        без расписания (если day_of_week NULL),
        // либо конкретный слот доступности для сотрудника.
        // main.js ожидает массив, где каждый элемент - это
        потенциальная строка в таблице расписания.
        $staff_schedule_data[] = [
            'staff_id' => (int)$row['staff_id'],
            'first_name' => htmlspecialchars($row['first_name']),
            'last_name' => htmlspecialchars($row['last_name']),
            'specialization' =>
htmlspecialchars($row['specialization'] ?? '-'),
            'day_of_week' => $row['day_of_week'], // Может быть NULL,
            если у сотрудника нет записей в availability
            'start_time' => $row['start_time'],    // Может быть NULL
            'end_time' => $row['end_time']         // Может быть NULL
        ];
    }
    $result->free();

    if (empty($staff_schedule_data)) {
        // Это условие сработает, если таблица staff пуста
        $response['error'] = 'Сотрудники не найдены в базе данных.';
    } else {
        // main.js (v5_modal_fix) ожидает, что 'data' будет массивом
        объектов,
        // где каждый объект может быть использован для построения
        строки в таблице расписания.
        // Он сам группирует по staff_id для отображения.
        $response['data'] = $staff_schedule_data;
    }
} else {
    $response['error'] = 'Ошибка SQL-запроса при получении расписания:
' . htmlspecialchars($conn->error);
}

if (isset($conn) && is_object($conn) && method_exists($conn, 'close'))
{
    $conn->close();
}
echo json_encode($response);
?>

```

Файл index.php:

```

<?php
// index.php

session_start();
$page_title = "Главная";
include 'includes/header.php';
require_once 'includes/db_connect.php';
?>

```

```

<section class="hero-section">
    <div class="hero-overlay"></div>
    <div class="container hero-content">
        <h1>Добро пожаловать в BeautySalon</h1>
        <p>Ваш оазис красоты и релаксации в самом сердце
города.</p>
        <a href="<?php echo $base_url; ?>booking.php"
class="btn btn-primary btn-lg futuristic-button">Записаться
онлайн</a>
    </div>
</section>

<section class="about-us-short">
    <div class="container">
        <h2>О Нас</h2>
        <p>Мы предлагаем широкий спектр услуг для вашей красоты
и здоровья. Наши мастера – профессионалы своего дела, использующие
только лучшую косметику и современные технологии. Погрузитесь в
атмосферу заботы и комфорта.</p>
    </div>
</section>

<section class="featured-services">
    <div class="container">
        <h2>Популярные Услуги</h2>
        <div class="services-grid">
            <?php
                // Выбираем три случайные услуги из таблицы
services
                $sql = "
                    SELECT
                        id,
                        name,
                        description,
                        price
                    FROM services
                    ORDER BY RAND()
                    LIMIT 3
                ";
                $result = $conn->query($sql);

                if ($result && $result->num_rows > 0) {
                    while ($row = $result->fetch_assoc()) {
                        $id
                            = (int)$row['id'];
                        $name
                            =
htmlspecialchars($row['name']);
                        // обрезаем описание до 100 символов
                        $shortDesc
                            =
htmlspecialchars(mb_substr($row['description'], 0, 100)) . '...';
                        $price
                            =
htmlspecialchars($row['price']);
                        ?>

```

```

        <div class="service-card-featured">
            <div class="service-icon"><i class="fas
fa-spa"></i></div>

            <h3><?php echo $name; ?></h3>
            <p><?php echo $shortDesc; ?></p>
            <p class="price">Цена: <?php echo
$price; ?> руб.</p>
            <a href="<?php echo $base_url;
?>booking.php?service_id=<?php echo $id; ?>"
class="btn btn-secondary">Записаться</a>
        </div>
        <?php
    }
    } else {
        echo "
"
        <p>Популярные услуги скоро
появятся.</p>";
    }
    ?>
</div>
</section>

<section class="cta-section">
    <div class="container">
        <h2>Готовы преобразиться?</h2>
        <p>Запишитесь на процедуру уже сегодня и почувствуйте
разницу!</p>
        <a href="<?php echo $base_url; ?>services.php"
class="btn btn-outline-light futuristic-button">Все Услуги</a>
    </div>
</section>

<?php
$conn->close();
include 'includes/footer.php';
?>

```

Файл login.php:

```

<?php
$page_title = "Вход";
// Убедимся, что сессия стартует до любого вывода HTML
if (session_status() == PHP_SESSION_NONE) {
    session_start();
}
include 'includes/header.php'; // $base_url должен быть определен
здесь или ранее
require_once 'includes/db_connect.php';

$errors = [];

```

```

// Если $base_url не определен в header.php, определите его здесь
if (!isset($base_url)) {
    $script_name = $_SERVER['SCRIPT_NAME'];
    $project_folder_name = 'BeautySalon';
    if (strpos($script_name, '/' . $project_folder_name . '/') !==
false) {
        $base_url = '/' . $project_folder_name . '/';
    } else {
        $base_url = '/';
    }
}

// Если пользователь уже вошел, перенаправляем на страницу "Мои
записи"
if (isset($_SESSION['user_id'])) {
    header("Location: " . htmlspecialchars($base_url) .
"my_appointments.php");
    exit();
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = trim($_POST['email']);
    $password = $_POST['password'];

    if (empty($email)) {
        $errors[] = "Email обязателен для заполнения.";
    } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $errors[] = "Некорректный формат email.";
    }

    if (empty($password)) {
        $errors[] = "Пароль обязателен для заполнения.";
    }

    if (empty($errors)) {
        $stmt = $conn->prepare("SELECT user_id, first_name,
password_hash, role FROM users WHERE email = ?");
        if ($stmt) {
            $stmt->bind_param("s", $email);
            $stmt->execute();
            $result = $stmt->get_result();

            if ($result->num_rows == 1) {
                $user = $result->fetch_assoc();
                if (password_verify($password,
$user['password_hash'])) {
                    // Успешный вход
                    $_SESSION['user_id'] = $user['user_id'];
                    $_SESSION['user_first_name'] =
$user['first_name'];
                    $_SESSION['user_role'] = $user['role'];

                    // Перенаправление после успешного входа

```

```

        // Если было прервано бронирование, вернем на
        страницу бронирования
        if (isset($_SESSION['redirect_to_booking']) &&
$_SESSION['redirect_to_booking'] === true) {
            unset($_SESSION['redirect_to_booking']);
            $booking_params_query = '';
            if(isset($_SESSION['booking_params']) &&
!empty($_SESSION['booking_params'])) {
                $booking_params_query = '?' .
http_build_query($_SESSION['booking_params']);
                unset($_SESSION['booking_params']);
            }
            header("Location: " .
htmlspecialchars($base_url) . "booking.php" . $booking_params_query);
            exit();
        } else {
            header("Location: " .
htmlspecialchars($base_url) . "my_appointments.php");
            exit();
        }
    } else {
        $errors[] = "Неверный email или пароль.";
    }
} else {
    $errors[] = "Неверный email или пароль.";
}
$stmt->close();
} else {
    $errors[] = "Ошибка подготовки запроса: " . $conn->error;
}
}
$conn->close();
}
?>

```

```

<div class="container page-container auth-container">
    <h1 class="page-title">Вход в Аккаунт</h1>

    <?php if (isset($_GET['message'])): ?>
        <div class="alert alert-success"><?php echo
htmlspecialchars(urldecode($_GET['message'])); ?></div>
    <?php endif; ?>

    <?php if (!empty($errors)): ?>
        <div class="alert alert-danger">
            <ul>
                <?php foreach ($errors as $error): ?>
                    <li><?php echo htmlspecialchars($error); ?></li>
                <?php endforeach; ?>
            </ul>
        </div>
    <?php endif; ?>

```

```

        <form action="<?php echo htmlspecialchars($base_url .
'login.php'); ?>" method="POST" class="auth-form">
            <div class="form-group">
                <label for="email">Email</label>
                <input type="email" id="email" name="email" value="<?php
echo isset($_POST['email']) ? htmlspecialchars($_POST['email']) : '';
?>" required>
            </div>
            <div class="form-group">
                <label for="password">Пароль</label>
                <input type="password" id="password" name="password"
required>
            </div>
            <button type="submit" class="btn btn-primary futuristic-
button">Войти</button>
        </form>
        <p class="auth-link">Нет аккаунта? <a href="<?php echo
htmlspecialchars($base_url . 'register.php');
?>">Зарегистрироваться</a></p>
    </div>

<?php include 'includes/footer.php'; ?>

```

Файл logout.php:

```

<?php
// Строго в начале файла, до любого вывода HTML
if (session_status() == PHP_SESSION_NONE) {
    session_start();
}

// Определение $base_url, если он не определен глобально
// Эта логика должна быть такой же, как и в других ваших файлах
(header.php, index.php и т.д.)
if (!isset($base_url)) {
    $script_name = $_SERVER['SCRIPT_NAME'];
    $project_folder_name = 'BeautySalon'; // Убедитесь, что это имя
вашей папки проекта

    if (strpos($script_name, '/' . $project_folder_name . '/') !==
false) {
        $base_url = '/' . $project_folder_name . '/';
    }
    // Проверка, если скрипт в корневой папке проекта, которая
является корнем домена
    // Это более сложный случай, и может потребоваться более точное
определение
    // Для простоты, если не в подпапке BeautySalon, считаем корнем
'/'
    else {
        $base_url = '/';
    }
}

```

```

}

// 1. Очистить все переменные сессии
$_SESSION = array();

// 2. Если используется идентификатор сессии в cookie, удалить cookie
сессии
// Это важно для полного уничтожения сессии.
if (ini_get("session.use_cookies")) {
    $params = session_get_cookie_params();
    setcookie(session_name(), '', time() - 42000,
        $params["path"], $params["domain"],
        $params["secure"], $params["httponly"]
    );
}

// 3. Уничтожить сессию
// Это удаляет все данные, связанные с текущей сессией.
session_destroy();

// 4. Перенаправить на главную страницу или страницу входа
// Использование htmlspecialchars для $base_url здесь не обязательно,
т.к. это URL,
// но не повредит для единообразия.
header("Location: " . htmlspecialchars($base_url) . "index.php");
exit(); // Важно вызвать exit() после header("Location:") для
прекращения выполнения скрипта.
?>

```

Файл my_appointments.php:

```

<?php
// my_appointments.php

ini_set('display_errors', 1);
error_reporting(E_ALL);

$page_title = "Мои Записи";
if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

include 'includes/header.php';
require_once 'includes/db_connect.php';

if (!isset($_SESSION['user_id'])) {
    header("Location: " . $base_url . "login.php?message=" .
        urlencode("Пожалуйста, войдите, чтобы просмотреть свои записи."));
    exit();
}

$client_id = $_SESSION['user_id'];

```

```

$message    = '';
$error      = '';

// Обработка отмены, если была нажата кнопка «cancel»
if (isset($_GET['action'], $_GET['id']) && $_GET['action'] ===
'cancel') {
    $appointment_id_to_cancel = intval($_GET['id']);
    // Проверяем право на отмену
    $stmt_check = $conn->prepare("
        SELECT appointment_datetime
        FROM appointments
        WHERE appointment_id = ?
            AND client_id = ?
            AND status = 'scheduled'
    ");
    $stmt_check->bind_param("ii", $appointment_id_to_cancel,
$client_id);
    $stmt_check->execute();
    $res_check = $stmt_check->get_result();

    if ($res_check && $res_check->num_rows === 1) {
        $row          = $res_check->fetch_assoc();
        $dt            = new DateTime($row['appointment_datetime']);
        $now            = new DateTime();
        $interval       = $now->diff($dt);
        $hours_diff     = ($interval->days * 24) + $interval->h;

        if ($dt < $now) {
            $error = "Нельзя отменить прошедшую запись.";
        } elseif ($hours_diff < 24) {
            $error = "Отмена возможна не позднее чем за 24 часа.";
        } else {
            $stmt_cancel = $conn->prepare("
                UPDATE appointments
                SET status = 'cancelled'
                WHERE appointment_id = ?
                    AND client_id = ?
            ");
            $stmt_cancel->bind_param("ii", $appointment_id_to_cancel,
$client_id);
            if ($stmt_cancel->execute()) {
                $message = "Запись успешно отменена.";
            } else {
                $error = "Ошибка отмены: " . $stmt_cancel->error;
            }
        }
        $stmt_check->close();
    } else {
        $error = "Запись не найдена или уже отменена.";
    }
}

// Основной запрос: теперь правильно JOIN по services.name

```

```

$sql = "
    SELECT
        a.appointment_id,
        s.name AS service_name,
        st.first_name AS staff_first_name,
        st.last_name AS staff_last_name,
        a.appointment_datetime,
        a.status
    FROM appointments a
    LEFT JOIN services s ON a.service_id = s.id
    LEFT JOIN staff st ON a.staff_id = st.staff_id
    WHERE a.client_id = ?
    ORDER BY a.appointment_datetime DESC
";

$stmt = $conn->prepare($sql);
$stmt->bind_param("i", $client_id);
$stmt->execute();
$result = $stmt->get_result();
?>

<div class="container page-container">
    <h1 class="page-title">Мои Записи</h1>
    <p class="page-subtitle">Здесь вы можете просмотреть историю
    ваших записей и управлять предстоящими визитами.</p>

    <?php if ($message): ?>
        <div class="alert alert-success"><?php echo $message;
?></div>
    <?php endif; ?>
    <?php if ($error): ?>
        <div class="alert alert-danger"><?php echo $error;
?></div>
    <?php endif; ?>

    <?php if ($result && $result->num_rows > 0): ?>
        <div class="appointments-list">
            <?php while ($row = $result->fetch_assoc()): ?>
                <?php
                    $status_rus = '';
                    switch ($row['status']) {
                        case 'scheduled': $status_rus =
'Запланировано'; break;
                        case 'completed': $status_rus = 'Завершено';
break;
                        case 'cancelled': $status_rus = 'Отменено';
break;
                        case 'no-show': $status_rus = 'Неявка';
break;
                        default: $status_rus =
htmlspecialchars($row['status']);
                    }

```

```

        $appointment_dt = new
DateTime($row['appointment_datetime']);
        $is_past          = $appointment_dt < new
DateTime();
        ?>
        <div class="appointment-card status-<?php echo
htmlspecialchars($row['status']); ?>
            <?php echo $is_past ? 'past-appointment'
: ''; ?>">
            <h3><?php echo
htmlspecialchars($row['service_name']); ?></h3>
            <p><i class="fas fa-user-tie"></i>
                <strong>Мастер:</strong>
                <?php echo
htmlspecialchars($row['staff_first_name'] . ' ' .
$row['staff_last_name']); ?>
            </p>
            <p><i class="far fa-calendar-alt"></i>
                <strong>Дата и время:</strong>
                <?php echo $appointment_dt->format('d.m.Y
H:i'); ?>
            </p>
            <p><i class="fas fa-info-circle"></i>
                <strong>Статус:</strong>
                <span class="status-badge"><?php echo
$status_rus; ?></span>
            </p>
            <?php if ($row['status'] === 'scheduled' &&
!$is_past):
                $now_for_cancel = new DateTime();
                $diff_cancel     = $now_for_cancel-
>diff($appointment_dt);
                $hours_diff2     = ($diff_cancel->days *
24) + $diff_cancel->h;
                if ($hours_diff2 >= 24): ?>
                    <a href="<?php echo $base_url;
?>my_appointments.php?action=cancel&id=<?php echo
$row['appointment_id']; ?>"
                        class="btn btn-danger btn-sm
futuristic-button-sm"
                            onclick="return confirm('Вы
уверены, что хотите отменить эту запись?');">
                                <i class="fas fa-times-
circle"></i> Отменить запись
                            </a>
                <?php else: ?>
                    <p class="cannot-cancel-
notice"><small>Отмена возможна не позднее чем за 24 часа.</small></p>
                <?php
                endif;
            endif; ?>
        </div>

```

```

        <?php endwhile; ?>
    </div>
    <?php else: ?>
        <p>У вас пока нет записей. Хотите <a href="<?php echo
$base_url; ?>booking.php">записаться на услугу</a>?</p>
    <?php endif; ?>
</div>

```

```

<?php
$stmt->close();
$conn->close();
include 'includes/footer.php';

```

Файл privacy_policy.php:

```

<?php
$page_title = "Политика конфиденциальности";
if (session_status() == PHP_SESSION_NONE) {
    session_start();
}
include 'includes/header.php';

if (!isset($base_url)) {
    $script_name = $_SERVER['SCRIPT_NAME'];
    $project_folder_name = 'BeautySalon';
    if (strpos($script_name, '/' . $project_folder_name . '/') !==
false) {
        $base_url = '/' . $project_folder_name . '/';
    } else {
        $base_url = '/';
    }
}
?>

<div class="container page-container">
    <h1 class="page-title"><?php echo $page_title; ?></h1>
    <p class="page-subtitle">Дата последнего обновления: 21 мая 2025
г.</p>

    <div class="privacy-policy-content">
        <section>
            <h2>1. Введение</h2>
            <p>Добро пожаловать в "BeautySalon". Мы ценим вашу
конфиденциальность и стремимся защищать ваши персональные данные.
Настоящая Политика конфиденциальности объясняет, как мы собираем,
используем, раскрываем и защищаем вашу информацию, когда вы посещаете
наш веб-сайт https://skyddad.se/digital-sakerhet/integritet/tips-har-kan-du-kolla-om-en-webbsida-ar-saker-eller-ej/ и пользуетесь нашими
услугами.</p>
        </section>

        <section>

```

<h2>2. Сбор вашей информации</h2>

<p>Мы можем собирать информацию о вас различными способами. Информация, которую мы можем собирать на Сайте, включает:</p>

Личные данные: Имя, фамилия, адрес электронной почты, номер телефона, которые вы добровольно предоставляете нам при регистрации на Сайте или при записи на услуги.

Данные о посещениях: Информация, которую ваш браузер отправляет автоматически, такая как IP-адрес, тип браузера, операционная система, время доступа, и страницы, которые вы просматривали до и после посещения Сайта.

Данные об использовании услуг: Информация о выбранных вами услугах, мастерах, дате и времени записи.

</section>

<section>

<h2>3. Использование вашей информации</h2>

<p>Имея точную информацию о вас, мы можем предоставить вам бесперебойный, эффективный и индивидуальный опыт. В частности, мы можем использовать информацию, собранную о вас через Сайт, для:</p>

Создания и управления вашей учетной записью.
Обработки ваших записей на услуги и отправки подтверждений.

Отправки вам электронных писем о вашей учетной записи или заказах.

Улучшения работы Сайта и наших услуг.

Ответа на запросы клиентов и оказания поддержки.

Информирования вас о новых услугах, специальных предложениях и других новостях, которые могут вас заинтересовать (с вашего согласия).

</section>

<section>

<h2>4. Раскрытие вашей информации</h2>

<p>Мы не будем передавать вашу личную информацию третьим лицам без вашего согласия, за исключением случаев, предусмотренных законом, или как описано ниже:</p>

Поставщики услуг: Мы можем передавать вашу информацию сторонним поставщикам услуг, которые выполняют услуги от нашего имени, такие как обработка платежей, анализ данных, рассылка электронной почты, хостинг-услуги, обслуживание клиентов и маркетинговая помощь.

Требования законодательства: Мы можем раскрыть вашу информацию, если это необходимо для ответа на

судебный процесс, расследования возможного нарушения наших политик или для защиты прав, собственности и безопасности других лиц.

</section>

<section>

<h2>5. Безопасность вашей информации</h2>

<p>Мы используем административные, технические и физические меры безопасности для защиты вашей личной информации. Хотя мы предприняли разумные шаги для защиты личной информации, которую вы нам предоставляете, имейте в виду, что никакие меры безопасности не являются идеальными или непроницаемыми, и ни один метод передачи данных не может гарантировать отсутствие какого-либо перехвата или другого типа неправомерного использования.</p>

</section>

<section>

<h2>6. Ваши права</h2>

<p>Вы имеете право на доступ, исправление или удаление ваших персональных данных, которые мы храним. Вы также можете возразить против обработки ваших данных или запросить ограничение такой обработки. Для этого свяжитесь с нами по указанным ниже контактным данным.</p>

</section>

<section>

<h2>7. Изменения в настоящей Политике конфиденциальности</h2>

<p>Мы можем время от времени обновлять настоящую Политику конфиденциальности. Обновленная версия будет обозначена обновленной датой "Последнее обновление", и обновленная версия вступит в силу, как только станет доступной. Мы рекомендуем вам периодически просматривать эту Политику конфиденциальности, чтобы быть в курсе того, как мы защищаем вашу информацию.</p>

</section>

<section>

<h2>8. Свяжитесь с нами</h2>

<p>Если у вас есть вопросы или комментарии по поводу настоящей Политики конфиденциальности, пожалуйста, свяжитесь с нами:</p>

<p>Email: info@beautysalon.example.com</p>

<p>Телефон: +7 (495) 123-45-67</p>

<p>Адрес: г. Москва, ул. Примерная, д. 1, офис 101</p>

</section>

</div>

</div>

<style>

/* Стили специфичные для страницы Политики конфиденциальности */
.privacy-policy-content section {
margin-bottom: 2.5rem;
}

```

.privacy-policy-content h2 {
    font-family: var(--font-headings);
    color: var(--text-color-headings);
    font-size: 1.8rem;
    margin-bottom: 1rem;
    padding-bottom: 0.5rem;
    border-bottom: 2px solid var(--primary-color);
    display: inline-block;
}
.privacy-policy-content p,
.privacy-policy-content ul li {
    color: var(--text-color-muted);
    line-height: 1.7;
    margin-bottom: 0.8rem;
}
.privacy-policy-content ul {
    list-style-type: disc;
    padding-left: 25px;
    margin-bottom: 1rem;
}
.privacy-policy-content strong {
    color: var(--text-color-main);
}
</style>

```

```
<?php include 'includes/footer.php'; ?>
```

Файл register.php:

```

<?php
$page_title = "Регистрация";
include 'includes/header.php'; // Предполагается, что $base_url
определяется в header.php или ранее
require_once 'includes/db_connect.php';

$errors = [];
$success_message = '';

// Если $base_url не определен в header.php, определите его здесь
if (!isset($base_url)) {
    // Определяем $base_url относительно корня сайта.
    // Это может потребовать корректировки в зависимости от вашей
    структуры.
    // Пример: если скрипт в C:\xampp\htdocs\BeautySalon\register.php
    // и корень сайта http://localhost/BeautySalon/
    // то $base_url = '/BeautySalon/';
    // Если XAMPP настроен так, что BeautySalon это корень (например,
    через VirtualHost), то $base_url = '/';

    // Простой способ определения, если проект в подпапке htdocs
    $script_name = $_SERVER['SCRIPT_NAME']; //
    /BeautySalon/register.php

```

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

```

        $project_folder_name = 'BeautySalon'; // Имя вашей папки проекта
        if (strpos($script_name, '/' . $project_folder_name . '/') !==
false) {
            $base_url = '/' . $project_folder_name . '/';
        } else {
            $base_url = '/'; // Если в корне или другая конфигурация
        }
    }

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $first_name = trim($_POST['first_name']);
    $last_name = trim($_POST['last_name']);
    $email = trim($_POST['email']);
    $phone_number = trim($_POST['phone_number']); // Может быть пустым
    $password = $_POST['password'];
    $password_confirm = $_POST['password_confirm'];

    if (empty($first_name)) {
        $errors[] = "Имя обязательно для заполнения.";
    }
    if (empty($last_name)) {
        $errors[] = "Фамилия обязательна для заполнения.";
    }
    if (empty($email)) {
        $errors[] = "Email обязателен для заполнения.";
    } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $errors[] = "Некорректный формат email.";
    } else {
        // Проверка на существование email в базе
        $stmt_check = $conn->prepare("SELECT user_id FROM users WHERE
email = ?");
        if ($stmt_check) {
            $stmt_check->bind_param("s", $email);
            $stmt_check->execute();
            $result_check = $stmt_check->get_result();
            if ($result_check->num_rows > 0) {
                $errors[] = "Пользователь с таким email уже
существует.";
            }
            $stmt_check->close();
        } else {
            $errors[] = "Ошибка подготовки запроса для проверки email:
" . $conn->error;
        }
    }

    if (empty($password)) {
        $errors[] = "Пароль обязателен для заполнения.";
    } elseif (strlen($password) < 6) {
        $errors[] = "Пароль должен быть не менее 6 символов.";
    }
}

```

```

        if ($password !== $password_confirm) {
            $errors[] = "Пароли не совпадают.";
        }

        if (empty($errors)) {
            $password_hash = password_hash($password, PASSWORD_DEFAULT);

            // Обработка phone_number: если пустой, то NULL
            $phone_number_to_db = !empty($phone_number) ? $phone_number :
null;

            $stmt_insert = $conn->prepare("INSERT INTO users (first_name,
last_name, email, password_hash, phone_number) VALUES (?, ?, ?, ?,
?)");
            if ($stmt_insert) {
                $stmt_insert->bind_param("sssss", $first_name, $last_name,
$email, $password_hash, $phone_number_to_db);
                if ($stmt_insert->execute()) {
                    $success_message = "Регистрация прошла успешно! Теперь
вы можете <a href='" . htmlspecialchars($base_url) .
"login.php">войти</a>.";
                } else {
                    $errors[] = "Ошибка регистрации. Пожалуйста,
попробуйте позже. Ошибка: " . $stmt_insert->error;
                }
                $stmt_insert->close();
            } else {
                $errors[] = "Ошибка подготовки запроса для добавления
пользователя: " . $conn->error;
            }
        }
        $conn->close();
    }
    ?>

<div class="container page-container auth-container">
    <h1 class="page-title">Регистрация</h1>

    <?php if (!empty($errors)): ?>
        <div class="alert alert-danger">
            <ul>
                <?php foreach ($errors as $error): ?>
                    <li><?php echo htmlspecialchars($error); ?></li>
                <?php endforeach; ?>
            </ul>
        </div>
    <?php endif; ?>

    <?php if (!empty($success_message)): ?>
        <div class="alert alert-success"><?php echo $success_message;
?></div>
    <?php else: ?>

```

```

        <form action="<?php echo htmlspecialchars($base_url .
'register.php'); ?>" method="POST" class="auth-form">
            <div class="form-group">
                <label for="first_name">Имя</label>
                <input type="text" id="first_name" name="first_name"
value="<?php echo isset($_POST['first_name']) ?
htmlspecialchars($_POST['first_name']) : ''; ?>" required>
            </div>
            <div class="form-group">
                <label for="last_name">Фамилия</label>
                <input type="text" id="last_name" name="last_name"
value="<?php echo isset($_POST['last_name']) ?
htmlspecialchars($_POST['last_name']) : ''; ?>" required>
            </div>
            <div class="form-group">
                <label for="email">Email</label>
                <input type="email" id="email" name="email"
value="<?php echo isset($_POST['email']) ?
htmlspecialchars($_POST['email']) : ''; ?>" required>
            </div>
            <div class="form-group">
                <label for="phone_number">Номер телефона
(необязательно)</label>
                <input type="tel" id="phone_number"
name="phone_number" value="<?php echo isset($_POST['phone_number']) ?
htmlspecialchars($_POST['phone_number']) : ''; ?>" placeholder="+7
(XXX) XXX-XX-XX">
            </div>
            <div class="form-group">
                <label for="password">Пароль</label>
                <input type="password" id="password" name="password"
required>
            </div>
            <div class="form-group">
                <label for="password_confirm">Подтвердите
пароль</label>
                <input type="password" id="password_confirm"
name="password_confirm" required>
            </div>
            <button type="submit" class="btn btn-primary futuristic-
button">Зарегистрироваться</button>
        </form>
        <p class="auth-link">Уже есть аккаунт? <a href="<?php echo
htmlspecialchars($base_url . 'login.php'); ?>">Войти</a></p>
        <?php endif; ?>
    </div>

<?php include 'includes/footer.php'; ?>

```

Файл services.php:

```
<?php
```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090301.32.001 ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 96 |

```

// services.php

session_start();
$page_title    = "Наши Услуги";
$current_page  = "services";

include 'includes/header.php';
require_once 'includes/db_connect.php';

// Выбираем все услуги
$sql          = "
    SELECT
        id,
        category,
        name,
        description,
        duration,
        price,
        image_filename
    FROM services
    ORDER BY category, name
";
$result = $conn->query($sql);
?>

<div class="container page-container">
    <h1 class="page-title">Наши Услуги</h1>
    <p class="page-subtitle">
        Откройте для себя полный спектр процедур для вашей красоты и
        благополучия.
    </p>

    <?php
    if ($result && $result->num_rows > 0) {
        $current_category = '';
        while ($row = $result->fetch_assoc()) {
            // При смене категории – закрываем предыдущую сетку и
            открываем новую
            if ($current_category !== $row['category']) {
                if ($current_category !== '') {
                    echo '</div>'; // закрываем .category-services-
grid
                }
                $current_category = $row['category'];
                echo '<h2 class="category-title">'
                    . htmlspecialchars($current_category)
                    . '</h2>';
                echo '<div class="category-services-grid">';

                // *** ТУТ – путь берём прямо из image_filename, он уже
                'images/services/...'
                $img = $row['image_filename']

```

```

        ? $base_url . $row['image_filename']
        : $base_url . 'images/services/default.jpg';
    ?>

    <div class="service-card">
        "
            style="width:100%; border-radius:var(--border-
radius); margin-bottom:1rem;"
        >
        <h3><?php echo htmlspecialchars($row['name']); ?></h3>
        <p class="description"><?php echo
htmlspecialchars($row['description']); ?></p>
        <p>
            <strong>Продолжительность:</strong>
            <?php echo htmlspecialchars($row['duration']); ?>
        </p>
        <p class="price">
            <strong>Цена:</strong>
            <?php echo htmlspecialchars($row['price']); ?>
руб.
        </p>
        <a
            href="<?php echo htmlspecialchars($base_url .
'booking.php?service_id=' . $row['id']); ?>"
            class="btn btn-primary futuristic-button"
            >Записаться</a>
    </div>

    <?php
    }
    // Закрываем последнюю .category-services-grid
    if ($current_category !== '') {
        echo '</div>';
    }
    } else {
        echo '<p>Информация об услугах сейчас недоступна. Пожалуйста,
зайдите позже.</p>';
    }
    ?>

</div>

<?php
$conn->close();
include 'includes/footer.php';
?>

```

Файл team.php:

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090301.32.001 ПЗ | Лист |
| | | | | | | 98 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

```

<?php
?><!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Наша Команда | BeautySalon</title>
    <link rel="stylesheet" href="assets/css/style.css">
    <link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500
;600;700&family=Orbitron:wght@400;700&display=swap" rel="stylesheet">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css" integrity="sha512-
pQfmXUlhgqYq4+6q1F+BwFJLeUv7u+P5R9C3A7Ysjm5pjxsne6xxQZUBLbr4sHsz7Nv9W
0Z47ZRUQhOglXfKw==" crossorigin="anonymous" referrerpolicy="no-
referrer" />
</head>
<body>
<header class="main-header">
    <div class="container">
        <a href="index.php" class="logo">
            <i class="fa-solid fa-leaf logo-icon"></i>
            <span class="logo-text">BeautySalon</span>
        </a>
        <button class="mobile-nav-toggle" id="mobile-nav-toggle">
            <i class="fa fa-bars open-icon"></i>
            <i class="fa fa-times close-icon"></i>
        </button>
        <nav class="main-nav" id="main-nav">
            <ul>
                <li><a href="index.php"><i class="fa fa-home fa-
fw"></i>Главная</a></li>
                <li><a href="services.php"><i class="fa fa-scissors
fa-fw"></i>Услуги</a></li>
                <li><a href="team.php" class="active"><i class="fa fa-
users fa-fw"></i>Наша Команда</a></li>
                <li><a href="booking.php"><i class="fa fa-calendar fa-
fw"></i>Запись</a></li>
                <li><a href="contact.php"><i class="fa fa-envelope fa-
fw"></i>Контакты</a></li>
                <li><a href="my_appointments.php"><i class="fa fa-user
fa-fw"></i>Мои записи</a></li>
                <li><a href="logout.php"><i class="fa fa-sign-out-alt
fa-fw"></i>Выход</a></li>
            </ul>
        </nav>
    </div>
</header>

<main>

```

```

<section class="hero-section">
  <div class="hero-overlay"></div>
  <div class="hero-content">
    <h1>Наша Замечательная Команда</h1>
    <p>Профессионалы с богатым опытом и истинным творческим
подходом.</p>
  </div>
</section>

<div class="container page-container team-page-container">
  <div class="team-layout">
    <div class="team-member-card fade-in-section">
      <div class="team-member-photo-container">
        
      </div>
      <div class="team-member-info">
        <h3 class="team-member-name">Анна Соколова</h3>
        <p class="team-member-specialization">Топ-стилист,
Арт-директор</p>
        <p class="team-member-
experience"><strong>Опыт:</strong> 12 лет</p>
        <blockquote class="team-member-bio">
          12+ лет опыта. Победитель международных
конкурсов. Специализация: креативные трансформации и сложные
окрашивания.
        </blockquote>
        <a href="booking.php?specialist_id=1" class="btn
btn-primary btn-sm team-member-book-btn">Записаться к мастеру</a>
      </div>
    </div>

    <div class="team-member-card fade-in-section">
      <div class="team-member-photo-container">
        
      </div>
      <div class="team-member-info">
        <h3 class="team-member-name">Ольга Максимова</h3>
        <p class="team-member-specialization">Ведущий
мастер нейл-арта</p>
        <p class="team-member-
experience"><strong>Опыт:</strong> 9 лет</p>
        <blockquote class="team-member-bio">
          Эксперт по сложным дизайнам, аппаратному
маникюру и медицинскому педикюру. Более 9 лет в индустрии.
        </blockquote>
        <a href="booking.php?specialist_id=2" class="btn
btn-primary btn-sm team-member-book-btn">Записаться к мастеру</a>
      </div>
    </div>
  </div>
</div>

```

```

        <div class="team-member-card fade-in-section">
            <div class="team-member-photo-container">
                
            </div>
            <div class="team-member-info">
                <h3 class="team-member-name">Елена Ковалева</h3>
                <p class="team-member-specialization">Врач
дермато-косметолог</p>
                <p class="team-member-
experience"><strong>Опыт:</strong> 7 лет</p>
                <blockquote class="team-member-bio">
                    Высшее медицинское образование. Специалист по
инъекционным методикам, аппаратной косметологии и лечению кожи.
                </blockquote>
                <a href="booking.php?specialist_id=3" class="btn
btn-primary btn-sm team-member-book-btn">Записаться к мастеру</a>
            </div>
        </div>
    </div>
</main>

<footer class="main-footer">
    <div class="container footer-content-wrapper">
        <div class="footer-section footer-about">
            <span class="footer-logo-text"><i class="fa-solid fa-
leaf"></i>BeautySalon</span>
            <p class="footer-tagline">Ваш персональный оазис красоты и
релаксации. Мы стремимся предоставить вам лучший сервис и незабываемые
впечатления.</p>
            <div class="footer-contact-info">
                <p><i class="fa fa-map-marker-alt fa-fw"></i>г.
Москва, ул. Примерная, д. 1, офис 101</p>
                <p><i class="fa fa-phone fa-fw"></i>+7 (495) 123-45-
67</p>
                <p><i class="fa fa-envelope fa-fw"></i><a
href="mailto:info@beautysalon.ru">info@beautysalon.ru</a></p>
            </div>
        </div>
        <div class="footer-section footer-quick-links">
            <h4>Навигация</h4>
            <ul>
                <li><a href="index.php">Главная</a></li>
                <li><a href="services.php">Услуги</a></li>
                <li><a href="team.php">Наша Команда</a></li>
                <li><a href="booking.php">Запись</a></li>
                <li><a href="contact.php">Контакты</a></li>
            </ul>
        </div>
        <div class="footer-section footer-newsletter">

```

```

        <h4>Будьте в курсе</h4>
        <p>Подпишитесь на нашу рассылку, чтобы первыми узнавать о
специальных предложениях и новинках.</p>
        <form class="footer-newsletter-form" action="#"
method="post">
            <input type="email" name="email" placeholder="Введите
ваш email..." class="text-input newsletter-input">
            <button type="submit" class="btn newsletter-submit-
btn">Подписаться <i class="fa fa-arrow-right"></i></button>
        </form>
        <div class="footer-social-icons">
            <a href="#"><i class="fab fa-facebook-f"></i></a>
            <a href="#"><i class="fab fa-instagram"></i></a>
            <a href="#"><i class="fab fa-vk"></i></a>
            <a href="#"><i class="fab fa-telegram-plane"></i></a>
        </div>
    </div>
</div>
<div class="footer-bottom-bar">
    <p>&copy; 2025 BeautySalon. Все права защищены.</p>
    <p class="footer-credits">Разработка: <a href="#">Ваша
Компания</a></p>
</div>
</footer>

<script>
    const navToggle = document.getElementById('mobile-nav-toggle');
    const nav = document.getElementById('main-nav');
    navToggle.addEventListener('click', () => {
        nav.classList.toggle('active');
        navToggle.classList.toggle('active');
    });
</script>
</body>
</html>

```

Приложение В

(Обязательное)

Web-приложение для онлайн записи для салона красоты

Презентация

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090301.32.001 ПЗ | Лист |
| | | | | | | 103 |
| Изм. | Лист | № докум. | Подпись | Дата | | |



Web-приложение для салона красоты

Автор работы:

студент гр. 21ВВП2 Кирсанова Дарья Михайловна

Научный руководитель:

д.т.н., профессор каф. ВТ Зинкин Сергей Александрович

1

Актуальность разработки

- Быстрая и удобная онлайн-запись к понравившемуся мастеру на удобный для вас день и время;
- Автоматизация рабочих процессов;
- Преимущество салонов с данной технологией.

2

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

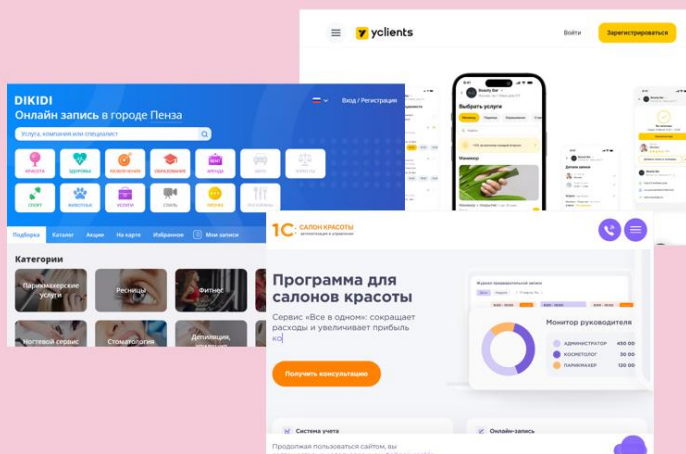
ПГУ 1.090301.32.001 ПЗ

Лист

104

Анализ существующих аналогов

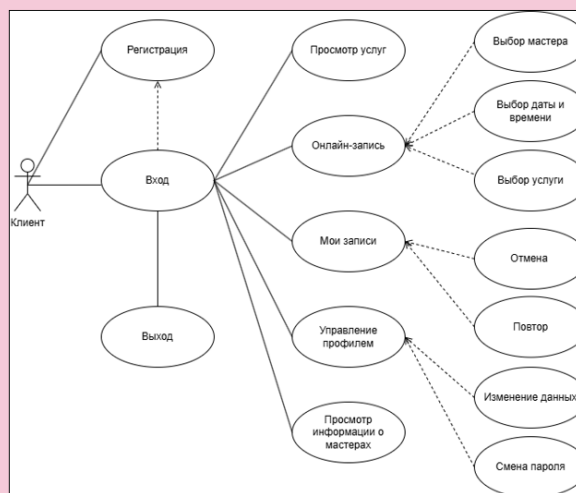
- «DIKIDI»
- «YCLIENTS»
- «salon1c.ru»



3

Возможности приложения

- просмотр списка услуг и их описания;
- просмотр информации о мастерах салона;
- онлайн-запись на услуги к выбранному мастеру на удобное время;
- просмотр своих предстоящих и прошедших записей;
- регистрация и управление своим профилем.



4

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

ПГУ 1.090301.32.001 ПЗ

Лист

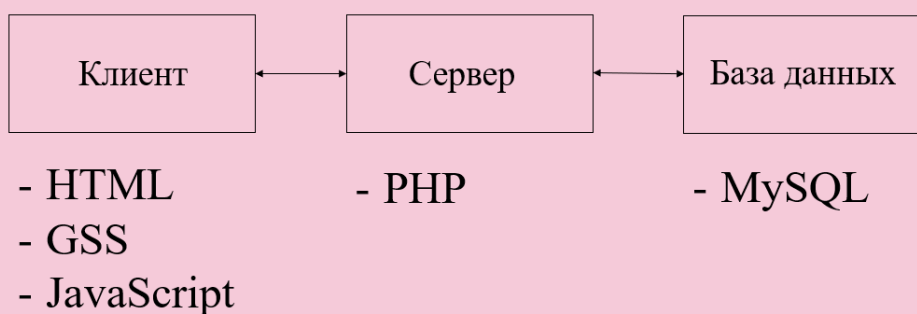
105

Выбор технологий и средств разработки приложения

- PHP;
- MySQL;
- HTML;
- CSS;
- JavaScript;
- XAMPP.

5

Проектирование информационной системы



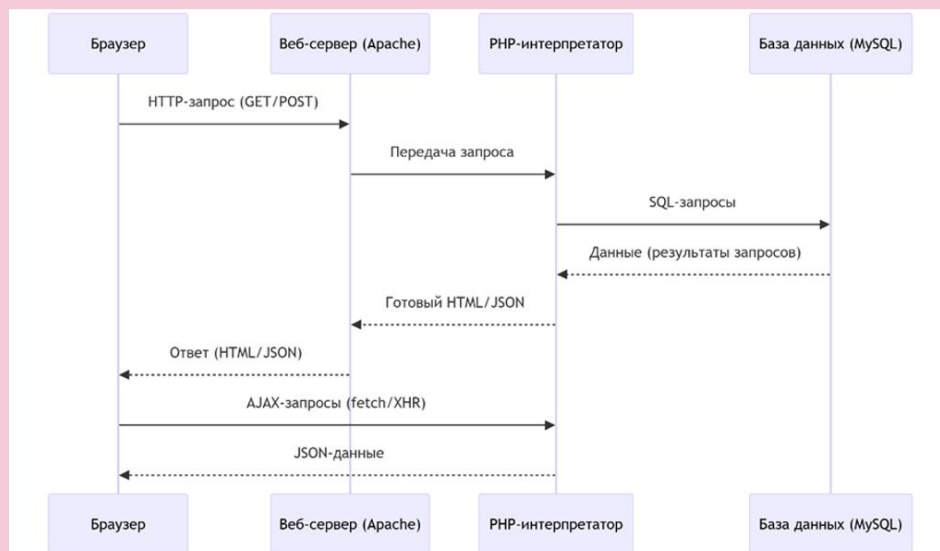
6

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

ПГУ 1.090301.32.001 ПЗ

Лист

106



7

Спасибо за внимание!

8

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

ПГУ 1.090301.32.001 ПЗ

Лист

107