МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет	*Кафедра*
вычислительной техники	«Вычислительная техника»

Специальность: 09.05.01 «Применение и эксплуатация автоматизированных систем специального назначения»

Специализация: «Эксплуатация вычислительных машин, комплексов, систем и сетей специального назначения»

ДИПЛОМНЫЙ ПРОЕКТ

на тему

ПРОГРАММА ДЛЯ БРОНИРОВАНИЯ РЖД БИЛЕТОВ

Студент _____ 18.06.2025 Мангутов Антон Анатольевич
(подпись, дата) (ФИО полностью)

Руководитель _____ 18.06.2025 Егоров В.Ю.
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр _____ 18.06.2025 Бычков А.С.
(подпись, дата) (фамилия, инициалы)

Рецензент гл. специалист _____ Васильев А.В.
(должность, место работы) (подпись, дата) (фамилия, инициалы)

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

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

Работа защищена с отметкой удовлетворительно

(протокол заседания ГЭК от 23.06.2025 № 10)

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет	*Кафедра*
Вычислительной техники	Вычислительная техника

«Утверждаю»
Заведующий кафедрой
Митрохин М.А.
«6» 05 2025 г.

ЗАДАНИЕ
НА ДИПЛОМНЫЙ ПРОЕКТ

1. Студент Мангутов Антон Анатольевич гр. 20ВОЭ1 факультета вычислительной техники.

2.Тема работы: «Программа для бронирования РЖД билетов».

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

3. Руководитель работы к.т.н., доцент каф. ВТ «ПГУ» Егоров В.Ю.

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

Разработать программу для бронирования РЖД билетов. Программа должна выполнять следующие функции: покупка билетов на поезд, возможность отмены заказа, автоматическое, мониторинг истории заказов, добавление, изменение и удаление рейсов. Приложение должно быть спроектировано с применением клиент-серверной архитектуры и состоять из трех связанных между собой частей: клиентская часть, серверная часть, база данных. Клиентская часть программы должна обладать графическим интерфейсом. Серверная часть должна включать в себя: сбор, обработку и анализ информации, собранной с информационных ресурсов. Взаимодействие между серверной и клиентской частями должно осуществляться с помощью Open Server.

5. Перечень подлежащих разработке вопросов.

Разработать приложение «Программа для бронирования РЖД билетов», которое будет способствовать удобному и быстрому приобретения железнодорожных билетов на поезда. Приложение должно быть спроектировано с применением клиент-серверной архитектуры и состоять из трех связанных между собой частей: клиентская часть, серверная часть, база данных. Взаимодействие между серверной и клиентской частями должно осуществляться с помощью Open Server

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	10	06.05.2025 – 09.05.2025	
2.	Проектирование системы и базы данных	10	10.05.2025 – 12.05.2025	
3.	Разработка системы и базы данных	30	13.05.2025 – 20.05.2025	
4.	Отладка и тестирование	10	21.05.2025 – 23.05.2025	
5.	Разработка экономической части	5	24.05.2025	
6.	Разработка БЖД раздела	5	25.05.2025	
7.	Оформление графической части	10	26.05.2025 – 28.05.2025	
8.	Оформление выпускной работы	20	29.05.2025 – 02.06.2025	

Дата выдачи задания 06.05.2025 г.

Руководитель дипломного проекта _____ Егоров В.Ю.
(подпись, дата)

Задание к исполнению принял студент _____ Мангутов А.А.
(подпись, дата)

Работу к защите допустить
Декан факультета _____ Фионова Л.Р.
(подпись, дата)
23.06.2025

Реферат

Дипломная работа содержит 139 листов, 33 рисунка, 10 таблиц, 7 использованных источников и 4 приложения.

СЕРВЕР, КЛИЕНТ, ФУНКЦИОНАЛ, РАЗРАБОТКА, ПРОГРАММА, СТРУКТУРА, ДАННЫЕ

Цель работы – разработать веб-приложение, которое обеспечит пользователей возможностью легко находить и бронировать билеты на поезда, а также управлять своими поездками..

Объектами разработки являются:

– серверное приложение работает с базой данных для получения информации о поездах, расписаниях, ценах и наличии мест;

– клиентское приложение для обработки действия пользователя, такие как нажатия кнопок, выбор мест и т. д, также отображает графический интерфейс и отправляет запросы на сервер.

В результате дипломного проектирования было разработано приложение, полностью соответствующее поставленной задаче в резуьтате которой, разработано веб-приложение, разработана серверная часть приложения, для сбора данных о поездах, расписаниях, ценах и наличии мест, создана клиентская часть приложения с графическим интерфейсом для и отображения результатов.

Содержание

Перечень сокращений и обначений 5
Введение 6
1 Программа для бронирования РЖД билетов 7
1.1 Предпроектный анализ 7
1.1.1 Актуальность 7
1.1.2 Пользователи и сценарии 7
1.1.3 Анализ аналогов 9
1.1.4 Выбор языка программирования 11
1.1.5 Выбор фреймворка для серверной части приложения 14
1.1.6 Хранение данных 16
1.1.7 Выбор СУБД 17
1.1.8 Выбор платформы и веб - сервера 18
1.1.9 Выбор библиотеки для реализации графического интерфейса 20
1.1.10 Функциональные требования 22
1.1.11 Выводы по предпроектному анализу 22
1.2 Проектирование и разработка 23
1.2.1 Разработка серверной части приложения 23
1.2.2 Разработка клиентской части приложения 27
1.3 Проектирование базы данных 32
1.3.1 Демонстрация функционала
1.3.2 Демонстрация структуры БД
2 Результаты испытаний 32
3 Оценка затрат на разработку программы для бронирования РЖД билетов 46
3.1 Организация производства разработки программы для бронирования РЖД билетов 45
3.2 Расчет себестоимости ПО для программы бронирования РЖД билетов .. 49
3.3 Оценка Эффективности программы бронирования РЖД билетов 52

3.4 Анализ рисков при разработке программы для бронирования РЖД билетов 53
4 Безопасность на транспортных объектах 54
4.1 Контроль за состоянием инфраструктуры 54
4.2 Профилактика аварийности 54
Эвакуация и действия при ЧС 55
Заключение 56
Список использованных источников 57
Приложение А – Структура БД 58
Приложение Б – Диаграммы UML, модели, структуры 63
Приложение В – Листинг программы 67
Приложение Г – Презентация 131

Перечень сокращений и обозначений

БД – база данных

ООП – объектно-ориентированное программирование

Патерн – шаблон проектирования

СУБД – система управления базами данных

ЧС – чрезвычайная ситуация

ЭВМ – электронно-вычислительная машина

API (Application Programming Interface) – программный интерфейс приложения

Backend – серверная часть приложения

Frontend – клиентская часть приложения

HTML (Hyper Text Markup Language) – язык гипертекстовой разметки

MVC (Model-View-Controller) – «Модель – Представление-Контроллер» или «Модель Вид – Контроллер»

ORM (Object-Relational Mapping) – объектно-реляционное отображение, или преобразование

REST (Representational State Transfer) – передача репрезентативного состояния или передача «самоописываемого» состояния

SPA (Single Page Application) – приложение, работающее в браузере без перезагрузки страницы

UML (Unified Modeling Language) – унифицирован язык моделирования.

URL (Uniform Resource Locator) – система унифицированных адресов электронных ресурсов.

Введение

Разработка сайта для регистрации РЖД билетов относится к сфере веб – разработки (Web Development). Сайт для регистрации билетов на поезда РЖД представляет собой важный инструмент, который позволяет пользователям быстро и удобно приобретать билеты, избегая очередей и лишних затрат времени.

Цель данной работы заключается в разработке функционального и интуитивно понятного веб-приложения, которое обеспечит пользователей возможностью легко находить и бронировать билеты на поезда, а также управлять своими поездками. В процессе работы будет рассмотрен ряд аспектов, включая проектирование пользовательского интерфейса, реализацию серверной логики, а также интеграцию с платежными системами.

Актуальность темы обусловлена растущим спросом на онлайн – сервисы в сфере транспортных услуг, а также необходимостью повышения качества обслуживания клиентов. В результате реализации данного проекта планируется создать платформу, которая не только упростит процесс покупки билетов, но и улучшит общий пользовательский опыт.

1 Программа для бронирования РЖД билетов

1.1 Предпроектный анализ

1.1.1 Актуальность

Тема "Программа для бронирования РЖД билетов" обладает высокой актуальностью, так как в настоящее время очень много людей пользуется данным вариантом для передвижения между городами.

Развитие цифровых технологий и рост мобильности населения создают высокий спрос на удобные и быстрые способы приобретения железнодорожных билетов. Регистрация на сайте РЖД значительно упрощает процесс планирования поездок, сокращает время покупки билета и исключает необходимость посещения железнодорожных касс или пунктов самообслуживания. В современных условиях, когда важны скорость обслуживания и комфорт, создание специализированного портала для регистрации и покупки билетов становится особенно актуальным: это повышает уровень удовлетворенности пассажиров и способствует более эффективной работе железнодорожной компании.

Кроме того, внедрение онлайн-сервиса укрепляет конкурентоспособность РЖД на рынке перевозок. В условиях растущего спроса на электронные сервисы данный сайт позволит расширить клиентскую базу, повысить лояльность пассажиров и обеспечить быстрый доступ к актуальной информации о расписаниях, ценах и специальных предложениях.

1.1.2 Пользователи и сценарии

Потенциальные пользователи сайта для регистрации и приобретения билетов РЖД делятся на несколько категорий:

– обычные пассажиры – пользователь заходит на сайт, выбирает направление и дату поездки, просматривает доступные рейсы, выбирает подходящий и оформляет покупку, при необходимости регистрируется или входит в личный кабинет для сохранения информации и истории заказов. Целью данного пользователя является быстро и удобно приобрести железнодорожный билет, а также ознакомиться с расписаниями и ценами;

– регулярные клиенты – вход в личный кабинет, быстрый выбор часто используемых маршрутов, просмотр истории покупок, оформление билетов с минимальными усилиями. Целью данного пользователя является автоматизация процесса покупки, использование сохранённых данных, оформление билетов заранее;

– малоподвижные пользователи – вход с мобильного телефона, облегчённый интерфейс, возможность оплаты через мобильные платёжные системы. Целью данного пользователя является возможность использования сервиса даже при ограниченном доступе к компьютерной технике, например, через мобильные устройства.

Основные сценарии использования включают:

– регистрацию нового пользователя;

– вход в существующий аккаунт;

– поиск и подбор маршрутов;

– выбор и оформление билетов;

– получение подтверждения и электронных билетов;

– возможность возврата и изменения заказов.

Такая структура сценариев позволяет обеспечить комфортное и функциональное использование сайта для всех категорий пользователей.

UML диаграмма вариантов использования представлена ниже на (рисунке 1).

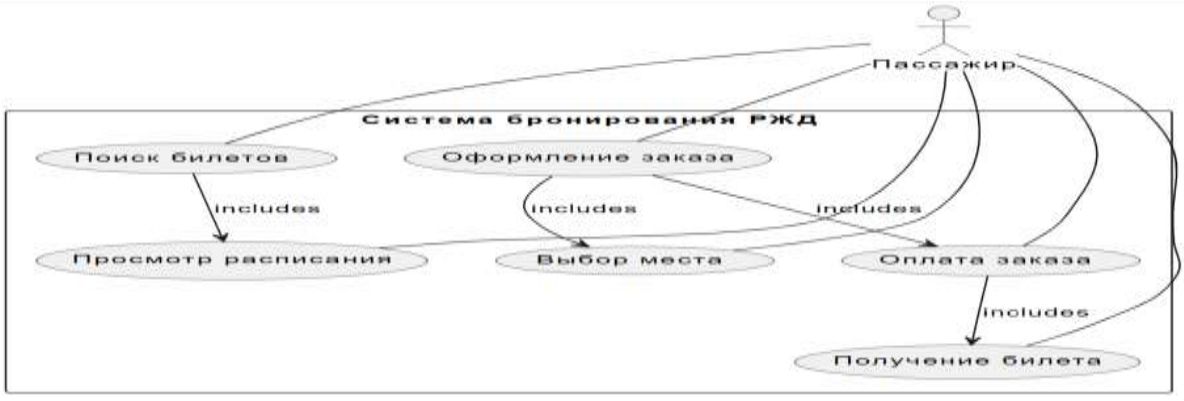

Рисунок 1 – Диаграмма вариантов использования

1.1.3 Анализ аналогов

Для того, чтобы определиться с функционалом разрабатываемой информационной-автоматизированной системы необходимо проанализировать аналоги. Такими системами являются: РЖД, Яндекс. Транспорт, Билет. Ру, Авито. Билеты, Туту. Ру.

Официальный сайт РЖД является самым надежным и удобным ресурсом для бронирования и покупки билетов. На сайте доступны все виды билетов, включая пластиковые, электронные и студенческие. Удобная система бронирования и наличие личного кабинета позволяют легко управлять своими поездками, а возможность оплаты по карте делает процесс покупки более комфортным. Цены на билеты соответствуют тарифам РЖД, что гарантирует актуальность информации. Также стоит отметить доступную и быструю техподдержку, которая помогает пользователям решать возникающие вопросы. Из недостатков можно выделить задержки с обновлением данных и не самый простой интерфейс, который новичку может показаться сложным.

Яндекс. Транспорт — один из самых популярных ресурсов для бронирования и покупки билетов на поезда РЖД. На платформе доступны все виды билетов, включая пластиковые и электронные. Удобная система бронирования, наличие личного кабинета и возможность оплаты по карте делают процесс покупки комфортным и простым. Цены соответствуют тарифам РЖД, и также доступны различные скидки. В дополнение, пользователи могут рассчитывать на доступную и быструю техподдержку, что способствует решению возникающих вопросов. К недостаткам этого сайта можно отнести ограниечение вариантов выбора по сравнению со специализированными сайтами. а также не всегда отображаются все возможные тарифы.

Билет. ру — это ресурс, специализирующийся на продаже билетов на поезда РЖД. На сайте доступны все виды билетов, включая пластиковые и электронные. Удобная система бронирования, наличие личного кабинета и возможность оплаты по карте делают процесс покупки простым и комфортным. Цены соответствуют

тарифам РЖД, и пользователи могут воспользоваться различными скидками. Техподдержка также доступна, что помогает пользователям решать возникающие вопросы. Из минусов можно выделить: комиссии за бронирование, не всегда можно купить билет в электронном виде.

Авито. Билеты — это ресурс, предлагающий услуги бронирования и покупки билетов на поезда РЖД. На сайте доступны все виды билетов, включая пластиковые и электронные. Удобная система бронирования, наличие личного кабинета и возможность оплаты по карте делают процесс покупки комфортным. Однако стоит отметить, что цены могут быть выше, чем на официальном сайте РЖД. Техподдержка также доступна, что помогает пользователям решать возникающие вопросы. Главными недостатками сайта Авито. Билеты являются: присутствие шанса мошенничества, не всегда есть гарантия легальности билета и его актуальности.

Сайт Туту. ру является одним из крупнейших сервисов для бронирования билетов в России и странах СНГ, предлагая пользователям возможность приобретать билеты на поезда РЖД, автобусы, авиарейсы и ж/д экспрессы. Платформа отличается удобным и интуитивно понятным интерфейсом, что делает процесс поиска и сравнения вариантов быстрым и простым. Пользователи могут заранее выбирать места и часто имеют возможность воспользоваться скидками и акциями. Однако у сервиса есть и недостатки: взимается комиссия за бронирование, а также иногда цены могут отличаться от официальных на сайте РЖД. В целом, tutu.ru представляет собой удобный инструмент для путешественников, несмотря на некоторые недостатки, такие как комиссии и возможные расхождения в ценах.

1.1.4 Выбор языка программирования

Существует множество языков программирования, которые могут использоваться для разработки веб-приложений.

Рассмотрим несколько из них и сравним их по некоторым критериям.

Плюсы JavaScript:

– один язык для Frontend и Backend;

– высокая производительность;

– большое сообщество и множество библиотек (npm);

– хорош для реального времени (WebSockets для уведомлений о билетах).

Минусы JavaScript:

– менее эффективен для сложных вычислений (по сравнению с Java /C#);

– требует аккуратной работы с памятью (риск утечек);

– менее строгая типизация (TypeScript решает частично).

Python имеет очень простой синтаксис и хорошо подходит для быстрой разработки прототипов. Он также имеет большое сообщество и множество библиотек и фреймворков для создания веб – приложений.

Плюсы Python:

– быстрая разработка (много встроенных решений в Django);

– хорош для аналитики и AI (если нужно внедрить рекомендации маршрутов);

– простота интеграции с другими сервисами (API, платежи).

Минусы Python:

– медленнее Java/C# (не идеален для высоких нагрузок без оптимизации);

– глобальная блокировка интерпретатора;

– подходит для небольших и средних проектов, но может потребовать масштабирования через микросервисы.

Плюсы Java:

– высокая производительность (JVM оптимизирована для нагрузок);

– многопоточность и масштабируемость (подходит для миллионов запросов);

– надёжность и безопасность (используется в банках и гос. системах);

– лёгкая интеграция с корпоративными системами РЖД.

Минусы Java:

– многословный синтаксис;

– сложнее в освоении (требует глубоких знаний Java);

– долгая разработка по сравнению с Python/Node.js;

– идеален для крупных и высоконагруженных систем (как сайт РЖД).

Плюсы C#:

– высокая производительность (близок к Java);

– хорошая интеграция с Microsoft-сервисами (если РЖД использует Azure);

– безопасность и стабильность (подходит для корпоративных решений);

Минусы C#:

– меньше open-source решений, чем у Java/Python;

– привязка к экосистеме Microsoft (Core кроссплатформенный);

– хороший выбор, если в компании уже используется стек Microsoft.

Плюсы PHP:

– простота развёртывания (хостинги поддерживают PHP "из коробки");

– быстрая разработка (Laravel имеет удобный ORM и шаблоны).

Минусы PHP:

– менее производительный, чем Java/C#/Node.js;

– подходит для небольших проектов.

Каждый из этих языков программирования имеет свои плюсы и минусы, и выбор зависит от конкретных потребностей проекта. Java – оптимальный выбор для крупных и высоконагруженных проектов, C# - для высокой производительности, а также стабильной и безопасной работы Python – для быстрой разработки прототипов, Java Script – подходит как для Frontend, так и для Backend, PHP – для веб-сайтов, которые работают с базами данных. В любом случае выбор языка зависит от требований конкретного проекта и уровня квалификации разработчиков.

1.1.5 Выбор фреймворка для серверной части приложения

Существует множество фреймворков для PHP, каждый со своими особенностями и преимуществами. Рассмотрим некоторые из них:

– jQuery – это легкая и кроссбраузерная JavaScript–библиотека, которая упрощает манипуляцию DOM, обработку событий и анимацию с помощью обширной экосистемы плагинов;

– React – это библиотека, разработанная Facebook, использующая компонентный подход, виртуальный DOM для повышения производительности и односторонний поток данных для создания современных пользовательских интерфейсов. Он предлагает множество инструментов для создания масштабируемых приложений, включая контейнеры зависимостей, компоненты, события и др;

– Angular – это мощный фреймворк, разработанный Google, обеспечивающий двухстороннюю привязку данных, инъекцию зависимостей и расширенную систему маршрутизации для создания полноценных веб-приложений;

– Yii – это еще один высокопроизводительный фреймворк, написанный на PHP. Он поддерживает шаблон проектирования MVC и имеет специальные инструменты для обработки форм, работы с базами данных и аутентификации. Он также предоставляет возможность использовать компоненты от других разработчиков, что делает его серьезным конкурентом других фреймворков.

В итоге выбор фреймворка определяется уровнем опыта разработчика и требованиями конкретного проекта, при этом все перечисленные инструменты предлагают широкий функционал и возможности для быстрой и безопасной разработки приложений. Есть несколько причин, по которым был выбран фреймворк jQuery.

jQuery – имеет интуитивно понятный и лаконичный синтаксис, что облегчает изучение и использование, а также автоматически обрабатывает различия между браузерами, что позволяет избежать проблем с совместимостью.

1.1.6 Хранение данных

phpMyAdmin – это популярное веб-приложение, разработанное на PHP, предназначенное для управления базами данных MySQL и MariaDB через удобный графический интерфейс. Оно позволяет пользователям выполнять широкий спектр операций с базами данных без необходимости использовать командную строку, логотип phpMyAdmin изображен ниже, на рисунке 2.

Рисунок - 2

Ключевые возможности phpMyAdmin включают создание, удаление и редактирование баз данных; работу с таблицами (создание, изменение, удаление); выполнение произвольных SQL – запросов, импорт и экспорт данных в различных форматах, таких как SQL, CSV, XML и другие, управление пользователями и их правами доступа, поддержку множества языков интерфейса, а также создание резервных копий баз данных.

Плюсы phpMyAdmin состоят в его простоте использования, что делает управление базами данных доступным даже для новичков, в богатом функционале, необходимом для работы с различными аспектами баз данных, а

также в наличии обширного сообщества пользователей и разработчиков, обеспечивающего поддержку и документацию.

Минусы включают вопросы безопасности — при неправильной настройке приложение может стать уязвимым для атак — и возможное снижение производительности при работе с очень крупными базами данных.

1.1.7 Выбор СУБД

На сегодняшний день существует множество типов баз данных и систем управления ими, которые можно разделить на два основных класса: реляционные (SQL) и нереляционные (NoSQL). Главное отличие между ними заключается в способе представления данных.

В нереляционных базах данных информация обычно хранится в виде словарей, где ключом является первичный ключ сущности, а значением — все остальные поля таблицы. Этот подход позволяет избежать строгой типизации и фиксированного формата данных. Кроме того, NoSQL базы данных могут эффективно управлять иерархическими структурами данных, тогда как в SQL-базах для этого часто требуется создание дополнительных таблиц. Наиболее популярные NoSQL СУБД включают MongoDB, Redis.

Реляционные базы данных, также известные как табличные, делятся на два типа: строковые и колоночные. Основное различие между ними заключается в способе хранения данных.

В колоночных базах данных данные каждой колонки хранятся отдельно, что позволяет эффективно сжимать информацию и отлично подходит для аналитических задач и ведения журналов. Примером может служить Amazon Redshift, хранилище данных, позволяющее анализировать большие объёмы данных с помощью стандартного SQL.

В строковых базах данных данные сохраняются по строкам, что увеличивает время обработки и выборки, но обеспечивает быстрый доступ к отдельным записям для чтения, изменения или удаления, а некоторые из них даже

превосходят по эффективности решения NoSQL. Ниже перечислены наиболее популярные СУБД и краткая информация о них:

– MySQL – Это одна из самых популярных открытых реляционных СУБД, широко используемая в веб-разработке и приложениях. Обладает высокой скоростью работы, надежностью и поддержкой транзакций. Подходит для небольших и средних проектов, а также для масштабируемых решений. Преимущества включают высокую скорость обработки простых запросов, простоту настройки и администрирования, широкую поддержку со стороны хостинг-провайдеров, хорошую документацию и активное сообщество, а также встроенную поддержку репликации. Из недостатков можно выделить высокую стоимость, сложность в использовании и наличие лицензионных ограничений;

– СУБД PostgreSQL – Это мощная и расширяемая реляционная база данных с открытым исходным кодом. Отличается поддержкой сложных запросов, расширяемыми типами данных и стабильностью. Предпочитается для систем, требующих высокой надежности и обработки сложных данных. К преимуществам можно отнести то, что данная СУБД обладает полной поддержкой стандартов SQL, расширяемостью, отличной работой с JSON, поддержкой оконных функций, CTE и сложных JOIN, а также надежной системой транзакций (ACID), к недостаткам можно отнести высокое требование ресурсов и сложнсть в настройке для новичков , а также меньшее наличие готовых SaaS-решений по сравнению с MySQL.

1.1.8 Выбор платформы и веб-сервера

В качестве платформы использовался PhpMyAdmin, в качестве сервера использовался локальный OpenServer.

Для реализации поставленных задач в рамках дипломного проекта была выбрана платформа phpMyAdmin, являющаяся очень удобным и популярным веб – приложением для управления базами данных на основе MySQL. Данная платформа обеспечивает интуитивно понятный интерфейс для выполнения различных операций, таких как создание, изменение и удаление баз данных и таблиц, выполнение SQL – запросов, импорт и экспорт данных, а также

управление правами пользователей. Благодаря этому, использование phpMyAdmin значительно ускоряет процесс разработки и облегчает администрирование базы данных, что особенно важно при проведении тестирования и отладки.

В качестве веб-сервера для локальной разработки был использован программный комплекс OpenServer. Это легкий и многофункциональный серверный стек, предназначенный для быстрого запуска и тестирования веб – приложений на локальной машине. OpenServer включает в себя все необходимые компоненты для работы веб – сервера: Apache или Nginx, PHP, MySQL/MariaDB, а также инструменты для управления этими компонентами.

Одним из главных преимуществ OpenServer является его простота в использовании. Пользователю не требуется глубоких знаний в области настройки серверных компонентов: установка и запуск происходят в несколько кликов, а интерфейс управления позволяет легко переключать версии PHP, настройка параметров серверов, а также управление базами данных дается очень просто. Это делает OpenServer особенно удобным для студентов и начинающих разработчиков.

Кроме того, OpenServer поддерживает автоматическую настройку рабочего окружения, что позволяет быстро запускать несколько проектов одновременно, управлять доменными именами и портами, а также выполнять резервное копирование и восстановление данных. Встроенные инструменты позволяют контролировать работу серверных служб, просматривать логи и анализировать работу приложений.

Использование OpenServer обеспечивает стабильную работу локальной среды, что важно при разработке и тестировании проекта. Возможность быстро настраивать окружение и легко вносить изменения значительно повышает эффективность работы. Также, благодаря популярности и широкой поддержке, OpenServer хорошо интегрируется с другими инструментами разработки, такими как phpMyAdmin, что делает его оптимальным выбором для учебных целей и профессиональной работы.

Таким образом, выбранная платформа phpMyAdmin в совокупности с серверным комплексом OpenServer обеспечивают удобную, стабильную и функциональную среду для разработки, тестирования и демонстрации работы базы данных и связанного с ней программного обеспечения.

1.1.9 Выбор библиотеки для реализации графического интерфейса

Для создания удобного и привлекательного пользовательского интерфейса в рамках проекта был проведён анализ доступных библиотек и фреймворков. Главным требованием являлась возможность адаптивности под различные устройства интеграция с сервером. В результате были выбраны следующие библиотеки:

– Bootstrap — это популярный фронтенд – фреймворк, разработанный компанией Twitter, предназначенный для быстрого создания адаптивных и современно выглядящих веб-интерфейсов. Он включает в себя готовые компоненты, такие как формы, кнопки, таблицы, навигационные панели и модальные окна, а также систему сеток для адаптивного расположения элементов. Использование Bootstrap позволяет разработчику быстро создавать пользовательский интерфейс, который одинаково хорошо отображается на различных устройствах — от мобильных телефонов до больших мониторов. Благодаря обширной документации и активному сообществу, интеграция Bootstrap в проект является простой и эффективной;

– Material UI – это библиотека компонентов, реализующая дизайн – гайдлайны Material Design от Google. Она предоставляет современный и эстетичный внешний вид, а также богатый набор элементов пользовательского интерфейса, таких как карточки, диалоговые окна, меню и кнопки с анимациями. Material UI хорошо подходит для проектов, требующих современного и единообразного визуального стиля, а также обеспечивает хорошую совместимость с React и другими популярными фреймворками;

– Ant Design – это фреймворк компонентов, разработанный компанией Alibaba, ориентированный на создание корпоративных и административных

интерфейсов. Он включает в себя широкий набор элементов, таких как таблицы, формы, навигация и уведомления, а также обеспечивает высокую степень настройки внешнего вида. Благодаря своей модульной структуре и богатству компонентов Ant Design позволяет создавать сложные и функциональные интерфейсы с минимальными затратами времени.

Исходя из всего, был выбран Bootstrap. Bootstrap отлично подходит для построения адаптивных интерфейсов, которые одинаково хорошо выглядят на различных устройствах – от смартфонов до больших мониторов – благодаря системе сеток, которая облегчит работу с различными разрешениями и платформами. Это особенно важно в современном мире, где пользователи используют множество разных устройств.

Еще одним большим плюсом является обширная документация, богатая примерами и обучающими материалами, а также активное сообщество разработчиков, обеспечивающее быструю поддержку и помощь при возникновении вопросов. Такой уровень поддержки делает работу с фреймворком более комфортной и понятной даже для новичков. Все эти особенности делают Bootstrap одним из лучших вариантов для быстрого и эффективного построения простых и главное функциональных интерфейсов.

1.1.10 Функциональные требования

Требования к проекту на основе пользовательских сценариев, разработанных ранее:

– главные цели проекта:

1) обеспечение удобного поиска и бронирования билетов;

2) иметь возможность просмотра расписания и доступных маршрутов;

3) предоставить возможность покупки билетов онлайн;

4) обеспечить простое управление личным кабинетом пользователя;

– основные функции (функциональные требования):

1) регистрация и авторизация пользователей;

2) поиск маршрутов по дате и времени;

3) просмотр расписаний и наличия билетов;

4) бронирование билетов;

5) создание и управление личным кабинетом (просмотр заказов, история покупок);

6) отмена и возврат билетов;

– важные нюансы:

1) обеспечение безопасности данных;

2) обеспечение высокой доступности;

3) поддержка мобильных устройств.

1.1.11 Выводы по предпроектному анализу:

– показана актуальность выбранной темы;

– описаны функциональные требования к разрабатываемому приложению;

– проведен анализ и выбраны инструменты для разработки.

1.2 Проектирование и разработка

Сайт состоит из OpenServer сервера, клиентского и серверного приложения. Сервер собирает, обрабатывает и хранит информацию, интерпретатором выступает PHP, а веб сервером является Apache. Клиентами могут быть браузер, Firefox/Chrome, а также desktop – приложения.

На рисунке 3 представлена UML диаграмма развертывания, которая демонстрирует рабочую конфигурацию проекта.

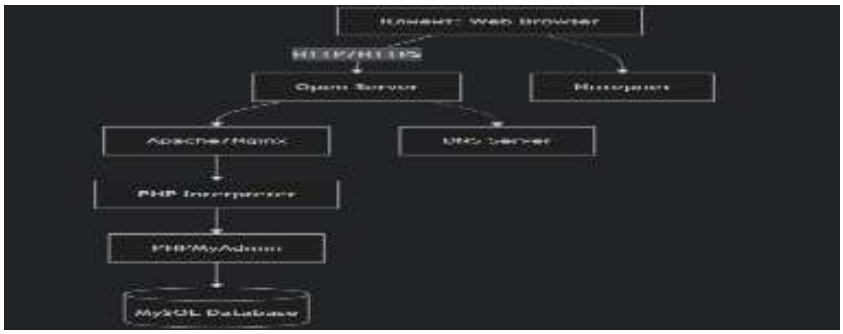

Рисунок 3 – UML диаграмма развертывания

1.2.1 Разработка серверной части приложения

Backend приложение разрабатывается на фреймворке Bootstrap. Был создан проект командой «composer create – project Bootstrap backend». Было создано базовое приложение с минимальным функционалом. Для проверки можно было перейти по следующему URL «https://getindex.ru/back/». В данный момент ссылка защищена. Базовая страница приведена на рисунке 4.

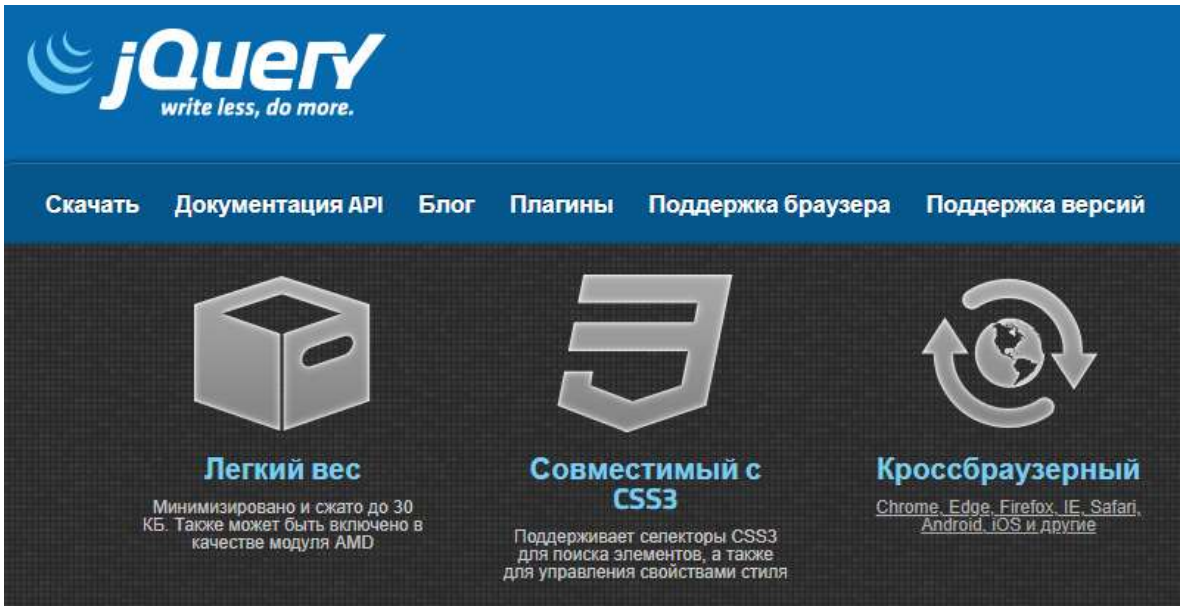

Рисунок 4 – Стартовая страница jQuery

После этого была добавлена библиотека jQuery – это быстрая, лёгкая и многофункциональная JavaScript – библиотека.

Все настройки приложения задаются в конфигурационном файле «.env».

Ниже приведена часть файла .env:

```
define('DB_HOST', 'localhost');
define('DB_USER', 'train_tickets');
define('DB_PASS', 'your_password');
define('DB_NAME', 'train_tickets');
```

Входной точкой в backend приложение являются запросы от frontend приложения. Эти запросы инициируют взаимодействие между двумя частями архитектуры, и их правильная обработка критически важна для функционирования всей системы. Запросы обрабатываются механизмом маршрутизации jQuery.

Маршруты для API определяются в файле routes.php. Этот файл служит своего рода картой, которая связывает URL – адреса входящих запросов с соответствующими функциями или методами обработки на сервере, а также для определения всех маршрутов, относящихся к API. Для определения маршрута применяется метод Route, с указанием нужного HTTP – метода (например, get, post, put, delete) и маршрута. Например, вызов Route::get('user', [AuthController::class, 'user']) создает маршрут, который отвечает на GET-запросы по пути /user и передает их в метод user контроллера AuthController.

Для взаимодействия с API в jQuery используют методы $.ajax(), $.get(), $.post(). Они позволяют отправлять запросы на сервер, получать и обрабатывать ответы асинхронно, при этом обновляя DOM в соответствии с полученными данными. AJAX – это технология для асинхронных запросов к серверу без перезагрузки страницы. В jQuery есть встроенные методы для выполнения AJAX – запросов:

– $.ajax() – универсальный метод, позволяющий настроить любой тип запроса;

– $.get() – для GET – запросов (отправляет асинхронный HTTP GET – запрос на указанный URL. Используется для получения данных с сервера;

– $.post() – для POST – запросов. Используется для отправки данных на сервер.

Пример AJAX – запроса можно увидеть ниже:

```
$.ajax({
  url: '/api/endpoint',
  method: 'PUT',
  data: { key: 'value' },
  success: function(response) {
    console.log('Обновлено');
  }
});
```

CRUD (Create, Read, Update, Delete) – это основные операции для работы с данными. В AJAX их реализуют с помощью различных HTTP – методов при

взаимодействии с API или сервером. С помощью них можно выполнять дополнительные задачи, например аутентификация пользователей или выполнение проверки прав доступа с обработкой ошибок.

DOM (Document Object Model) в jQuery – структура HTML-документа, с которой можно работать через библиотеку, которая предоставляет собой методы для доступа, модификации и создания элементов, а также для обработки событий.

В jQuery для доступа к DOM – элементам используется функция $(). Ей передают селектор, который определяет, какие элементы нужно выбрать. Например, чтобы выбрать все элементы <p> в документе, пишут: $('p').

На рисунке 5 можно просмотреть различные запросы, их маршрутизацию, и какие контроллеры и модели взаимодействуют.

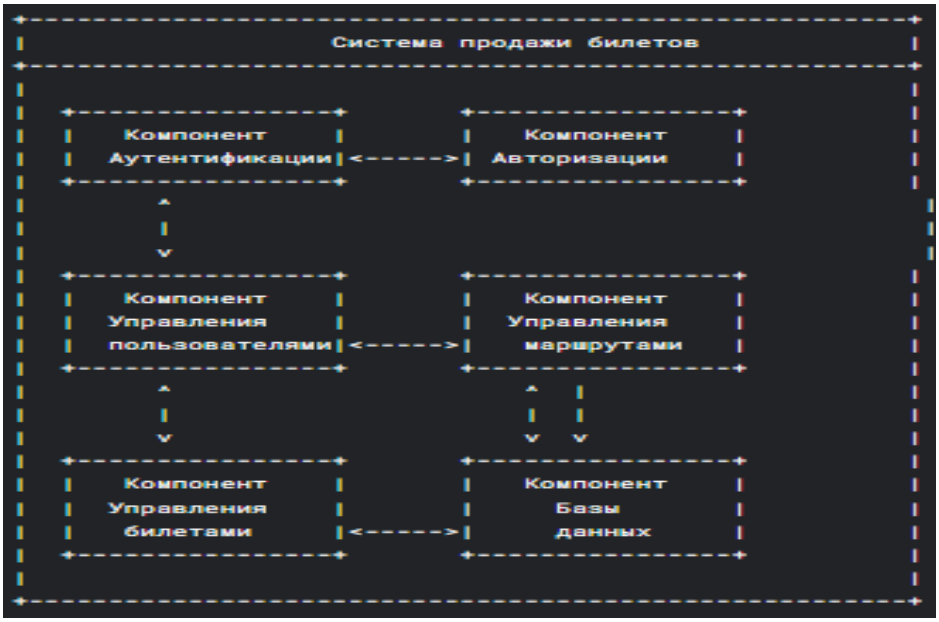

Рисунок 5 – UML диаграмма компонентов

Компонент аутентификации:

– отвечает за вход пользователей в систему;

– проверяет учетные данные;

– взаимодействует с компонентом авторизации.

Компонент авторизации:

– проверяет права доступа (например, requireAdmin());

– управляет ролями пользователей;

– взаимодействует с компонентом управления пользователями.

Компонент Управления пользователями:

– crud операции с пользователями;

– взаимодействует с базой данных.

Компонент Управления маршрутами:

– добавление/редактирование/удаление маршрутов;

– валидация данных маршрутов;

– проверка связанных билетов перед удалением;

– взаимодействует с компонентом билетов и БД.

Компонент Управления билетами:

– обработка операций с билетами;

– проверка доступности мест;

– взаимодействует с компонентом маршрутов.

Компонент Базы данных:

– хранение данных о пользователях, маршрутах и билетах;

– обеспечение транзакций;

– предоставление данных другим компонентам;

– дополнительные элементы:

– интерфейс администратора;

– система flash – сообщений для уведомлений;

– механизм перенаправлений (redirect).

Связи между компонентами показывают основные взаимодействия и потоки данных в системе.

1.2.2 Разработка клиентской части приложения

Разработка нашей клиентской части началась с подготовки окружения. Чтобы не было никаких проблем в дальнейшем, мы устанавливаем последнюю версию Node.js и создан новый проект.

Далее мы подключаем Bootstrap к нашему проекту через CDN (Content Delivery Network). Чтобы это сделать, нужно вставить в HTML – файл ссылки на необходимые файлы CSS и JS, предоставляемые Bootstrap. Выглядит это следующим образом:

```
<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<title>Покупка билетов РЖД</title>
</head>
<body>
<!-- Контент страницы
</body>
</html>
```

Для обработки событий мы используем JavaScript и jQuery с использованием AJAX – запросов для получаения информации о поездах.

AJAX – это технология, которая позволяет обновлять данные на веб-странице без полной перезагрузки. Это особенно полезное приспособление, так как при помощи данной технологии мы сможем получаеть данные о поездах в реальном времени. Ниже представлен код демонстрирующий обработку событий:

```
$(document).ready(function() {
 $('form').on('submit', function(event) {
 event.preventDefault();
 let departure = $('#departure').val();
 let arrival = $('#arrival').val();
 let date = $('#date').val();

 $.ajax({
```

```
url: 'api/search', // URL вашего API
method: 'GET',
data: {
departure: departure,
arrival: arrival,
date: date
},
success: function(data) {
// Обработка полученных данных и отображение результатов
},
error: function() {
alert('Произошла ошибка при поиске поездов.');
}
});
});
```

Для реализации данной технологии необходимо пройти через несколько этапов, первым из которых является создание объекта XMLHttpRequest, который необходим для создания HTTP – запросов к серверу. При создании нового экземпляра этого объекта, браузер подготавливает все необходимое для работы с запросами. Следующим этапом является настройка запроса (xhr.open('GET', '/api/data', true)) в котором метод open() три параметра: метод запроса(GET, POST, PUT, DELETE), URL (куда отправляется запрос), асинхронность (true, false).

Следующим этапом была реализована маршрутизация. Маршрутизация – это процесс обработки URL и определения компонента, отображаемого на странице.

Ниже часть кода, который показывает, как реализована маршрутизация в проекте:

```
// script.js
function loadPage(page) {
 const app = document.getElementById('app');
 switch (page) {
 case 'home':
 app.innerHTML = '<h1>Главная страница</h1><p>Добро пожаловать на
```

```
наш сайт!</p>';
 break;
 case 'about':
 app.innerHTML = '<h1>О нас</h1><p>Мы - команда разработчиков.</p>';
 break;
 case 'contact':
 app.innerHTML = '<h1>Контакты</h1><p>Свяжитесь с нами по
электронной почте.</p>';
 break;
 default:
 app.innerHTML = '<h1>404</h1><p>Страница не найдена.</p>';
 }
```

Функция loadPage отвечает за отображение содержимого на странице. Она принимает параметр page, который определяет, какую страницу нужно загрузить. Компонент Route используется для указания компонентов, которые должны отображаться при переходе по определенным маршрутам.

Во время отправки формы вызывается метод submit, в котором идет обращение к API backend приложения с помощью await fetch, если ответ пришел, то с помощью useDispatch происходит установка состояния для хранения введенного URL – адреса.

```
const response = await fetch
('https://getindex.ru/back/api/V1/site', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({
                url             })
});
```

Далее происходит перезагрузка страницы с перенаправлением на компонент страницы анализа. Из сохраненного состояния вытягивается введенный URL – адрес, и дальше начинается установка всех состояний компонентов, доставая значения из json пришедшего от backend приложения, на рисунке 6 можно увидеть структуру клиентской части.

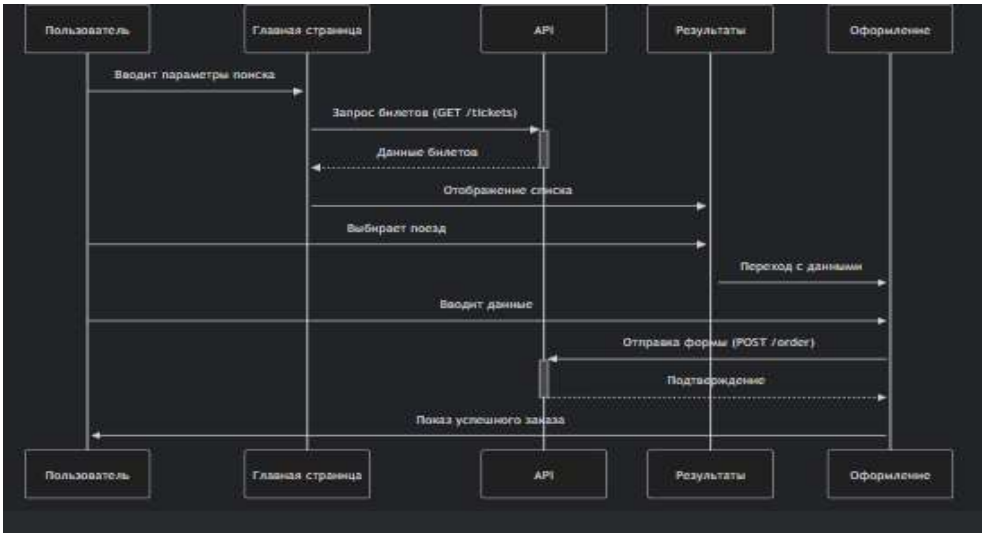

Рисунок 6 – Структура клиентской части приложения

1.3 Проектирование базы данных

База данных реализована в Mermaid и состоит из трех нормализованных таблиц, связанных через внешние ключи, а схему базы данных можно увидеть на рисунке 7. Структура таблиц обеспечивает:

– хранение информации о пользователях;

– получение информации о маршрутах;

– получение информации о билетах.

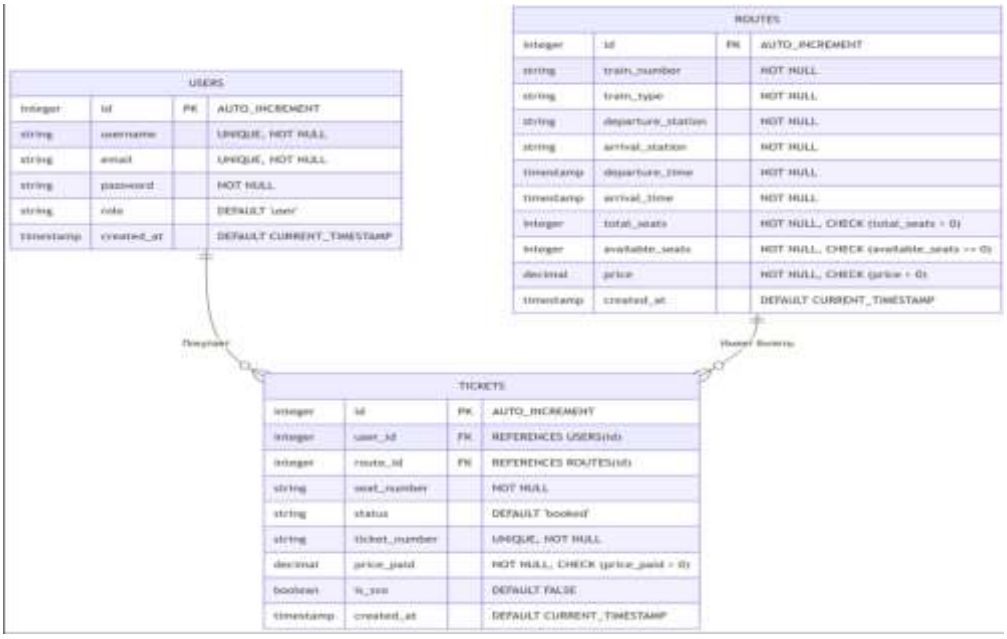

Рисунок 7 – Схема базы данных

Рассмотрим сущности: users, tickets, routes, показанные в таблицах 1, 2, 3

Таблица 1 – Сущность users (пользователи)

Поле	Тип данных	Ограничения	Описание
Id	INTEGER	PRIMARY KEY, AUTO_INCREMENT	Уникальный идентификатор
Username	STRING	UNIQUE, NOT_NULL	Имя пользователя
Email	STRING	UNIQUE, NOT_NULL	Почта пользователя
Password	STRING	NOT_NULL	Пароль
Role	STRING	DEFAULT ‘USER’	Роль
Created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Дата создания

Таблица 2 – Сущность tickets (билеты)

Поле	Тип данных	Ограничения	Описание
Id	INTEGER	PRIMARY KEY, AUTO_INCREMENT	Уникальный идентификатор
User_id	INTEGER	REFERENCES_USERS(ID)	идентификатор пользователя
Route_id	INTEGER	REFERENCES_ROUTES(ID) NOT_NULL	идентификатор маршрута

Поле	Тип данных	Ограничения	Описание
Seat_number	STRING	NOT_NULL	Номер места
Status	STRING	DEFAULT“BOOKED”	Статус
Ticket_number	STRING	UNIQUE,NOT_NULL	Номер билета
Price_paid	DECIMAL	NOT_NULL	Цена билета
Is_svo	BOOLEAN	DEFAULT_FALSE	Участник “специальной военной операции”
Created_at	TIMESTAMP	DEFAULT_CURRENT_TIMESTAMP	Дата создания

Таблица 3 – Сущность routes (маршруты)

Поле	Тип данных	Ограничения	Описание
Id	INTEGER	PRIMARY KEY, AUTO_INCREMENT	Уникальный идентификатор
Train_number	STRING	NOT_NULL	Номер поезда
Train_type	STRING	NOT_NULL	Тип поезда
Departure_station	STRING	NOT_NULL	Город отправления
Arrival_station	STRING	NOT_NULL	Город прибытия
Departure_time	TIMESTAMP	NOT_NULL	Время отбытия

Поле	Тип данных	Ограничения	Описание
			поезда
Arrival_time	TIMESTAMP	NOT_NULL	Время прибытия поезда
Total_seats	INTEGER	NOT_NULL	Общее количество мест
Available_seats	INTEGER	NOT_NULL	Количество свободных мест
Price	DECIMAL	NOT_NULL	Цена
Created_at	TIMESTAMP	NOT_NULL	Дата создания

2 Результаты испытаний

Окно регистрации можно увидеть ниже, на рисунке А.1.

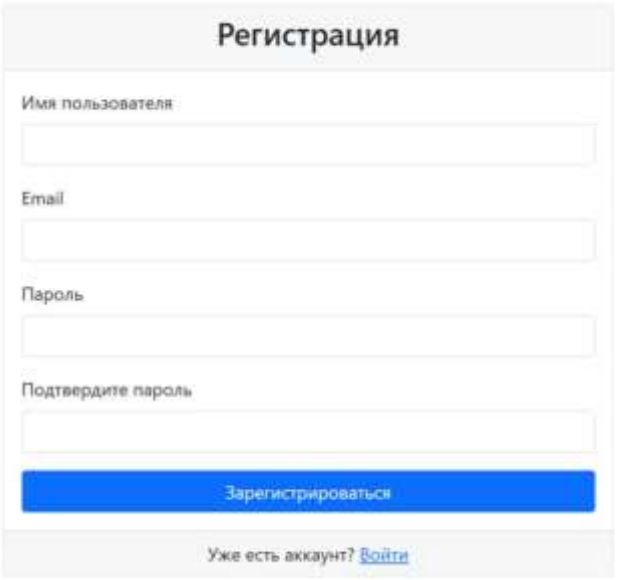

Рисунок А.1 – Окно регистрации

Регистрацию пользователя можно увидеть ниже, на рисунке А.2.

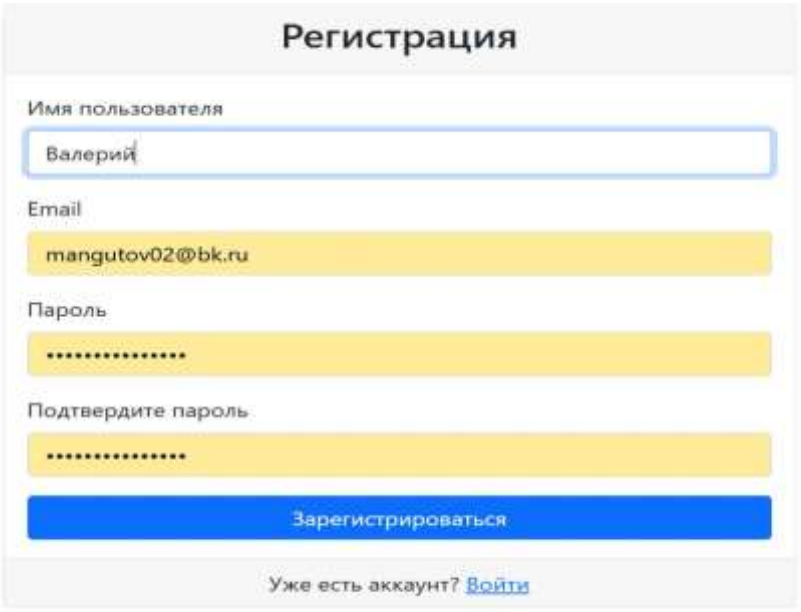

Рисунок А.2 – Регистрация пользователя

На рисунке А.3 можно увидеть успешный результат регистрации.

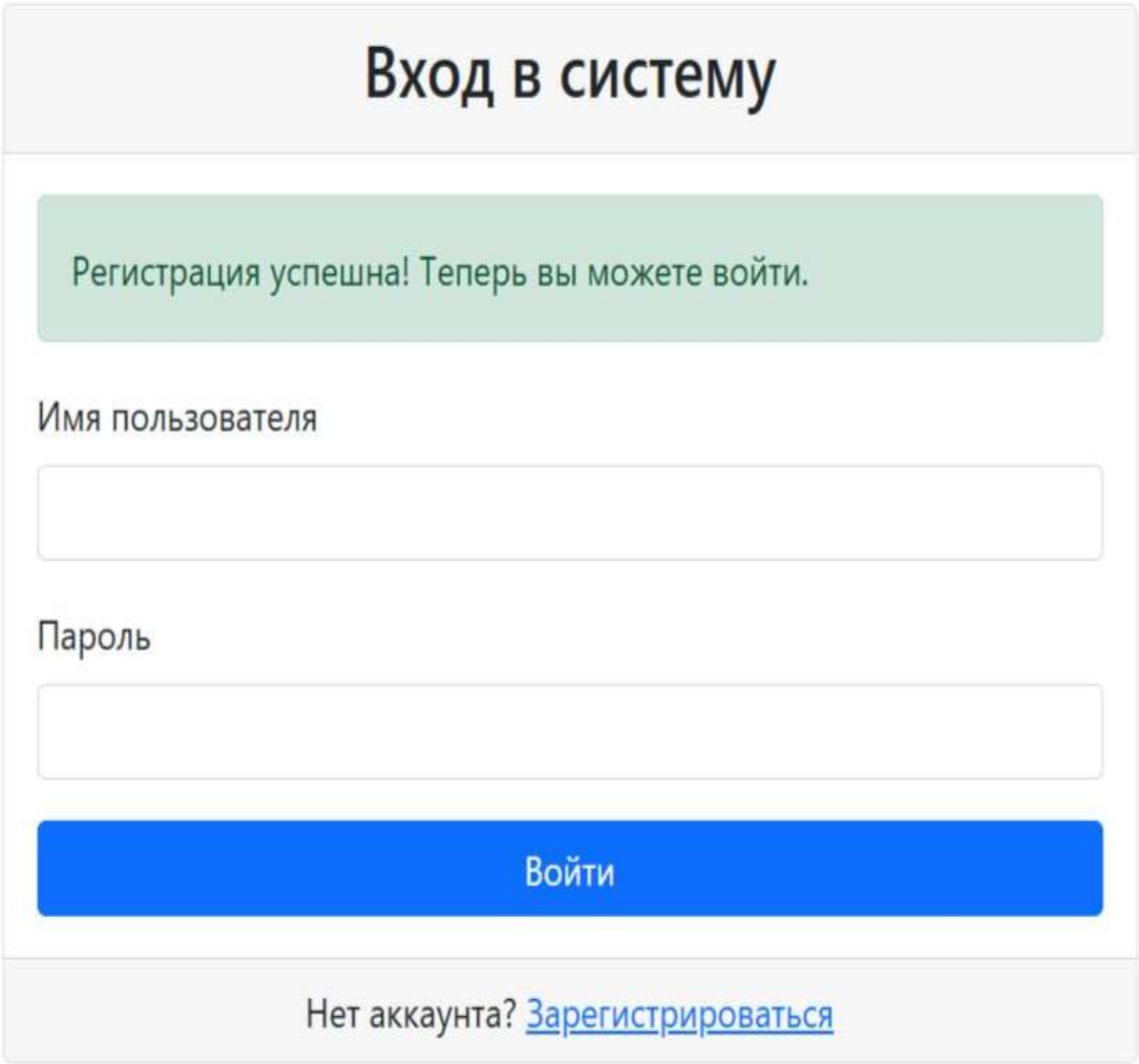

Рисунок А.3 – Успешная регистрация

Заполнение полей можно увидеть ниже, на рисунке А.4.

.

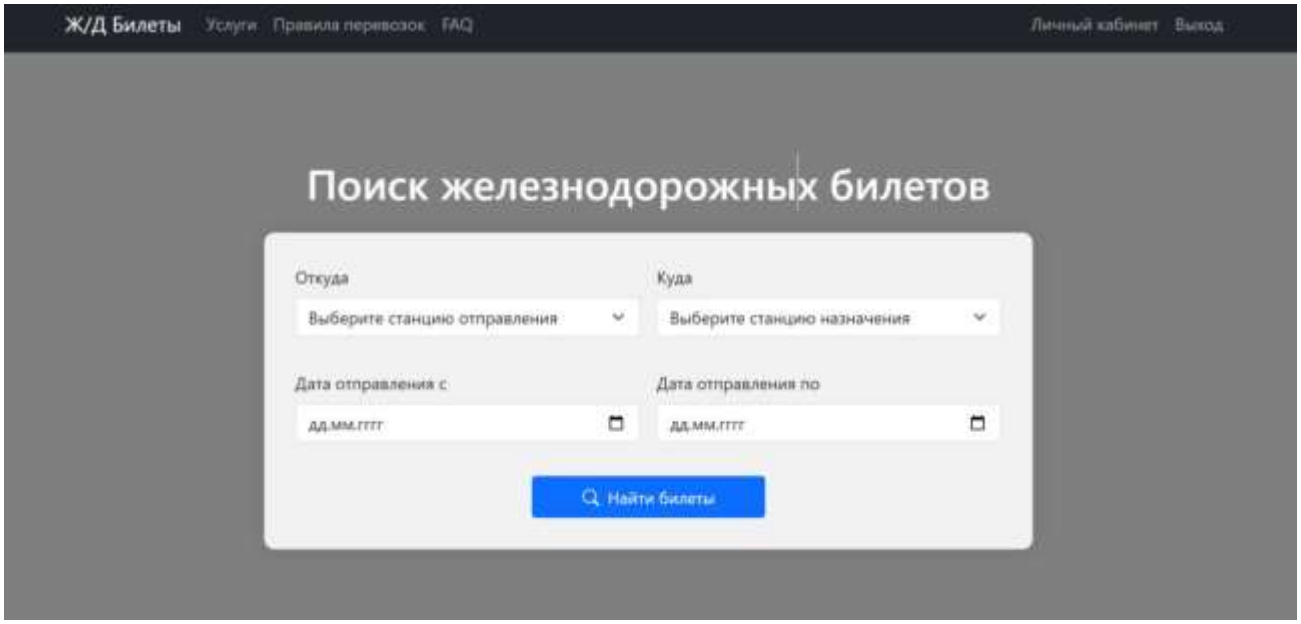

Рисунок А.4 – Заполнение полей

Перечень городов можно увидеть ниже, на рисунке А.5.

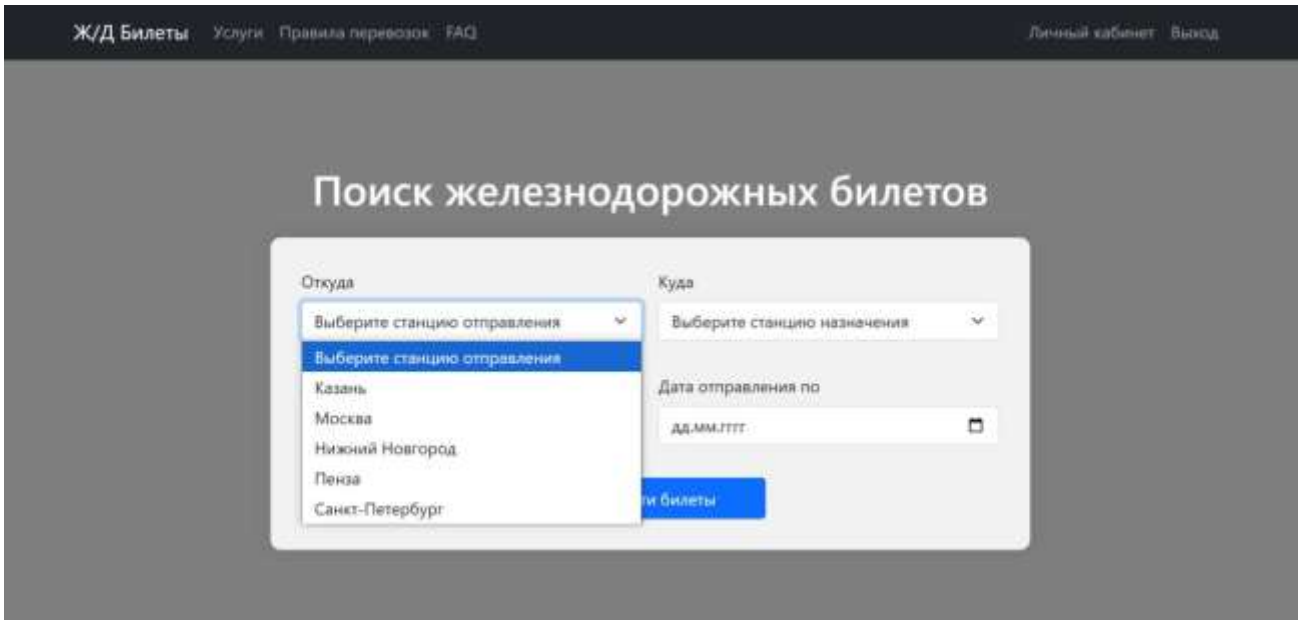

Рисунок А.5 – Выборка городов

Окно выбора даты можно увидеть ниже, на рисунке А.6.

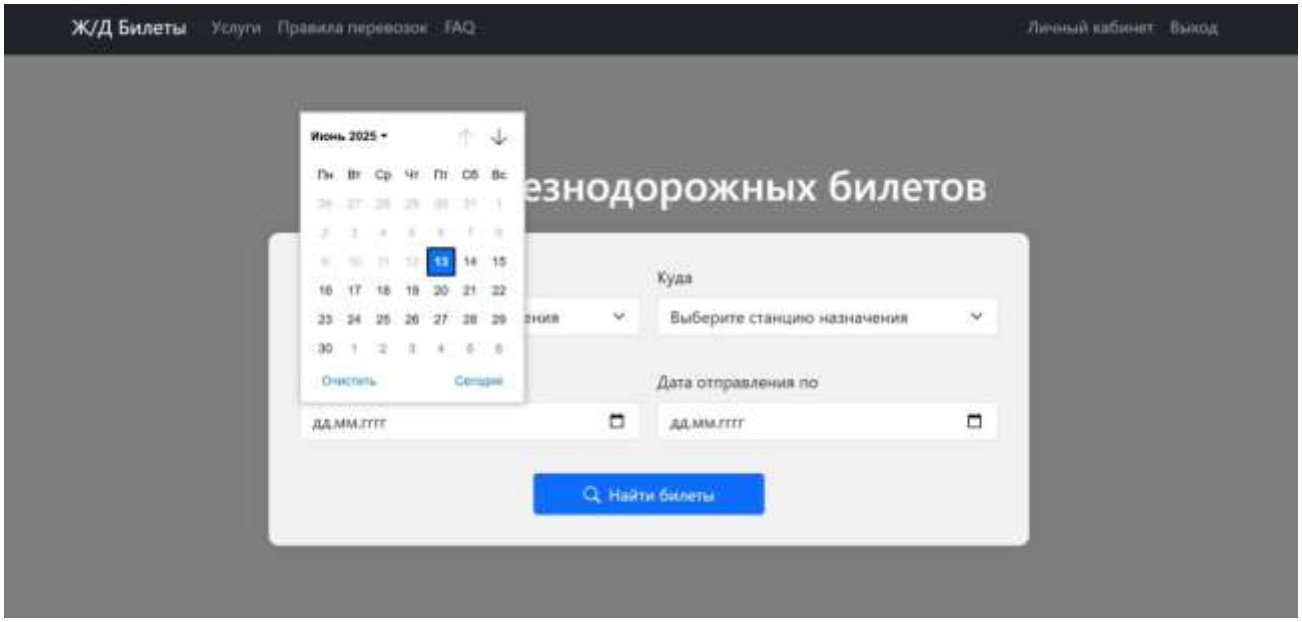

Рисунок А.6 – Выбор даты

Обязательное заполнение полей можно увидеть ниже, на рисунке А.7.

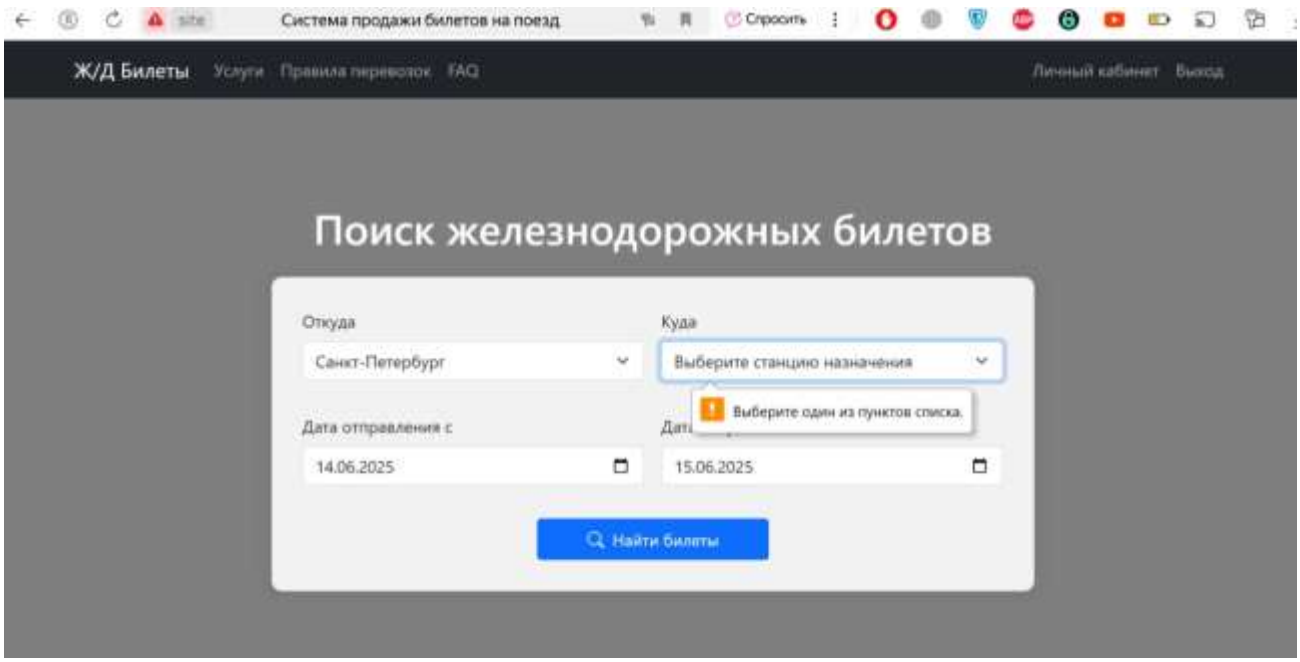

Рисунок А.7 – Обязательный выбор города и даты

Результаты выбора городов и даты вылета, прилета, можно увидеть на рисунке А.8.

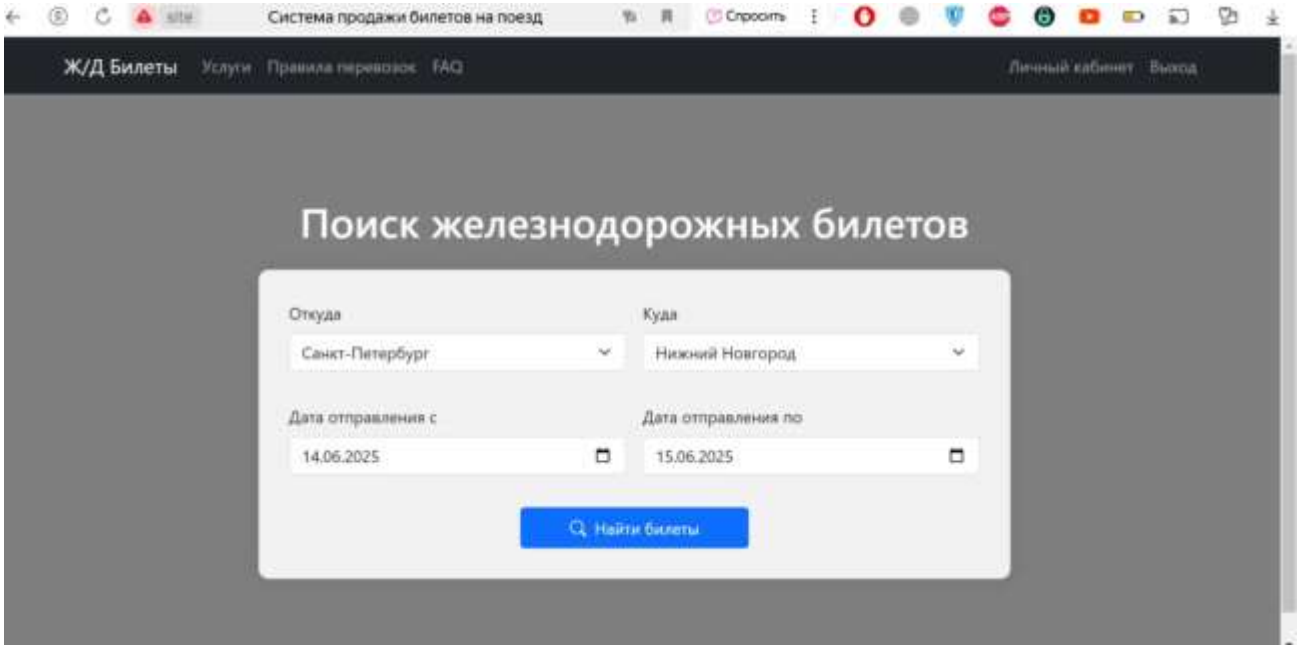

Рисунок А.8 – Выбраны города и даты вылета и прилета

Результаты поиска выбранного рейса можно увидеть ниже, на рисунке А.9.

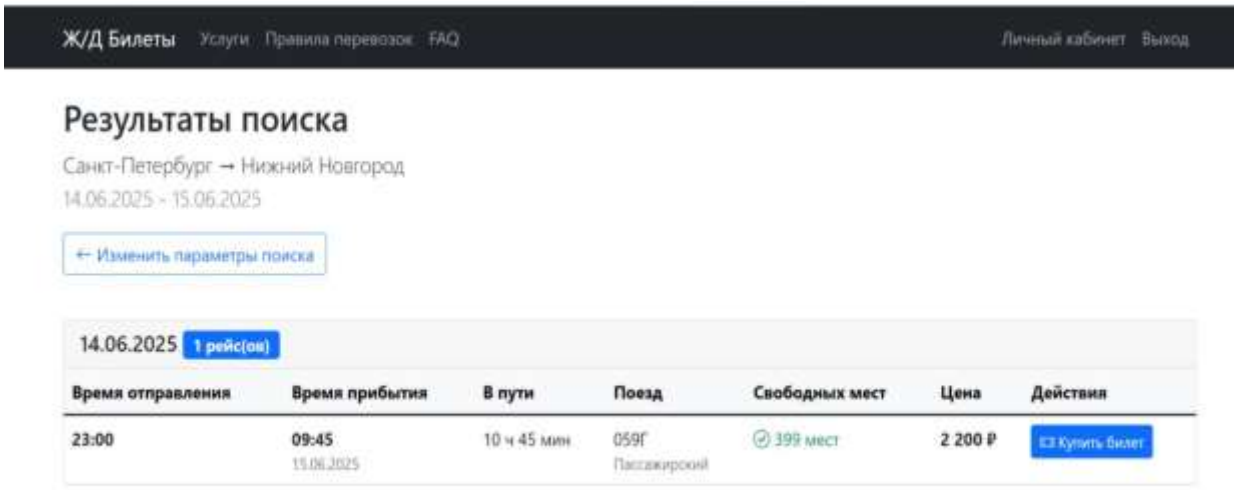

Рисунок А.9 – Результаты поиска

Результат покупки билета можно увидеть ниже, на рисунке А.10.

Рисунок А.10 – Покупка билета

Окно регистрации можно увидеть ниже, на рисунке А.11.

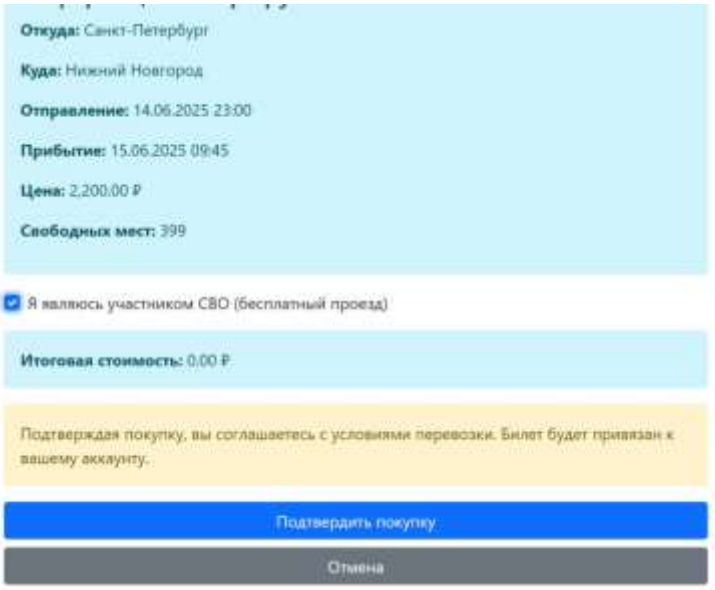

Рисунок А.11 – Специальные условия для участников СВО

Результат покупки можно увидеть ниже, на рисунке А.12.

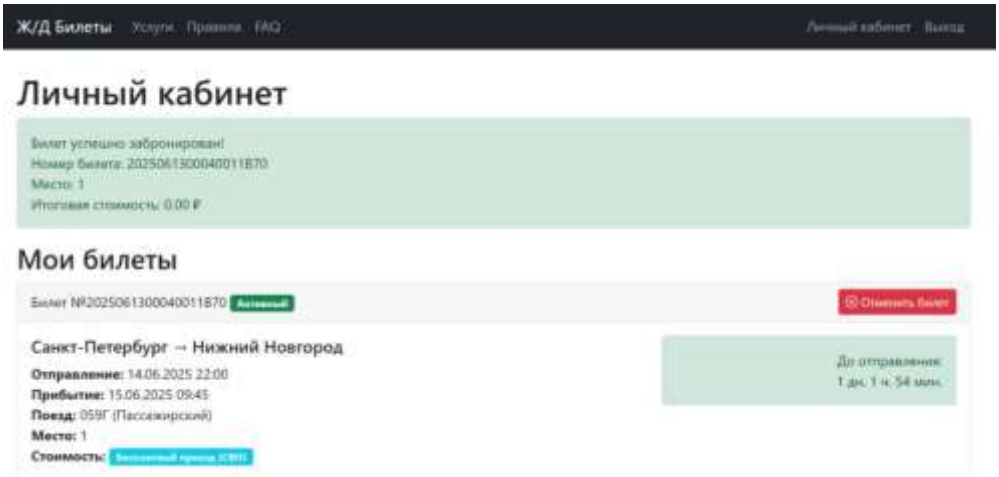

Рисунок А.12 – Успешная покупка билета

Отмену покупки билета можно увидеть ниже, на рисунке А.13.

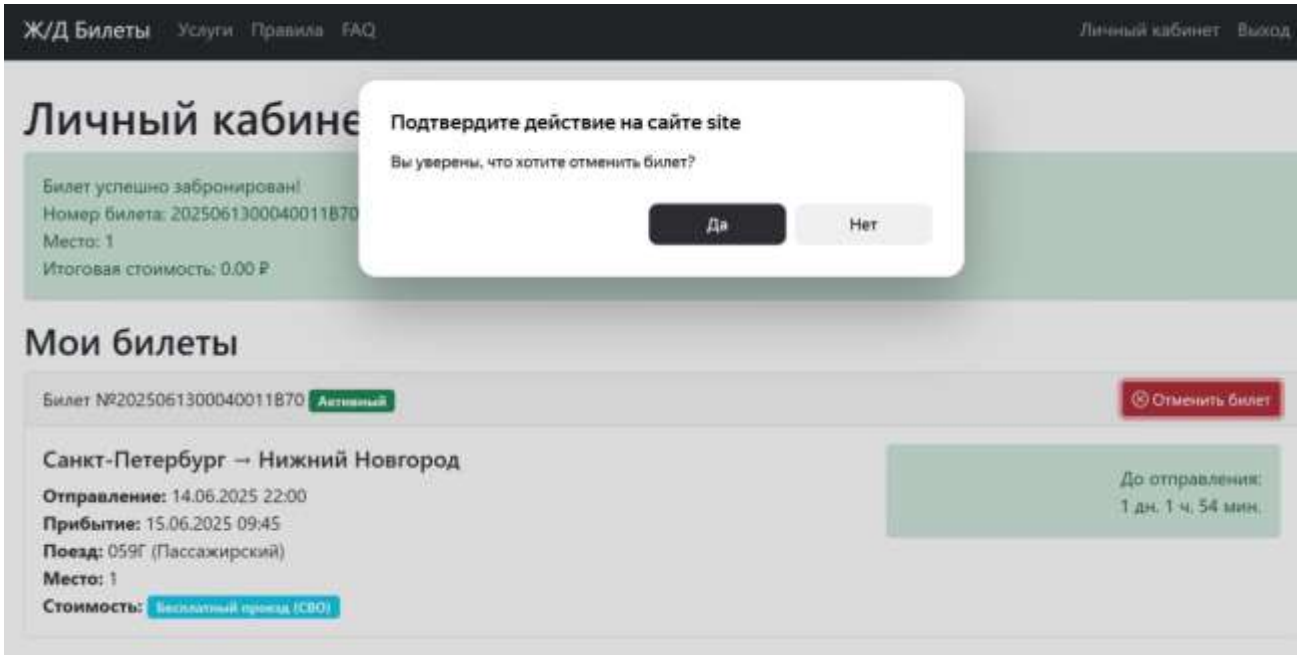

Рисунок А.13 – Отменяем покупку билета

Результат отмены билетаи можно увидеть ниже, на рисунке А.14.

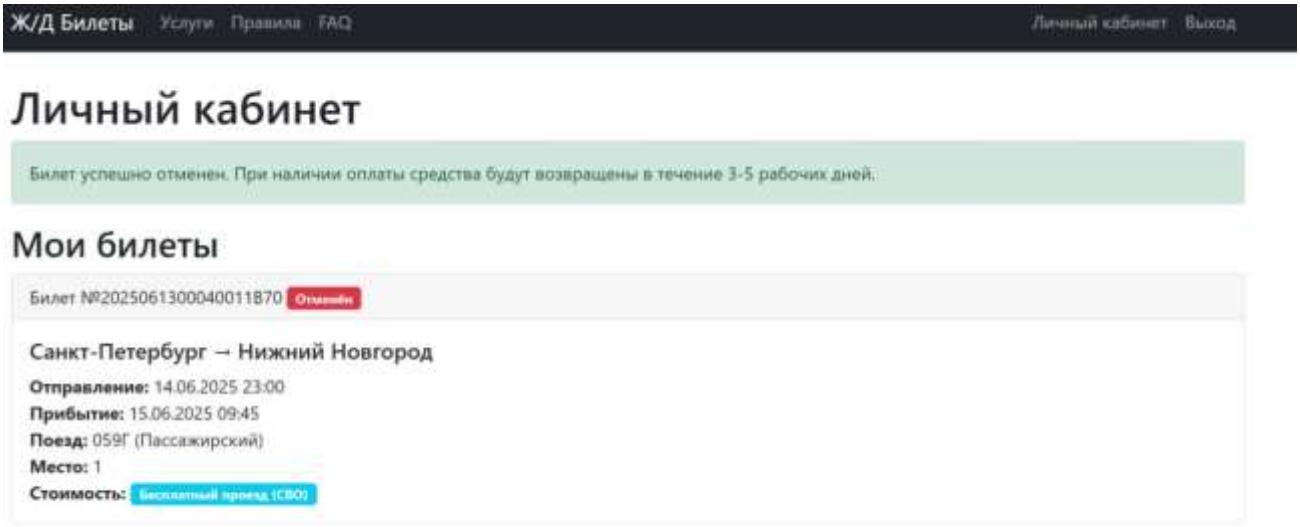

Рисунок А.14 – Билет отменен

Результат покупки еще одного можно увидеть ниже, на рисунке А.15.

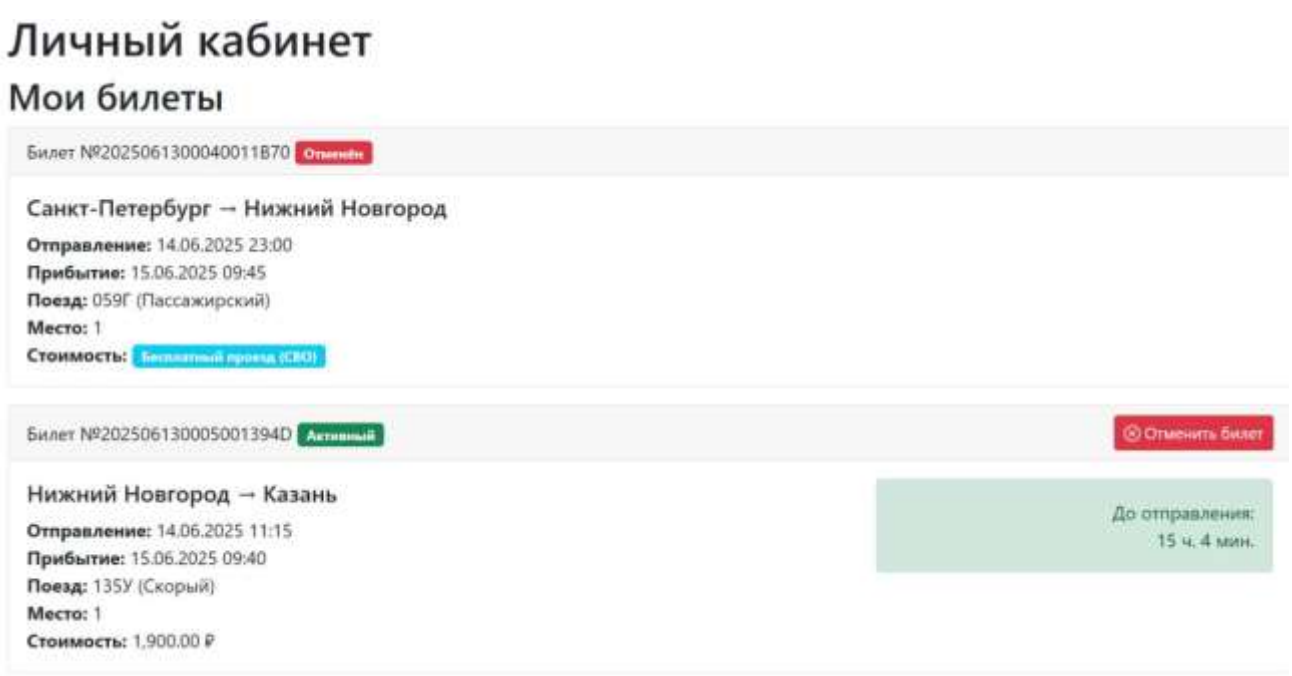

Рисунок А.15 – Покупка еще одного билета

Услуги сервиса можно увидеть ниже, на рисунке А.16.

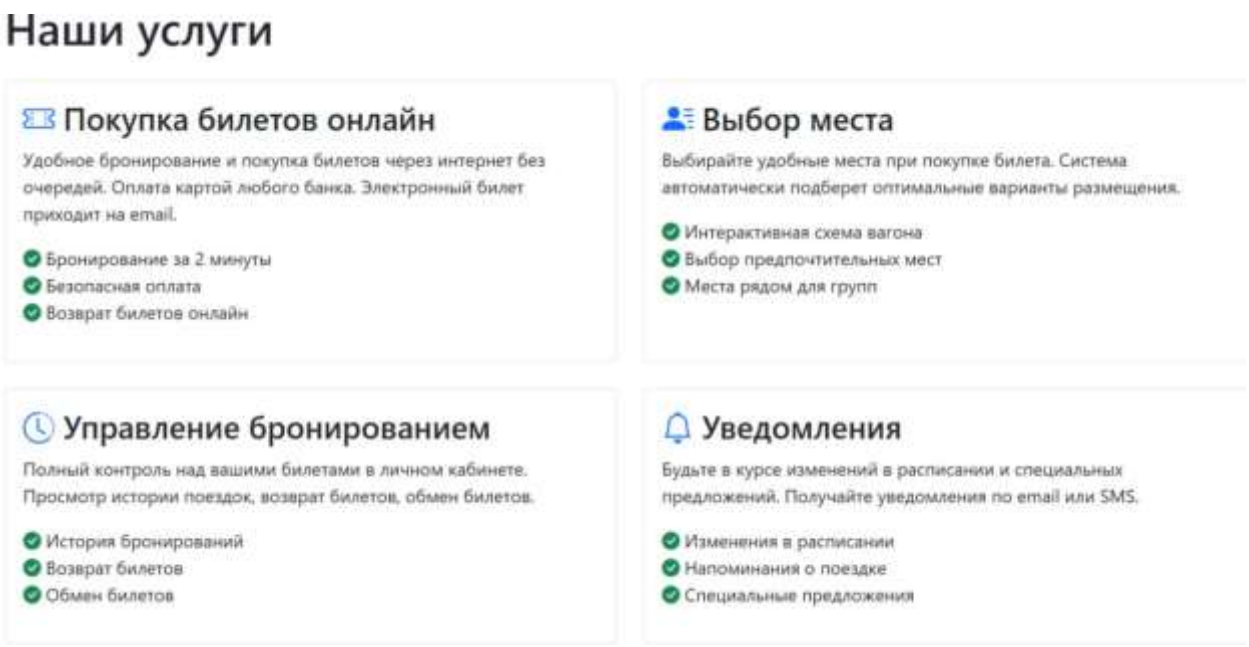

Рисунок А.16 – Услуги сервиса

Перечень дополнительных услуг можно увидеть ниже, на рисунке А.17.

Дополнительные услуги

Услуга	Описание	Стоимость
Страхование	Страхование от несчастных случаев на время поездки	от 100 ₽
Питание в пути	Заказ питания с доставкой к месту	от 300 ₽
VIP-зал ожидания	Доступ в комфортабельный зал ожидания	от 500 ₽
Перевозка багажа	Дополнительный багаж сверх нормы	от 200 ₽

Рисунок А.17 – Дополнительные услуги

Общие положения можно увидеть ниже, на рисунке А.18.

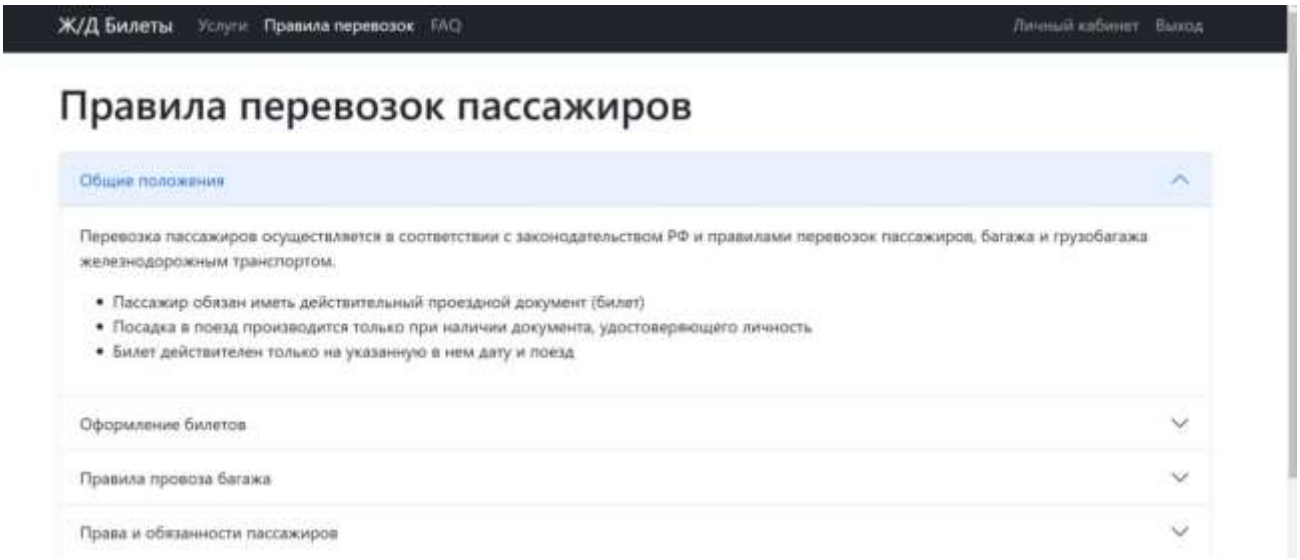

Рисунок А.18 – Правила перевозок пассажирова. Общие положения

Правила оформления билетов можно увидеть ниже, на рисунке А.19.

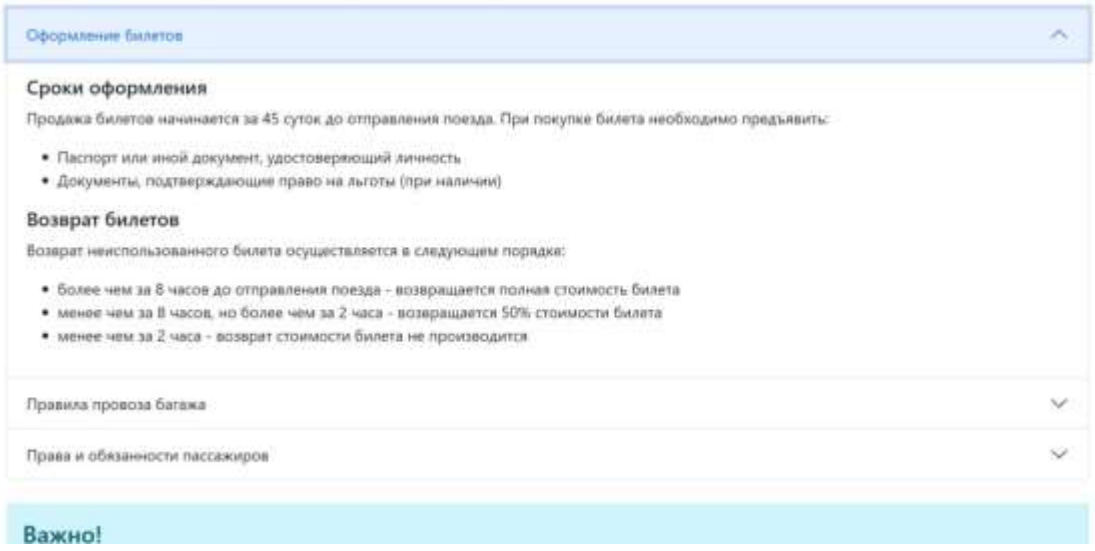

Рисунок А.19 - Правила перевозок пассажирова. Оформление билетов

Правила провоза багажа можно увидеть ниже, на рисунке А.20.

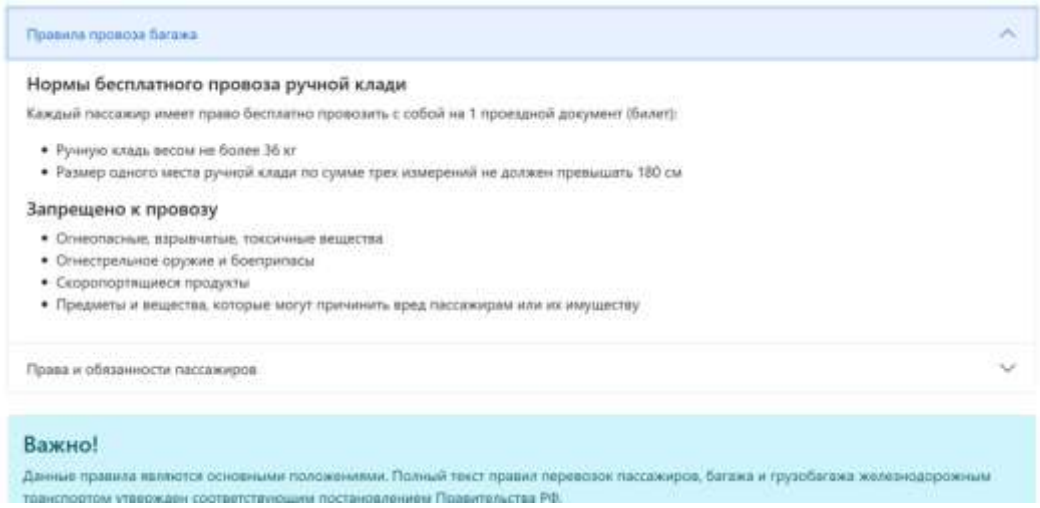

Рисунок А.20 - Правила перевозок пассажирова. Правила провоза багажа

Правила и обязанности пассажиров можно увидеть ниже, на рисунке А.20.

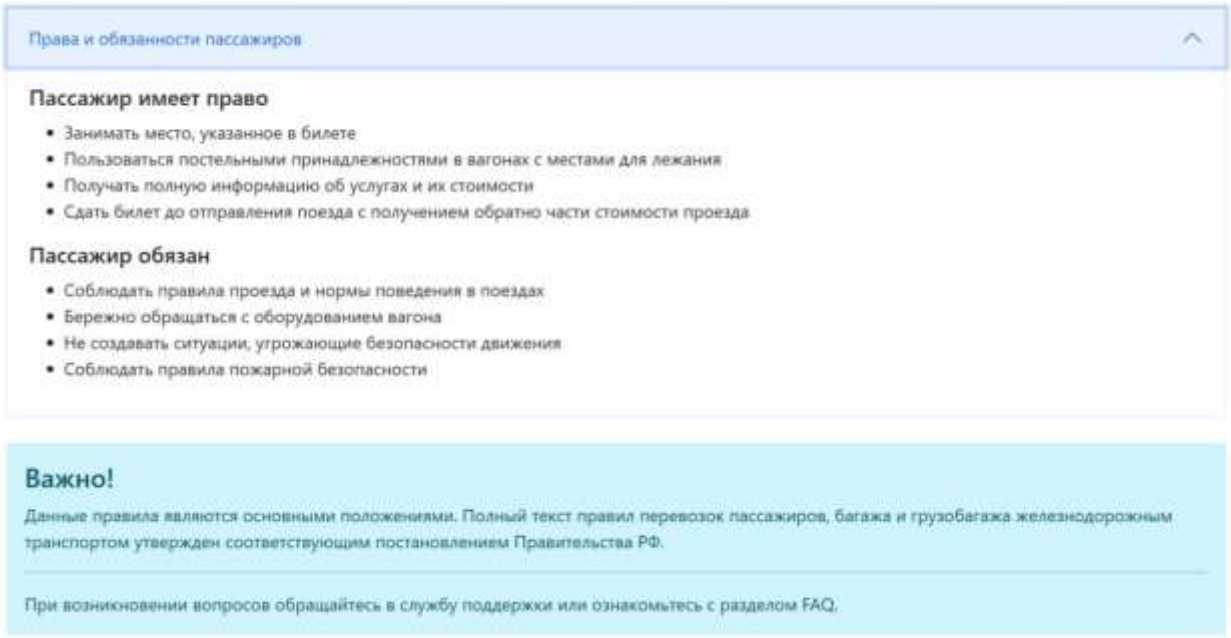

Рисунок А.20 - Правила перевозок пассажирова. Правила и обязанности пассажиров.

Часто задаваемые вопросы касаемые покупки и бронирования билетов можно увидеть ниже, на рисунке А.21.

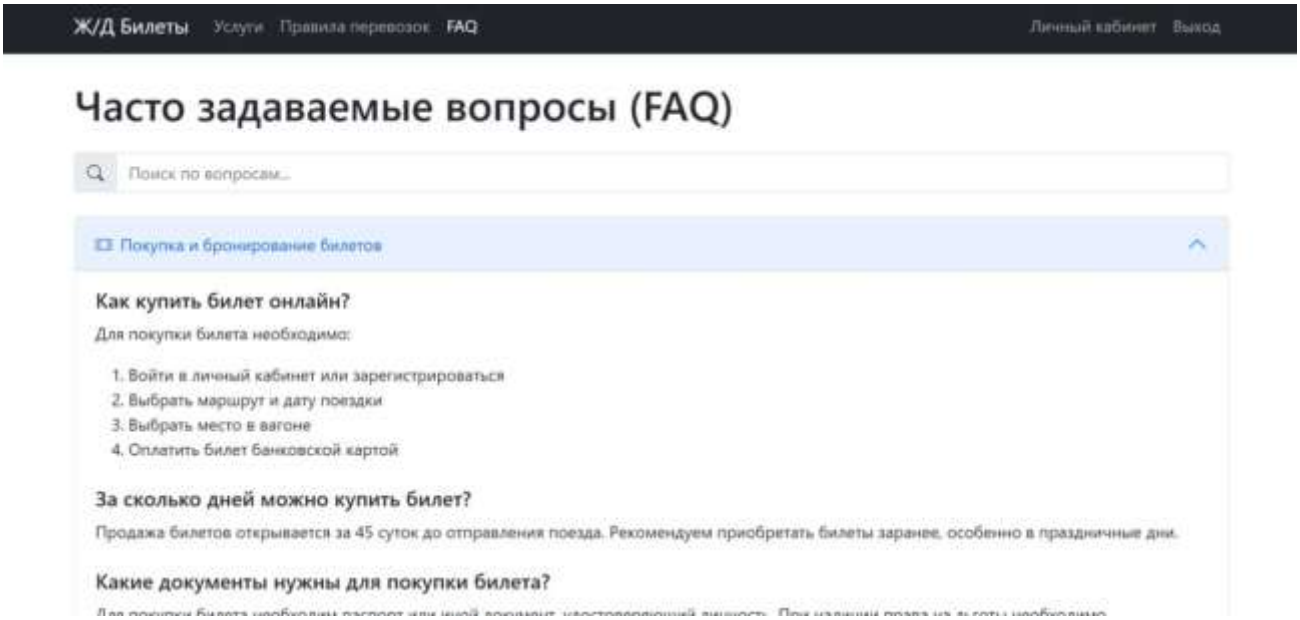

Рисунок А.21 – Часто задаваемые вопросы. Покупка и бронирование билетов.

Часто задаваемые вопросы касаемые способа оплаты можно увидеть ниже, на рисунке А.22.

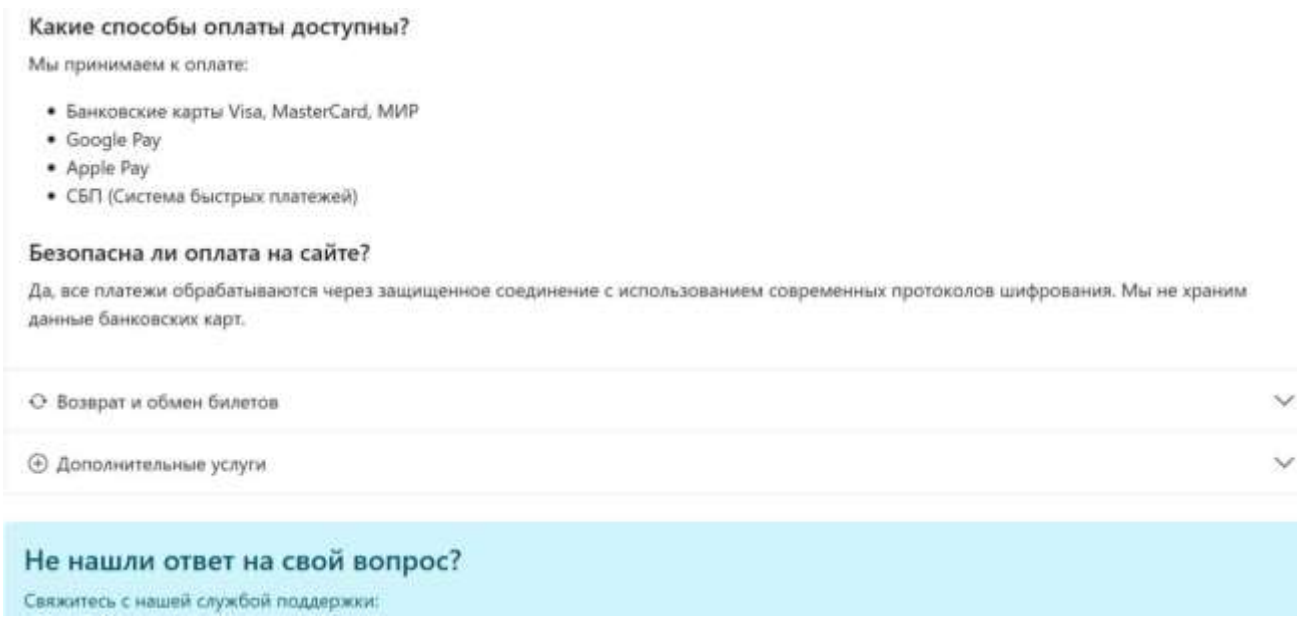

Рисунок А.22 – Часто задаваемые вопросы. Способы оплаты.

Условия возврата билета можно увидеть ниже, на рисунке А.23.

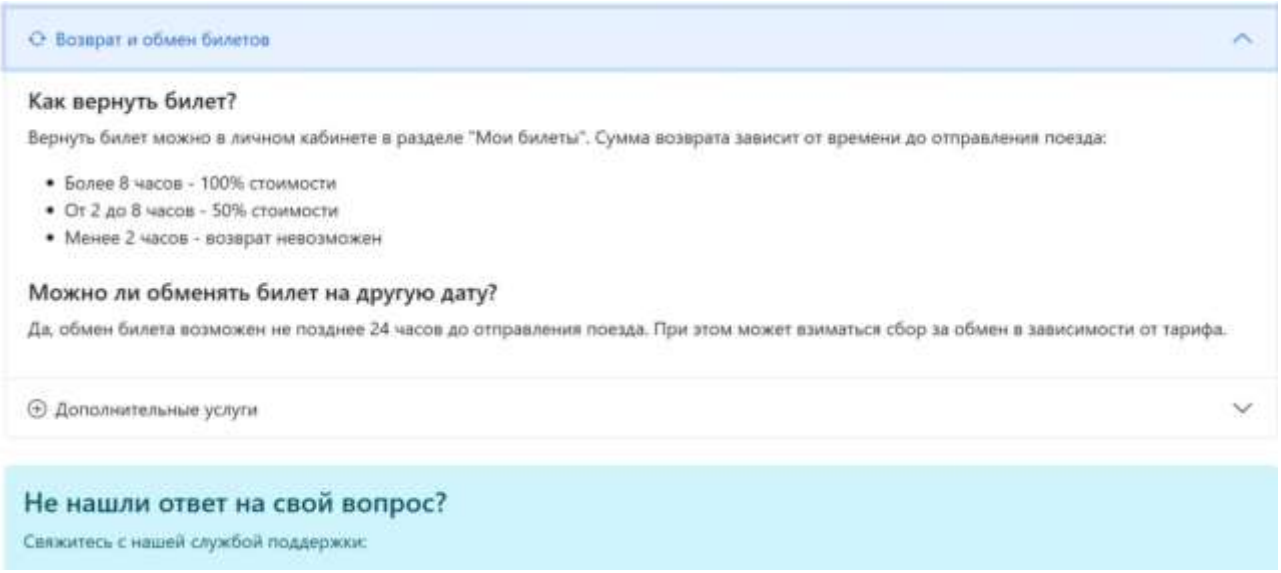

Рисунок А.23 – Часто задаваемые вопросы. Условия возврата билета.

Список дополнительных услуг можно увидеть ниже, на рисунке А.24.

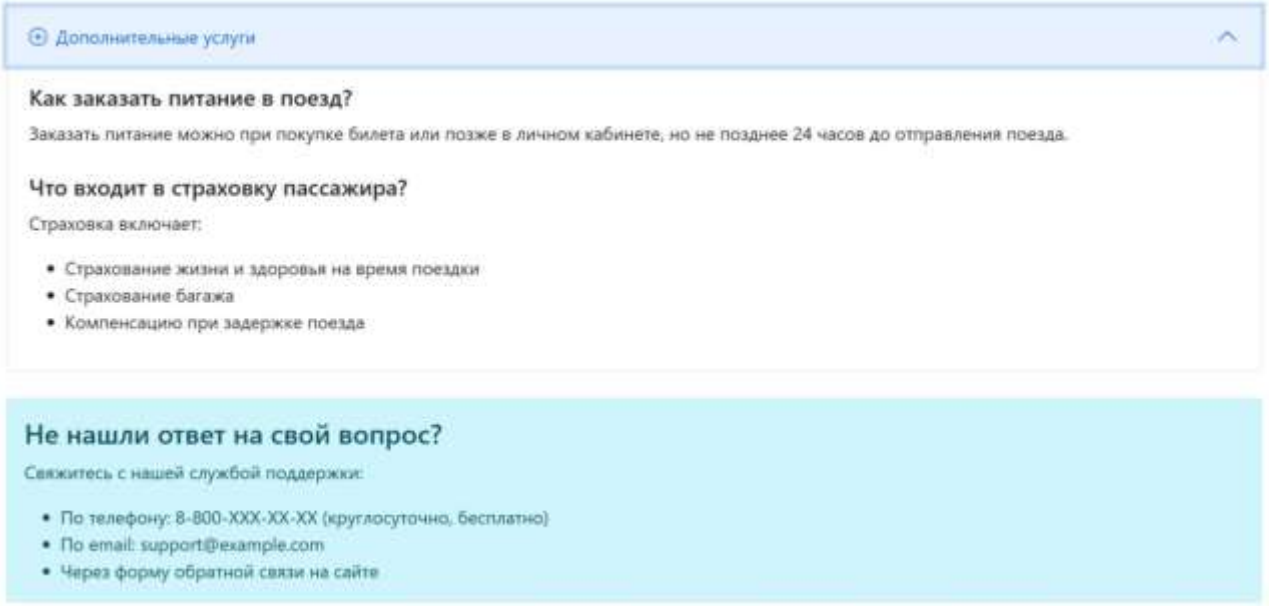

Рисунок А.24 – Часто задаваемые вопросы. Дополнительные услуги.

3 Оценка затрат на разработку программы для бронирования РЖД билетов

3.1 Организация производства разработки программы для бронирования РЖД билетов

Разработка информационной-автоматизированной системы осуществлена на базе площадью 28 кв.м.

Для создания системы были закуплены вычислительные устройства, лицензионное программное обеспечение и дополнительные ресурсы.

Данные оборудование и лицензионное программное обеспечение представлены соответственно в таблицах 4 и 5.

Таблица 4 – Потребность в оборудовании

Оборудование	Источник	Количество	Ориентировочная цена за единицу, рублей
Ноутбук Msi Modern 14 C12MO-688RU (9S7-14J111-688)	ООО «ДНС»	4	71000

Таблица 5 – Потребность в лицензионном программном обеспечении

Лицензионное программное обеспечение	Источник	Цена, рублей	Срок действия лицензии, дней
Microsoft Windows 10	https://майкрософт.официальный.рус/	–	бессрочная
WebStorm	https://www.jetbrains.com/webstorm/	–	Бессрочная
Антивирус Avast	https://www.avast.ru/store	2500	365

Перечень дополнительных материалов, источники их приобретения, объем, и ориентировочные цены приведены в таблице 6.

Таблица 6 – Потребность в дополнительных материалах

Материал	Источник	Количество, штук	Ориентировочная цена за единицу, рублей
Стол компьютерный	ООО «ДНС»	4	4000
Simple 110x75x60 см ЛДСП цвет белый шагрень	Гипермаркет «Лента»	4	1500
Компьютерное кресло ARDOR GAMING Chaos Guard 400M черный	ООО «ДНС»	4	12000
Канцтовары	Гипермаркет	4	600

	«Лента»		

Разработка программы для бронирования РЖД билетов осуществлялась командой разработчиков, представленных в таблице 7.

Таблица 7 – Состав коллектива разработчиков

Специалист	Количество, человек	Описание занятости
Руководитель проекта	1	– планирование и контроль сроков разработки; – координация работы команды (программиста, тестировщика, дизайнера и др); – постановка задач и приоритезация этапов разработки; – общение с заказчиком, сбор требований и согласование изменений – – управление бюджетом и ресурсами проекта; – анализ рисков и решение проблем, возникающих в процессе работы.
Специалист	Количество, человек	Описание занятости
Программист	1	– написание кода сайта (фронтенд и/или бэкенд, в зависимости от специализации); – реализация функционала по техническому заданию (ТЗ); – оптимизация производительности и безопасности сайта; – интеграция с внешними сервисами

		(платежи, API, базы данных и т. д). – исправление ошибок, выявленных тестировщиком; – настройка серверной части (если нет отдельного DevOps-специалиста);
Тестировщик	1	– проверка работоспособности сайта на разных устройствах и браузерах; – поиск багов и ошибок в функционале, интерфейсе, логике работы; – составление тест-кейсов и отчетов по найденным проблемам; – проверка корректности обработки данных (ввод форм, валидация и т. д); – тестирование нагрузки и отказоустойчивости (при необходимости); – контроль качества перед сдачей проекта.
Дизайнер	1	– разработка визуальной концепции сайта; – оптимизация графики.

Данные по основной заработной плате разработчиков программы представлены в таблице 8.

Таблица 8 – Сведения о заработной плате

Специалист	Количество, человек	Сумма заработной платы, рублей
Руководитель проекта	1	100000
Программист	1	82000
Тестировщик	1	45000
Дизайнер	1	80000

3.2 Расчет себестоимости ПО для программы бронирования РЖД билетов

Себестоимость разработки информационной-автоматизированной системы включает следующие расходы:

– на приобретение оборудования;

– на приобретение лицензионного программного обеспечения;

– на дополнительные материалы;

– на заработную плату;

– на обслуживание программы.

Затраты на оборудование составят 284000 рублей и представлены в таблице 4.

Затраты на программное обеспечение составят 2500 рублей и представлены в таблице 5.

Затраты на дополнительные материалы составят 72400 рублей и представлены в таблице 6.

Затраты на заработную плату включают затраты на основную заработную плату разработчиков программы и отчисления на страховые взносы.

При пятидневной рабочей неделе с учетом выходных дней среднее число рабочих дней в месяце составляет около 21 дня. Для завершения всего проекта

требуется около двух месяцев. Таким образом, затраты на заработную плату разработчиков составят 307000 рублей в месяц, 614000 рублей за весь период.

Затраты на обслуживание включают следующие статьи:

– амортизация оборудования, равная 10% от его стоимости;

– расходы на обслуживание оборудования, составляющие 15% от амортизационных начислений;

– расходы на содержание помещения. Амортизация оборудования составляет 20% от стоимости оборудования. Было приобретено 4 ноутбука по 71000 рублей.

Таким образом затраты на амортизацию оборудования в месяц составят:

$$\frac{(4 * 71000) * 0{,}1}{12} = 2366 \text{ руб.}$$

Обслуживание оборудования составляет 15% от амортизационных отчислений.

Таким образом затраты на обслуживание оборудования составят:

$$2366 * 0{,}15 = 355 \text{ руб.}$$

Накладные расходы составляют 75% от суммы заработной платы. Таким образом, расходы на накладные затраты равны:

$$614000 * 0{,}75 = 460500 \text{ руб.}$$

Для аренды небольшого офиса для сотрудников потребуется около 40000, с учетом коммуналки, платы за интернет, уборки и офисной мебели потребуется 172000 из которых на коммуналку потребуется 5000, на интернет около 3000, на уборку порядка 4000, а за офисную мебель придется заплатить 120000.

Результат расчетов по статьям затрат на обслуживание приведены ниже в таблице 9.

Таблица 9 – Затраты на обслуживание

Статья затрат	Показатель отчислений	Сумма отчислений в месяц, рублей	Итоговая сумма отчислений за период разработки, рублей
Амортизация оборудования	10%	2366	4732
Обслуживание оборудования	15%	355	710
Накладные расходы	75%	460500	921000
Аренда офиса и расходы на его содержание	–	172000	224000
Итого, рублей		635221	1150442

Себестоимость разработки определяется как сумма затрат, включающая все статьи затрат на реализацию проекта, представленное в таблице 10.

Таблица 10 – Общая сумма затрат

Статья затрат	Сумма отчислений, рублей
Приобретение оборудования	284000
Приобретение лицензионного программного обеспечения	2500
Затраты на заработную плату	307000
Статья затрат	Сумма отчислений, рублей
Затраты на дополнительные	72400

Статья затрат	Сумма отчислений, рублей
материалы	
Затраты на обслуживание	1150442
Итого:	1284442

Таким образом, сумма затрат на разработку проекта составит 1284442 рублей.

3.3 Оценка эффективности программы бронирования РЖД билетов

Эффективность сайта РЖД для бронирования билетов можно оценить по нескольким ключевым аспектам. Интерфейс сайта должен быть интуитивно понятным и легким в навигации. Эффективный сайт предлагает пользователям простой процесс поиска и бронирования билетов, минимизируя количество шагов и время, необходимое для завершения покупки. Скорость загрузки страниц также критически важна для удержания пользователей – если сайт загружается слишком медленно, это может привести к потере клиентов, которые предпочитают более быстрые и отзывчивые платформы.

Функциональность сайта включает такие возможности, как поиск по различным критериям (дата, направление, тип поезда), выбор мест, опции для возврата билетов. Сайт должен обеспечивать защиту личных данных пользователей и безопасные способы оплаты, так как от этого зависит уровень доверия пользователей к продукту. Поддержка пользователей также играет значимую роль – наличие доступной и оперативной службы поддержки, а также раздела FAQ помогает пользователям быстро находить ответы на свои вопросы и решать возникающие проблемы. Кроме того, аналитика и сбор обратной связи позволяют постоянно улучшать функционал и пользовательский опыт на основе данных о поведении клиентов и их отзывах.

3.4 Анализ рисков при разработке программы для бронирования РЖД билетов

Анализ рисков при разработке сайта для бронирования билетов РЖД включает в себя выявление потенциальных угроз и проблем, которые могут возникнуть на различных этапах проекта. Среди технических рисков выделяются возможные ошибки в программном обеспечении, такие как баги и недоработки в коде, которые могут привести к сбоям в работе системы и негативно сказаться на пользовательском опыте. Также существует риск неполной интеграции новой системы с уже существующими базами данных и сервисами, что может вызвать задержки и дополнительные затраты.

Также, при высоких нагрузках, например, в праздничные дни, сайт может работать медленно или даже выйти из строя из – за проблем с производительностью. Финансовые риски связаны с возможным перерасходом бюджета, так как непредвиденные расходы на разработку, тестирование и внедрение могут привести к превышению запланированных средств. Также существует риск низкой рентабельности, если сайт не привлечет достаточное количество пользователей, что повлияет на покрытие затрат и окупаемость проекта. В области безопасности важным аспектом является риск утечки данных, так как хранение и обработка личных данных пользователей создают угрозу нарушения конфиденциальности, что может повлечь за собой юридические последствия и потерю доверия клиентов. Также существует опасность кибератак, которые могут привести к сбоям в работе сайта и утечке информации. Организационные риски связаны с недостатком квалифицированных кадров, что может снизить качество конечного продукта из – за отсутствия необходимого опыта у команды разработчиков. Кроме того, неэффективное управление проектом, включая плохую организацию работы и недостаточное планирование, может вызвать задержки и перерасход ресурсов, что негативно скажется на общем ходе реализации проекта.

4 Безопасность на транспортных объектах

4.1 Контроль за состоянием инфраструктуры

Контроль за состоянием инфраструктуры на дорогах РЖД представляет собой комплексную систему, объединенную в регулярные проверки и современные технологические методы мониторинга. Плановые и внеплановые инспекции путей, мостов и тоннелей проводят квалифицированные специалисты и специализированные бригады, они оценивают состояние железнодорожного полотна и прочих элементов. Также ко всему этому применяются системы контроля: датчики, дроны обеспечивающие мониторинг в реальном времени и своевременное выявление дефектов. На основании собранных данных разрабатываются графики обслуживания и ремонта для устранения выявленных проблем. Неотъемлемой частью является обучение работников современным методам контроля, это повышает их профессиональный уровень и эффективность работы. Такой интегрированный подход к проверкам и обслуживанию инфраструктуры значительно повышает безопасность и надежность железнодорожных перевозок.

4.2 Профилактика аварийности

Профилактика аварийности на железных дорогах осуществляется с помощью различных методов, которые способствуют снижению числа инцидентов и обеспечивают безопасное движение. Для предотвращения аварий производят анализ прошлых, этот анализ позволяет выявить закономерности и причины происшествий, например, человеческий фактор или какие то, технические неисправности, неблагоприятные условия окружающей среды и другое. Важно обучать персонал по методам, включающим тренинги по правильным действиям в экстренных ситуациях, взаимодействию с пассажирами и мерам по предотвращению аварий. Также активно внедряются современные системы мониторинга и автоматизации, обеспечивающие контроль за состоянием

подвижного состава и инфраструктуры, а средства связи и сигнализации повышают оперативность реагирования на потенциальные угрозы. Регулярные технические осмотры вагонов и путей помогают как можно скорее выявлять и устранять неисправности, тем самым предотвращая их развитие и риск возникновения аварийных ситуаций. Важным аспектом является формирование культуры безопасности среди работников и пассажиров, что способствует повышению внимания и ответственности. Также важным является сотрудничество с правоохранительными органами и службами безопасности для быстрого реагирования на угрозы и борьбы с правонарушениями. В итоге, важно развивать технологии и совершенствовать методы для уменьшения риска возникновения ЧС.

4.3 Эвакуация и действия при ЧС

Эвакуация и действия при чрезвычайных ситуациях (ЧС) являются ключевыми элементами для безопасности населения. Для эффективной эвакуации необходимы план и подготовка, в которую входит разработка разных видов мероприятий. Плановая эвакуация предполагает заранее подготовленные мероприятия, проводимые при угрозе или возникновении ЧС, а экстренная – реализуется срочно в случае внезапных угроз, таких как пожар или землетрясение. Процессы эвакуации состоят из нескольких этапов: информирование населения о предстоящей опасности, сбор информации о местах, требующих эвакуации, а также организация маршрутов, пунктов сбора и транспортировки. Необходимыми действиями при ЧС являются: спокойствие, использование альтернативные выходы, помощь паникующим и нуждающимся. В организациях и учреждениях необходимы различные планы действий на разные сценарии, а также отработки этих планов позволяют подготовить людей к быстрому и скоординированному реагированию. Если следовать этим рекомендациям, то можно поднять уровень готовности и эффективности действий при ЧС, обеспечивая защиту и спокойствие населения в экстремальных ситуациях.

Заключение

В ходе выполнения дипломного проекта была изучена предметная область транспортных услуг, в результате чего было сформулировано техническое задание на разработку программы для бронирования РЖД билетов. Далее были выбраны средства разработки, с помощью которых была разработана программа.

Программа для бронирования РЖД билетов реализована в соответствии с техническим заданием.

В разделе экономики была произведена оценка затрат на разработку и внедрение приложения.

В дальнейшей разработке данной программы будут: система оплаты, реализация API, добавление уведомлений по email, добавление системы поиска маршрутов с пересадками, добавление системы скидок и промокодов, добавление печати билетов в PDF

Список использованных источников

1. Документация к фреймворку Bootstrap. [Электронный ресурс] / Режим доступа: https://getbootstrap.com/docs/5.0/getting-started/introduction/ свободный (дата обращения 29.05.2025).
2. Документация к библиотеке jQuery. [Электронный ресурс] / Режим доступа: https://api.jquery.com/ свободный. (дата обращения 29.05.2025).
3. Документация по OpenServer. [Электронный ресурс] / Режим доступа: https://delphinpro.ru/a/7 свободный. (дата обращения 29.05.2025).
4. Документация по phpMyAdmin. [Электронный ресурс] / Режим доступаhttps://www.phpmyadmin.net/docs/ свободный (дата обращения 29.05.2025).
5. Янцев В. В. «JavaScript. Креативное программирование»
6. Дэниел Нихтер. «Эффективная работа MySQL: лучшие практики и методы»
7. Никсон Робин «Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5» [2019].

Приложение А

(обязательное)

Программа для покупки РЖД билетов

Структура БД

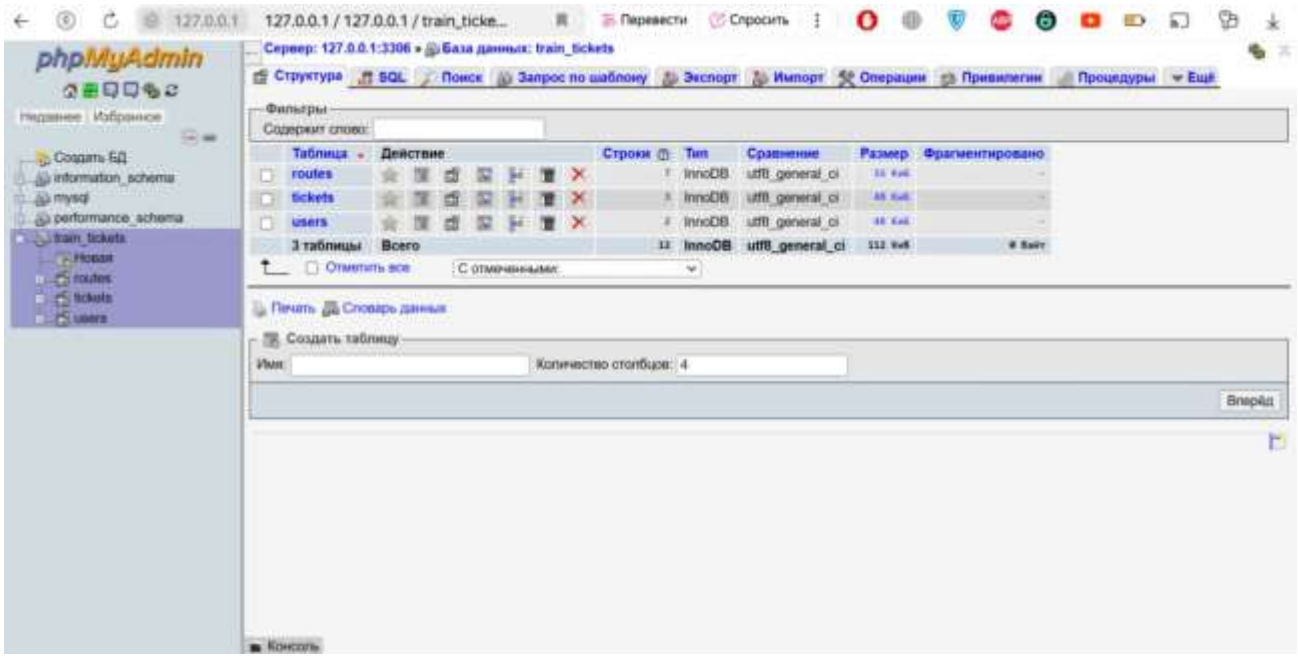

Рисунок Б.1 – Главная панель

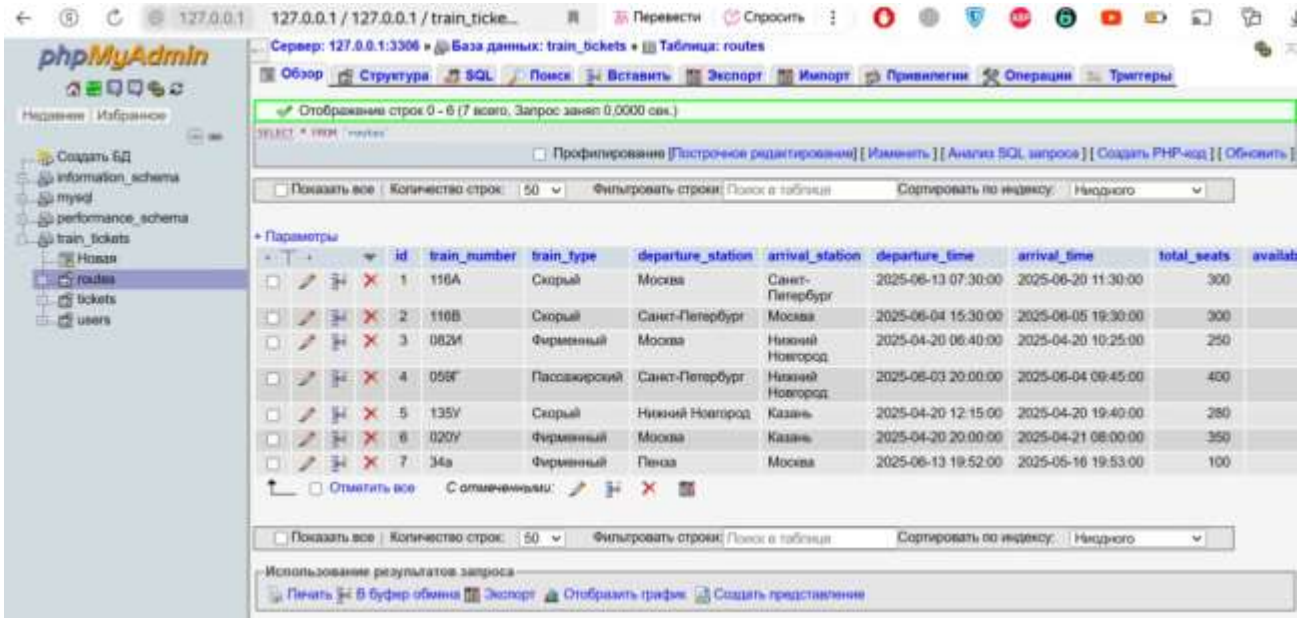

Рисунок Б.2 – Обзор таблицы routes

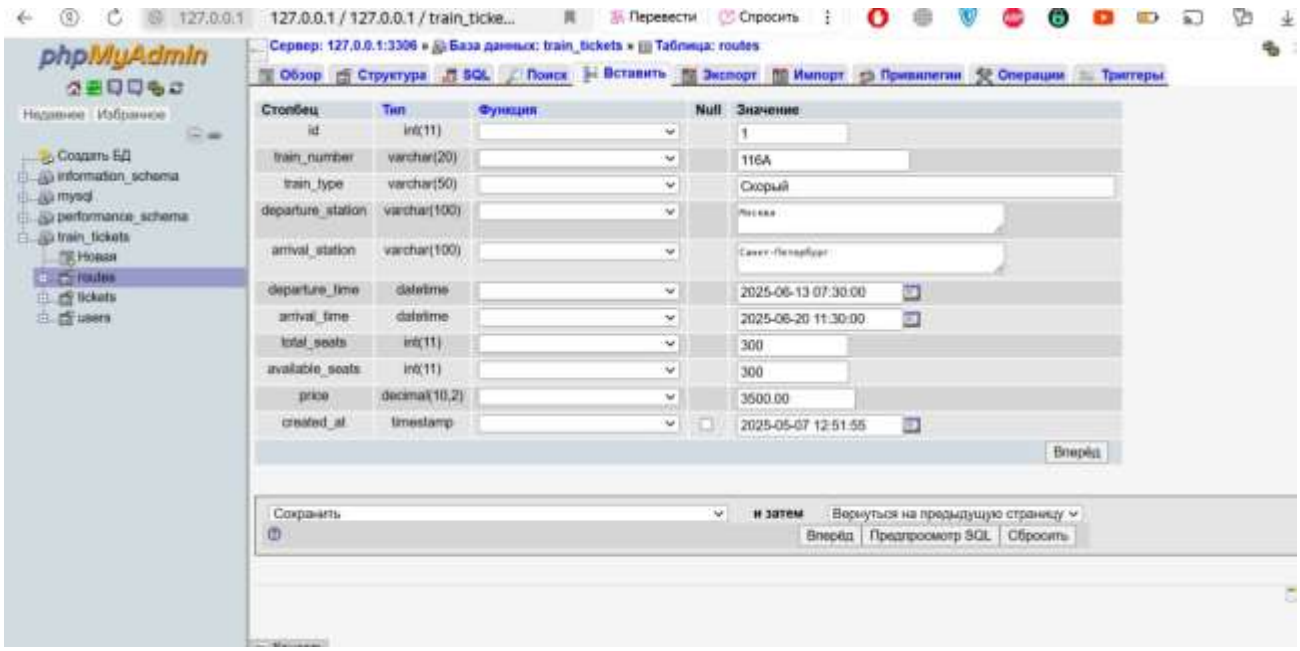

Рисунок Б.3 – редактор таблицы routes

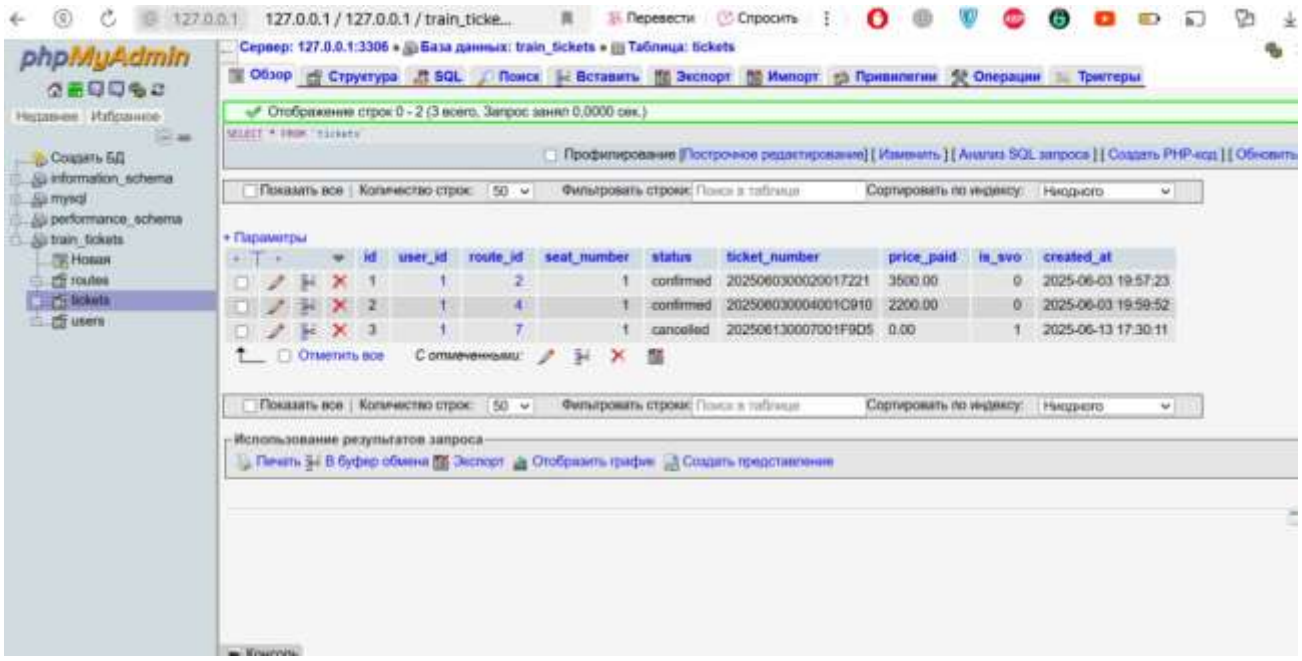

Рисунок Б.4 – Обзор таблицы tickets

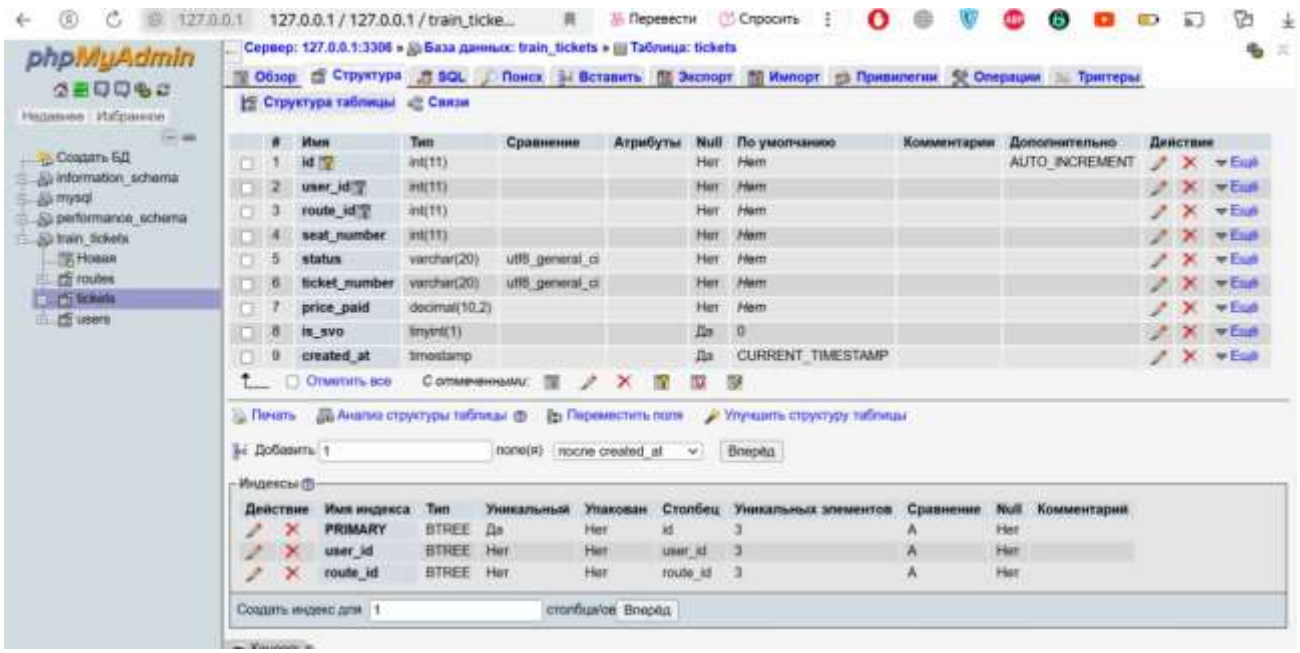

Рисунок Б.5 – Редактор таблицы tickets

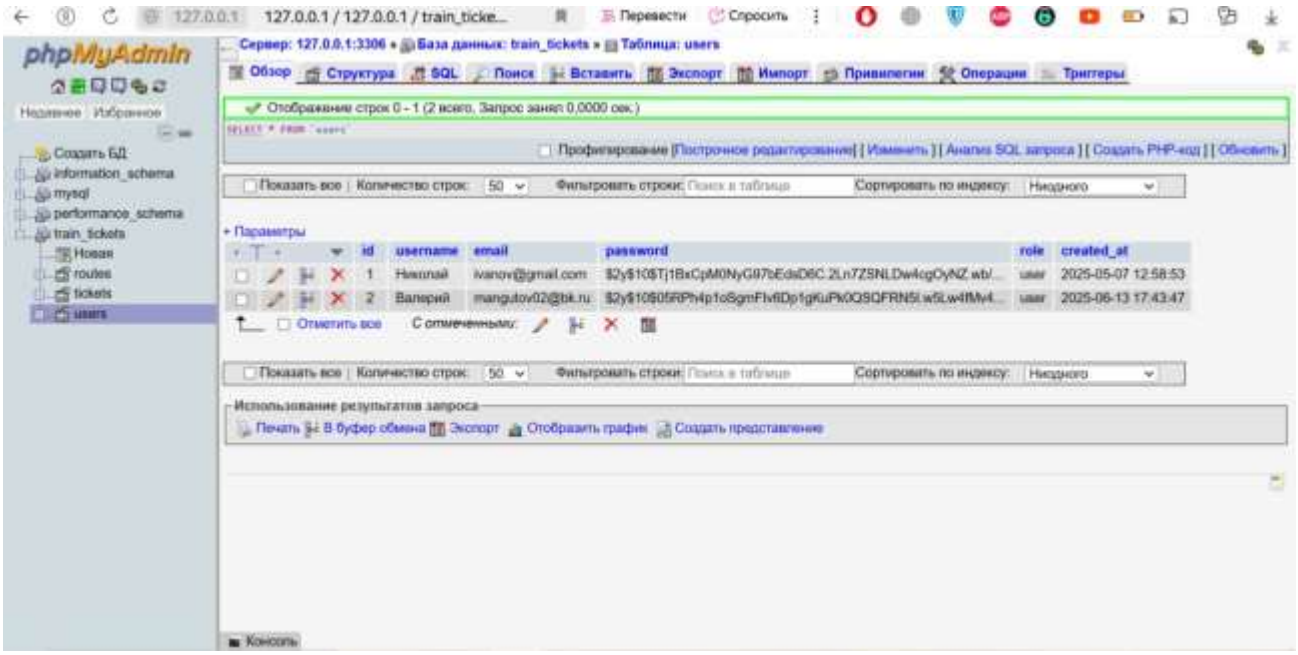

Рисунок Б.6 – Обзор таблицы users

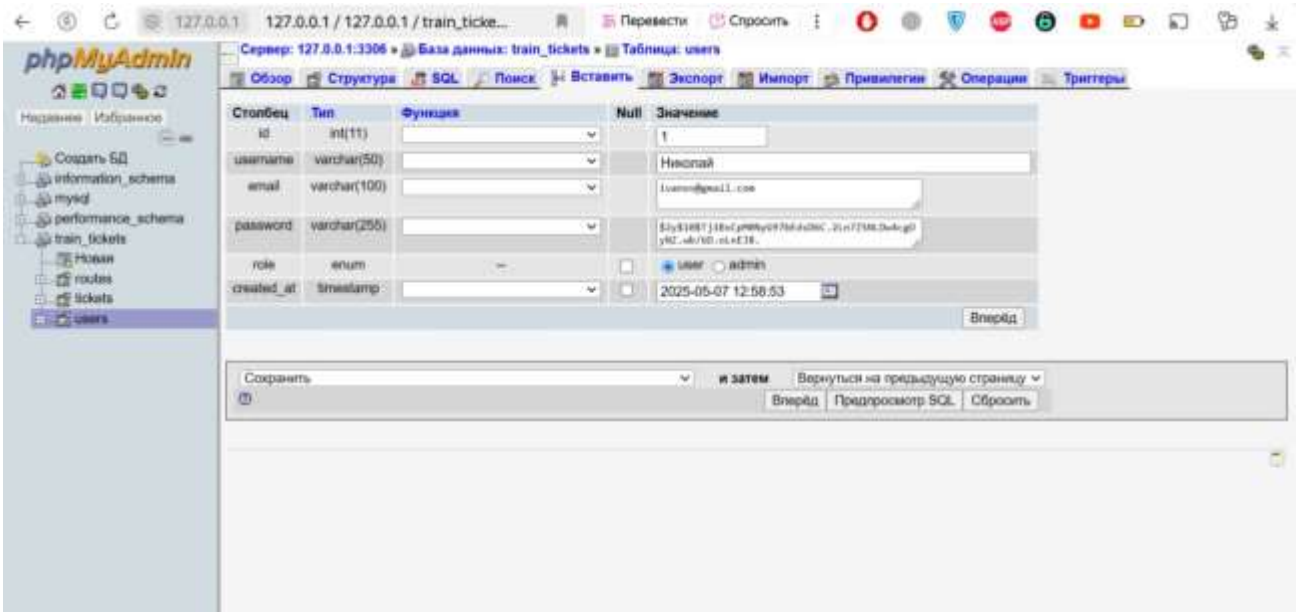

Рисунок Б.7 – Редактор таблицы users

Приложение Б

(обязательное)

Программа для регистрации РЖД билетов

Диаграммы

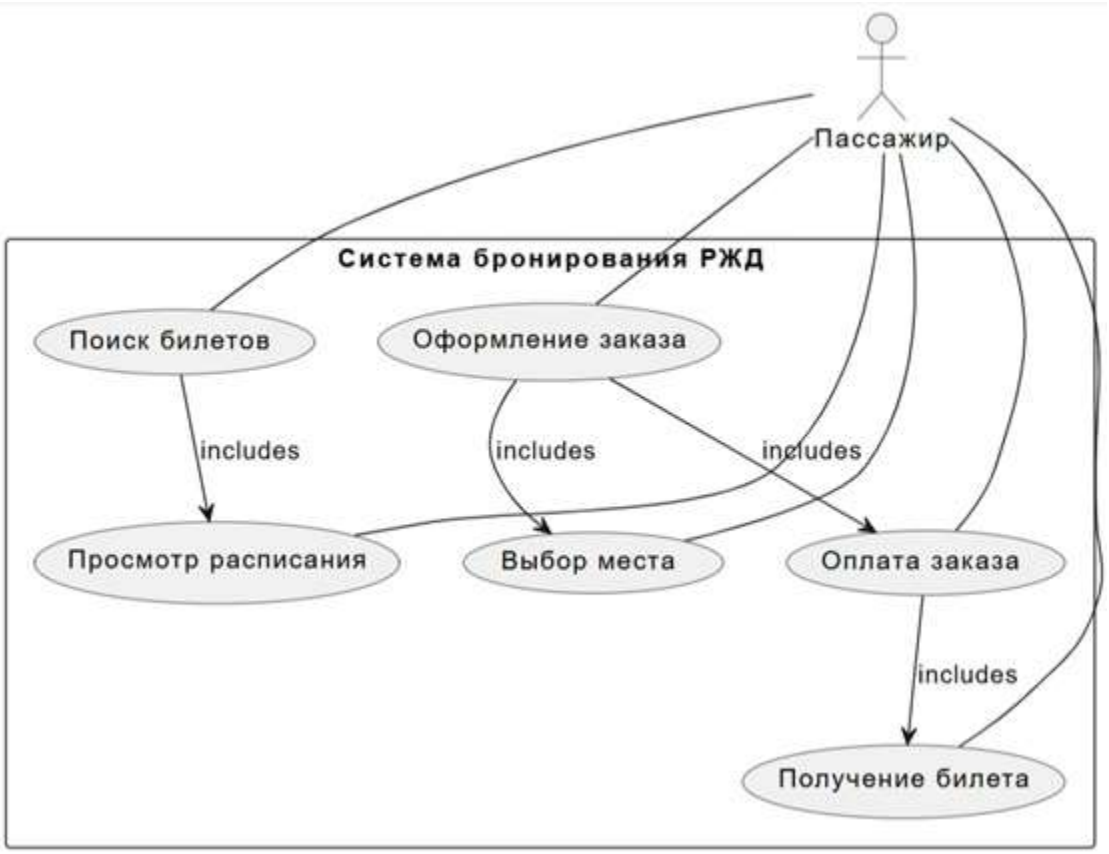

Рисунок В.1 –Диаграмма вариантов использования

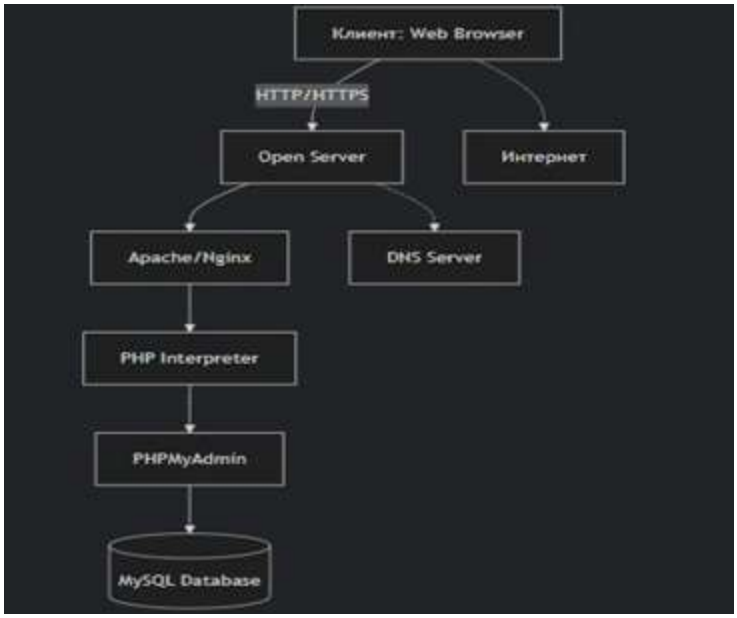

Рисунок В.2 – UML диаграмма развертывания

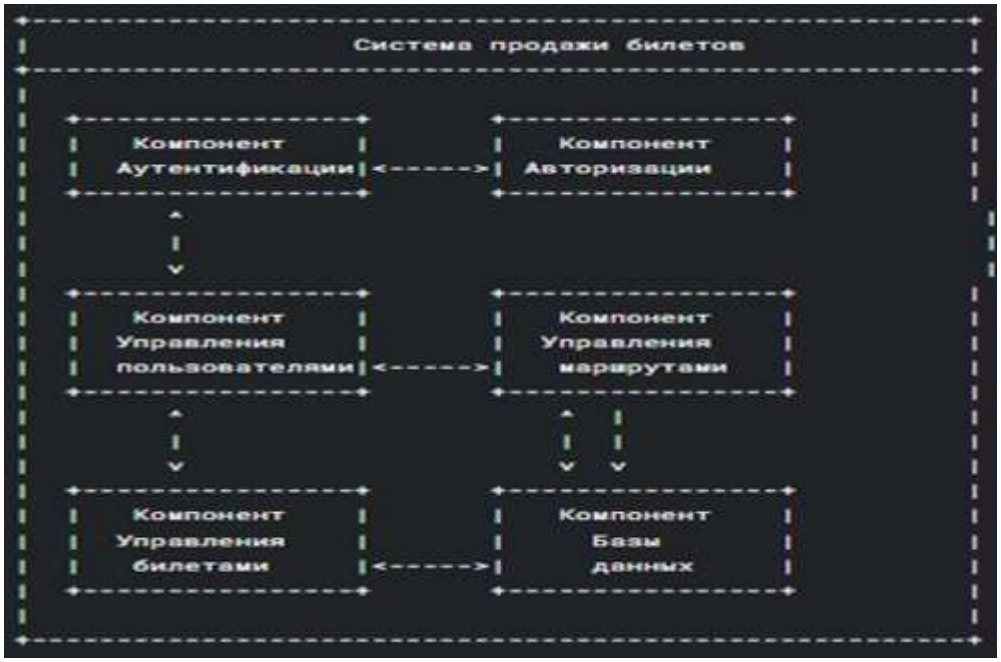

Рисунок В.3 – UML диаграмма компонентов

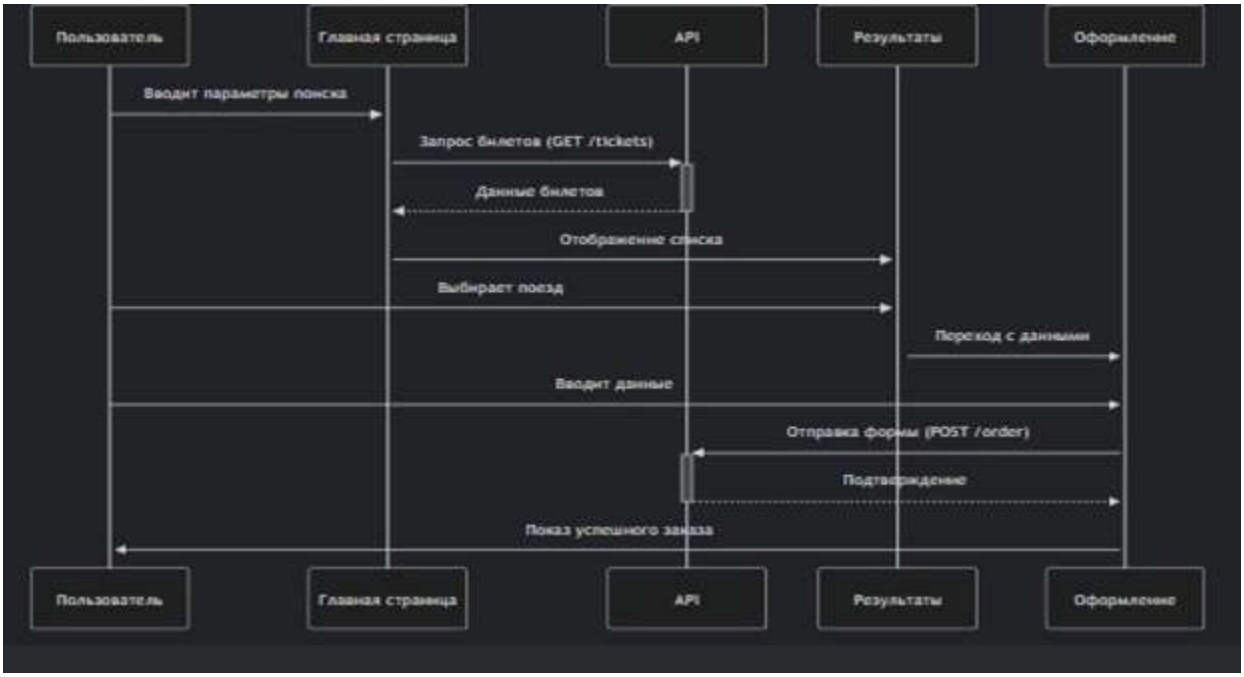

Рисунок В.4 – Структура клиентской части приложения

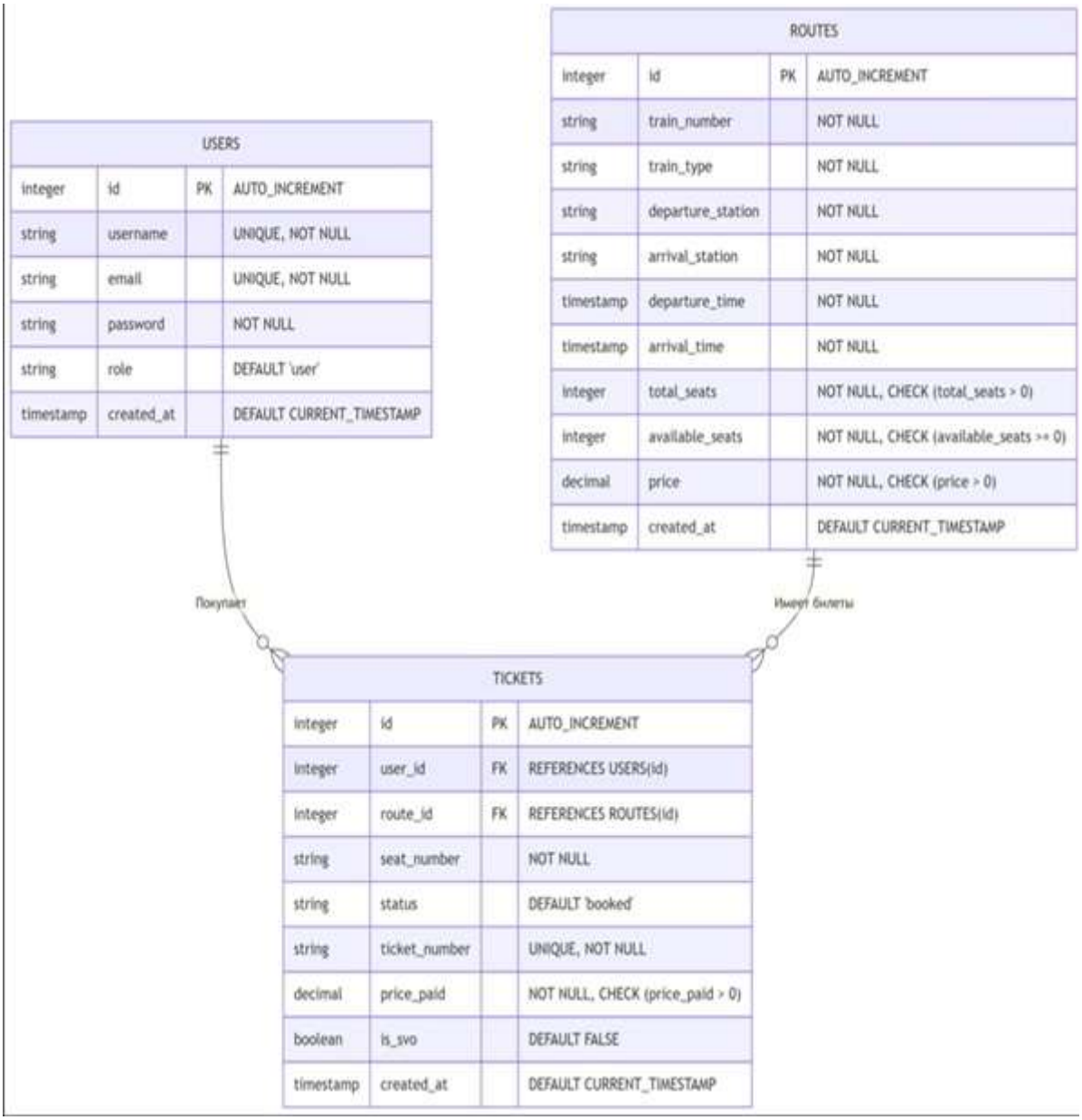

Рисунок В.5 – Структура базы данных

Приложение В

(обязательное)

Программа для регистрации РЖД билетов

Листинг программы

Файл book.php

```
<?php
require_once 'config/database.php';
require_once 'includes/functions.php';

requireLogin();

if (!isset($_GET['route_id'])) {
    flashMessage('Маршрут не выбран', 'danger');
    redirect('/');
}

$route_id = (int)$_GET['route_id'];

// Получаем информацию о маршруте
$stmt = $pdo->prepare("SELECT * FROM routes WHERE id = ? AND
departure_time > NOW()");
$stmt->execute([$route_id]);
$route = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$route) {
    flashMessage('Маршрут не найден или уже отправлен', 'danger');
    redirect('/');
}

if ($route['available_seats'] <= 0) {
    flashMessage('К сожалению, все места на этот маршрут уже
заняты', 'danger');
    redirect('/');
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    try {
        $pdo->beginTransaction();

        // Проверяем еще раз количество мест
        $stmt = $pdo->prepare("SELECT available_seats FROM routes
WHERE id = ? FOR UPDATE");
        $stmt->execute([$route_id]);
        $current_seats = $stmt->fetchColumn();

        if ($current_seats <= 0) {
            throw new Exception('Места уже заняты');
        }

        // Находим свободное место
        $stmt = $pdo->prepare("
            SELECT MIN(t.seat_number) + 1 as next_seat
            FROM tickets t
```

```
            WHERE t.route_id = ?
            AND NOT EXISTS (
                SELECT 1
                FROM tickets
                WHERE route_id = t.route_id
                AND seat_number = t.seat_number + 1
                AND status != 'cancelled'
            )
            UNION
            SELECT 1
            ORDER BY next_seat
            LIMIT 1
        ");
        $stmt->execute([$route_id]);
        $seat_number = $stmt->fetchColumn();

        if (!$seat_number) {
            $seat_number = 1;
        }

        // Рассчитываем финальную стоимость
        $final_price = $route['price'];
        $is_svo = isset($_POST['is_svo']) && $_POST['is_svo'] ==
1;

        // Применяем скидку 100% для участников СВО
        if ($is_svo) {
            $final_price = 0; // Полная скидка 100%
        }

        // Создаем уникальный номер билета
        $random_part = strtoupper(substr(md5(uniqid()), 0, 4)); //
4 случайных символа
        $ticket_number = date('Ymd') .
                         str_pad($route_id, 4, '0', STR_PAD_LEFT) .
                         str_pad($seat_number, 3, '0',
STR_PAD_LEFT) .
                         $random_part;

        // Проверяем уникальность номера билета
        $stmt = $pdo->prepare("SELECT COUNT(*) FROM tickets WHERE
ticket_number = ?");
        $stmt->execute([$ticket_number]);
        if ($stmt->fetchColumn() > 0) {
            throw new Exception('Ошибка генерации номера билета.
Пожалуйста, попробуйте еще раз.');
        }

        // Создаем билет
        $stmt = $pdo->prepare("
            INSERT INTO tickets (
                user_id,
```

```
                route_id,
                seat_number,
                status,
                ticket_number,
                price_paid,
                is_svo
            )
            VALUES (?, ?, ?, 'confirmed', ?, ?, ?)
        ");
        $stmt->execute([
            $_SESSION['user_id'],
            $route_id,
            $seat_number,
            $ticket_number,
            $final_price,
            $is_svo
        ]);

        // Обновляем количество свободных мест
        $stmt = $pdo->prepare("
            UPDATE routes
            SET available_seats = available_seats - 1
            WHERE id = ?
        ");
        $stmt->execute([$route_id]);

        $pdo->commit();

        // Формируем сообщение об успешной покупке
        $success_message = sprintf(
            'Билет успешно забронирован!<br>Номер билета:
%s<br>Место: %d<br>Итоговая стоимость: %.2f ₽',
            $ticket_number,
            $seat_number,
            $final_price
        );

        flashMessage($success_message);
        redirect('/profile.php');

    } catch (Exception $e) {
        $pdo->rollBack();
        flashMessage('Ошибка при бронировании билета: ' . $e-
>getMessage(), 'danger');
        redirect('/');
    }
}
?>

<!DOCTYPE html>
<html lang="ru">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Покупка билета - Система продажи билетов</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootst
rap.min.css" rel="stylesheet">
</head>
<body>
    <div class="container">
        <div class="row justify-content-center mt-5">
            <div class="col-md-8">
                <div class="card">
                    <div class="card-header">
                        <h3 class="text-center">Подтверждение
покупки билета</h3>
                    </div>
                    <div class="card-body">
                        <?= displayFlashMessage() ?>

                        <div class="alert alert-info">
                            <h4>Информация о маршруте:</h4>
                            <p><strong>Откуда:</strong> <?=
sanitize($route['departure_station']) ?></p>
                            <p><strong>Куда:</strong> <?=
sanitize($route['arrival_station']) ?></p>
                            <p><strong>Отправление:</strong> <?=
date('d.m.Y H:i', strtotime($route['departure_time'])) ?></p>
                            <p><strong>Прибытие:</strong> <?=
date('d.m.Y H:i', strtotime($route['arrival_time'])) ?></p>
                            <p><strong>Цена:</strong> <?=
number_format($route['price'], 2) ?> ₽</p>
                            <p><strong>Свободных мест:</strong>
<?= $route['available_seats'] ?></p>
                        </div>

                        <form method="POST" action="">
                            <div class="mb-3">
                                <div class="form-check">
                                    <input type="checkbox"
class="form-check-input" id="is_svo" name="is_svo" value="1">
                                    <label class="form-check-
label" for="is_svo">Я являюсь участником СВО (бесплатный
проезд)</label>
                                </div>
                            </div>

                            <div class="alert alert-info mb-3"
id="price_info">
                                <p class="mb-0">
                                    <strong>Итоговая стоимость:
</strong>
```

```
                                    <span id="final_price"><?=
number_format($route['price'], 2) ?></span> ₽
                                </p>
                            </div>

                            <div class="alert alert-warning">
                                <p class="mb-0">
                                    Подтверждая покупку, вы
соглашаетесь с условиями перевозки.
                                    Билет будет привязан к вашему
аккаунту.
                                </p>
                            </div>

                            <div class="d-grid gap-2">
                                <button type="submit" class="btn
btn-primary">Подтвердить покупку</button>
                                <a href="/" class="btn btn-
secondary">Отмена</a>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstra
p.bundle.min.js"></script>
    <script>

document.getElementById('is_svo').addEventListener('change',
function() {
            const basePrice = <?= $route['price'] ?>;
            const finalPrice = this.checked ? 0 : basePrice; //
Полная скидка 100%
            document.getElementById('final_price').textContent =
finalPrice.toFixed(2);
        });
    </script>
</body>
     </html>
```

Файл cancel ticket.php

```
<?php
require_once 'config/database.php';
require_once 'includes/functions.php';

requireLogin();

if (!isset($_GET['ticket_id'])) {
```

```
    flashMessage('Билет не выбран',
'danger');
    redirect('/profile.php');
}

$ticket_id = (int)$_GET['ticket_id'];

try {
    $pdo->beginTransaction();

    // Получаем информацию о билете
    $stmt = $pdo->prepare("
        SELECT t.*, r.departure_time
        FROM tickets t
        JOIN routes r ON t.route_id = r.id
        WHERE t.id = ? AND t.user_id = ? AND
t.status = 'confirmed'
        FOR UPDATE
    ");
    $stmt->execute([$ticket_id,
$_SESSION['user_id']]);
    $ticket = $stmt-
>fetch(PDO::FETCH_ASSOC);

    if (!$ticket) {
        throw new Exception('Билет не найден
или уже отменен');
    }

    // Проверяем, что до отправления
осталось больше часа
    $departure_time = new
DateTime($ticket['departure_time']);
    $now = new DateTime();
    $interval = $now->diff($departure_time);

    if ($departure_time <= $now) {
        throw new Exception('Невозможно
отменить билет: поезд уже отправлен');
    }

    if ($interval->days == 0 && $interval->h
< 1) {
        throw new Exception('Невозможно
отменить билет менее чем за час до
отправления');
    }

    // Отменяем билет
    $stmt = $pdo->prepare("
        UPDATE tickets
        SET status = 'cancelled'
```

```
        WHERE id = ?
    ");
    $stmt->execute([$ticket_id]);

    // Возвращаем место в кассу
    $stmt = $pdo->prepare("
        UPDATE routes
        SET available_seats =
available_seats + 1
        WHERE id = ?
    ");
    $stmt->execute([$ticket['route_id']]);

    $pdo->commit();

    flashMessage('Билет успешно отменен. При
наличии оплаты средства будут возвращены в
течение 3-5 рабочих дней.');
    redirect('/profile.php');

} catch (Exception $e) {
    $pdo->rollBack();
    flashMessage('Ошибка при отмене билета:
' . $e->getMessage(), 'danger');
    redirect('/profile.php');
}
                seat_number,
```

Файл index.php

```
<?php
require_once 'config/database.php';
require_once 'includes/functions.php';

// Получение списка станций для выпадающих
списков
$stations_query = "SELECT DISTINCT
departure_station FROM routes
                  UNION
                  SELECT DISTINCT
arrival_station FROM routes
                  ORDER BY departure_station";
$stmt = $pdo->query($stations_query);
$stations = $stmt->fetchAll(PDO::FETCH_COLUMN);
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1.0">
    <title>Система продажи билетов на
поезд</title>
```

```
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1
.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.7.2/font/bootstrap-icons.css">
    <style>
        .hero-section {
            background: linear-gradient(rgba(0,
0, 0, 0.5), rgba(0, 0, 0, 0.5)),
url('assets/images/train-hero.jpg');
            background-size: cover;
            background-position: center;
            padding: 100px 0;
            color: white;
        }
        .search-form {
            background: rgba(255, 255, 255,
0.9);
            padding: 30px;
            border-radius: 10px;
            box-shadow: 0 0 20px rgba(0, 0, 0,
0.1);
        }
        .search-button {
            height: 40px;
        }
    </style>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-
dark bg-dark">
        <div class="container">
            <a class="navbar-brand" href="/">Ж/Д
Билеты</a>
            <button class="navbar-toggler"
type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav">
                <span class="navbar-toggler-
icon"></span>
            </button>
            <div class="collapse navbar-
collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link"
href="/services.php">Услуги</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="/rules.php">Правила перевозок</a>
                    </li>
```

```
                    <li class="nav-item">
                        <a class="nav-link"
href="/faq.php">FAQ</a>
                    </li>
                </ul>
                <ul class="navbar-nav ms-auto">
                    <?php if (isLoggedIn()): ?>
                        <?php if (isAdmin()): ?>
                            <li class="nav-
item">
                                <a class="nav-
link" href="/admin/">Панель администратора</a>
                            </li>
                        <?php endif; ?>
                        <li class="nav-item">
                            <a class="nav-link"
href="/profile.php">Личный кабинет</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link"
href="/logout.php">Выход</a>
                        </li>
                    <?php else: ?>
                        <li class="nav-item">
                            <a class="nav-link"
href="/login.php">Вход</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link"
href="/register.php">Регистрация</a>
                        </li>
                    <?php endif; ?>
                </ul>
            </div>
        </div>
    </nav>

    <section class="hero-section">
        <div class="container">
            <div class="row justify-content-
center">
                <div class="col-md-8">
                    <h1 class="text-center mb-
4">Поиск железнодорожных билетов</h1>
                    <div class="search-form">
                        <form
action="search_results.php" method="GET">
                            <div class="row g-
3">
                                <div class="col-
md-6 mb-3">
```

```
                                    <label
for="from" class="form-label text-
dark">Откуда</label>
                                    <select
class="form-select" id="from" name="from"
required>
                                        <option
value="">Выберите станцию отправления</option>
                                        <?php
foreach ($stations as $station): ?>

<option value="<?= htmlspecialchars($station)
?>">

<?= htmlspecialchars($station) ?>

</option>
                                        <?php
endforeach; ?>
                                    </select>
                                </div>
                                <div class="col-
md-6 mb-3">
                                    <label
for="to" class="form-label text-
dark">Куда</label>
                                    <select
class="form-select" id="to" name="to" required>
                                        <option
value="">Выберите станцию назначения</option>
                                        <?php
foreach ($stations as $station): ?>

<option value="<?= htmlspecialchars($station)
?>">

<?= htmlspecialchars($station) ?>

</option>
                                        <?php
endforeach; ?>
                                    </select>
                                </div>
                                <div class="col-
md-6 mb-3">
                                    <label
for="date_from" class="form-label text-
dark">Дата отправления с</label>
                                    <input
type="date" class="form-control" id="date_from"
name="date_from" required
```

```
min="<?php echo date('Y-m-d'); ?>">
                                    </div>
                                    <div class="col-
md-6 mb-3">
                                        <label
for="date_to" class="form-label text-dark">Дата
отправления по</label>
                                        <input
type="date" class="form-control" id="date_to"
name="date_to" required

min="<?php echo date('Y-m-d'); ?>">
                                    </div>
                                    <div class="col-
12 text-center">
                                        <button
type="submit" class="btn btn-primary search-
button px-5">
                                            <i
class="bi bi-search me-2"></i>Найти билеты
                                        </button>
                                    </div>
                                </div>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
        </section>

        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.
3/dist/js/bootstrap.bundle.min.js"></script>
        <script>
            // Валидация дат

document.getElementById('date_from').addEventLis
tener('change', function() {

document.getElementById('date_to').min =
this.value;
            });


document.getElementById('date_to').addEventListe
ner('change', function() {
                if (this.value <
document.getElementById('date_from').value) {
                    this.value =
document.getElementById('date_from').value;
                }
```

```
        });
    </script>
</body>
     </html>
```

Файл login.php

```
<?php
require_once 'config/database.php';
require_once 'includes/functions.php';

if (isLoggedIn()) {
    redirect('/');
}

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

    $errors = [];

    if (empty($username) || empty($password)) {
        $errors[] = 'Все поля обязательны для заполнения';
    }

    if (empty($errors)) {
        $stmt = $pdo->prepare("SELECT * FROM users WHERE username
= ?");
        $stmt->execute([$username]);
        $user = $stmt->fetch(PDO::FETCH_ASSOC);

        if ($user && password_verify($password,
$user['password'])) {
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['username'] = $user['username'];
            $_SESSION['role'] = $user['role'];

            flashMessage('Добро пожаловать, ' . $user['username']
. '!');
            redirect('/');
        } else {
            $errors[] = 'Неверное имя пользователя или пароль';
        }
    }
}
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
```

```
    <title>Вход - Система продажи билетов</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootst
rap.min.css" rel="stylesheet">
</head>
<body>
    <div class="container">
        <div class="row justify-content-center mt-5">
            <div class="col-md-6">
                <div class="card">
                    <div class="card-header">
                        <h3 class="text-center">Вход в
систему</h3>
                    </div>
                    <div class="card-body">
                        <?= displayFlashMessage() ?>

                        <?php if (!empty($errors)): ?>
                            <div class="alert alert-danger">
                                <ul class="mb-0">
                                    <?php foreach ($errors as
$error): ?>
                                        <li><?= $error ?></li>
                                    <?php endforeach; ?>
                                </ul>
                            </div>
                        <?php endif; ?>

                        <form method="POST" action="">
                            <div class="mb-3">
                                <label for="username" class="form-
label">Имя пользователя</label>
                                <input type="text" class="form-
control" id="username" name="username"
                                       value="<?= isset($username)
? sanitize($username) : '' ?>" required>
                            </div>

                            <div class="mb-3">
                                <label for="password" class="form-
label">Пароль</label>
                                <input type="password"
class="form-control" id="password" name="password" required>
                            </div>

                            <div class="d-grid">
                                <button type="submit" class="btn
btn-primary">Войти</button>
                            </div>
                        </form>
                    </div>
                    <div class="card-footer text-center">
```

```
                        Нет аккаунта? <a
href="/register.php">Зарегистрироваться</a>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstra
p.bundle.min.js"></script>
</body>
</html>
```

Файл logout.php

```
<?php
require_once 'includes/functions.php';

session_start();
session_destroy();
flashMessage('Вы успешно вышли из системы');
     redirect('/login.php');
```

Файл profile.php

```
 <?php
require_once 'config/database.php';
require_once 'includes/functions.php';

requireLogin();

// Получаем билеты пользователя
$stmt = $pdo->prepare("
    SELECT t.*,
           r.departure_station,
           r.arrival_station,
           r.departure_time,
           r.arrival_time,
           r.train_number,
           r.train_type
    FROM tickets t
    JOIN routes r ON t.route_id = r.id
    WHERE t.user_id = ?
    ORDER BY r.departure_time DESC
");
$stmt->execute([$_SESSION['user_id']]);
$tickets = $stmt->fetchAll(PDO::FETCH_ASSOC);

// Обработка отмены билета
if (isset($_POST['cancel_ticket'])) {
    $ticket_id = (int)$_POST['ticket_id'];
```

```
    try {
        $pdo->beginTransaction();

        // Проверяем, принадлежит ли билет пользователю
        $stmt = $pdo->prepare("
            SELECT t.*, r.departure_time
            FROM tickets t
            JOIN routes r ON t.route_id = r.id
            WHERE t.id = ? AND t.user_id = ?
            FOR UPDATE
        ");
        $stmt->execute([$ticket_id, $_SESSION['user_id']]);
        $ticket = $stmt->fetch(PDO::FETCH_ASSOC);

        if (!$ticket) {
            throw new Exception('Билет не найден');
        }

        // Проверяем, не прошло ли время отправления
        if (strtotime($ticket['departure_time']) <= time()) {
            throw new Exception('Нельзя отменить билет после
отправления поезда');
        }

        // Отменяем билет
        $stmt = $pdo->prepare("UPDATE tickets SET status =
'cancelled' WHERE id = ?");
        $stmt->execute([$ticket_id]);

        // Возвращаем место в пул свободных
        $stmt = $pdo->prepare("
            UPDATE routes
            SET available_seats = available_seats + 1
            WHERE id = ?
        ");
        $stmt->execute([$ticket['route_id']]);

        $pdo->commit();
        flashMessage('Билет успешно отменен');
        redirect('/profile.php');

    } catch (Exception $e) {
        $pdo->rollBack();
        flashMessage($e->getMessage(), 'danger');
        redirect('/profile.php');
    }
}
?>

<!DOCTYPE html>
<html lang="ru">
```

```
<head>
    <meta charset="UTF-8">
    <meta  name="viewport"  content="width=device-width,  initial-
scale=1.0">
    <title>Личный кабинет - Система продажи билетов</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootst
rap.min.css" rel="stylesheet">
    <link                                          rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.7.2/font/bootstrap-icons.css">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark mb-4">
        <div class="container">
            <a class="navbar-brand" href="/">Ж/Д Билеты</a>
            <button  class="navbar-toggler"  type="button"  data-bs-
toggle="collapse" data-bs-target="#navbarNav">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav me-auto">
                    <li class="nav-item">
                        <a                             class="nav-link"
href="services.php">Услуги</a>
                    </li>
                    <li class="nav-item">
                        <a                             class="nav-link"
href="rules.php">Правила</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="faq.php">FAQ</a>
                    </li>
                </ul>
                <ul class="navbar-nav">
                    <?php if (isLoggedIn()): ?>
                        <?php if (isAdmin()): ?>
                            <li class="nav-item">
                                <a                     class="nav-link"
href="admin/">Панель администратора</a>
                            </li>
                        <?php endif; ?>
                        <li class="nav-item">
                            <a                         class="nav-link"
href="profile.php">Личный кабинет</a>
                        </li>
                        <li class="nav-item">
                            <a                         class="nav-link"
href="logout.php">Выход</a>
                        </li>
                    <?php else: ?>
                        <li class="nav-item">
```

```
                            <a                          class="nav-link"
href="login.php">Вход</a>
                        </li>
                        <li class="nav-item">
                            <a                          class="nav-link"
href="register.php">Регистрация</a>
                        </li>
                    <?php endif; ?>
                </ul>
            </div>
        </div>
    </nav>

    <div class="container my-4">
        <h1>Личный кабинет</h1>

        <?= displayFlashMessage() ?>

        <div class="row">
            <div class="col-md-12">
                <h2>Мои билеты</h2>
                <?php if (empty($tickets)): ?>
                    <div class="alert alert-info">
                        У вас пока нет купленных билетов.
                        <a    href="/"    class="alert-link">Найти
билеты</a>
                    </div>
                <?php else: ?>
                    <?php foreach ($tickets as $ticket): ?>
                        <?php
                        $departure              =               new
DateTime($ticket['departure_time']);
                        $arrival                =               new
DateTime($ticket['arrival_time']);
                        $now = new DateTime();
                        $canCancel = $departure > $now &&
                                     $ticket['status']          ===
'confirmed' &&
                                     $departure->modify('-1 hour') >
$now;
                        ?>
                        <div class="card mb-3">
                            <div     class="card-header     d-flex
justify-content-between align-items-center">
                                <span>
                                    Билет                       №<?=
htmlspecialchars($ticket['ticket_number']) ?>
                                    <?php   if   ($ticket['status']
=== 'confirmed'): ?>
                                        <span   class="badge   bg-
success">Активный</span>
```

```
                                    <?php                    elseif
($ticket['status'] === 'cancelled'): ?>
                                        <span   class="badge   bg-
danger">Отменён</span>
                                    <?php endif; ?>
                                </span>
                                <?php if ($canCancel): ?>
                                    <a
href="cancel_ticket.php?ticket_id=<?= $ticket['id'] ?>"
                                       class="btn  btn-danger  btn-
sm"
                                       onclick="return  confirm('Вы
уверены, что хотите отменить билет?')">
                                        <i     class="bi     bi-x-
circle"></i> Отменить билет
                                    </a>
                                <?php endif; ?>
                            </div>
                            <div class="card-body">
                                <div class="row">
                                    <div class="col-md-8">
                                        <h5 class="card-title">
                                            <?=
htmlspecialchars($ticket['departure_station']) ?> →
                                            <?=
htmlspecialchars($ticket['arrival_station']) ?>
                                        </h5>
                                        <p class="card-text">

<strong>Отправление:</strong>
                                            <?=         $departure-
>format('d.m.Y H:i') ?><br>

<strong>Прибытие:</strong>
                                            <?=           $arrival-
>format('d.m.Y H:i') ?><br>

<strong>Поезд:</strong>
                                            <?=
htmlspecialchars($ticket['train_number']) ?>
                                             (<?=
htmlspecialchars($ticket['train_type']) ?>)<br>

<strong>Место:</strong> <?= $ticket['seat_number'] ?><br>

<strong>Стоимость:</strong>
                                            <?php                if
($ticket['is_svo']): ?>
                                                <span  class="badge
bg-info">Бесплатный проезд (СВО)</span>
                                            <?php else: ?>
```

```
                                                    <?=
number_format($ticket['price_paid'], 2) ?> ₽
                                                <?php endif; ?>
                                            </p>
                                        </div>
                                        <div   class="col-md-4   text-
end">
                                            <?php                    if
($ticket['status'] === 'confirmed'): ?>
                                                <?php   if   ($departure
<= $now): ?>
                                                    <span  class="badge
bg-secondary">Поезд отправлен</span>
                                                <?php else: ?>
                                                    <div   class="alert
alert-success mb-0">
                                                        До
отправления:<br>
                                                        <?php
                                                        $interval    =
$now->diff($departure);
                                                        if  ($interval-
>days > 0) {
                                                            echo
$interval->days . ' дн. ';
                                                        }
                                                        echo
$interval->h . ' ч. ' . $interval->i . ' мин.';
                                                        ?>
                                                    </div>
                                                <?php endif; ?>
                                            <?php endif; ?>
                                        </div>
                                    </div>
                                </div>
                            </div>
                        <?php endforeach; ?>
                    <?php endif; ?>
                </div>
            </div>
        </div>

        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstra
p.bundle.min.js"></script>
</body>
                            </html>
```

Файл register.php

```
<?php
require_once 'config/database.php';
require_once 'includes/functions.php';
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Правила перевозок - Система продажи билетов на
поезд</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootst
rap.min.css" rel="stylesheet">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.7.2/font/bootstrap-icons.css">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark mb-4">
        <div class="container">
            <a class="navbar-brand" href="/">Ж/Д Билеты</a>
            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link"
href="/services.php">Услуги</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link active"
href="/rules.php">Правила перевозок</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="/faq.php">FAQ</a>
                    </li>
                </ul>
                <ul class="navbar-nav ms-auto">
                    <?php if (isLoggedIn()): ?>
                        <?php if (isAdmin()): ?>
                            <li class="nav-item">
                                <a class="nav-link"
href="/admin/">Панель администратора</a>
                            </li>
                        <?php endif; ?>
                        <li class="nav-item">
```

```
                            <a class="nav-link"
href="/profile.php">Личный кабинет</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link"
href="/logout.php">Выход</a>
                        </li>
                    <?php else: ?>
                        <li class="nav-item">
                            <a class="nav-link"
href="/login.php">Вход</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link"
href="/register.php">Регистрация</a>
                        </li>
                    <?php endif; ?>
                </ul>
            </div>
        </div>
    </nav>

    <div class="container">
        <h1 class="mb-4">Правила перевозок пассажиров</h1>

        <div class="accordion" id="rulesAccordion">
            <div class="accordion-item">
                <h2 class="accordion-header" id="headingOne">
                    <button class="accordion-button" type="button"
data-bs-toggle="collapse" data-bs-target="#collapseOne">
                        Общие положения
                    </button>
                </h2>
                <div id="collapseOne" class="accordion-collapse
collapse show" data-bs-parent="#rulesAccordion">
                    <div class="accordion-body">
                        <p>Перевозка пассажиров осуществляется в
соответствии с законодательством РФ и правилами перевозок
пассажиров, багажа и грузобагажа железнодорожным транспортом.</p>
                        <ul>
                            <li>Пассажир обязан иметь
действительный проездной документ (билет)</li>
                            <li>Посадка в поезд производится
только при наличии документа, удостоверяющего личность</li>
                            <li>Билет действителен только на
указанную в нем дату и поезд</li>
                        </ul>
                    </div>
                </div>
            </div>

            <div class="accordion-item">
```

```
                <h2 class="accordion-header" id="headingTwo">
                    <button class="accordion-button collapsed"
type="button" data-bs-toggle="collapse" data-bs-
target="#collapseTwo">
                        Оформление билетов
                    </button>
                </h2>
                <div id="collapseTwo" class="accordion-collapse
collapse" data-bs-parent="#rulesAccordion">
                    <div class="accordion-body">
                        <h5>Сроки оформления</h5>
                        <p>Продажа билетов начинается за 45 суток
до отправления поезда. При покупке билета необходимо
предъявить:</p>
                        <ul>
                            <li>Паспорт или иной документ,
удостоверяющий личность</li>
                            <li>Документы, подтверждающие право на
льготы (при наличии)</li>
                        </ul>

                        <h5 class="mt-3">Возврат билетов</h5>
                        <p>Возврат неиспользованного билета
осуществляется в следующем порядке:</p>
                        <ul>
                            <li>более чем за 8 часов до
отправления поезда - возвращается полная стоимость билета</li>
                            <li>менее чем за 8 часов, но более чем
за 2 часа - возвращается 50% стоимости билета</li>
                            <li>менее чем за 2 часа - возврат
стоимости билета не производится</li>
                        </ul>
                    </div>
                </div>
            </div>

            <div class="accordion-item">
                <h2 class="accordion-header" id="headingThree">
                    <button class="accordion-button collapsed"
type="button" data-bs-toggle="collapse" data-bs-
target="#collapseThree">
                        Правила провоза багажа
                    </button>
                </h2>
                <div id="collapseThree" class="accordion-collapse
collapse" data-bs-parent="#rulesAccordion">
                    <div class="accordion-body">
                        <h5>Нормы бесплатного провоза ручной
клади</h5>
                        <p>Каждый пассажир имеет право бесплатно
провозить с собой на 1 проездной документ (билет):</p>
                        <ul>
```

```
                                <li>Ручную кладь весом не более 36
кг</li>
                                <li>Размер одного места ручной клади
по сумме трех измерений не должен превышать 180 см</li>
                            </ul>

                            <h5 class="mt-3">Запрещено к провозу</h5>
                            <ul>
                                <li>Огнеопасные, взрывчатые, токсичные
вещества</li>
                                <li>Огнестрельное оружие и
боеприпасы</li>
                                <li>Скоропортящиеся продукты</li>
                                <li>Предметы и вещества, которые могут
причинить вред пассажирам или их имуществу</li>
                            </ul>
                        </div>
                    </div>
                </div>

                <div class="accordion-item">
                    <h2 class="accordion-header" id="headingFour">
                        <button class="accordion-button collapsed"
type="button" data-bs-toggle="collapse" data-bs-
target="#collapseFour">
                            Права и обязанности пассажиров
                        </button>
                    </h2>
                    <div id="collapseFour" class="accordion-collapse
collapse" data-bs-parent="#rulesAccordion">
                        <div class="accordion-body">
                            <h5>Пассажир имеет право</h5>
                            <ul>
                                <li>Занимать место, указанное в
билете</li>
                                <li>Пользоваться постельными
принадлежностями в вагонах с местами для лежания</li>
                                <li>Получать полную информацию об
услугах и их стоимости</li>
                                <li>Сдать билет до отправления поезда
с получением обратно части стоимости проезда</li>
                            </ul>

                            <h5 class="mt-3">Пассажир обязан</h5>
                            <ul>
                                <li>Соблюдать правила проезда и нормы
поведения в поездах</li>
                                <li>Бережно обращаться с оборудованием
вагона</li>
                                <li>Не создавать ситуации, угрожающие
безопасности движения</li>
```

```
                            <li>Соблюдать правила пожарной
безопасности</li>
                        </ul>
                    </div>
                </div>
            </div>
        </div>

        <div class="alert alert-info mt-4">
            <h4 class="alert-heading">Важно!</h4>
            <p>Данные правила являются основными положениями.
Полный текст правил перевозок пассажиров, багажа и грузобагажа
железнодорожным транспортом утвержден соответствующим
постановлением Правительства РФ.</p>
            <hr>
            <p class="mb-0">При возникновении вопросов обращайтесь
в службу поддержки или ознакомьтесь с разделом FAQ.</p>
        </div>
    </div>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstra
p.bundle.min.js"></script>
</body>
</html>
```

Файл search results.php

```
<?php
require_once 'config/database.php';
require_once 'includes/functions.php';

// Получение параметров поиска
$from = isset($_GET['from']) ? $_GET['from'] : '';
$to = isset($_GET['to']) ? $_GET['to'] : '';
$date_from = isset($_GET['date_from']) ? $_GET['date_from'] :
'';
$date_to = isset($_GET['date_to']) ? $_GET['date_to'] : '';

// Подготовка запроса для поиска маршрутов
$query = "SELECT * FROM routes
          WHERE departure_station = :from
          AND arrival_station = :to
          AND DATE(departure_time) BETWEEN :date_from AND
:date_to
          AND departure_time > NOW()
          ORDER BY departure_time";
```

```
$stmt = $pdo->prepare($query);
$stmt->execute([
    ':from' => $from,
    ':to' => $to,
    ':date_from' => $date_from,
    ':date_to' => $date_to
]);
$routes = $stmt->fetchAll(PDO::FETCH_ASSOC);

// Группировка маршрутов по датам
$routes_by_date = [];
foreach ($routes as $route) {
    $date = date('Y-m-d',
strtotime($route['departure_time']));
    if (!isset($routes_by_date[$date])) {
        $routes_by_date[$date] = [];
    }
    $routes_by_date[$date][] = $route;
}
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Результаты поиска - Система продажи билетов на
поезд</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/b
ootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.7.2/font/bootstrap-icons.css">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark
mb-4">
        <div class="container">
            <a class="navbar-brand" href="/">Ж/Д Билеты</a>
            <button class="navbar-toggler" type="button"
data-bs-toggle="collapse" data-bs-target="#navbarNav">
```

```
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse"
id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link"
href="/services.php">Услуги</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="/rules.php">Правила перевозок</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="/faq.php">FAQ</a>
                    </li>
                </ul>
                <ul class="navbar-nav ms-auto">
                    <?php if (isLoggedIn()): ?>
                        <?php if (isAdmin()): ?>
                            <li class="nav-item">
                                <a class="nav-link"
href="/admin/">Панель администратора</a>
                            </li>
                        <?php endif; ?>
                        <li class="nav-item">
                            <a class="nav-link"
href="/profile.php">Личный кабинет</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link"
href="/logout.php">Выход</a>
                        </li>
                    <?php else: ?>
                        <li class="nav-item">
                            <a class="nav-link"
href="/login.php">Вход</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link"
href="/register.php">Регистрация</a>
                        </li>
```

```
                    <?php endif; ?>
                </ul>
            </div>
        </div>
    </nav>

    <div class="container">
        <div class="row mb-4">
            <div class="col">
                <h2>Результаты поиска</h2>
                <p class="lead">
                    <?= htmlspecialchars($from) ?> → <?=
htmlspecialchars($to) ?><br>
                    <span class="text-muted">
                        <?= date('d.m.Y',
strtotime($date_from)) ?> - <?= date('d.m.Y',
strtotime($date_to)) ?>
                    </span>
                </p>
                <a href="/" class="btn btn-outline-primary
mb-3">
                    <i class="bi bi-arrow-left"></i> Изменить
параметры поиска
                </a>
            </div>
        </div>

        <?php if (empty($routes)): ?>
            <div class="alert alert-info">
                <h4 class="alert-heading">Маршруты не
найдены</h4>
                <p>К сожалению, на выбранные даты нет
доступных маршрутов. Попробуйте изменить параметры
поиска.</p>
                <hr>
                <p class="mb-0">Рекомендуем проверить другие
даты или выбрать другие станции.</p>
            </div>
        <?php else: ?>
            <?php foreach ($routes_by_date as $date =>
$day_routes): ?>
                <div class="card mb-4">
                    <div class="card-header">
```

```
                        <h3 class="card-title h5 mb-0">
                            <?= date('d.m.Y',
strtotime($date)) ?>
                            <span class="badge bg-
primary"><?= count($day_routes) ?> рейс(ов)</span>
                        </h3>
                    </div>
                    <div class="card-body p-0">
                        <div class="table-responsive">
                            <table class="table table-hover
mb-0">
                                <thead class="table-light">
                                    <tr>
                                        <th>Время
отправления</th>
                                        <th>Время
прибытия</th>
                                        <th>В пути</th>
                                        <th>Поезд</th>
                                        <th>Свободных
мест</th>
                                        <th>Цена</th>
                                        <th>Действия</th>
                                    </tr>
                                </thead>
                                <tbody>
                                    <?php foreach
($day_routes as $route): ?>
                                        <?php
                                        $departure = new
DateTime($route['departure_time']);
                                        $arrival = new
DateTime($route['arrival_time']);
                                        $duration =
$departure->diff($arrival);
                                        $duration_text = '';

                                        if ($duration->d > 0)
{
                                            $duration_text .=
$duration->d . ' д ';
                                        }
                                        if ($duration->h > 0
|| $duration->d > 0) {
```

```
                                                $duration_text .=
$duration->h . ' ч ';
                                            }
                                            $duration_text .=
$duration->i . ' мин';
                                            ?>
                                            <tr>
                                                <td>
                                                    <strong><?=
$departure->format('H:i') ?></strong>
                                                </td>
                                                <td>
                                                    <strong><?=
$arrival->format('H:i') ?></strong>
                                                    <?php if
($departure->format('Y-m-d') !== $arrival->format('Y-m-d')):
?>
                                                        <div
class="text-muted small">
                                                            <?=
$arrival->format('d.m.Y') ?>
                                                        </div>
                                                    <?php endif;
?>
                                                </td>
                                                <td>
                                                    <?=
$duration_text ?>
                                                </td>
                                                <td>
                                                    <?=
htmlspecialchars($route['train_number']) ?>
                                                    <div
class="text-muted small"><?=
htmlspecialchars($route['train_type']) ?></div>
                                                </td>
                                                <td>
                                                    <?php if
($route['available_seats'] > 10): ?>
                                                        <span
class="text-success">
                                                            <i
class="bi bi-check-circle"></i> <?= $route['available_seats']
?> мест
                                                        </span>
```

```
                                                    <?php elseif
($route['available_seats'] > 0): ?>
                                                        <span
class="text-warning">
                                                            <i
class="bi bi-exclamation-circle"></i> Осталось <?=
$route['available_seats'] ?> мест
                                                        </span>
                                                    <?php else:
?>
                                                        <span
class="text-danger">
                                                            <i
class="bi bi-x-circle"></i> Нет мест
                                                        </span>
                                                    <?php endif;
?>
                                                </td>
                                                <td>
                                                    <strong><?=
number_format($route['price'], 0, ',', ' ') ?> ₽</strong>
                                                </td>
                                                <td>
                                                    <?php if
(isLoggedIn()): ?>
                                                        <?php if
($route['available_seats'] > 0): ?>
                                                            <a
href="book.php?route_id=<?= $route['id'] ?>" class="btn btn-
primary btn-sm">

<i class="bi bi-ticket-perforated"></i> Купить билет
                                                            </a>
                                                        <?php
else: ?>

<button class="btn btn-secondary btn-sm" disabled>Нет
мест</button>
                                                        <?php
endif; ?>
                                                    <?php else:
?>
                                                        <a
href="login.php" class="btn btn-outline-primary btn-sm">

Войдите для покупки
```

```
                                                    </a>
                                                <?php endif;
    ?>
                                                    </td>
                                                </tr>
                                            <?php endforeach; ?>
                                        </tbody>
                                    </table>
                                </div>
                            </div>
                        </div>
                    <?php endforeach; ?>
                <?php endif; ?>
            </div>

            <script
        src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/boo
        tstrap.bundle.min.js"></script>
        </body>
        </html>
```

Файл services.php

```
<?php
require_once 'config/database.php';
require_once 'includes/functions.php';
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Услуги - Система продажи билетов на поезд</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootst
rap.min.css" rel="stylesheet">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.7.2/font/bootstrap-icons.css">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark mb-4">
        <div class="container">
            <a class="navbar-brand" href="/">Ж/Д Билеты</a>
            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav">
                <span class="navbar-toggler-icon"></span>
            </button>
```

```
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link active"
href="/services.php">Услуги</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="/rules.php">Правила перевозок</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="/faq.php">FAQ</a>
                    </li>
                </ul>
                <ul class="navbar-nav ms-auto">
                    <?php if (isLoggedIn()): ?>
                        <?php if (isAdmin()): ?>
                            <li class="nav-item">
                                <a class="nav-link"
href="/admin/">Панель администратора</a>
                            </li>
                        <?php endif; ?>
                        <li class="nav-item">
                            <a class="nav-link"
href="/profile.php">Личный кабинет</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link"
href="/logout.php">Выход</a>
                        </li>
                    <?php else: ?>
                        <li class="nav-item">
                            <a class="nav-link"
href="/login.php">Вход</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link"
href="/register.php">Регистрация</a>
                        </li>
                    <?php endif; ?>
                </ul>
            </div>
        </div>
    </nav>

    <div class="container">
        <h1 class="mb-4">Наши услуги</h1>

        <div class="row g-4">
            <div class="col-md-6">
                <div class="card h-100">
```

```
                    <div class="card-body">
                        <h3 class="card-title">
                            <i class="bi bi-ticket-perforated
text-primary"></i>
                            Покупка билетов онлайн
                        </h3>
                        <p class="card-text">
                            Удобное бронирование и покупка билетов
через интернет без очередей.
                            Оплата картой любого банка.
Электронный билет приходит на email.
                        </p>
                        <ul class="list-unstyled">
                            <li><i class="bi bi-check-circle-fill
text-success"></i> Бронирование за 2 минуты</li>
                            <li><i class="bi bi-check-circle-fill
text-success"></i> Безопасная оплата</li>
                            <li><i class="bi bi-check-circle-fill
text-success"></i> Возврат билетов онлайн</li>
                        </ul>
                    </div>
                </div>
            </div>

            <div class="col-md-6">
                <div class="card h-100">
                    <div class="card-body">
                        <h3 class="card-title">
                            <i class="bi bi-person-lines-fill
text-primary"></i>
                            Выбор места
                        </h3>
                        <p class="card-text">
                            Выбирайте удобные места при покупке
билета.
                            Система автоматически подберет
оптимальные варианты размещения.
                        </p>
                        <ul class="list-unstyled">
                            <li><i class="bi bi-check-circle-fill
text-success"></i> Интерактивная схема вагона</li>
                            <li><i class="bi bi-check-circle-fill
text-success"></i> Выбор предпочтительных мест</li>
                            <li><i class="bi bi-check-circle-fill
text-success"></i> Места рядом для групп</li>
                        </ul>
                    </div>
                </div>
            </div>

            <div class="col-md-6">
                <div class="card h-100">
```

```
                    <div class="card-body">
                        <h3 class="card-title">
                            <i class="bi bi-clock-history text-
primary"></i>
                            Управление бронированием
                        </h3>
                        <p class="card-text">
                            Полный контроль над вашими билетами в
личном кабинете.
                            Просмотр истории поездок, возврат
билетов, обмен билетов.
                        </p>
                        <ul class="list-unstyled">
                            <li><i class="bi bi-check-circle-fill
text-success"></i> История бронирований</li>
                            <li><i class="bi bi-check-circle-fill
text-success"></i> Возврат билетов</li>
                            <li><i class="bi bi-check-circle-fill
text-success"></i> Обмен билетов</li>
                        </ul>
                    </div>
                </div>
            </div>

            <div class="col-md-6">
                <div class="card h-100">
                    <div class="card-body">
                        <h3 class="card-title">
                            <i class="bi bi-bell text-
primary"></i>
                            Уведомления
                        </h3>
                        <p class="card-text">
                            Будьте в курсе изменений в расписании
и специальных предложений.
                            Получайте уведомления по email или
SMS.
                        </p>
                        <ul class="list-unstyled">
                            <li><i class="bi bi-check-circle-fill
text-success"></i> Изменения в расписании</li>
                            <li><i class="bi bi-check-circle-fill
text-success"></i> Напоминания о поездке</li>
                            <li><i class="bi bi-check-circle-fill
text-success"></i> Специальные предложения</li>
                        </ul>
                    </div>
                </div>
            </div>
        </div>

        <div class="mt-5">
```

```
            <h2 class="mb-4">Дополнительные услуги</h2>

            <div class="table-responsive">
                <table class="table table-striped">
                    <thead>
                        <tr>
                            <th>Услуга</th>
                            <th>Описание</th>
                            <th>Стоимость</th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>
                            <td>Страхование</td>
                            <td>Страхование от несчастных случаев
на время поездки</td>
                            <td>от 100 ₽</td>
                        </tr>
                        <tr>
                            <td>Питание в пути</td>
                            <td>Заказ питания с доставкой к
месту</td>
                            <td>от 300 ₽</td>
                        </tr>
                        <tr>
                            <td>VIP-зал ожидания</td>
                            <td>Доступ в комфортабельный зал
ожидания</td>
                            <td>от 500 ₽</td>
                        </tr>
                        <tr>
                            <td>Перевозка багажа</td>
                            <td>Дополнительный багаж сверх
нормы</td>
                            <td>от 200 ₽</td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </div>
    </div>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstra
p.bundle.min.js"></script>
</body>
</html>
Файл index.php
<?php
require_once '../config/database.php';
require_once '../includes/functions.php';
```

```
requireLogin();
requireAdmin();

// Получаем статистику
$stats = [
    'routes' => $pdo->query("SELECT COUNT(*) FROM routes")-
>fetchColumn(),
    'users' => $pdo->query("SELECT COUNT(*) FROM users WHERE role
= 'user'")->fetchColumn(),
    'tickets' => $pdo->query("SELECT COUNT(*) FROM tickets WHERE
status = 'confirmed'")->fetchColumn(),
    'revenue' => $pdo->query("SELECT COALESCE(SUM(price_paid), 0)
FROM tickets WHERE status = 'confirmed'")->fetchColumn()
];

// Получаем последние билеты
$stmt = $pdo->query("
    SELECT t.*, u.username,
           r.departure_station, r.arrival_station,
           r.departure_time, r.train_number
    FROM tickets t
    JOIN users u ON t.user_id = u.id
    JOIN routes r ON t.route_id = r.id
    WHERE t.status = 'confirmed'
    ORDER BY t.created_at DESC
    LIMIT 5
");
$recent_tickets = $stmt->fetchAll(PDO::FETCH_ASSOC);

// Получаем популярные маршруты
$stmt = $pdo->query("
    SELECT r.*,
           COUNT(t.id) as tickets_count,
           (r.total_seats - r.available_seats) as seats_taken
    FROM routes r
    LEFT JOIN tickets t ON r.id = t.route_id AND t.status =
'confirmed'
    GROUP BY r.id
    ORDER BY tickets_count DESC
    LIMIT 5
");
$popular_routes = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Панель администратора - Система продажи билетов</title>
```

```
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootst
rap.min.css" rel="stylesheet">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.7.2/font/bootstrap-icons.css">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark mb-4">
        <div class="container">
            <a class="navbar-brand" href="/admin/">Панель
администратора</a>
            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link"
href="/admin/routes.php">Управление маршрутами</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="/admin/users.php">Управление пользователями</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="/admin/tickets.php">Управление билетами</a>
                    </li>
                </ul>
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item">
                        <a class="nav-link" href="/">На сайт</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="/logout.php">Выход</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <div class="container">
        <h1 class="mb-4">Панель администратора</h1>

        <?= displayFlashMessage() ?>

        <div class="row mb-4">
            <div class="col-md-3">
                <div class="card text-white bg-primary">
```

```
                    <div class="card-body">
                        <h5 class="card-title">Маршрутов</h5>
                        <p class="card-text display-4"><?=
number_format($stats['routes']) ?></p>
                    </div>
                </div>
            </div>
            <div class="col-md-3">
                <div class="card text-white bg-success">
                    <div class="card-body">
                        <h5 class="card-title">Пользователей</h5>
                        <p class="card-text display-4"><?=
number_format($stats['users']) ?></p>
                    </div>
                </div>
            </div>
            <div class="col-md-3">
                <div class="card text-white bg-info">
                    <div class="card-body">
                        <h5 class="card-title">Всего билетов</h5>
                        <p class="card-text display-4"><?=
number_format($stats['tickets']) ?></p>
                    </div>
                </div>
            </div>
            <div class="col-md-3">
                <div class="card text-white bg-warning">
                    <div class="card-body">
                        <h5 class="card-title">Общая выручка</h5>
                        <p class="card-text display-4"><?=
number_format($stats['revenue'], 2) ?> ₽</p>
                    </div>
                </div>
            </div>
        </div>

        <div class="row">
            <div class="col-md-6">
                <div class="card mb-4">
                    <div class="card-header">
                        <h3 class="card-title mb-0">Последние
билеты</h3>
                    </div>
                    <div class="card-body">
                        <?php if (empty($recent_tickets)): ?>
                            <p class="text-muted">Нет проданных
билетов</p>
                        <?php else: ?>
                            <div class="table-responsive">
                                <table class="table table-
striped">
                                    <thead>
```

```
                                        <tr>
                                            <th>Пользователь</th>
                                            <th>Маршрут</th>
                                            <th>Дата</th>
                                        </tr>
                                    </thead>
                                    <tbody>
                                        <?php foreach
($recent_tickets as $ticket): ?>
                                            <tr>
                                                <td><?=
htmlspecialchars($ticket['username']) ?></td>
                                                <td>
                                                    <?=
htmlspecialchars($ticket['departure_station']) ?> →
                                                    <?=
htmlspecialchars($ticket['arrival_station']) ?><br>
                                                    <small
class="text-muted">
                                                        Поезд <?=
htmlspecialchars($ticket['train_number']) ?>
                                                    </small>
                                                </td>
                                                <td>
                                                    <?=
date('d.m.Y H:i', strtotime($ticket['departure_time'])) ?>
                                                </td>
                                            </tr>
                                        <?php endforeach; ?>
                                    </tbody>
                                </table>
                            </div>
                        <?php endif; ?>
                    </div>
                </div>

                <div class="col-md-6">
                    <div class="card mb-4">
                        <div class="card-header">
                            <h3 class="card-title mb-0">Популярные
маршруты</h3>
                        </div>
                        <div class="card-body">
                            <?php if (empty($popular_routes)): ?>
                                <p class="text-muted">Нет данных о
популярных маршрутах</p>
                            <?php else: ?>
                                <div class="table-responsive">
                                    <table class="table table-
striped">
                                        <thead>
```

```
                                        <tr>
                                            <th>Маршрут</th>
                                            <th>Билетов
продано</th>
                                            <th>Загруженность</th>
                                        </tr>
                                    </thead>
                                    <tbody>
                                        <?php foreach
($popular_routes as $route): ?>
                                            <tr>
                                                <td>
                                                    <?=
htmlspecialchars($route['departure_station']) ?> →
                                                    <?=
htmlspecialchars($route['arrival_station']) ?><br>
                                                    <small
class="text-muted">
                                                        Поезд <?=
htmlspecialchars($route['train_number']) ?>
                                                    </small>
                                                </td>
                                                <td><?=
$route['tickets_count'] ?></td>
                                                <td>
                                                    <?php
                                                    $percentage =
($route['seats_taken'] / $route['total_seats']) * 100;
                                                    $bg_class =
$percentage > 80 ? 'bg-danger' :

($percentage > 50 ? 'bg-warning' : 'bg-success');
                                                    ?>
                                                    <div
class="progress">
                                                        <div

class="progress-bar <?= $bg_class ?>"

role="progressbar"

style="width: <?= $percentage ?>%"
                                                             aria-

valuenow="<?= $percentage ?>"
                                                             aria-

valuemin="0"
                                                             aria-

valuemax="100">
                                                            <?=

round($percentage) ?>%
                                                        </div>
                                                    </div>
                                                </td>
```

```
                                        </tr>
                                    <?php endforeach; ?>
                                </tbody>
                            </table>
                        </div>
                    <?php endif; ?>
                </div>
            </div>
        </div>
    </div>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstra
p.bundle.min.js"></script>
</body>
</html>
```

Файл routes.php

```
<?php
require_once '../config/database.php';
require_once '../includes/functions.php';

requireLogin();
requireAdmin();

// Обработка удаления маршрута
if (isset($_POST['delete_route'])) {
    $route_id = (int)$_POST['route_id'];

    try {
        $pdo->beginTransaction();

        // Проверяем, есть ли билеты на этот маршрут
        $stmt = $pdo->prepare("SELECT COUNT(*) FROM tickets WHERE
route_id = ?");
        $stmt->execute([$route_id]);
        if ($stmt->fetchColumn() > 0) {
            throw new Exception('Нельзя удалить маршрут, на
который уже куплены билеты');
        }

        // Удаляем маршрут
        $stmt = $pdo->prepare("DELETE FROM routes WHERE id = ?");
        $stmt->execute([$route_id]);

        $pdo->commit();
        flashMessage('Маршрут успешно удален');

    } catch (Exception $e) {
        $pdo->rollBack();
```

```
            flashMessage($e->getMessage(), 'danger');
        }

        redirect('/admin/routes.php');
    }

    // Обработка добавления/редактирования маршрута
    if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
    isset($_POST['save_route'])) {
        $route_id = isset($_POST['route_id']) ?
    (int)$_POST['route_id'] : null;
        $train_number = trim($_POST['train_number']);
        $train_type = trim($_POST['train_type']);
        $departure_station = trim($_POST['departure_station']);
        $arrival_station = trim($_POST['arrival_station']);
        $departure_time = $_POST['departure_time'];
        $arrival_time = $_POST['arrival_time'];
        $total_seats = (int)$_POST['total_seats'];
        $price = (float)$_POST['price'];

        $errors = [];

        if (empty($train_number)) {
            $errors[] = 'Номер поезда обязателен';
        }

        if (empty($train_type)) {
            $errors[] = 'Тип поезда обязателен';
        }

        if (empty($departure_station) || empty($arrival_station)) {
            $errors[] = 'Станции отправления и прибытия обязательны';
        }

        if (empty($departure_time) || empty($arrival_time)) {
            $errors[] = 'Время отправления и прибытия обязательны';
        } elseif (strtotime($departure_time) >=
    strtotime($arrival_time)) {
            $errors[] = 'Время прибытия должно быть позже времени
    отправления';
        }

        if ($total_seats <= 0) {
            $errors[] = 'Количество мест должно быть положительным
    числом';
        }

        if ($price <= 0) {
            $errors[] = 'Цена должна быть положительным числом';
        }

        if (empty($errors)) {
```

```
        try {
            if ($route_id) {
                // Обновление существующего маршрута
                $stmt = $pdo->prepare("
                    UPDATE routes
                    SET train_number = ?, train_type = ?,
                        departure_station = ?, arrival_station =
?,
                        departure_time = ?, arrival_time = ?,
                        total_seats = ?, available_seats =
total_seats -
                            (SELECT COUNT(*) FROM tickets WHERE
route_id = ? AND status != 'cancelled'),
                        price = ?
                    WHERE id = ?
                ");
                $stmt->execute([
                    $train_number, $train_type,
                    $departure_station, $arrival_station,
                    $departure_time, $arrival_time,
                    $total_seats, $route_id,
                    $price, $route_id
                ]);
                flashMessage('Маршрут успешно обновлен');
            } else {
                // Добавление нового маршрута
                $stmt = $pdo->prepare("
                    INSERT INTO routes
                    (train_number, train_type, departure_station,
arrival_station,
                     departure_time, arrival_time, total_seats,
available_seats, price)
                    VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)
                ");
                $stmt->execute([
                    $train_number, $train_type,
                    $departure_station, $arrival_station,
                    $departure_time, $arrival_time,
                    $total_seats, $total_seats,
                    $price
                ]);
                flashMessage('Маршрут успешно добавлен');
            }
            redirect('/admin/routes.php');
        } catch (PDOException $e) {
            $errors[] = 'Ошибка при сохранении маршрута: ' . $e-
>getMessage();
        }
    }
}

// Получение маршрута для редактирования
```

```
$edit_route = null;
if (isset($_GET['edit']) && is_numeric($_GET['edit'])) {
    $stmt = $pdo->prepare("SELECT * FROM routes WHERE id = ?");
    $stmt->execute([(int)$_GET['edit']]);
    $edit_route = $stmt->fetch(PDO::FETCH_ASSOC);
}

// Получаем список маршрутов с подсчетом проданных билетов
$stmt = $pdo->prepare("
    SELECT r.*,
           COALESCE(COUNT(CASE WHEN t.status = 'confirmed' THEN 1
END), 0) as tickets_sold
    FROM routes r
    LEFT JOIN tickets t ON r.id = t.route_id
    GROUP BY r.id
    ORDER BY r.departure_time DESC
");
$stmt->execute();
$routes = $stmt->fetchAll(PDO::FETCH_ASSOC);

// Список типов поездов для выбора
$train_types = ['Скорый', 'Фирменный', 'Пассажирский',
'Экспресс'];
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Управление маршрутами - Система продажи билетов</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootst
rap.min.css" rel="stylesheet">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark mb-4">
        <div class="container">
            <a class="navbar-brand" href="/admin/">Панель
администратора</a>
            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link active"
href="/admin/routes.php">Управление маршрутами</a>
                    </li>
                    <li class="nav-item">
```

```
                        <a class="nav-link"
href="/admin/users.php">Управление пользователями</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="/admin/tickets.php">Управление билетами</a>
                    </li>
                </ul>
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item">
                        <a class="nav-link" href="/">На сайт</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="/logout.php">Выход</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <div class="container">
        <h1 class="mb-4">Управление маршрутами</h1>

        <?= displayFlashMessage() ?>

        <?php if (!empty($errors)): ?>
            <div class="alert alert-danger">
                <ul class="mb-0">
                    <?php foreach ($errors as $error): ?>
                        <li><?= $error ?></li>
                    <?php endforeach; ?>
                </ul>
            </div>
        <?php endif; ?>

        <div class="card mb-4">
            <div class="card-header">
                <h3 class="card-title mb-0">
                    <?= $edit_route ? 'Редактировать маршрут' :
'Добавить маршрут' ?>
                </h3>
            </div>
            <div class="card-body">
                <form method="POST" action="">
                    <?php if ($edit_route): ?>
                        <input type="hidden" name="route_id"
value="<?= $edit_route['id'] ?>">
                    <?php endif; ?>

                    <div class="row">
                        <div class="col-md-6 mb-3">
```

```
                            <label for="train_number" class="form-
label">Номер поезда</label>
                            <input type="text" class="form-
control" id="train_number" name="train_number"
                                   value="<?= $edit_route ?
sanitize($edit_route['train_number']) : '' ?>" required>
                        </div>

                        <div class="col-md-6 mb-3">
                            <label for="train_type" class="form-
label">Тип поезда</label>
                            <select class="form-select"
id="train_type" name="train_type" required>
                                <option value="">Выберите тип
поезда</option>
                                <?php foreach ($train_types as
$type): ?>
                                    <option value="<?= $type ?>"
<?= $edit_route && $edit_route['train_type'] === $type ?
'selected' : '' ?>>
                                        <?= $type ?>
                                    </option>
                                <?php endforeach; ?>
                            </select>
                        </div>
                    </div>

                    <div class="row">
                        <div class="col-md-6 mb-3">
                            <label for="departure_station"
class="form-label">Станция отправления</label>
                            <input type="text" class="form-
control" id="departure_station" name="departure_station"
                                   value="<?= $edit_route ?
sanitize($edit_route['departure_station']) : '' ?>" required>
                        </div>

                        <div class="col-md-6 mb-3">
                            <label for="arrival_station"
class="form-label">Станция прибытия</label>
                            <input type="text" class="form-
control" id="arrival_station" name="arrival_station"
                                   value="<?= $edit_route ?
sanitize($edit_route['arrival_station']) : '' ?>" required>
                        </div>
                    </div>

                    <div class="row">
                        <div class="col-md-6 mb-3">
                            <label for="departure_time"
class="form-label">Время отправления</label>
```

```
                                <input type="datetime-local"
class="form-control" id="departure_time" name="departure_time"
                                       value="<?= $edit_route ?
date('Y-m-d\TH:i', strtotime($edit_route['departure_time'])) : ''
?>" required>
                            </div>

                            <div class="col-md-6 mb-3">
                                <label for="arrival_time" class="form-
label">Время прибытия</label>
                                <input type="datetime-local"
class="form-control" id="arrival_time" name="arrival_time"
                                       value="<?= $edit_route ?
date('Y-m-d\TH:i', strtotime($edit_route['arrival_time'])) : ''
?>" required>
                            </div>
                        </div>

                        <div class="row">
                            <div class="col-md-6 mb-3">
                                <label for="total_seats" class="form-
label">Всего мест</label>
                                <input type="number" class="form-
control" id="total_seats" name="total_seats"
                                       value="<?= $edit_route ?
$edit_route['total_seats'] : '100' ?>" required min="1">
                            </div>

                            <div class="col-md-6 mb-3">
                                <label for="price" class="form-
label">Цена</label>
                                <div class="input-group">
                                    <input type="number" class="form-
control" id="price" name="price" step="0.01"
                                           value="<?= $edit_route ?
$edit_route['price'] : '1000.00' ?>" required min="0">
                                    <span class="input-group-
text">₽</span>
                                </div>
                            </div>
                        </div>

                        <div class="d-grid gap-2">
                            <button type="submit" name="save_route"
class="btn btn-primary">
                                <?= $edit_route ? 'Сохранить
изменения' : 'Добавить маршрут' ?>
                            </button>
                            <?php if ($edit_route): ?>
                                <a href="/admin/routes.php" class="btn
btn-secondary">Отмена</a>
                            <?php endif; ?>
```

```
                    </div>
                </form>
            </div>
        </div>

        <div class="card">
            <div class="card-header">
                <h3 class="card-title mb-0">Список маршрутов</h3>
            </div>
            <div class="card-body">
                <div class="table-responsive">
                    <table class="table table-striped">
                        <thead>
                            <tr>
                                <th>ID</th>
                                <th>Поезд</th>
                                <th>Маршрут</th>
                                <th>Отправление</th>
                                <th>Прибытие</th>
                                <th>В пути</th>
                                <th>Места</th>
                                <th>Цена</th>
                                <th>Действия</th>
                            </tr>
                        </thead>
                        <tbody>
                            <?php foreach ($routes as $route): ?>
                                <?php
                                $departure = new
DateTime($route['departure_time']);
                                $arrival = new
DateTime($route['arrival_time']);
                                $duration = $departure-
>diff($arrival);
                                $duration_text = '';

                                if ($duration->d > 0) {
                                    $duration_text .= $duration->d
. ' д ';
                                }
                                if ($duration->h > 0 || $duration-
>d > 0) {
                                    $duration_text .= $duration->h
. ' ч ';
                                }
                                $duration_text .= $duration->i . '
мин';
                                ?>
                                <tr>
                                    <td><?= $route['id'] ?></td>
                                    <td>
```

```
                                        <?=
htmlspecialchars($route['train_number']) ?><br>
                                        <small class="text-
muted"><?= htmlspecialchars($route['train_type']) ?></small>
                                    </td>
                                    <td>
                                        <?=
htmlspecialchars($route['departure_station']) ?> →
                                        <?=
htmlspecialchars($route['arrival_station']) ?>
                                    </td>
                                    <td><?= date('d.m.Y H:i',
strtotime($route['departure_time'])) ?></td>
                                    <td><?= date('d.m.Y H:i',
strtotime($route['arrival_time'])) ?></td>
                                    <td><?= $duration_text ?></td>
                                    <td>
                                        <?=
$route['available_seats'] ?> / <?= $route['total_seats'] ?><br>
                                        <small class="text-
muted">продано: <?= $route['tickets_sold'] ?></small>
                                    </td>
                                    <td><?=
number_format($route['price'], 2) ?>₽</td>
                                    <td>
                                        <div class="btn-group">
                                            <a href="?edit=<?=
$route['id'] ?>" class="btn btn-primary btn-sm">
                                                Редактировать
                                            </a>
                                            <?php if
($route['tickets_sold'] == 0): ?>
                                                <form
method="POST" action="" class="d-inline"

onsubmit="return confirm('Вы уверены, что хотите удалить этот
маршрут?')">
                                                    <input
type="hidden" name="route_id" value="<?= $route['id'] ?>">
                                                    <button
type="submit" name="delete_route" class="btn btn-danger btn-sm">
                                                        Удалить
                                                    </button>
                                                </form>
                                            <?php endif; ?>
                                        </div>
                                    </td>
                                </tr>
                            <?php endforeach; ?>
                        </tbody>
                    </table>
                </div>
```

```
            </div>
        </div>
    </div>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstra
p.bundle.min.js"></script>
</body>
</html>

Файл tickets.php
<?php
require_once '../config/database.php';
require_once '../includes/functions.php';

requireLogin();
requireAdmin();

// Обработка изменения статуса билета
if (isset($_POST['change_status'])) {
    $ticket_id = (int)$_POST['ticket_id'];
    $new_status = $_POST['status'];

    if (!in_array($new_status, ['pending', 'confirmed',
'cancelled'])) {
        flashMessage('Неверный статус билета', 'danger');
        redirect('/admin/tickets.php');
    }

    try {
        $pdo->beginTransaction();

        // Получаем информацию о билете
        $stmt = $pdo->prepare("
            SELECT t.*, r.available_seats
            FROM tickets t
            JOIN routes r ON t.route_id = r.id
            WHERE t.id = ?
            FOR UPDATE
        ");
        $stmt->execute([$ticket_id]);
        $ticket = $stmt->fetch(PDO::FETCH_ASSOC);

        if (!$ticket) {
            throw new Exception('Билет не найден');
        }

        // Если меняем статус на "отменен", увеличиваем количество
свободных мест
        if ($new_status === 'cancelled' && $ticket['status'] !==
'cancelled') {
            $stmt = $pdo->prepare("
```

```
                UPDATE routes
                SET available_seats = available_seats + 1
                WHERE id = ?
            ");
            $stmt->execute([$ticket['route_id']]);
        }
        // Если меняем с "отменен" на другой статус, уменьшаем
количество свободных мест
        elseif ($ticket['status'] === 'cancelled' && $new_status
!== 'cancelled') {
            if ($ticket['available_seats'] <= 0) {
                throw new Exception('Нет свободных мест для
восстановления билета');
            }

            $stmt = $pdo->prepare("
                UPDATE routes
                SET available_seats = available_seats - 1
                WHERE id = ?
            ");
            $stmt->execute([$ticket['route_id']]);
        }

        // Обновляем статус билета
        $stmt = $pdo->prepare("UPDATE tickets SET status = ? WHERE
id = ?");
        $stmt->execute([$new_status, $ticket_id]);

        $pdo->commit();
        flashMessage('Статус билета успешно изменен');

    } catch (Exception $e) {
        $pdo->rollBack();
        flashMessage($e->getMessage(), 'danger');
    }

    redirect('/admin/tickets.php');
}

// Получение списка билетов с фильтрацией
$where = [];
$params = [];

if (isset($_GET['status']) && in_array($_GET['status'],
['pending', 'confirmed', 'cancelled'])) {
    $where[] = "t.status = ?";
    $params[] = $_GET['status'];
}

if (!empty($_GET['search'])) {
    $search = '%' . $_GET['search'] . '%';
    $where[] = "(
```

```
        u.username LIKE ? OR
        u.email LIKE ? OR
        r.departure_station LIKE ? OR
        r.arrival_station LIKE ?
    )";
    $params = array_merge($params, [$search, $search, $search,
$search]);
}

$sql = "
    SELECT t.*, u.username, u.email,
           r.departure_station, r.arrival_station,
           r.departure_time, r.arrival_time, r.price
    FROM tickets t
    JOIN users u ON t.user_id = u.id
    JOIN routes r ON t.route_id = r.id
";

if (!empty($where)) {
    $sql .= " WHERE " . implode(" AND ", $where);
}

$sql .= " ORDER BY t.created_at DESC";

$stmt = $pdo->prepare($sql);
$stmt->execute($params);
$tickets = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Управление билетами - Система продажи билетов</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootst
rap.min.css" rel="stylesheet">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark mb-4">
        <div class="container">
            <a class="navbar-brand" href="/admin/">Панель
администратора</a>
            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
```

```
                        <a class="nav-link"
href="/admin/routes.php">Управление маршрутами</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="/admin/users.php">Управление пользователями</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link active"
href="/admin/tickets.php">Управление билетами</a>
                    </li>
                </ul>
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item">
                        <a class="nav-link" href="/">На сайт</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="/logout.php">Выход</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <div class="container">
        <h1 class="mb-4">Управление билетами</h1>

        <?= displayFlashMessage() ?>

        <div class="card">
            <div class="card-header">
                <div class="row align-items-center">
                    <div class="col">
                        <h3 class="card-title mb-0">Список
билетов</h3>
                    </div>
                    <div class="col-auto">
                        <form class="d-flex" method="GET"
action="">
                            <select name="status" class="form-
select me-2">
                                <option value="">Все
статусы</option>
                                <option value="pending" <?=
isset($_GET['status']) && $_GET['status'] === 'pending' ?
'selected' : '' ?>>
                                    Ожидает подтверждения
                                </option>
                                <option value="confirmed" <?=
isset($_GET['status']) && $_GET['status'] === 'confirmed' ?
'selected' : '' ?>>
```

```
                                    Подтвержден
                                </option>
                                <option value="cancelled" <?=
isset($_GET['status']) && $_GET['status'] === 'cancelled' ?
'selected' : '' ?>>
                                    Отменен
                                </option>
                            </select>
                            <input type="search" name="search"
class="form-control me-2" placeholder="Поиск..."
                                   value="<?=
isset($_GET['search']) ? sanitize($_GET['search']) : '' ?>">
                            <button type="submit" class="btn btn-
primary">Найти</button>
                        </form>
                    </div>
                </div>
            </div>
            <div class="card-body">
                <div class="table-responsive">
                    <table class="table table-striped">
                        <thead>
                            <tr>
                                <th>ID</th>
                                <th>Пользователь</th>
                                <th>Маршрут</th>
                                <th>Отправление</th>
                                <th>Прибытие</th>
                                <th>Место</th>
                                <th>Статус</th>
                                <th>Цена</th>
                                <th>Дата создания</th>
                                <th>Действия</th>
                            </tr>
                        </thead>
                        <tbody>
                            <?php foreach ($tickets as $ticket):
?>
                                <tr>
                                    <td><?= $ticket['id'] ?></td>
                                    <td>
                                        <?=
sanitize($ticket['username']) ?>
                                        <br>
                                        <small class="text-
muted"><?= sanitize($ticket['email']) ?></small>
                                    </td>
                                    <td>
                                        <?=
sanitize($ticket['departure_station']) ?> →
                                        <?=
sanitize($ticket['arrival_station']) ?>
```

```
                                        </td>
                                        <td><?= date('d.m.Y H:i',
strtotime($ticket['departure_time'])) ?></td>
                                        <td><?= date('d.m.Y H:i',
strtotime($ticket['arrival_time'])) ?></td>
                                        <td><?= $ticket['seat_number']
?></td>
                                        <td>
                                            <?php
                                            $status_class = [
                                                'pending' =>
'warning',
                                                'confirmed' =>
'success',
                                                'cancelled' =>
'danger'
                                            ][$ticket['status']];

                                            $status_text = [
                                                'pending' => 'Ожидает
подтверждения',
                                                'confirmed' =>
'Подтвержден',
                                                'cancelled' =>
'Отменен'
                                            ][$ticket['status']];
                                            ?>
                                            <span class="badge bg-<?=
$status_class ?>"><?= $status_text ?></span>
                                        </td>
                                        <td><?=
number_format($ticket['price'], 2) ?> ₽</td>
                                        <td><?= date('d.m.Y H:i',
strtotime($ticket['created_at'])) ?></td>
                                        <td>
                                            <div class="btn-group">
                                                <?php if
($ticket['status'] !== 'confirmed'): ?>
                                                    <form
method="POST" action="" class="d-inline">
                                                        <input
type="hidden" name="ticket_id" value="<?= $ticket['id'] ?>">
                                                        <input
type="hidden" name="status" value="confirmed">
                                                        <button
type="submit" name="change_status"
class="btn btn-success btn-sm"
onclick="return confirm('Подтвердить билет?')">
Подтвердить
```

```
                                                    </button>
                                                </form>
                                            <?php endif; ?>

                                            <?php if
($ticket['status'] !== 'cancelled'): ?>
                                                <form
method="POST" action="" class="d-inline">
                                                    <input
type="hidden" name="ticket_id" value="<?= $ticket['id'] ?>">
                                                    <input
type="hidden" name="status" value="cancelled">
                                                    <button
type="submit" name="change_status"

class="btn btn-danger btn-sm"

onclick="return confirm('Отменить билет?')">
                                                        Отменить
                                                    </button>
                                                </form>
                                            <?php endif; ?>
                                        </div>
                                    </td>
                                </tr>
                            <?php endforeach; ?>
                        </tbody>
                    </table>
                </div>
            </div>
        </div>
    </div>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstra
p.bundle.min.js"></script>
</body>
</html>
```

Файл users.php

```
<?php
require_once '../config/database.php';
require_once '../includes/functions.php';

requireLogin();
requireAdmin();

// Обработка изменения роли пользователя
if (isset($_POST['change_role'])) {
    $user_id = (int)$_POST['user_id'];
    $new_role = $_POST['role'] === 'admin' ? 'admin' : 'user';
```

```
    try {
        // Нельзя понизить роль самому себе
        if ($user_id === $_SESSION['user_id']) {
            throw new Exception('Вы не можете изменить свою
собственную роль');
        }

        $stmt = $pdo->prepare("UPDATE users SET role = ? WHERE id
= ?");
        $stmt->execute([$new_role, $user_id]);

        flashMessage('Роль пользователя успешно изменена');
    } catch (Exception $e) {
        flashMessage($e->getMessage(), 'danger');
    }

    redirect('/admin/users.php');
}

// Обработка блокировки/разблокировки пользователя
if (isset($_POST['toggle_block'])) {
    $user_id = (int)$_POST['user_id'];
    $is_blocked = (bool)$_POST['is_blocked'];

    try {
        // Нельзя заблокировать самого себя
        if ($user_id === $_SESSION['user_id']) {
            throw new Exception('Вы не можете заблокировать самого
себя');
        }

        $stmt = $pdo->prepare("UPDATE users SET is_blocked = ?
WHERE id = ?");
        $stmt->execute([$is_blocked ? 0 : 1, $user_id]);

        flashMessage($is_blocked ? 'Пользователь разблокирован' :
'Пользователь заблокирован');
    } catch (Exception $e) {
        flashMessage($e->getMessage(), 'danger');
    }

    redirect('/admin/users.php');
}

// Получение списка пользователей с их статистикой
$stmt = $pdo->query("
    SELECT u.*,
           COUNT(t.id) as total_tickets,
           SUM(CASE WHEN t.status = 'confirmed' THEN 1 ELSE 0 END)
as active_tickets
    FROM users u
    LEFT JOIN tickets t ON u.id = t.user_id
```

```
    GROUP BY u.id
    ORDER BY u.created_at DESC
");
$users = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Управление пользователями - Система продажи
билетов</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootst
rap.min.css" rel="stylesheet">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark mb-4">
        <div class="container">
            <a class="navbar-brand" href="/admin/">Панель
администратора</a>
            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link"
href="/admin/routes.php">Управление маршрутами</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link active"
href="/admin/users.php">Управление пользователями</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="/admin/tickets.php">Управление билетами</a>
                    </li>
                </ul>
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item">
                        <a class="nav-link" href="/">На сайт</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="/logout.php">Выход</a>
                    </li>
                </ul>
            </div>
```

```
        </div>
    </nav>

    <div class="container">
        <h1 class="mb-4">Управление пользователями</h1>

        <?= displayFlashMessage() ?>

        <div class="card">
            <div class="card-header">
                <h3 class="card-title mb-0">Список
пользователей</h3>
            </div>
            <div class="card-body">
                <div class="table-responsive">
                    <table class="table table-striped">
                        <thead>
                            <tr>
                                <th>ID</th>
                                <th>Имя пользователя</th>
                                <th>Email</th>
                                <th>Роль</th>
                                <th>Статус</th>
                                <th>Билетов</th>
                                <th>Дата регистрации</th>
                                <th>Действия</th>
                            </tr>
                        </thead>
                        <tbody>
                            <?php foreach ($users as $user): ?>
                                <tr>
                                    <td><?= $user['id'] ?></td>
                                    <td><?=
sanitize($user['username']) ?></td>
                                    <td><?=
sanitize($user['email']) ?></td>
                                    <td>
                                        <span class="badge bg-<?=
$user['role'] === 'admin' ? 'danger' : 'primary' ?>">
                                            <?= $user['role'] ===
'admin' ? 'Администратор' : 'Пользователь' ?>
                                        </span>
                                    </td>
                                    <td>
                                        <?php if
(isset($user['is_blocked']) && $user['is_blocked']): ?>
                                            <span class="badge bg-
danger">Заблокирован</span>
                                        <?php else: ?>
                                            <span class="badge bg-
success">Активен</span>
                                        <?php endif; ?>
```

```
                                    </td>
                                    <td>
                                        <?=
$user['active_tickets'] ?> активных
                                        (всего: <?=
$user['total_tickets'] ?>)
                                    </td>
                                    <td><?= date('d.m.Y H:i',
strtotime($user['created_at'])) ?></td>
                                    <td>
                                        <?php if ($user['id'] !==
$_SESSION['user_id']): ?>
                                            <div class="btn-
group">
                                                <form
method="POST" action="" class="d-inline">
                                                    <input
type="hidden" name="user_id" value="<?= $user['id'] ?>">
                                                    <input
type="hidden" name="role"

value="<?= $user['role'] === 'admin' ? 'user' : 'admin' ?>">
                                                    <button
type="submit" name="change_role"

class="btn btn-warning btn-sm"

onclick="return confirm('Вы уверены, что хотите изменить роль
этого пользователя?')">
                                                        <?=
$user['role'] === 'admin' ? 'Снять админа' : 'Сделать админом' ?>
                                                    </button>
                                                </form>

                                                <form
method="POST" action="" class="d-inline">
                                                    <input
type="hidden" name="user_id" value="<?= $user['id'] ?>">
                                                    <input
type="hidden" name="is_blocked"

value="<?= isset($user['is_blocked']) ? $user['is_blocked'] : 0
?>">
                                                    <button
type="submit" name="toggle_block"

class="btn btn-<?= isset($user['is_blocked']) &&
$user['is_blocked'] ? 'success' : 'danger' ?> btn-sm"

onclick="return confirm('Вы уверены?')">
```

```
                                                        <?=
isset($user['is_blocked']) && $user['is_blocked'] ?
'Разблокировать' : 'Заблокировать' ?>
                                                            </button>
                                                        </form>
                                                    </div>
                                                <?php endif; ?>
                                            </td>
                                        </tr>
                                    <?php endforeach; ?>
                                </tbody>
                            </table>
                        </div>
                    </div>
                </div>
            </div>

            <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstra
p.bundle.min.js"></script>
</body>
</html>
```

Файл database.php

```
<?php
define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASS', '');
define('DB_NAME', 'train_tickets');

try {
    $pdo = new PDO(
        "mysql:host=" . DB_HOST . ";dbname=" . DB_NAME,
        DB_USER,
        DB_PASS,
        [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]
    );
} catch(PDOException $e) {
    die("Connection failed: " . $e->getMessage());
}
```

Файл functions.php

```
<?php
session_start();

function isLoggedIn() {
    return isset($_SESSION['user_id']);
}

function isAdmin() {
    return isset($_SESSION['role']) && $_SESSION['role'] ===
'admin';
}
```

```
function redirect($path) {
    header("Location: $path");
    exit();
}

function sanitize($input) {
    return htmlspecialchars(strip_tags($input), ENT_QUOTES, 'UTF-
8');
}

function flashMessage($message, $type = 'success') {
    $_SESSION['flash'] = [
        'message' => $message,
        'type' => $type
    ];
}

function displayFlashMessage() {
    if (isset($_SESSION['flash'])) {
        $message = $_SESSION['flash']['message'];
        $type = $_SESSION['flash']['type'];
        unset($_SESSION['flash']);
        return "<div class='alert alert-
{$type}'>{$message}</div>";
    }
    return '';
}

function requireLogin() {
    if (!isLoggedIn()) {
        flashMessage('Пожалуйста, войдите в систему', 'danger');
        redirect('/login.php');
    }
}

function requireAdmin() {
    if (!isAdmin()) {
        flashMessage('Доступ запрещен', 'danger');
        redirect('/index.php');
    }
     }
method="POST" action="" class="d-inline">
                                                    <input
type="hidden" name="user_id" value="<?= $user['id'] ?>">
                                                    <input
type="hidden" name="is_blocked"

value="<?= isset($user['is_blocked']) ? $user['is_blocked'] : 0
?>">
                                                    <button
type="submit" name="toggle_block"
```

```
class="btn btn-<?= isset($user['is_blocked']) &&
$user['is_blocked'] ? 'success' : 'danger' ?> btn-sm"

onclick="return confirm('Вы уверены?')">
```

Приложение Г

(обязательное)

Программа для регистрации РЖД билетов

Презентация