**Министерство образования и науки Кыргызской**

**Республики**

**Кыргызский Государственный Технический Университет**

**им. И. Раззакова**

**Институт электроники и телекоммуникаций**

**кафедра *«Информационные системы и технологии***

***им. акад. А. Жайнакова»***

**направления: 710200 Информационные системы и технологии**

**КУРСОВОЙ ПРОЕКТ**

**по дисциплине**

**«Проектирование и разработка веб-приложений**

**BECKEND»**

**на тему «Веб-приложение для онлайн-**

**резервирования мест в кинотеатре»**

**Разработал: студент гр. ИСТ(б)-1-23 \_\_\_\_\_\_\_\_\_\_\_\_ М.М.Сактанбаева**

**(подпись)**

**Проверила: к.ф.-м.н., доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Н.Аманкулова**

**(подпись)**

**Бишкек-2025**

**ЗАДАНИЯ**

на выполнение курсового проекта по дисциплине  
«Проектирование и разработка веб-приложений (Backend)»

**Студент:** Сактанбаева Мираида Мирлановна

**Группа:** ИСТ(б)-1-23

**Тема курсового проекта:** «Веб-приложение для онлайн-

резервирования мест в кинотеатре»

**Исходные данные для проектирования:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

* Требования к функциональности;
* Технические условия;
* Ограничения по технологиям и инструментам;
* Дополнительные условия (при наличии).

**Цель курсового проекта:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  
(формулируется чётко, например: разработка веб-приложения для онлайн-бронирования услуг.)

**Задачи проекта:**  
(перечислить ключевые задачи, например:)

* Проектирование архитектуры веб-приложения;
* Разработка клиентской части на JavaScript/React;
* Разработка серверной части на Node.js/Express;
* Реализация взаимодействия с базой данных;
* Организация аутентификации пользователей (если требуется).

**Требования к разрабатываемому приложению:**  
(указать минимальный функционал, например:)

* Регистрация и вход пользователя;
* Просмотр, создание, редактирование, удаление данных;
* Отправка запросов к серверу через API;
* Корректная обработка ошибок.

**Используемые технологии:**

* Frontend: HTML, CSS, JavaScript (или React, Vue и др.);
* Backend: Node.js, Express.js;
* База данных: MongoDB, PostgreSQL и др.

**График выполнения проекта:**

| Этап | Содержание работ | Срок выполнения |
| --- | --- | --- |
| 1 | Аналитический этап (исследование, постановка задачи) | 01/ 02/ 2025 |
| 2 | Проектировочный этап (архитектура, выбор технологий) | 05/ 03 / 2025 |
| 3 | Разработка приложения | 10/ 04 / 2025 |
| 4 | Тестирование и отладка | 15 / 05 / 2025\_\_ |
| 5 | Подготовка пояснительной записки | 20/ 05 / 2025 |
| 6 | Защита проекта | 28/ 05 / 2025 |

**Руководитель курсового проекта:**  
Фамилия, инициалы: Аманкулова Н.А.   
Должность: Заведующая кафедрой «Информационные системы и технологии им. академика А.Жайнкова»

**Подписи:**

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /ФИО/ 27 «май**» 2025**г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /ФИО/ 27 «май**» 2025**г.

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc200324251)

[**ГЛАВА 1. ОБЪЕКТНО- ОРИЕНТИРОВАННЫЙ АНАЛИЗ**](#_Toc200324252)

[**ПРЕДМЕТНОЙ ОБЛАСТИ** 5](#_Toc200324253)

[**1.1. Анализ существующих решений в области онлайн-бронирования билетов в кино** 5](#_Toc200324254)

[**1.2. Требования к функциональности и дизайну веб-приложения** 7](#_Toc200324255)

[**1.3. Проектирование структуры и интерфейса веб-сайта** 10](#_Toc200324256)

[**1.4. Выбор технологий и инструментов разработки** 12](#_Toc200324257)

[**ГЛАВА 2. РАЗРАБОТКА И ТЕСТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ** 15](#_Toc200324258)

[**2.1. Реализация фронтенд-части веб-приложения** 15](#_Toc200324259)

[**2.2. Реализация бэкенд-части.** 20](#_Toc200324260)

[**2.3. Наполнение приложения данными о киносеансах и залах** 23](#_Toc200324261)

[**2.4. Тестирование системы бронирования и устранение ошибок** 24](#_Toc200324262)

[**2.5. Адаптация веб-приложения для мобильных устройств** 25](#_Toc200324263)

[**ЗАКЛЮЧЕНИЕ** 26](#_Toc200324264)

[**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ** 27](#_Toc200324265)

# **ВВЕДЕНИЕ**

В наше время люди всё чаще предпочитают бронировать билеты в кино через интернет, чтобы сэкономить время и выбрать удобные места заранее. Поэтому создание веб-приложения для онлайн-резервирования мест в кинотеатре является актуальной задачей. Такое приложение позволяет пользователям быстро находить нужный фильм, выбирать время сеанса и бронировать места прямо с компьютера или телефона [1].

**Цель проекта** — разработать удобное и понятное веб-приложение, с помощью которого пользователи смогут бронировать места в кинотеатре в режиме онлайн.

**Задачи проекта:**

* создать клиентскую часть (фронтенд), которая отображает фильмы, залы и места;
* реализовать серверную часть (бэкенд) для обработки запросов и работы с базой данных;
* обеспечить взаимодействие между фронтендом и бэкендом;
* протестировать работу приложения и устранить возможные ошибки.

В процессе разработки использовались такие технологии, как:

* **JavaScript** для создания интерфейса;
* **Node.js** с **Express** для серверной части;
* **SQL Server** для хранения данных;
* **Tailwind CSS** для стилизации элементов;

# **ГЛАВА 1. ОБЪЕКТНО- ОРИЕНТИРОВАННЫЙ АНАЛИЗ**

# **ПРЕДМЕТНОЙ ОБЛАСТИ**

## **1.1. Анализ существующих решений в области онлайн-бронирования билетов в кино**

В современном цифровом обществе технологии онлайн-бронирования билетов в кино стали неотъемлемой частью сферы развлечений. Пользователям важно иметь удобный и быстрый доступ к информации о фильмах и возможность мгновенно забронировать место в зале. На данный момент существует множество платформ, реализующих подобный функционал. Наиболее известными являются такие сервисы, как Киноход, Рамблер.Касса, Яндекс.Афиша, а также официальные сайты сетей кинотеатров, таких как Синема Парк, Каро, Формула Кино и другие. Кроме того, активно развиваются зарубежные решения, например, Fandango и Atom Tickets, ориентированные на аудиторию США и других стран.

Анализ указанных систем показывает, что они обладают схожим набором ключевых функций. Основными возможностями таких сервисов являются: выбор кинотеатра, просмотр расписания сеансов, отображение информации о фильмах (описание, жанр, возрастное ограничение, трейлер), выбор конкретного фильма и времени показа, визуальное представление схемы зала с возможностью интерактивного выбора места, бронирование или покупка билета, а также последующая оплата через электронные платёжные системы. Преимуществами большинства существующих решений является высокая степень автоматизации процесса покупки билетов, что делает использование подобных сервисов удобным и быстрым. Большинство платформ предлагают интуитивно понятный пользовательский интерфейс, адаптированный как под веб-браузеры, так и под мобильные устройства.

Интерактивная схема зала позволяет пользователям точно определить доступные места и выбрать наиболее подходящие [13].

Поддержка различных платёжных методов, включая банковские карты и электронные кошельки, ускоряет процесс покупки. Наличие личного кабинета позволяет отслеживать историю покупок, возвращать билеты при необходимости, а также управлять персональными данными.

Несмотря на широкую функциональность, многие системы бронирования имеют недостатки. Часто интерфейс перегружен, особенно в сервисах с разными видами мероприятий, что затрудняет навигацию. В пиковые часы возможны сбои: схема зала может загружаться медленно, возникают трудности при оплате. Некоторые платформы требуют обязательной регистрации или установки приложения, что ограничивает удобство. Также многие сервисы ориентированы только на крупные сети, исключая частные и региональные кинотеатры.

Общие черты всех платформ — онлайн-бронирование мест, оплата через интернет, информация о фильмах и интерактивная схема зала. Все сервисы стремятся упростить процесс покупки билетов без необходимости посещения кассы. Отличия заключаются в деталях: Яндекс.Афиша работает как агрегатор и перенаправляет на сайты партнёров, тогда как Киноход и сайты сетей обеспечивают полный цикл покупки. Некоторые интерфейсы просты и удобны, другие содержат дополнительную информацию — обзоры, рецензии и трейлеры. Зарубежные сервисы, например Fandango, предлагают дополнительные функции, такие как оценка фильмов и предзаказ закусок.

Проведённый анализ демонстрирует, что рынок онлайн-бронирования билетов в кино насыщен и активно развивается. Однако, несмотря на наличие множества платформ, остаются определённые трудности, связанные с удобством навигации, стабильностью работы, уровнем персонализации и универсальностью решений. Внимательное изучение существующих решений позволяет выявить как эффективные подходы к реализации онлайн-бронирования, так и недостатки, характерные для большинства систем, что важно учитывать при дальнейшем исследовании и разработке в данной области [13].

## **1.2. Требования к функциональности и дизайну веб-приложения**

Создание веб-приложения для онлайн-резервирования мест в кинотеатре требует чёткого понимания не только технологических решений, но и потребностей конечного пользователя. От того, насколько удобно, понятно и приятно будет взаимодействовать с системой, зависит её успех. Поэтому при проектировании такого веб-приложения важно уделить внимание как функциональным, так и дизайнерским требованиям. В данном разделе рассматриваются ключевые аспекты, влияющие на комфорт пользователей, администраторов и устойчивость самой системы.

Функциональность — это «скелет» веб-приложения, обеспечивающий выполнение основной задачи: быстрое и удобное бронирование мест в кинотеатре. Его основные функции, которые должны быть реализованы:

1. Регистрация и авторизация пользователей

* Возможность создания личного кабинета с использованием email, номера телефона или авторизации через социальные сети.
* Защита данных и безопасность входа (включая хэширование паролей, капчу при множественных попытках входа).

2. Просмотр доступных фильмов

* Список текущих фильмов с афишей, описанием, трейлером и продолжительностью.

3. Выбор кинотеатра и сеанса

* Поддержка нескольких кинотеатров и залов.

4. Интерактивная схема зала

* Графическое отображение всех рядов и мест.
* Свободные, занятые места отображаются в разных цветах.
* Реализация выбора сразу нескольких мест.
* Поддержка мобильных свайпов и масштабирования [8].

Наряду с функциональностью, важно обеспечить:

* Надёжность: система должна быть устойчивой к сбоям, иметь защиту от потери данных.
* Производительность: даже при большом числе пользователей бронирование должно происходить мгновенно.
* Масштабируемость: возможность расширения — добавление новых кинотеатров, залов, городов.
* Безопасность: защита от SQL-инъекций, CSRF, XSS и других видов атак.
* Кроссбраузерность и адаптивность: поддержка всех популярных браузеров и мобильных устройств.

Дизайн интерфейса играет ключевую роль в восприятии системы пользователем. Независимо от того, насколько функционально и технологично реализовано веб-приложение, именно внешний вид и удобство управления определяют, захочет ли пользователь вернуться и использовать сервис повторно. Один из главных принципов — это простота и интуитивность. Интерфейс должен быть понятен даже тем, кто впервые пользуется подобными системами. Оптимальным считается подход, при котором для совершения бронирования требуется минимальное количество действий: выбор фильма, затем зала, потом мест и завершение оплаты. Логичность и последовательность шагов повышают удобство использования и снижают риск ошибок.

Визуальный стиль современных систем бронирования стремится к минимализму, чистоте линий, читаемой типографике и использованию акцентных цветов для привлечения внимания к основным действиям. Анимации при переходах между экранами или при выборе мест делают взаимодействие более живым и приятным. При разработке интерфейса рекомендуется использовать готовые дизайн-системы, такие как Tailwind CSS или Material UI, так как они обеспечивают единообразие стиля и позволяют быстрее реализовать адаптивный и современный внешний вид [8].

Особое внимание должно уделяться адаптивности интерфейса. Пользователи всё чаще обращаются к сервису с мобильных устройств, поэтому система должна автоматически подстраиваться под разные размеры экранов. Интерактивная схема зала должна быть масштабируемой, чтобы пользователь мог легко приблизить нужную часть, а все элементы управления — такими, чтобы на них было удобно нажимать даже с экрана смартфона.

Немаловажно обеспечить визуальную обратную связь. После каждого действия пользователь должен получать чёткое подтверждение: успешно ли выбрано место, прошла ли оплата, или возникла ошибка. Это повышает уровень доверия и помогает избежать недоразумений. Также необходимо учитывать контрастность интерфейса: цветовая палитра не должна утомлять глаза, текст должен хорошо читаться на фоне. Желательно предусмотреть возможность переключения между светлой и тёмной темами.

Небольшие анимации — например, при выборе места или при открытии информации о фильме — делают использование системы более приятным и динамичным.

Если система разрабатывается для конкретной сети кинотеатров, важно учесть элементы фирменного стиля — логотип, цвета, особенности визуальной идентичности. Это делает систему узнаваемой, создаёт у пользователя ощущение доверия и принадлежности к конкретному бренду, а также позволяет сохранить корпоративную целостность в оформлении цифровых сервисов.

Разрабатываемое веб-приложение должно быть не только технически работоспособным, но и удобным, логичным, визуально привлекательным и безопасным. Только сочетание этих факторов позволит создать конкурентоспособный продукт, который обеспечит пользователю комфортное и быстрое бронирование билетов, а администрации — удобное управление расписанием, залами и статистикой. Такой подход повысит лояльность клиентов, улучшит продажи и укрепит репутацию кинотеатра в цифровой среде [8].

## **1.3. Проектирование структуры и интерфейса веб-сайта**

Процесс создания веб-приложения для онлайн-бронирования мест в кинотеатре начинается с тщательного проектирования его структуры и пользовательского интерфейса. На этом этапе определяется, как пользователи будут взаимодействовать с системой, какие функции будут распределены по страницам, как будет организована навигация и как данные будут передаваться между клиентской и серверной частью приложения.

Архитектура веб-приложения, как правило, включает две основные составляющие: клиентскую и серверную стороны. Пользовательский интерфейс (фронтенд) отвечает за визуальное отображение элементов, обработку действий пользователя и передачу информации на сервер. Он реализуется с помощью HTML, CSS и JavaScript, а также современных библиотек и фреймворков, таких как React, Vue или Angular. Серверная часть (бэкенд) обеспечивает выполнение бизнес-логики, обработку запросов, работу с базой данных и другими сервисами. Для этой цели могут использоваться такие технологии, как Node.js с Express, Python с Django или Flask, Java с Spring Boot и другие. Важнейшим компонентом системы также является база данных, в которой хранятся сведения о фильмах, залах, расписании сеансов, забронированных местах и пользователях [9].

Веб-приложение должно включать ряд функциональных страниц, обеспечивающих комфортную и логичную навигацию. На главной странице пользователь видит афишу текущих фильмов в виде слайдера, имеет возможность воспользоваться быстрым поиском и сразу перейти к бронированию. Далее представлена страница со списком всех фильмов, на которой пользователь может ознакомиться с краткой информацией о каждом фильме и при желании перейти к более подробному описанию. Страница информации о фильме содержит трейлер, постер, краткое содержание, сведения о жанре и актёрах. Отсюда пользователь может начать процесс бронирования, перейдя к выбору времени сеанса и зала [11].

На этапе выбора зала реализуется интерактивное отображение свободных и занятых мест, позволяющее выбрать одно или несколько кресел и подтвердить бронирование.

Особое внимание в процессе проектирования должно уделяться удобству и интуитивной понятности интерфейса. Навигация должна быть логичной и доступной с любой страницы, чтобы пользователь всегда понимал, где он находится и как вернуться назад. Интерфейс следует оформлять в минималистичном стиле, избегая перегрузки страницы избыточной информацией. Основные элементы управления — кнопки, формы, уведомления — должны быть чётко выделены с помощью цветовой и визуальной иерархии. Пользователь должен без усилий различать активные элементы, ошибки ввода и выбранные объекты. Также рекомендуется внедрение микровзаимодействий, таких как анимации при наведении на кнопки, плавные переходы между страницами, всплывающие уведомления и подсказки, обеспечивающие более живой и отзывчивый интерфейс.

Продуманная структура и качественный пользовательский интерфейс напрямую влияют на успешность веб-приложения, повышая удобство и скорость взаимодействия, снижая количество ошибок и увеличивая лояльность пользователей. Всё это способствует не только улучшению пользовательского опыта, но и повышению эффективности работы самой системы бронирования В современных условиях высокой конкуренции удобство использования, эстетика и логика взаимодействия с приложением определяют удовлетворённость пользователей и их готовность возвращаться к сервису вновь [11].

## **1.4. Выбор технологий и инструментов разработки**

Выбор технологий и инструментов разработки играет ключевую роль в успешной реализации веб-приложения для онлайн-резервирования мест в кинотеатре. На данном этапе важно учесть не только технические характеристики, но и популярность, активность сообщества, наличие документации, удобство разработки, масштабируемость и безопасность.

Так как проект представляет собой динамичное веб-приложение с клиентской и серверной частью, основной упор делается на использование современных веб-технологий, обеспечивающих высокую скорость работы, удобный пользовательский интерфейс и надёжную обработку данных [3].

Для реализации клиентской части веб-приложения были выбраны следующие технологии:

* **HTML5** и **CSS3** — основа разметки и оформления страниц. HTML используется для создания структуры веб-страниц, а CSS — для их стилизации и визуального оформления. Использование последних стандартов позволяет реализовать адаптивный и семантически корректный интерфейс.
* **JavaScript** — основной язык программирования, отвечающий за интерактивность и динамическое поведение элементов на странице.
* **React** — библиотека JavaScript для построения пользовательских интерфейсов. React обеспечивает высокую производительность за счёт виртуального DOM, удобную компонентную архитектуру и быструю разработку благодаря большому количеству готовых решений и расширений.
* **Tailwind CSS** — современная утилитарная CSS-библиотека, позволяющая быстро и гибко настраивать стиль без необходимости писать кастомные CSS-файлы. Использование Tailwind делает разработку более продуктивной, а интерфейс — визуально единообразным и адаптивным [4].
* **Axios** — библиотека для выполнения HTTP-запросов.

Используется для связи между клиентом и сервером, особенно при отправке данных о бронированиях и получении информации о фильмах, сеансах, доступных местах и т. д [10].

Для реализации серверной логики веб-приложения выбраны следующие инструменты:

* **Node.js** — серверная среда выполнения JavaScript, позволяющая обрабатывать запросы на стороне сервера. Отличается высокой производительностью, поддержкой асинхронной обработки и большим выбором модулей.
* **Express.js** — фреймворк для Node.js, предоставляющий простой и гибкий способ создания REST API. С его помощью можно быстро организовать маршрутизацию, обработку запросов, middleware и структуру проекта.
* **bcrypt.js** — библиотека для хеширования паролей. Обеспечивает дополнительную защиту данных пользователей при регистрации и входе в систему.

#### База данных

Для хранения информации о фильмах, пользователях, бронированиях и сеансах используется:

* **MySQL** — реляционная система управления базами данных (СУБД), обладающая высокой производительностью, стабильностью и гибкой системой запросов. MySQL позволяет эффективно организовать связи между таблицами, использовать индексацию и обеспечивать целостность данных.
* **Sequelize ORM** — объектно-реляционное отображение (ORM) для Node.js, предназначенное для работы с MySQL. Облегчает взаимодействие с базой данных, позволяя писать запросы на JavaScript вместо SQL, а также управлять миграциями и ассоциациями между моделями (например, пользователи и их бронирования) [6].

Для упрощения и ускорения процесса разработки использовались следующие вспомогательные инструменты:

* **Visual Studio Code** — линейка продуктов Microsoft, которая включает интегрированную среду разработки программного обеспечения. Она предоставляет возможность внедрять, тестировать, компилировать все веб-проекты. Это JS, HTML, PHP редакторы исходного кода с возможностью рефакторинга между файлами. Встроенный отладчик работает как отладчик исходного кода и как отладчик машинного уровня. Однако эта IDE достаточно тяжеловесна. Чтобы выполнить какую-нибудь простую задачу, потребуется много ресурсов. Поэтому иногда удобнее использовать более легкий редактор.
* **Git и GitHub** — система контроля версий и платформа для хранения и совместной разработки кода. Использование Git позволяет отслеживать изменения, откатываться к предыдущим версиям, а GitHub предоставляет удобный интерфейс для ведения репозитория.

Дополнительно при разработке были учтены инструменты и методы, направленные на:

* **Адаптивную вёрстку** — чтобы веб-приложение одинаково корректно отображалось как на экранах компьютеров, так и на мобильных устройствах;
* **Валидацию форм** — проверка вводимых пользователями данных как на клиентской, так и на серверной стороне;
* **Обработку ошибок** — реализация механизмов уведомлений, логирования и резервных сценариев;
* **Защиту от основных уязвимостей** — таких как SQL-инъекции, XSS, CSRF, благодаря грамотной структуре кода и использованию надёжных библиотек [7].

# **ГЛАВА 2. РАЗРАБОТКА И ТЕСТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ**

## **2.1. Реализация фронтенд-части веб-приложения**

В рамках разработки веб-приложения для онлайн-бронирования мест в кинотеатре была реализована клиентская часть (frontend), обеспечивающая удобный пользовательский интерфейс и интуитивное взаимодействие с системой. Структура проекта организована таким образом, чтобы изолировать визуальный интерфейс от остальной логики, что упрощает разработку, поддержку и возможное расширение приложения в будущем.

Клиентская часть размещена в отдельной директории frontend/ и разработана с использованием CSS-фреймворка Tailwind и языка JavaScript. Tailwind CSS позволил быстро и гибко оформлять элементы интерфейса, добиваясь современного и адаптивного дизайна. JavaScript используется для динамического отображения информации, управления событиями и отправки запросов к серверу с помощью REST API.

App.js — основной файл маршрутизации и корневой логики изображено на (Рис.2.1.1).

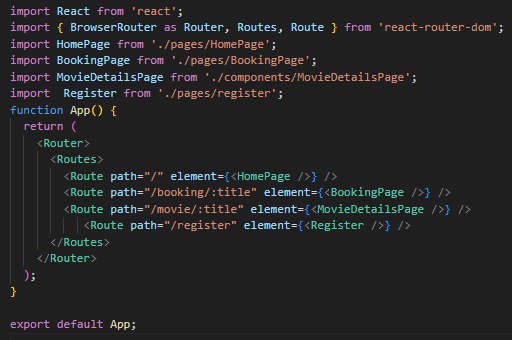


Рис.2.1.1. Пример App.js.

На рисунке 2.1.1 изображен файл App.js, который настраивает маршруты в React-приложении, то есть определяет, какая страница будет открываться по какому адресу. Он подключает главную страницу, страницу бронирования, страницу с деталями фильма и страницу регистрации, чтобы пользователь мог свободно переходить между ними.

Отображение подробной информации о фильме она изображена на (Рис.2.1.2).

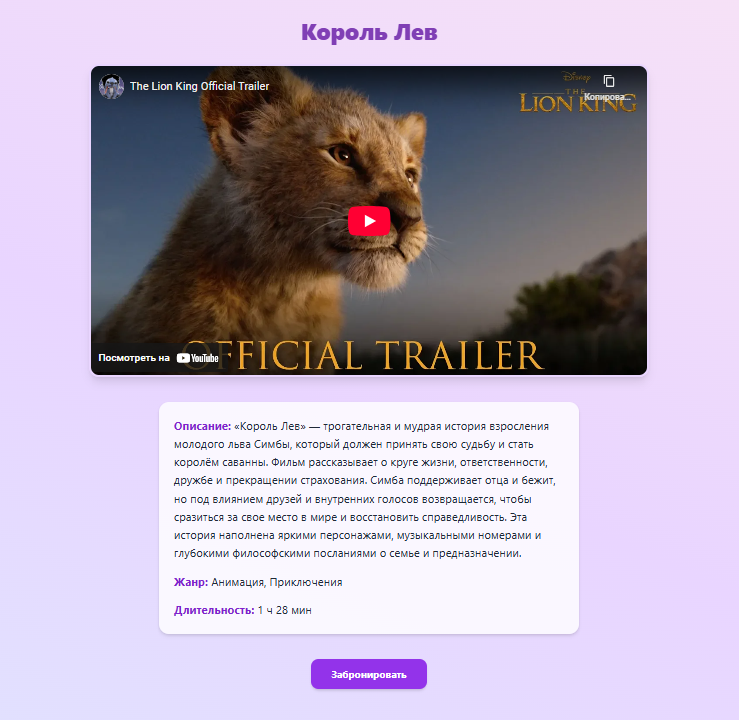


Рис.2.1.2. Отображение одной карточки.

На (Рис.2.1.2) представлено отображение подробной информации о выбранном фильме: его название, трейлер, описание, жанр и длительность. Также он показывает кнопку для перехода к бронированию мест на этот фильм.

Регистрация и вход пользователя осуществляется с помощью формы (Рис.2.1.3).

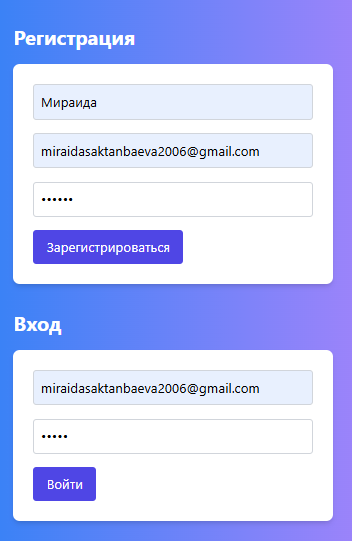


Рис.2.1.3. Регистрационная форма.

Как показано на (Рис.2.1.3) представлен компонент, отвечающий за регистрацию нового пользователя в системе. Форма включает в себя поля для ввода имени, адреса электронной почты, пароля и кнопки «Забронировать». В случае, если пользователь ранее зарегистрировался, он может перейти на форму входа через соответствующую кнопку «Войти».

Главная страница изображена на (Рис.2.1.4).

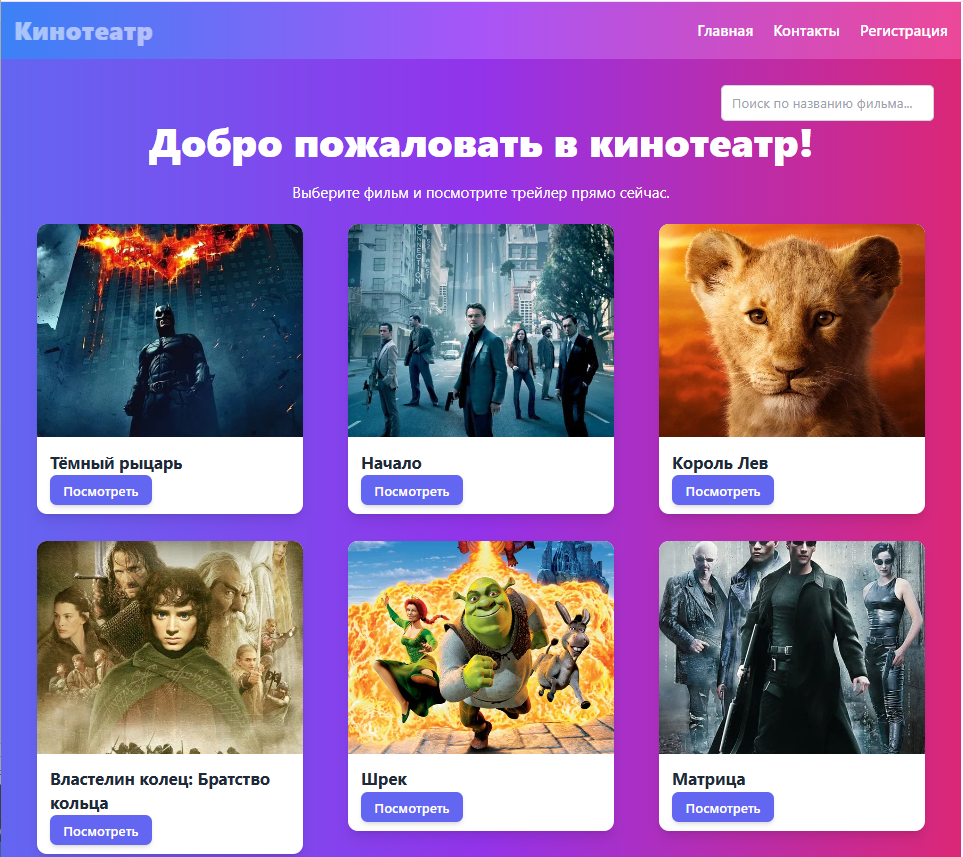


Рис.2.1.4. Главная страница.

Главная страница, которая изображена на (Рис.2.1.4) представляет собой первый экран, с которым сталкивается пользователь при входе в веб-приложение. На данной странице отображается навигационное меню, позволяющее быстро перейти к ключевым функциям. Также представлен блок с актуальными фильмами, включающей постеры и краткие описания популярных новинок. У каждого фильма есть кнопки «Подробнее» и «Забронировать», ведущие на страницы с описанием и выбором мест. Также с главной страницы можно перейти к регистрации.

Пример одного из визуальных компонентов для бронирования приведено на (Рис.2.1.5).

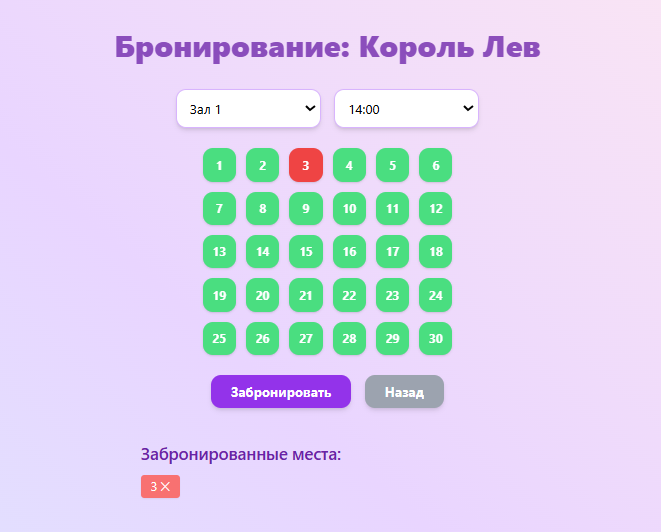


Рис.2.1.5. Бронирование мест.

Бронирование показано на (рис. 2.1.5) и проходит в несколько шагов: выбор фильма, зала и времени, затем — мест. Выбранные места можно отменить повторным нажатием. После этого нажимается кнопка «Забронировать» и, при необходимости, «Назад» для возврата.

Далее подтверждение бронирования отображено на (Рис.2.1.6).

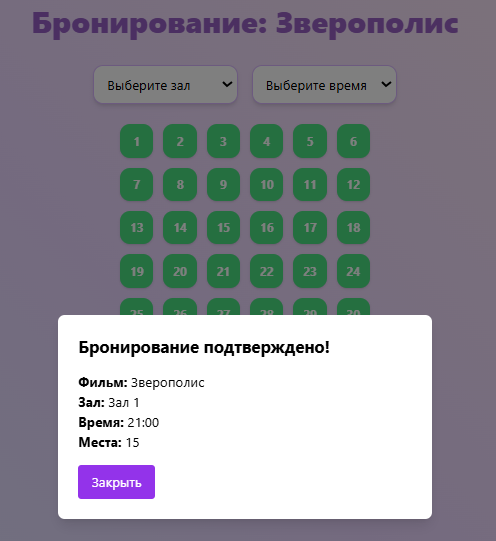


Рис.2.1.6. Подтверждение брони.

Согласно (Рис.2.1.6) после выбора мест пользователю показывается информация о бронировании с возможностью закрыть окно подтверждения.

## **2.2. Реализация бэкенд-части.**

Реализация бэкенд-части веб-приложения включает создание серверной логики, обеспечивающей обработку запросов от клиента, взаимодействие с базой данных и выполнение бизнес-правил системы. Серверная часть отвечает за приём данных о бронировании, управление информацией о фильмах, залах и сеансах, а также за авторизацию и регистрацию пользователей. Для этого используется набор модулей и маршрутов. На (Рис.2.2.1) ниже представлена структура папок и файлов.

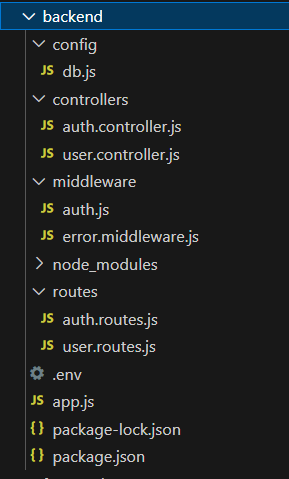


Рис.2.2.1. Структура backend (серверная часть).

На (Рис.2.2.1) показана структура backend (серверной части), демонстрирующая организацию кода, где каждая папка и файл имеют своё назначение: маршруты, модели данных, контроллеры и вспомогательные функции. Такой подход обеспечивает удобство разработки, сопровождения и масштабирования приложения.

Пример файла с кодом для регистрации новых пользователей и входа в систему отображена на (Рис.2.2.2).

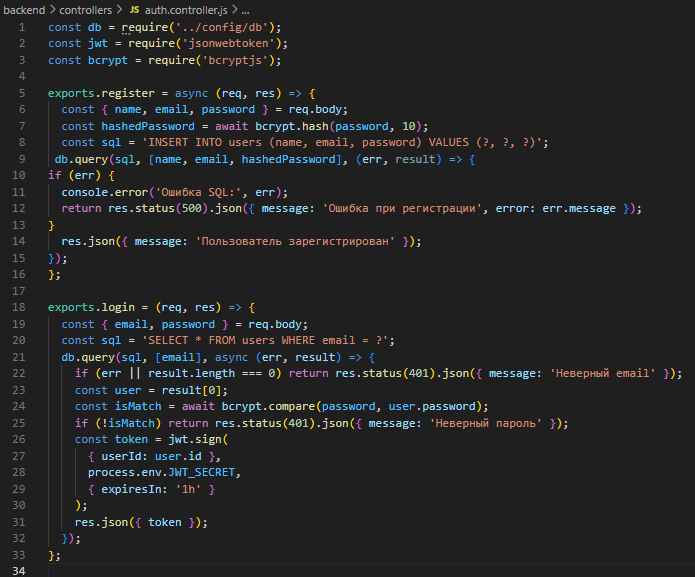


Рис.2.2.2 - Отвечает за регистрацию, вход в систему.

Как изображено на (Рис. 2.2.2), в этом файле реализована серверная логика для регистрации и входа пользователей. При регистрации данные пользователя принимаются с клиентской стороны, пароль надежно шифруется с помощью bcrypt и сохраняется в базе данных. При входе происходит поиск пользователя по email, затем введённый пароль сравнивается с сохранённым хэшом.

В этом файле настроены маршруты для регистрации, входа, валидации данных и получения профиля пользователя с защитой через JWT (Рис.2.2.3).

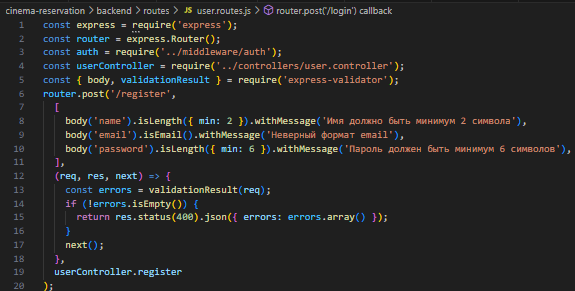


Рис.2.2.3. Обработка запросов регистрации, входа.

Как показано на (Рис. 2.2.3), в этом файле определены маршруты для регистрации и входа пользователей с проверкой данных через express-validator и обработкой ошибок.

Главный файл сервера, который настраивает работу приложения, обрабатывает запросы и управляет маршрутами показана на (Рис.2.2.4).

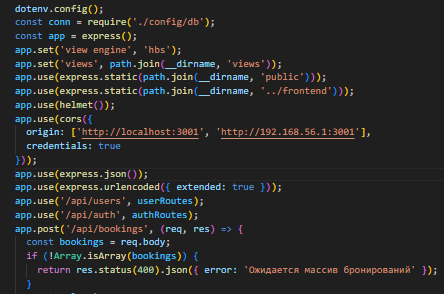


Рис. 2.2.4. Основной файл сервера.

Как показано на (Рис.2.2.4) этот код настраивает сервер Express, разрешает запросы с указанных адресов, подключает маршруты пользователей.

## **2.3. Наполнение приложения данными о киносеансах и залах**

Для корректной работы веб-приложения по онлайн-бронированию мест в кинотеатре важно обеспечить полноценное и структурированное наполнение информацией о киносеансах и залах. Эти данные формируют основу функционала, позволяя пользователям выбирать фильмы, время сеансов и места для посещения, как это показано на (Рис.2.3.1).

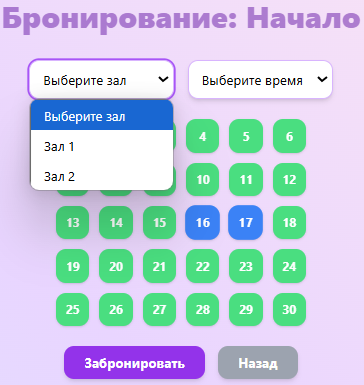


Рис. 2.3.1. Основные данные о фильмах, сеансах и залах.

На (Рис.2.3.1) показан интерфейс выбора зала, времени сеанса и доступных мест для бронирования.

Информация о киносеансах включает в себя название фильма, его описание и длительность, а также точное время начала показа. Для удобства восприятия пользователям предоставляются изображения — афиши, и видеоматериалы — трейлеры, что помогает сделать выбор более информированным и наглядным.

Каждый сеанс закреплён за конкретным залом и временем, чтобы избежать пересечений. Данные хранятся в связанных таблицах базы данных. Для пользователей доступна визуальная схема зала с выбором и бронированием мест в реальном времени, где занятые места автоматически блокируются.

## **2.4. Тестирование системы бронирования и устранение ошибок**

Тестирование является важным этапом разработки веб-приложения, позволяющим выявить ошибки, баги и несоответствия требованиям до запуска продукта в эксплуатацию. В процессе тестирования проверяется, насколько корректно работают все функции приложения, удобно ли оно для пользователя и соответствует ли заданным техническим требованиям.

Таблица 1. Результаты тестирования программы

|  |  |
| --- | --- |
| Выполняемое действие | Результат |
| Вход с данными username: miraida, password: pass123 | Успешный вход, перенаправление на главную страницу с отображением «Добро пожаловать в кинотеатр!» |
| Регистрация нового пользователя Анны, email anna@example.com, password: 456pass | Регистрация прошла успешно, пользователь автоматически авторизован, отображается домашняя страница |
| Повторная регистрация пользователя Анны с тем же email | Регистрация не выполняется, выводится сообщение об ошибке — «Ошибка при регистрации» |
| Просмотр списка фильмов на главной странице | Отображается список из первых 15 фильмов с корректными карточками и изображениями |
| Выбор места в зале и бронирование | Место успешно выделяется, после подтверждения брони информация отправляется на сервер и отображается сообщение об успешном бронировании |

## **2.5. Адаптация веб-приложения для мобильных устройств**

Современные пользователи всё чаще заходят в онлайн-сервисы через смартфоны и планшеты, поэтому очень важно, чтобы веб-приложения корректно отображались и были удобны для использования на разных устройствах с разными размерами экранов. Для приложений онлайн-бронирования мест в кинотеатре особенно важно обеспечить такую адаптивность интерфейса, чтобы пользователи могли без проблем выбирать сеансы и места, независимо от того, с какого устройства они заходят. Обычно для этого применяются технологии HTML5 и CSS3, а также популярные фреймворки, например, Tailwind CSS, которые помогают создавать гибкие и отзывчивые интерфейсы.

Ключевые элементы адаптивности включают удобное меню, которое на мобильных устройствах превращается в компактный "бургер", упрощённое отображение схемы зала, позволяющее выбирать места касанием пальца, а также крупные и легко нажимаемые кнопки и формы. Чтобы не загромождать экран на небольших устройствах, часто скрываются второстепенные элементы, оставляя только самую важную информацию.

Такая адаптация помогает сделать процесс бронирования более удобным и доступным в любом месте и в любое время, что значительно расширяет аудиторию пользователей и повышает комфорт взаимодействия с приложением [5].

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения проекта по разработке веб-приложения для онлайн-бронирования мест в кинотеатре был реализован комплексный и функциональный продукт, который сочетает в себе удобный пользовательский интерфейс и надежную серверную логику. Разделение на клиентскую и серверную части позволило обеспечить масштабируемость и удобство дальнейшего сопровождения приложения.

Использование современных технологий, таких как React для фронтенда и Node.js с Express для бэкенда, обеспечило гибкость и высокую производительность системы. Важной частью проекта стала продуманная структура базы данных, которая эффективно хранит информацию о фильмах, сеансах и бронированиях, обеспечивая целостность данных и быструю обработку запросов.

В ходе разработки особое внимание уделялось удобству и безопасности пользователей: реализованы формы регистрации и авторизации, а также механизмы обработки ошибок и валидации данных. Проведенное тестирование показало стабильную работу приложения, корректное отображение информации и правильное взаимодействие между клиентом и сервером.

Данный проект является хорошей основой для дальнейшего расширения и внедрения новых функций, таких как интеграция с платежными системами, уведомления пользователям, а также улучшение интерфейса. В целом, проделанная работа демонстрирует успешное применение современных веб-технологий для создания эффективных и удобных решений в области онлайн-сервисов [12].

# **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

* 1. 4 преимущества онлайн-покупки билетов в кинотеатр. [Электронный ресурс]. – Режим доступа: <https://jjtv.kz/ru/4-preimushchestva-onlayn-pokupki-biletov-v-kinoteatr>
  2. Creately. Онлайн бронирование билетов в кино. [Электронный ресурс]. – Режим доступа: <https://creately.com/diagram/example/O55dyCE4IKj/onlajn-bronirovanie-biletov-v-kino>
  3. Live Typing. Разработка приложений для кинотеатра и кино-сервисов. [Электронный ресурс]. – Режим доступа: <https://livetyping.com/ru/razrabotka-prilojeniy-kino-tematiki>
  4. Tailwind CSS. [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Tailwind_CSS>
  5. Адаптация сайта под мобильные устройства. [Электронный ресурс]. – Режим доступа: <https://www.ihc.ru/articles/kak-adaptirovat-sajt-pod-mobilnye-ustrojstva.html>
  6. Выполняемые на сервере веб-фреймворки. [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/ru/docs/Learn_web_development/Extensions/Server-side/First_steps/Web_frameworks>

ДЛЯ ТЕХНИЧЕСКОГО АНАЛИЗА. [Электронный ресурс]. – Режим доступа: <https://elib.bsu.by/bitstream/123456789/261612/1/51-52.pdf>

* 1. Инструменты для веб-разработки. [Электронный ресурс]. – Режим доступа: <https://edisonstudio.ru/blog/about-razrabotka/instrumenty-dlya-veb-razrabotki/>
  2. Как создать онлайн кинотеатр: технологии и решения для бизнеса. [Электронный ресурс]. – Режим доступа: <https://flussonic.ru/blog/news/kak-sozdat-onlayn-kinoteatr>
  3. РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ "КИНОТЕАТРА" С ИСПОЛЬЗОВАНИЕМ JAVASCRIPT, PHP И MYSQL. [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/razrabotka-web-prilozheniya-kinoteatra-s-ispolzovaniem-javascript-php-i-mysql?utm_source=chatgpt.com>
  4. РАЗРАБОТКА КЛИЕНТСКОЙ ЧАСТИ ВЕБ-ПРИЛОЖЕНИЯ
  5. Разработка приложений кино-тематики. [Электронный ресурс]. – Режим доступа: <https://livetyping.com/ru/razrabotka-prilojeniy-kino-tematiki?utm_source=chatgpt.com>
  6. Создание и сопровождение web-приложения для просмотра фильмов. [Электронный ресурс]. – Режим доступа: <https://lpi.sfu-kras.ru/files/vkr_strukov_i.p._isit.pdf>
  7. Создать приложение для бронирования билетов. [Электронный ресурс]. – Режим доступа: <https://livetyping.com/ru/kak-sozdat-prilozhenie-dlya-bronirovaniya-biletov>
  8. Хэррон, Д. Node.js: разработка серверных веб-приложений на JavaScript / Д. Хэррон. – М.: ДМК Пресс, 2020. – 288 с.