

Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное
учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт управления бизнес-процессами

Кафедра бизнес-информатики и моделирования бизнес-процессов

КУРСОВАЯ РАБОТА

по дисциплине «Проектирование, разработка и оптимизация
web-приложений»

«Создание web-приложения с применением современных
web-технологий по тематике «Театр»

Преподаватель

подпись, дата

Никитин А.И.

Студент

подпись, дата

Ершов Н.В.

Красноярск 2024

СОДЕРЖАНИЕ

Введение	3
1 Теоретическая часть	4
1.1 HTML И CSS	4
1.2 JavaScript	5
1.3 PHP	7
1.4 MySQL	9
2 Практическая часть	12
2.1 Требования к работе	12
2.2 Создание базы данных	13
2.3 Регистрация и авторизация	15
2.4 Главная страница	17
2.5 Личный кабинет	20
Заключение	29
Список использованных источников	30

Введение

В современном мире веб-технологии играют ключевую роль в развитии различных сфер деятельности, включая искусство и культуру. Театр, как одна из самых древних форм искусства, также не остался в стороне от цифровой трансформации. Создание веб-приложения, посвященного театру, представляет собой уникальную возможность не только для популяризации театрального искусства, но и для улучшения взаимодействия с аудиторией, расширения доступа к информации о спектаклях, актерах и театральных событиях.

Целью данной курсовой работы является проектирование и разработка веб-приложения с использованием современных веб-технологий, которое будет служить платформой для театров и их зрителей. В процессе работы будут рассмотрены основные аспекты проектирования, разработки и оптимизации веб-приложений, а также современные инструменты и технологии, такие как HTML5, CSS3, JavaScript и фреймворки для создания пользовательских интерфейсов.

Актуальность темы обусловлена необходимостью интеграции театрального искусства в цифровое пространство, что позволит не только сохранить культурное наследие, но и привлечь новую аудиторию. Веб-приложение будет включать функционал для публикации информации о текущих и предстоящих спектаклях, возможности онлайн-бронирования билетов, а также разделы с новостями и событиями театра.

Таким образом, данная работа направлена на создание инновационного решения, которое объединит театры и зрителей в едином цифровом пространстве, способствуя развитию театральной культуры в условиях современных реалий.

1 Теоретическая часть

1.1 HTML и CSS

HTML — это язык разметки, который используется для создания структуры веб-страниц. Он позволяет организовывать контент, добавляя различные элементы, такие как заголовки, абзацы, списки, изображения и ссылки:

1. Структура документа: Каждый HTML-документ имеет определённую структуру, которая включает заголовок (где указываются метаданные) и тело (где размещается видимый контент).

2. Элементы: HTML состоит из различных элементов, каждый из которых выполняет свою функцию. Например, заголовки обозначают важные темы, а абзацы — текстовые блоки.

3. Атрибуты: Элементы могут иметь атрибуты, которые добавляют дополнительную информацию, например, ссылки могут указывать на адреса веб-страниц.

CSS — это язык стилей, который используется для оформления веб-страниц. Он позволяет изменять внешний вид элементов, таких как цвета, шрифты и расположение на странице:

1. Стилизация: С помощью CSS можно задавать стили для различных элементов HTML. Например, можно изменить цвет текста, размер шрифта или фон элемента.

2. Селекторы: CSS использует селекторы для определения, к каким элементам применять стили. Это может быть сделано по тегам, классам или идентификаторам.

3. Организация стилей: CSS может быть включен в HTML-документ несколькими способами: встроенные стили (внутри документа), внутренние стили (в разделе заголовка) и внешние стили (через отдельный файл).

Понимание основ HTML и CSS является ключевым для создания веб-сайтов. Эти технологии позволяют разработчикам не только формировать контент, но и делать его доступным и привлекательным для пользователей.

1.2 JavaScript

JavaScript — это высокоуровневый, интерпретируемый язык программирования, который стал неотъемлемой частью веб-разработки. Он был создан в 1995 году Бренданом Айком и с тех пор значительно эволюционировал, став одним из основных языков для создания интерактивных и динамичных веб-приложений. JavaScript позволяет разработчикам добавлять функциональность на веб-страницы, взаимодействовать с пользователями и управлять данными в реальном времени.

Основные характеристики JavaScript:

JavaScript является языком с динамической типизацией, что означает, что переменные могут принимать значения различных типов без необходимости явного указания типа данных. Это делает язык гибким и удобным для быстрой разработки. Кроме того, JavaScript поддерживает объектно-ориентированное программирование, что позволяет создавать сложные структуры данных и повторно использовать код через наследование и модули.

Одной из ключевых особенностей JavaScript является его асинхронная природа. Это позволяет выполнять операции, такие как запросы к серверу, не блокируя основной поток выполнения программы. Асинхронные функции и промисы значительно упрощают работу с асинхронным кодом, позволяя разработчикам писать более чистый и читаемый код.

JavaScript тесно интегрирован с HTML и CSS, что позволяет разработчикам изменять содержимое веб-страниц на лету. С помощью Document Object Model (DOM) JavaScript может взаимодействовать с элементами HTML, изменять их свойства и стили. Например, можно добавить обработчики событий для кнопок, чтобы реагировать на действия пользователя, такие как клики или наведение курсора.

Кроме того, JavaScript может изменять стили элементов CSS в реальном времени. Это дает возможность создавать анимации, менять цвета и размеры элементов в ответ на действия пользователя, что значительно улучшает пользовательский опыт.

С течением времени вокруг JavaScript возникло множество библиотек и фреймворков, которые упрощают разработку веб-приложений. Одной из самых популярных библиотек является jQuery, которая предоставляет удобный интерфейс для работы с DOM и упрощает выполнение сложных задач с минимальным количеством кода.

Среди фреймворков выделяются React, Angular и Vue.js. Эти инструменты позволяют создавать сложные одностраничные приложения (SPA), где пользовательский интерфейс обновляется динамически без перезагрузки страницы. React, например, использует концепцию компонентов, что делает код более модульным и легко поддерживаемым.

Хотя JavaScript изначально был разработан для работы в браузере, он также нашел применение на стороне сервера благодаря платформе Node.js. Node.js позволяет разработчикам использовать JavaScript для создания серверных приложений, что обеспечивает единый стек технологий для фронтенда и бэкенда. Это упрощает процесс разработки и позволяет использовать одни и те же языковые конструкции на обеих сторонах.

Node.js поддерживает асинхронное программирование и позволяет обрабатывать множество запросов одновременно, что делает его отличным выбором для создания высокопроизводительных приложений. С помощью Node.js разработчики могут создавать API, работать с базами данных и обрабатывать файлы.

JavaScript стал неотъемлемой частью современной веб-разработки благодаря своей гибкости, мощным возможностям и широкому сообществу разработчиков. Он позволяет создавать интерактивные и динамичные веб-приложения, которые улучшают взаимодействие пользователей с контентом. С постоянными обновлениями языка и появлением новых библиотек и фреймворков JavaScript продолжает оставаться актуальным инструментом в арсенале веб-разработчиков. Важно отметить, что изучение JavaScript открывает двери к множеству возможностей в области разработки программного обеспечения как для фронтенда, так и для бэкенда.

1.3 PHP

PHP (Hypertext Preprocessor) — это широко используемый язык программирования, специально разработанный для создания динамических веб-страниц и веб-приложений. Он был создан в 1994 году датским программистом Расмусом Лердорфом и с тех пор стал одним из самых популярных языков для серверного программирования. PHP является языком с открытым исходным кодом, что позволяет разработчикам свободно использовать и модифицировать его под свои нужды.

PHP — это язык с интерпретируемым синтаксисом, что делает его простым в освоении и использовании. Он поддерживает множество парадигм программирования, включая процедурное, объектно-ориентированное и функциональное программирование. Это позволяет разработчикам выбирать наиболее подходящий стиль программирования для их проектов.

Одной из ключевых особенностей PHP является его способность взаимодействовать с различными базами данных, такими как MySQL, PostgreSQL и SQLite. Это делает его идеальным выбором для разработки веб-приложений, требующих хранения и обработки данных. PHP также поддерживает работу с различными форматами данных, включая JSON и XML, что упрощает интеграцию с внешними API и сервисами.

PHP часто используется в сочетании с HTML для создания динамического контента. Код PHP может быть встроен непосредственно в HTML-код, что позволяет генерировать страницы на лету в зависимости от пользовательских запросов или данных из базы данных. Например, можно создать страницу, которая отображает список товаров из базы данных, изменяя содержимое в зависимости от выбранной категории или фильтров.

PHP также поддерживает обработку форм, что позволяет разработчикам получать данные от пользователей, обрабатывать их и возвращать результаты. Это делает PHP идеальным языком для создания интерактивных веб-приложений, таких как интернет-магазины, блоги и социальные сети.

Существует множество фреймворков на основе PHP, которые значительно

упрощают процесс разработки. Одним из самых популярных является Laravel, который предлагает мощные инструменты для работы с маршрутизацией, аутентификацией и базами данных. Laravel также включает в себя систему шаблонов Blade, которая позволяет разработчикам создавать чистый и удобочитаемый код.

Другие известные фреймворки включают Symfony, CodeIgniter и Yii. Каждый из них имеет свои особенности и преимущества, позволяя разработчикам выбирать наиболее подходящий инструмент в зависимости от требований проекта.

Кроме того, PHP является основой для многих систем управления контентом (CMS), таких как WordPress, Joomla и Drupal. Эти платформы позволяют пользователям без глубоких знаний программирования создавать и управлять веб-сайтами. WordPress, в частности, стал самой популярной CMS в мире, обеспечивая более 40% всех сайтов в интернете.

Несмотря на свою популярность, PHP иногда подвергается критике за проблемы с безопасностью. Однако многие из этих проблем связаны с неправильным использованием языка или недостаточной защитой приложений со стороны разработчиков. Для повышения безопасности веб-приложений на PHP важно следовать лучшим практикам, таким как использование подготовленных запросов для работы с базами данных, регулярное обновление библиотек и фреймворков, а также применение методов аутентификации и авторизации пользователей.

Что касается производительности, PHP постоянно развивается. С выходом новых версий языка (например, PHP 7) были значительно улучшены скорость выполнения кода и использование памяти. Это делает его конкурентоспособным по сравнению с другими языками серверного программирования.

PHP остается одним из самых популярных языков для веб-разработки благодаря своей простоте, гибкости и мощным возможностям. Он идеально подходит для создания динамических веб-приложений и взаимодействия с базами данных. С поддержкой множества фреймворков и систем управления

контентом PHP предоставляет разработчикам широкий выбор инструментов для реализации их идей.

Несмотря на некоторые критические замечания по поводу безопасности и производительности, правильное использование PHP и следование современным стандартам разработки позволяют создавать надежные и масштабируемые приложения. В конечном итоге изучение PHP открывает множество возможностей для разработчиков в области веб-технологий и позволяет им создавать качественные решения для бизнеса и пользователей.

1.4 MySQL

MySQL — это одна из самых популярных систем управления базами данных (СУБД), основанная на реляционной модели. Она была разработана шведской компанией MySQL AB в начале 1990-х годов и с тех пор стала стандартом для хранения и управления данными в веб-приложениях. MySQL является системой с открытым исходным кодом, что делает её доступной для разработчиков и организаций по всему миру. В 2008 году MySQL была приобретена компанией Sun Microsystems, а затем в 2010 году — Oracle Corporation, что обеспечило дальнейшее развитие и поддержку этой СУБД.

MySQL поддерживает стандартный язык SQL (Structured Query Language), который используется для выполнения запросов к базе данных. Это позволяет разработчикам легко создавать, изменять и извлекать данные из баз данных. Одной из ключевых особенностей MySQL является её высокая производительность и способность обрабатывать большие объёмы данных. Благодаря использованию различных механизмов хранения, таких как InnoDB и MyISAM, MySQL может оптимизировать выполнение запросов и обеспечивать надежность данных.

MySQL также предлагает мощные инструменты для работы с транзакциями, что позволяет гарантировать целостность данных даже в случае сбоя системы. Система управления версиями InnoDB поддерживает механизмы блокировок на уровне строк, что минимизирует конфликты между параллельными транзакциями и повышает производительность.

Архитектура MySQL состоит из нескольких компонентов, включая серверную часть, клиентские библиотеки и интерфейсы для работы с различными языками программирования. Серверная часть отвечает за обработку запросов, управление данными и взаимодействие с клиентами. Клиентские библиотеки позволяют разработчикам подключаться к серверу и выполнять SQL-запросы из своих приложений.

Система также поддерживает различные API, включая ODBC (Open Database Connectivity) и JDBC (Java Database Connectivity), что делает её совместимой с широким спектром языков программирования, таких как PHP, Python, Java и C#. Это позволяет разработчикам интегрировать MySQL в свои приложения и использовать её возможности для хранения и обработки данных.

MySQL широко используется в различных сферах, включая веб-разработку, бизнес-приложения, аналитические системы и многое другое. Она является основой для многих популярных систем управления контентом (CMS), таких как WordPress, Joomla и Drupal. Эти платформы позволяют пользователям создавать и управлять веб-сайтами без необходимости глубоких знаний программирования, а MySQL обеспечивает надежное хранение данных.

Кроме того, MySQL активно используется в разработке веб-приложений благодаря своей способности обрабатывать запросы в реальном времени. Многие компании, такие как Facebook, Twitter и YouTube, используют MySQL для хранения пользовательских данных и обеспечения высокой производительности своих сервисов.

Безопасность является важным аспектом при работе с базами данных. MySQL предлагает различные механизмы для защиты данных, включая аутентификацию пользователей, шифрование соединений и управление правами доступа. Администраторы могут настраивать уровни доступа для пользователей и групп, что позволяет ограничивать доступ к конфиденциальной информации.

Кроме того, MySQL поддерживает регулярное резервное копирование данных и восстановление после сбоев. Это позволяет организациям минимизировать риски потери данных и обеспечивать их целостность.

Инструменты для мониторинга производительности также доступны в MySQL, что позволяет администраторам отслеживать состояние сервера и оптимизировать выполнение запросов.

MySQL остается одной из самых популярных систем управления базами данных благодаря своей простоте, гибкости и высокой производительности. Она идеально подходит для создания веб-приложений и управления большими объемами данных. С поддержкой различных языков программирования и платформ MySQL предоставляет разработчикам широкий выбор инструментов для реализации их идей.

Несмотря на некоторые вызовы в области безопасности, правильное использование MySQL и следование современным стандартам разработки позволяют создавать надежные и масштабируемые приложения. В конечном итоге изучение MySQL открывает множество возможностей для разработчиков в области работы с данными и позволяет им создавать качественные решения для бизнеса и пользователей.

2 Практическая часть

2.1 Требования к работе

Необходимо разработать web-приложения с применением современных web-технологий по тематике «Театр», используя технологии HTML, CSS, JavaScript, PHP и MySQL. Кроме того для своего проекта необходимо разработать базу данных для управления пользователями. Реализовать добавление, редактирование, просмотр и удаление пользователей. Также предусмотреть функции авторизации и аутентификации.

Описание реализации:

- неавторизированные пользователи должны иметь возможность регистрироваться. (форма добавления пользователя)
- авторизированные пользователи разделить на 2 группы: администраторы и авторизированные пользователи.
- администраторы имеют право просматривать, добавлять, редактировать и удалять пользователей.
- авторизированные пользователи имеют право просматривать и редактировать только свои данные.
- при регистрации пользователю присваивается статус авторизированного пользователя.

Требования к проекту:

- должна быть предусмотрена авторизация и регистрация пользователей;
- зарегистрированный пользователь имеет право редактировать свои данные и просматривать свои услуги;
- администраторы имеют право редактировать пользователей и добавлять услуги;
- неавторизированный пользователи имеют право просматривать список услуг;
- сайт должен быть выдержан в определенном стиле и корректно

открываться на любом разрешении;

- общий функционал может быть продуман самостоятельно;
- важно продемонстрировать не большой объем функционала, а минимальные требования к проекту – он может быть максимально простым, но при этом содержать все необходимые функции;
- отдельные бонусы предусмотрены за доп. функционал и красивый дизайн.

В качестве инструментов в работе будет использован текстовый редактор VScode, приложение для создания локального сервера OPEN server, в котором также можно найти веб-приложение phpMyAdmin для администрирования СУБД MySQL.

2.2 Создание базы данных

Перед началом работы необходимо запустить приложение XAMPP Control Panel и подождать. С помощью встроенного функционала (Рисунок 1) создалась база данных.

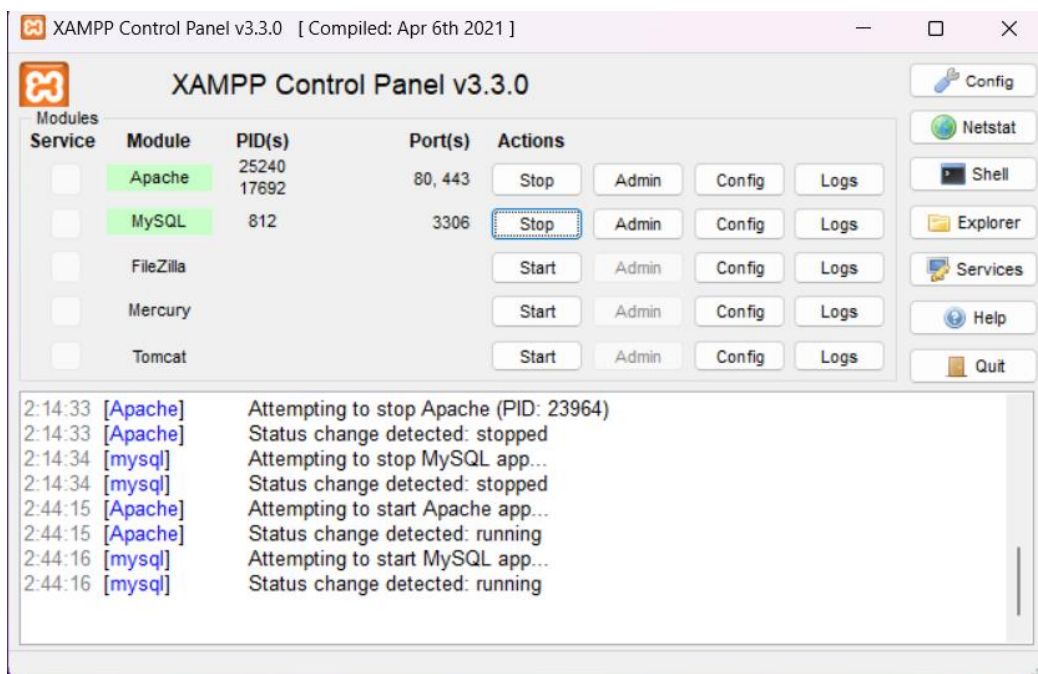


Рисунок 1 – запуск серверов через XAMPP Control Panel

Далее следует открыть веб-приложение phpMyAdmin (рис.2), в котором

можно создать базу данных через вспомогательные кнопки или напрямую через КОНСОЛЬ.

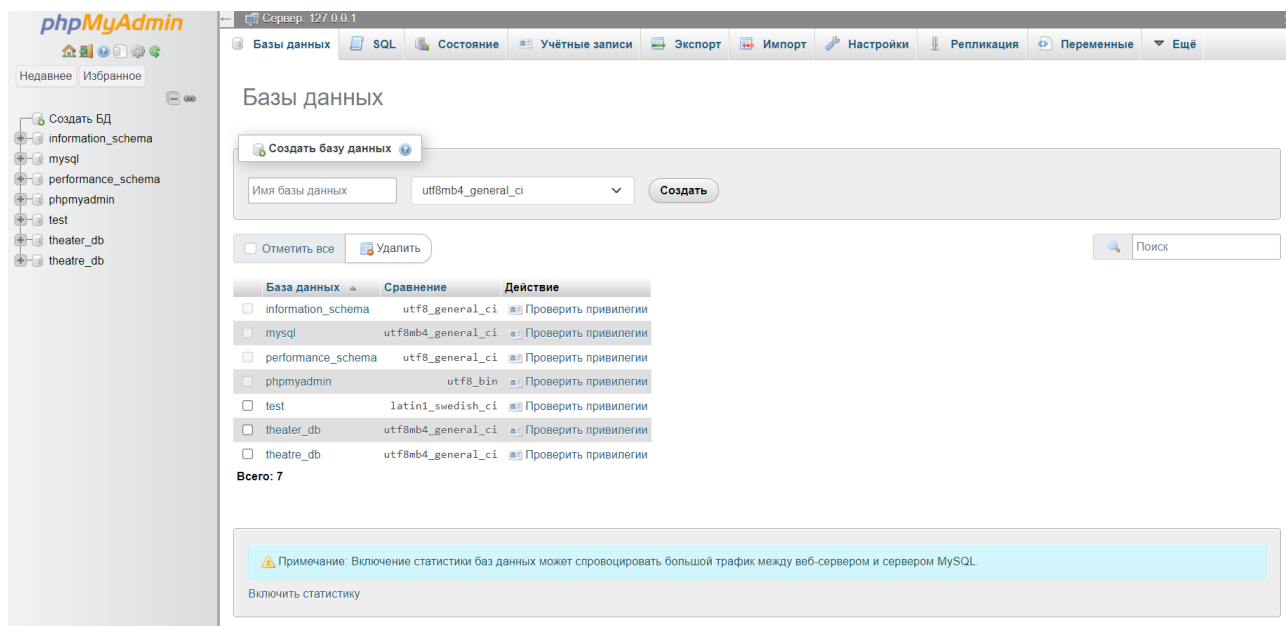


Рисунок 2 – веб-приложение phpMyAdmin

После создания базы данных следует проверить корректность ее подключения. Для этого можно использовать серверный язык php. В текстовом редакторе достаточно ввести следующий код:

```
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "";
5  $dbname = "base";
6
7  $conn = new mysqli($servername, $username, $password, $dbname);
8
9  if ($conn->connect_error) {
10     die("Connection failed: " . $conn->connect_error);
11 }
12 ?>
13
```

Рисунок 3 – код подключения

Теперь, при ошибке, связанной с подключением базы данных, выведется

сообщение “Connection failed”.

2.3 Главная страница

Попасть на главную страницу могут как авторизованные, так и неавторизованные пользователи. Сам документ называется `index.php` и содержит весь основной контент об веб-приложении Medical Services, используемом в качестве примера для курсовой работы.

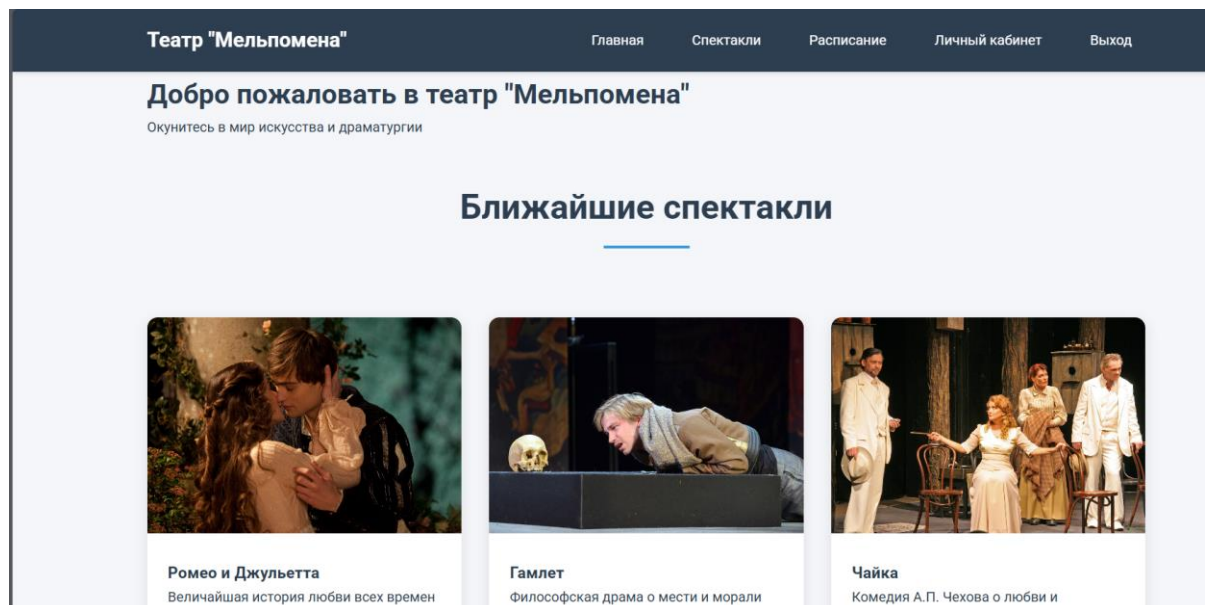


Рисунок 4 – главная страница для обычного пользователя

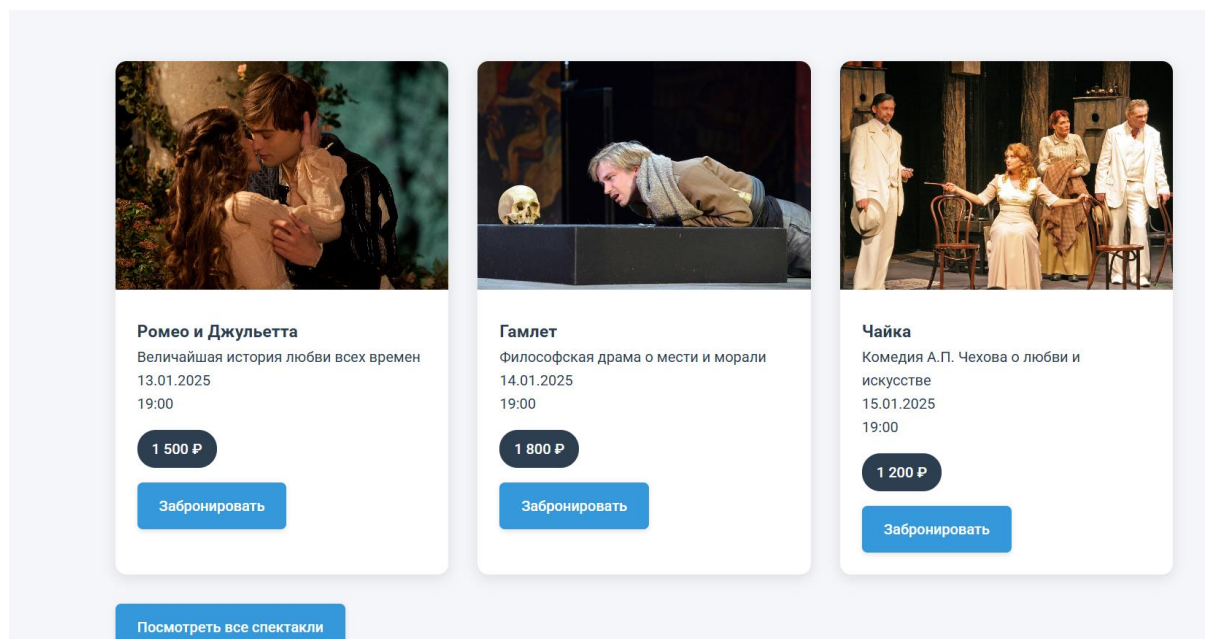


Рисунок 5- главная страница для администратора

Листинг index.php:

```
index.php
1  <?php
2  require_once 'includes/config.php';
3
4  // Отображение ошибок
5  error_reporting(E_ALL);
6  ini_set('display_errors', 1);
7
8  // Получение списка ближайших спектаклей
9  $sql = "SELECT s.*, ss.show_date, ss.show_time, ss.price, ss.available_seats
10         FROM shows s
11         LEFT JOIN show_schedules ss ON s.id = ss.show_id
12         ORDER BY ss.show_date ASC
13         LIMIT 3";
14  $result = $conn->query($sql);
15
16  if (!$result) {
17      die("Ошибка запроса: " . $conn->error);
18  }
19
20  // Проверка наличия данных
21  if ($result->num_rows == 0) {
22      die("Нет данных в таблице shows");
23  }
24  ?>
25
26  <!DOCTYPE html>
27  <html lang="ru">
28  <head>
29      <meta charset="UTF-8">
30      <meta name="viewport" content="width=device-width, initial-scale=1.0">
31      <title>Театр "Мельпомена"</title>
32      <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500;700&display=swap" rel="stylesheet">
33      <link rel="stylesheet" href="css/style.css">
34  </head>
35  <body>
36      <?php include 'includes/header.php'; ?>
37
38      <main>
39          <div class="hero">
40              <div class="container">
41                  <h1>Добро пожаловать в театр "Мельпомена"</h1>
42                  <p>Окунитесь в мир искусства и драматургии</p>
43              </div>
44          </div>
45
46          <section class="upcoming-shows">
47              <div class="container">
48                  <h2 class="section-title">Ближайшие спектакли</h2>
49                  <div class="shows-grid">
50                      <?php while($show = $result->fetch_assoc()): ?>
51                          <div class="show-card">
52                              ">
53                              <div class="show-info">
54                                  <h3><?php echo escape($show['title']); ?></h3>
55                                  <p><?php echo escape($show['description']); ?></p>
56                                  <div class="show-meta">
57                                      <p class="show-date">
58                                          <i class="far fa-calendar"></i>
59                                          <?php echo date('d.m.Y', strtotime($show['show_date'])); ?>
60                                      </p>
61                                      <p class="show-time">
62                                          <i class="far fa-clock"></i>
63                                          <?php echo date('H:i', strtotime($show['show_time'])); ?>
64                                      </p>
65                                      <p class="price-tag"><?php echo number_format($show['price'], 0, ',', ' '); ?> Р</p>
66                                  </div>
51                          </div>
52                      </div>
53                  </div>
54              </div>
55          </section>
56      </main>
57  </body>
58  </html>
```



```

66     </div>
67     <?php if($show['available_seats'] > 0): ?>
68         <a href="/lab10/booking.php?id=<?php echo $show['id']; ?>" class="btn btn-primary">Забронировать</a>
69     <?php else: ?>
70         <button class="btn btn-danger" disabled>Нет мест</button>
71     <?php endif; ?>
72 </div>
73 </div>
74 <?php endwhile; ?>
75 </div>
76 <div class="text-center">
77     <a href="/lab10/shows.php" class="btn btn-primary">Посмотреть все спектакли</a>
78 </div>
79 </div>
80 </section>
81
82 <section class="about-section">
83     <div class="container">
84         <h2 class="section-title">О нашем театре</h2>
85         <div class="about-content">
86             <div class="about-text">
87                 <p>Театр "Мельпомена" - это место, где классическое искусство встречается с современными постановками.
88                 Мы стремимся создавать незабываемые впечатления и дарить нашим зрителям яркие эмоции.</p>
89                 <p>Наш репертуар включает как классические произведения, так и современные постановки.
90                 Мы гордимся нашей труппой профессиональных актеров и режиссеров.</p>
91             </div>
92         </div>
93     </div>
94 </section>
95 </main>
96
97 <?php include 'includes/footer.php'; ?>
98 <script src="js/main.js"></script>
99 </body>
100 </html>

```

2.4 Регистрация и авторизация

Документ называется login.php. Он представляет собой белый фон с окном по середине при помощи которого пользователь сможет авторизироваться.

Вход в систему

Email:

Пароль:

Войти

Нет аккаунта? [Зарегистрируйтесь](#)

Рисунок 6 – окно авторизации

Листинг login.php:

```
1 <?php
2 require_once '../includes/config.php';
3
4 $error = '';
5
6 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
7     $email = $conn->real_escape_string($_POST['email']);
8     $password = $_POST['password'];
9
10    $sql = "SELECT id, username, password, role FROM users WHERE email = ?";
11    if ($stmt = $conn->prepare($sql)) {
12        $stmt->bind_param("s", $email);
13        $stmt->execute();
14        $result = $stmt->get_result();
15
16        if ($result->num_rows == 1) {
17            $user = $result->fetch_assoc();
18            if (password_verify($password, $user['password'])) {
19                $_SESSION['user_id'] = $user['id'];
20                $_SESSION['username'] = $user['username'];
21                $_SESSION['role'] = $user['role'];
22
23                header("Location: ../index.php");
24                exit();
25            } else {
26                $error = "Неверный пароль!";
27            }
28        } else {
29            $error = "Пользователь с таким email не найден!";
30        }
31        $stmt->close();
32    }
33 }
34 ?>
35
36 <!DOCTYPE html>
37 <html lang="ru">
38 <head>
39     <meta charset="UTF-8">
40     <meta name="viewport" content="width=device-width, initial-scale=1.0">
41     <title>Вход - Театр "Мельпомена"</title>
42     <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500;700&display=swap" rel="stylesheet">
43     <link rel="stylesheet" href="../css/style.css">
44 </head>
45 <body>
46     <?php include '../includes/header.php'; ?>
47
48     <main class="container">
49         <div class="form-container">
50             <h2>Вход в систему</h2>
51
52             <?php if ($error): ?>
53                 <div class="alert alert-error">
54                     <?php echo $error; ?>
55                 </div>
56             <?php endif; ?>
57
58             <form method="POST" action="">
59                 <div class="form-group">
60                     <label for="email">Email:</label>
61                     <input type="email" id="email" name="email" required>
62                 </div>
63
64                 <div class="form-group">
65                     <label for="password">Пароль:</label>
66                     <input type="password" id="password" name="password" required>
67                 </div>
68
69                 <button type="submit" class="btn btn-primary">Войти</button>
```

```

70     </form>
71
72     <p class="mt-3">
73         Нет аккаунта? <a href="register.php">Зарегистрируйтесь</a>
74     </p>
75 </div>
76 </main>
77
78 <?php include '../includes/footer.php'; ?>
79 </body>
80 </html>

```

Также со вкладки авторизация можно перейти на вкладку регистрация. Для этого используется файл registration.php.

Регистрация

Имя пользователя:

Email:

Пароль:

Подтвердите пароль:

[Зарегистрироваться](#)

Уже есть аккаунт? [Войдите](#)

Рисунок 7 -окно регистрации

Листинг страницы register.php:

```

1  <?php
2  require_once '../includes/config.php';
3
4  $error = '';
5  $success = '';
6
7  if ($_SERVER['REQUEST_METHOD'] == 'POST') {
8      $username = $conn->real_escape_string($_POST['username']);
9      $email = $conn->real_escape_string($_POST['email']);
10     $password = $_POST['password'];
11     $confirm_password = $_POST['confirm_password'];
12
13     // Проверка паролей
14     if ($password != $confirm_password) {
15         $error = "Пароли не совпадают!";
16     } else {
17         // Проверка существования email
18         $check_email = $conn->query("SELECT id FROM users WHERE email = '$email'");
19         if ($check_email->num_rows > 0) {
20             $error = "Пользователь с таким email уже существует!";
21         } else {
22             // Хеширование пароля
23             $hashed_password = password_hash($password, PASSWORD_DEFAULT);
24
25             // Добавление пользователя
26             $sql = "INSERT INTO users (username, email, password) VALUES (?, ?, ?)";
27             if ($stmt = $conn->prepare($sql)) {
28                 $stmt->bind_param("sss", $username, $email, $hashed_password);
29                 if ($stmt->execute()) {
30                     $success = "Регистрация успешна! Теперь вы можете войти.";
31                     header("refresh:2;url=login.php");
32                 } else {
33                     $error = "Что-то пошло не так. Попробуйте позже.";
34                 }
35                 $stmt->close();
36             }
37         }
38     }
39 }
40 ?>
41
42 <!DOCTYPE html>
43 <html lang="ru">
44 <head>
45     <meta charset="UTF-8">
46     <meta name="viewport" content="width=device-width, initial-scale=1.0">
47     <title>Регистрация - Театр "Мельпомена"</title>
48     <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500;700&display=swap" rel="stylesheet">
49     <link rel="stylesheet" href="../css/style.css">
50 </head>
51 <body>
52     <?php include '../includes/header.php'; ?>
53
54     <main class="container">
55         <div class="form-container">
56             <h2>Регистрация</h2>
57
58             <?php if ($error): ?>
59                 <div class="alert alert-error">
60                     <?php echo $error; ?>
61                 </div>
62             <?php endif; ?>
63
64             <?php if ($success): ?>
65                 <div class="alert alert-success">
66                     <?php echo $success; ?>

```

```

67     </div>
68     <?php endif; ?>
69
70     <form method="POST" action="">
71         <div class="form-group">
72             <label for="username">Имя пользователя:</label>
73             <input type="text" id="username" name="username" required>
74         </div>
75
76         <div class="form-group">
77             <label for="email">Email:</label>
78             <input type="email" id="email" name="email" required>
79         </div>
80
81         <div class="form-group">
82             <label for="password">Пароль:</label>
83             <input type="password" id="password" name="password" required>
84         </div>
85
86         <div class="form-group">
87             <label for="confirm_password">Подтвердите пароль:</label>
88             <input type="password" id="confirm_password" name="confirm_password" required>
89         </div>
90
91         <button type="submit" class="btn btn-primary">Зарегистрироваться</button>
92     </form>
93
94     <p class="mt-3">
95         Уже есть аккаунт? <a href="login.php">Войдите</a>
96     </p>
97 </div>
98 </main>
99
100 <?php include '../includes/footer.php'; ?>
101 </body>
102 </html>

```

2.5 Личный кабинет

При переходе в личный кабинет (файл profile.php) перед пользователем представлена возможность изменить необходимые данные. Также как и для регистрации, здесь используется функция, помогающая определить корректность ввода пароля.

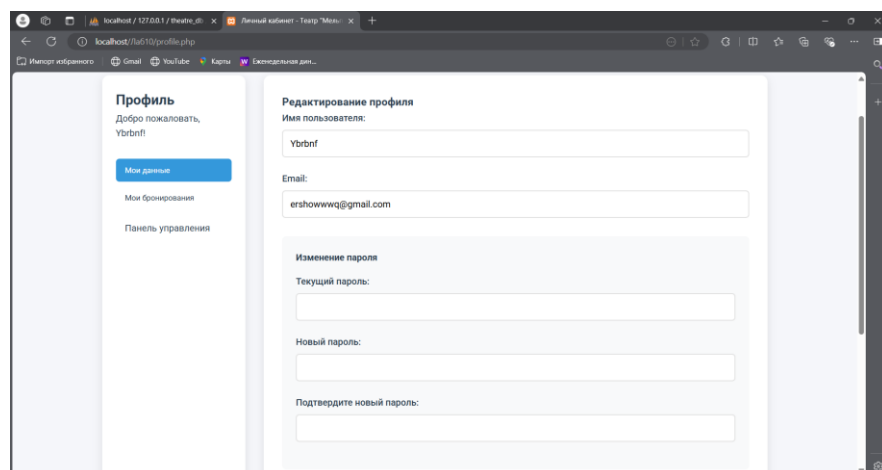


Рисунок 8 – личный кабинет для пользователя и его редактирование

Для объяснения принципа работы всей страницы стоит ознакомиться с кодом:

Листинг profile.php:

```
1  <?php
2  require_once 'includes/config.php';
3
4  // Проверка авторизации
5  if (!isLoggedIn()) {
6      header("Location: auth/login.php");
7      exit();
8  }
9
10 // Получение информации о пользователе
11 $user_id = $_SESSION['user_id'];
12 $sql = "SELECT * FROM users WHERE id = ?";
13 $stmt = $conn->prepare($sql);
14 $stmt->bind_param("i", $user_id);
15 $stmt->execute();
16 $user = $stmt->get_result()->fetch_assoc();
17
18 // Получение бронирований пользователя
19 $sql = "SELECT b.*, s.title, ss.show_date, ss.show_time, ss.price
20        FROM bookings b
21        JOIN show_schedules ss ON b.schedule_id = ss.id
22        JOIN shows s ON ss.show_id = s.id
23        WHERE b.user_id = ?
24        ORDER BY b.booking_date DESC";
25 $stmt = $conn->prepare($sql);
26 $stmt->bind_param("i", $user_id);
27 $stmt->execute();
28 $bookings = $stmt->get_result();
29
30 // Обработка обновления профиля
31 $success = '';
32 $error = '';
33
34 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
35     $username = $conn->real_escape_string($_POST['username']);
36     $email = $conn->real_escape_string($_POST['email']);
37     $current_password = $_POST['current_password'];
38     $new_password = $_POST['new_password'];
39     $confirm_password = $_POST['confirm_password'];
40
41     // Проверка текущего пароля
42     if (!empty($current_password)) {
43         if (password_verify($current_password, $user['password'])) {
44             if (!empty($new_password)) {
45                 if ($new_password === $confirm_password) {
46                     $hashed_password = password_hash($new_password, PASSWORD_DEFAULT);
47                     $sql = "UPDATE users SET username = ?, email = ?, password = ? WHERE id = ?";
48                     $stmt = $conn->prepare($sql);
49                     $stmt->bind_param("sssi", $username, $email, $hashed_password, $user_id);
50                 } else {
51                     $error = "Новые пароли не совпадают!";
52                 }
53             } else {
54                 $sql = "UPDATE users SET username = ?, email = ? WHERE id = ?";
55                 $stmt = $conn->prepare($sql);
56                 $stmt->bind_param("ssi", $username, $email, $user_id);
57             }
58
59             if (empty($error) && $stmt->execute()) {
60                 $success = "Профиль успешно обновлен!";
61                 // Обновление данных сессии
62                 $_SESSION['username'] = $username;
63             } else {
64                 $error = "Ошибка при обновлении профиля!";
65             }
66         } else {
67             $error = "Текущий пароль неверен!";
68         }
69     }
70 }
```

```

67         $error = "Неверный текущий пароль!";
68     }
69 } else {
70     $sql = "UPDATE users SET username = ?, email = ? WHERE id = ?";
71     $stmt = $conn->prepare($sql);
72     $stmt->bind_param("ssi", $username, $email, $user_id);
73     if ($stmt->execute()) {
74         $success = "Профиль успешно обновлен!";
75         $_SESSION['username'] = $username;
76     } else {
77         $error = "Ошибка при обновлении профиля!";
78     }
79 }
80 }
81 ?>
82
83 <!DOCTYPE html>
84 <html lang="ru">
85 <head>
86     <meta charset="UTF-8">
87     <meta name="viewport" content="width=device-width, initial-scale=1.0">
88     <title>Личный кабинет - Театр "Мельпомена"</title>
89     <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500;700&display=swap" rel="stylesheet">
90     <link rel="stylesheet" href="/lab10/css/style.css">
91 </head>
92 <body>
93     <?php include 'includes/header.php'; ?>
94
95     <main class="container">
96         <div class="profile-wrapper">
97             <div class="profile-sidebar">
98                 <div class="user-info">
99                     <div class="user-info">
100                         <h2>Профиль</h2>
101                         <p>Добро пожаловать, <?php echo escape($user['username']); ?>!</p>
102                     </div>
103
104                     <div class="profile-menu">
105                         <button class="menu-item active" data-tab="profile">Мои данные</button>
106                         <button class="menu-item" data-tab="bookings">Мои бронирования</button>
107                         <?php if (isAdmin()): ?>
108                             <a href="/lab10/admin/" class="menu-item">Панель управления</a>
109                         <?php endif; ?>
110                     </div>
111                 </div>
112
113                 <div class="profile-content">
114                     <div class="tab-content" id="profile">
115                         <h3>Редактирование профиля</h3>
116
117                         <?php if ($success): ?>
118                             <div class="alert alert-success"><?php echo $success; ?></div>
119                         <?php endif; ?>
120
121                         <?php if ($error): ?>
122                             <div class="alert alert-error"><?php echo $error; ?></div>
123                         <?php endif; ?>
124
125                         <form method="POST" action="" class="profile-form">
126                             <div class="form-group">
127                                 <label for="username">Имя пользователя:</label>
128                                 <input type="text" id="username" name="username" value="<?php echo escape($user['username']); ?>" required>

```

```

128 </div>
129
130 <div class="form-group">
131     <label for="email">Email:</label>
132     <input type="email" id="email" name="email" value="<?php echo escape($user['email']); ?>" required>
133 </div>
134
135 <div class="password-change">
136     <h4>Изменение пароля</h4>
137     <div class="form-group">
138         <label for="current_password">Текущий пароль:</label>
139         <input type="password" id="current_password" name="current_password">
140     </div>
141
142     <div class="form-group">
143         <label for="new_password">Новый пароль:</label>
144         <input type="password" id="new_password" name="new_password">
145     </div>
146
147     <div class="form-group">
148         <label for="confirm_password">Подтвердите новый пароль:</label>
149         <input type="password" id="confirm_password" name="confirm_password">
150     </div>
151 </div>
152
153 <button type="submit" class="btn btn-primary">Сохранить изменения</button>
154 </form>
155 </div>
156
157 <div class="tab-content" id="bookings" style="display: none;">
158     <h3>История бронирований</h3>
159
160     <?php if ($bookings->num_rows > 0): ?>
161         <div class="bookings-list">
162             <?php while ($booking = $bookings->fetch_assoc()): ?>
163                 <div class="booking-card">
164                     <div class="booking-header">
165                         <h4><?php echo escape($booking['title']); ?></h4>
166                         <span class="status status-<?php echo $booking['status']; ?>">
167                             <?php
168                             switch($booking['status']) {
169                                 case 'pending': echo 'Ожидает подтверждения'; break;
170                                 case 'confirmed': echo 'Подтверждено'; break;
171                                 case 'cancelled': echo 'Отменено'; break;
172                             }
173                             ?>
174                         </span>
175                     </div>
176                     <div class="booking-details">
177                         <p><strong>Дата:</strong> <?php echo date('d.m.Y', strtotime($booking['show_date'])); ?></p>
178                         <p><strong>Время:</strong> <?php echo date('H:i', strtotime($booking['show_time'])); ?></p>
179                         <p><strong>Количество мест:</strong> <?php echo $booking['seats_booked']; ?></p>
180                         <p><strong>Стоимость:</strong> <?php echo number_format($booking['total_price'], 0, ',', ' '); ?>
181                         <p><strong>Дата бронирования:</strong> <?php echo date('d.m.Y H:i', strtotime($booking['booking_
182                     </div>
183                     <?php if ($booking['status'] == 'pending'): ?>
184                         <form method="POST" action="/lab10/cancel_booking.php" onsubmit="return confirm('Вы уверены, что
185                         <input type="hidden" name="booking_id" value="<?php echo $booking['id']; ?>">
186                         <button type="button" class="btn btn-danger">Отменить бронирование</button>

```



```

187         </form>
188         <?php endif; ?>
189     </div>
190     <?php endwhile; ?>
191 </div>
192 <?php else: ?>
193     <p class="no-bookings">Вас пока нет бронирований</p>
194 <?php endif; ?>
195 </div>
196 </div>
197 </div>
198 </main>
199
200 <?php include 'includes/footer.php'; ?>
201 <script src="/наб10/js/main.js"></script>
202
203 <style>
204 .profile-wrapper {
205     display: grid;
206     grid-template-columns: 250px 1fr;
207     gap: 2rem;
208     margin: 2rem 0;
209 }
210
211 .profile-sidebar {
212     background: white;
213     border-radius: 12px;
214     padding: 1.5rem;
215     box-shadow: 0 4px 15px rgba(0,0,0,0.1);
216 }
217
218 .user-info {
219     margin-bottom: 2rem;
220 }
221
222 .profile-menu {
223     display: flex;
224     flex-direction: column;
225     gap: 0.5rem;
226 }
227
228 .menu-item {
229     padding: 0.75rem 1rem;
230     border: none;
231     background: none;
232     text-align: left;
233     cursor: pointer;
234     border-radius: 6px;
235     transition: all 0.3s;
236     color: var(--text-color);
237     text-decoration: none;
238 }
239
240 .menu-item:hover,
241 .menu-item.active {
242     background: var(--secondary-color);
243     color: white;
244 }
245
246 .profile-content {
247     background: white;
248     border-radius: 12px;
249     padding: 2rem;
250     box-shadow: 0 4px 15px rgba(0,0,0,0.1);

```

```

251 }
252
253 .password-change {
254     margin: 2rem 0;
255     padding: 1.5rem;
256     background: var(--gray-light);
257     border-radius: 8px;
258 }
259
260 .password-change h4 {
261     margin-bottom: 1rem;
262     color: var(--primary-color);
263 }
264
265 .bookings-list {
266     display: grid;
267     gap: 1rem;
268 }
269
270 .booking-card {
271     border: 1px solid #eee;
272     border-radius: 8px;
273     padding: 1.5rem;
274     transition: all 0.3s;
275 }
276
277 .booking-card:hover {
278     border-color: var(--secondary-color);
279     transform: translateX(5px);
280 }
281
282 .booking-header {
283     display: flex;
284     justify-content: space-between;
285 }
286
287 .booking-header {
288     display: flex;
289     justify-content: space-between;
290     align-items: start;
291     margin-bottom: 1rem;
292     padding-bottom: 1rem;
293     border-bottom: 1px solid #eee;
294 }
295
296 .booking-header h4 {
297     margin: 0;
298     color: var(--primary-color);
299 }
300
301 .status {
302     padding: 0.25rem 0.5rem;
303     border-radius: 4px;
304     font-size: 0.875rem;
305 }
306
307 .status-pending {
308     background: #fff3cd;
309     color: #856404;
310 }
311
312 .status-confirmed {
313     background: #d4edda;
314     color: #155724;
315 }
316
317 .status-cancelled {
318     background: #f8d7da;
319     color: #856404;
320 }

```

< config.php | init.sql >

< config.php | init.sql >

```

314     color: #721c24;
315 }
316
317 .booking-details p {
318     margin-bottom: 0.5rem;
319 }
320
321 .no-bookings {
322     text-align: center;
323     padding: 2rem;
324     color: var(--gray);
325     font-style: italic;
326 }
327
328 @media (max-width: 768px) {
329     .profile-wrapper {
330         grid-template-columns: 1fr;
331     }
332
333     .profile-menu {
334         flex-direction: row;
335         flex-wrap: wrap;
336     }
337 }
338 </style>
339
340 <script>
341 document.addEventListener('DOMContentLoaded', function() {
342     const menuItems = document.querySelectorAll('.menu-item');
343     const tabContents = document.querySelectorAll('.tab-content');
344
345     menuItems.forEach(item => {
346         item.addEventListener('click', function() {
347             const tabId = this.dataset.tab;
348
349             // Активация кнопки меню
350             menuItems.forEach(mi => mi.classList.remove('active'));
351             this.classList.add('active');
352
353             // Показ соответствующего контента
354             tabContents.forEach(tab => {
355                 tab.style.display = tab.id === tabId ? 'block' : 'none';
356             });
357         });
358     });
359 </script>
360
361 </body>
362 </html>

```

Далее рассмотрим созданную базу данных.

Созданная база данных имеет следующие таблицы:

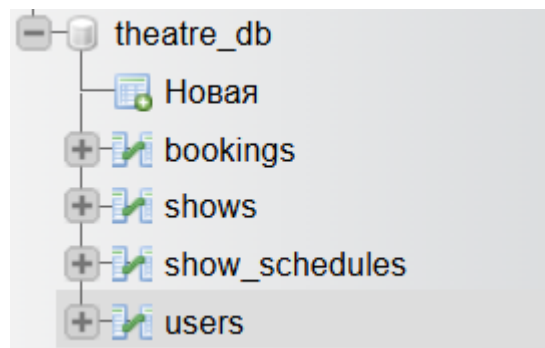


Рисунок 9 – база данных










			id	show_id	show_date	show_time	price	total_seats	available_seats	created_at
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	1	1	2025-01-13	19:00:00	1500.00	100	99 2025-01-12 22:53:25
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	2	2	2025-01-14	19:00:00	1800.00	100	100 2025-01-12 22:53:25
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	3	3	2025-01-15	19:00:00	1200.00	100	100 2025-01-12 22:53:25

Рисунок 10 – Таблица выложенных спектаклей на сайт



			id	username	email	password	role	created_at
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	1	Ybrbnf	ershowwwq@gmail.com	\$2y\$10\$7vgYc/LfVTyh9EOMpc.Vj.q7Hr2hjoOF5WNOjdM6WDA...	admin 2025-01-12 22:53:39

Рисунок 11 – Таблица пользователей и админов

Заключение

В ходе работы была проведена комплексная разработка, включающая проектирование, реализацию и оптимизацию веб-приложения, ориентированного на пользователей, интересующихся театральным искусством.

Мы рассмотрели современные технологии, такие как HTML5, CSS3, JavaScript и фреймворки, такие как React или Vue.js, которые позволили создать интерактивный и удобный интерфейс. В результате анализа требований пользователей и функциональных возможностей приложения была разработана архитектура, обеспечивающая высокую производительность и отзывчивость.

В процессе разработки были реализованы ключевые функции, включая систему бронирования билетов, каталог спектаклей, а также возможность оставлять отзывы и комментарии. Это не только улучшает взаимодействие пользователей с приложением, но и способствует созданию сообщества любителей театра.

Оптимизация приложения проводилась с учетом принципов SEO и производительности, что позволило значительно улучшить время загрузки страниц и повысить видимость в поисковых системах.

В заключение, разработанное веб-приложение не только отвечает современным требованиям пользователей, но и демонстрирует возможности применения актуальных технологий в сфере культурных проектов. Полученные знания и навыки будут полезны для дальнейшей работы в области разработки веб-приложений, а также для создания новых проектов, направленных на популяризацию театрального искусства.

Таким образом, данная работа подтверждает важность интеграции современных технологий в культурные инициативы и открывает новые горизонты для развития театральной сферы в цифровом пространстве.

Список использованных источников

1. Основы CSS: mdn web docs [Электронный источник]. – URL: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics [дата доступа 12.10.2024].
2. Основы HTML: mdn web docs [Электронный источник]. – URL: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/HTML_basics [дата доступа 12.10.2024].
3. Основы JavaScript: mdn web docs [Электронный источник]. – URL: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/JavaScript_basics [дата доступа 12.10.2024].
4. Подробнее о PHP [Электронный источник]. – URL: https://ru.hexlet.io/courses/php-setup-environment/lessons/what-is-php/theory_unit (дата обращения: 11.11.2024).
5. Что такое MySQL: Макхост [Электронный источник]. – URL: <https://mchost.ru/articles/chto-takoe-mysql/> [дата доступа 13.12.2024].
6. Что такое - PHP? [Электронный источник]. – URL: <https://archive-ipq-co.narod.ru/> (дата обращения: 13.12.2023).
7. Что такое HTML и CSS [Электронный источник]. – URL: <https://shneider-host.ru/blog/chto-takoe-html-i-css.html> (дата обращения: 11.12.2024).
8. Язык разметки HTML, таблицы стилей CSS и язык программирования JavaScript [Электронный источник]. – URL: https://web-creator.ru/articles/about_client_side (дата обращения: 21.10.2024).
9. PHP: википедия [Электронный источник]. – URL: <https://ru.wikipedia.org/wiki/PHP> [дата доступа 12.12.2024].