Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ

КАФЕДРА БИЗНЕС-ИНФОРМАТИКИ И МОДЕЛИРОВАНИЯ БИЗНЕС-ПРОЦЕССОВ

**КУРСОВАЯ РАБОТА**

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

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.И. Никитин

подпись, дата

Студент УБ22-09Б, 432216061 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К. А. Зайцева

подпись, дата

Красноярск 2024

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc181055353)

[1 Теоретическая часть 4](#_Toc181055354)

[1.1 HTML 4](#_Toc181055355)

[1.2 CSS 5](#_Toc181055356)

[1.3 JavaScript 7](#_Toc181055357)

[1.4 PHP 8](#_Toc181055358)

[1.5 MySQL 9](#_Toc181055359)

[2 Практическая часть 12](#_Toc181055360)

[2.1 Требования к web-приложению и его характеристика 12](#_Toc181055361)

[2.2 Создание базы данных 13](#_Toc181055362)

[2.3 Регистрация и авторизация 15](#_Toc181055363)

[2.4 Главная страница 20](#_Toc181055364)

[2.5 Дополнительные страницы 21](#_Toc181055365)

[2.5.1 Профиль пользователя 21](#_Toc181055366)

[2.5.2 Список книг 23](#_Toc181055367)

[2.5.3 Добавление книги 25](#_Toc181055368)

[2.6 Стилизация интерфейса с помощью CSS 27](#_Toc181055369)

[ЗАКЛЮЧЕНИЕ 33](#_Toc181055370)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 34](#_Toc181055371)

# ВВЕДЕНИЕ

В условиях развития цифровых технологий и увеличения популярности обмена информацией онлайн наблюдается рост интереса к платформам, которые способствуют свободному обмену книгами среди пользователей. Одним из таких направлений является буккроссинг — международное движение, призванное способствовать распространению знаний и литературных произведений, позволяя людям обмениваться книгами, находить единомышленников и развивать культуру чтения. Настоящий проект представляет собой разработку веб-приложения, выполняющего функции интернет-буккроссинга, который позволяет пользователям загружать книги, делиться ими, а также скачивать книги, загруженные другими пользователями. Основная цель проекта — создать удобную, безопасную и доступную платформу, которая бы способствовала обмену литературой и объединяла людей на основе их интересов к литературе.Цель: разработать сайт, используя технологии HTML, CSS, Javascript, PHP, MySQL на тему «Обмен книгами».

Объект исследования: создание web-приложения на тему «Обмен книгами».

Предмет исследования: процесс создания web-приложения на тему «Обмен книгами».

# Теоретическая часть

## **HTML**

HTML, также известный как Hypertext Markup Language, – это код, который используется для организации и отображения веб-страниц и их контента. Например, контент может быть структурирован в виде маркированных списков, параграфов или с использованием изображений и таблиц данных.

HTML – это не язык программирования, – это язык разметки, который помогает вашему браузеру отображать веб-страницы. HTML состоит из множества элементов, которые вы можете использовать, чтобы включать или оборачивать отдельные части контента, чтобы заставить его отображаться или действовать определенным образом. Ограждающие теги могут изменять текст или изображение, делать слова курсивом, изменять размер шрифта и так далее [1].

С его помощью можно описать структуру и содержимое страницы, определить различные элементы и их взаимосвязь. HTML является основным языком для создания содержимого веб-сайтов и организации информации для отображения в браузере. Основными компонентами HTML являются теги, которые определяют начало и конец элемента на веб-странице. Различные теги описывают разные элементы страницы, отображая текст, изображения, ссылки, списки, таблицы и другие объекты. Разработчики используют комбинацию тегов для создания структуры и внешнего вида страницы. Например, для создания заголовка страницы используется тег "<h1>", а для параграфа – тег "<p>". Ссылки создаются с помощью тега "<a>", а изображения – с помощью тега "<img>" [2]. Теги могут иметь атрибуты, которые дополняют или изменяют их поведение и внешний вид. Например, атрибут "href" определяет адрес, на который будет вести ссылка. Кроме основных элементов, HTML также поддерживает встраиваемые скрипты и стили. Скрипты позволяют добавить динамическое поведение на страницу, в то время как стили используются для определения внешнего вида элементов, об этом в параграфах 1.2 и 1.3.

HTML также позволяет организовать страницу в различные секции с помощью элементов, таких как "<header>", "<nav>", "<main>", "<section>", "<aside>" и "<footer>". Это помогает разбить страницу на логические блоки и облегчает навигацию и понимание структуры страницы. Для создания сложной таблицы данных можно использовать элементы "<table>", "<tr>", "<td>" и другие. Элементы формы, такие как "<input>", "<textarea>" и "<select>", позволяют создавать интерактивный контент и собирать информацию от пользователей.

Можно рассмотреть на примере, из чего состоит элемент абзаца. Запись «<p>Текст<p>» содержит открывающий тег <p> и закрывающий тег </p> в угловых скобках, контент (Текст), – все это вместе составляющие элемента абзаца [1].

HTML является открытым стандартом, что означает, что любой может использовать и расширять его. Новые версии HTML, такие как HTML5, вводят новые возможности и элементы, улучшающие пользовательский интерфейс, производительность и доступность и делающие разработку более простой и эффективной. HTML является основой веб-разработки и неотъемлемой частью создания веб-страниц. Он позволяет веб-разработчикам создавать красивые, удобочитаемые и уникальные страницы, а также обеспечивает доступность и интерактивность для пользователей. Знание HTML является необходимым для всех, кто интересуется созданием и разработкой веб-сайтов.

## **CSS**

Набор параметров форматирования, который применяется к элементам документа, чтобы изменить их внешний вид, называется стилем или CSS (Cascading Style Sheets, каскадные таблицы стилей). Современные издательские системы и текстовые редакторы давно поддерживают возможность работы со стилями, позволяя одним нажатием кнопки придать тексту заранее заданный вид. Создатели сайтов могут теперь использовать это, храня цвет, размеры текста и другие параметры в одном месте и легко «прикручивая» их к любому тегу. Стили также предлагают гораздо больше возможностей для форматирования, чем обычный HTML, что является дополнительным преимуществом [3].

Выделяют следующие преимущества использования каскадных таблиц стилей:

* Большое количество возможностей для веб-дизайнера, поскольку CSS – это стандарт W3C, разработанный специально для дизайна сайтов. Не все идеи можно реализовать с помощью HTML.
* Стили и их описания могут быть включены в отдельный файл. Таким образом, работа параллельно с дизайном и контентом сайта исключает путаницу.
* Из предыдущего пункта вытекают преимущества хранения стилей в одном файле (обычно style.css), а путь к нему указывается в начале документа. Таким образом, для изменения вида страницы достаточно использовать один файл.
* Поскольку таблицы стилей сохраняются в кэш-памяти, загрузка сайта происходит быстрее. Кроме того, можно значительно сократить количество кода, написанного с помощью CSS.
* Блочная ориентация облегчает разработку единого стиля для каждой страницы веб-сайта, поскольку все страницы объединены в одну таблицу. Кроме того, это позволяет более точно управлять оформлением, поскольку изменение одной строки изменит все представление на каждой странице.
* Таблицы стилей позволяют изменять представление страниц для различных устройств, что очень важно в настоящее время [4].

Чтобы задать красный цвет тексту во всех абзацах, можно написать следующий код: P {color: red;}.

Анатомия набора правил CSS, как можно видеть, проста:

* Существует селектор – элемент, к которому применяется изменение. В качестве селектора могут быть теги html, а также придуманные классы (.class), универсальный селектор (\*), идентификатор (#id), атрибут ([attr\*=value]).
* Объявление единственного правила, которое изменяет свойство элемента.
* Непосредственно сами свойства CSS, способы, которыми вы можете стилизовать определённый HTML-элемент.
* Справа от свойства, после двоеточия, у нас есть значение свойства, которое выбирает одно из множества возможных признаков для данного свойства.
* Каждый набор правил упакован в фигурных скобках.
* После объявления свойства CSS ставится двоеточие перед значением свойства. В каждом наборе правил необходимо использовать точку с запятой, чтобы отличать каждое объявление от следующего [5].

Важно отметить, что у CSS есть препроцессор SCSS или SASS. Это метаязык на основе CSS, предназначенный для увеличения уровня абстракции CSS-кода и упрощения файлов каскадных таблиц стилей, включён в состав языка разметки Haml.

## **JavaScript**

JavaScript (JS) – это высокоуровневый, интерпретируемый язык программирования, который применяется для создания динамических веб-страниц и веб-приложений. JS позволяет добавить интерактивность и функциональность на сайты, делая их более привлекательными для пользователей [6].

JavaScript разработан Brendan Eich в 1995 году для использования в Netscape Navigator. Сегодня он является одним из трех основных языков веб-разработки, вместе с HTML и CSS. JS может работать как на стороне клиента (front-end), так и на стороне сервера (back-end) с использованием платформы Node.js.

JavaScript включает в себя множество функций и возможностей. Он поддерживает объектно-ориентированное программирование (ООП), что позволяет разработчикам организовывать код вокруг объектов и их взаимодействия. Также JS поддерживает функциональное программирование, которое облегчает работу с функциями как с объектами первого класса.

JS имеет широкий набор встроенных функций и методов, которые позволяют работать с строками, числами, массивами, объектами и другими типами данных. Он также предоставляет возможности для манипуляции DOM (Document Object Model) - структура веб-страницы, что позволяет изменять содержимое и стиль элементов в реальном времени.

JS имеет поддержку асинхронного программирования с помощью промисов и асинхронных функций. Это особенно полезно при выполнении операций, которые могут занимать много времени, таких как загрузка данных с сервера или выполнение сложных вычислений. JS также позволяет создавать анимации, добавлять эффекты перехода и взаимодействия с элементами страницы через CSS-анимации и CSS-трансформации. Для работы с JS разработчикам доступны различные инструменты и фреймворки. Некоторые из самых популярных фреймворков включают React.js, Angular.js и Vue.js, которые предоставляют разнообразные инструменты и библиотеки для разработки сложных веб-приложений и компонентов пользовательского интерфейса [7].

## **PHP**

PHP, аббревиатура от «Hypertext Preprocessor», является языком программирования, предназначенным для веб-разработки. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML. Многие люди используют его для создания динамичных веб-страниц и приложений. PHP имеет множество возможностей для разработчиков и является одним из наиболее популярных языков программирования на сервере.

PHP является бесплатным и открытым языком программирования. В связи с тем, что он работает на сервере, весь код PHP выполняется на сервере перед тем, как данные отправляются на компьютер пользователя. Способность взаимодействовать с базами данных, обрабатывать формы и создавать динамический контент на веб-страницах являются основными преимуществами PHP.

PHP предлагает разработчикам множество встроенных функций, которые позволяют им создавать мощные веб-приложения. Он поддерживает объектно-ориентированное программирование (OOP), которое позволяет организовывать код вокруг объектов и их взаимодействия. Кроме того, PHP имеет много встроенных функций, которые позволяют работать с файлами, строками, массивами и другими типами данных.

Возможность работы с базами данных является одной из основных функций PHP. PHP работает вместе с несколькими системами управления базами данных (СУБД), такими как MySQL, PostgreSQL и SQLite. Это облегчает создание, извлечение, изменение и удаление данных из баз данных и использование их на веб-страницах.

При написании страницы сайта на html можно вставить что-либо на языке PHP просто написав такие «скобки»: «<?php /Блок PHP кода/ ?>» [8].

## **MySQL**

MySQL – это реляционная база данных (СУБД), предназначенная для обработки, организации и хранения больших количеств данных. Она предлагает широкий набор функций для удобной работы с данными и является одной из самых популярных и широко используемых СУБД в мире.

Ядро MySQL, MySQL Workbench, сервер MySQL и протокол командного интерфейса являются основными компонентами MySQL.

Ядро MySQL является важной частью СУБД и выполняет важные функции, такие как управление данными, обработка транзакций и поддержка языка запросов. С его помощью можно создавать, изменять и удалять базы данных, таблицы и индексы, а также выполнять сложные запросы на извлечение данных.

Программное обеспечение под названием сервер MySQL обрабатывает запросы клиентов и управляет доступом к данным. Он является ключевым компонентом, обеспечивающим масштабируемость и надежность MySQL. Различные модули и расширения, поддерживаемые сервером MySQL, позволяют расширять его функциональность и добавлять новые возможности.

Через командную строку или скрипты на языке Perl или PHP протокол командного интерфейса позволяет взаимодействовать с базой данных MySQL. Благодаря этому простому, но мощному интерфейсу администраторы могут управлять базами данных, выполнять запросы для проверки и обрабатывать данные. Поддержка транзакций и многопользовательский доступ являются важными компонентами MySQL. Транзакции гарантируют, что данные остаются защищенными и что информация будет правильно восстановлена в случае сбоев. MySQL также предлагает систему управления доступом, которая позволяет пользователям назначать различные уровни доступа к базам данных и таблицам, чтобы обеспечить безопасность данных.

MySQL популярен среди разработчиков и администраторов из-за его гибкой архитектуры и простоты использования. Веб-разработка, управление данными и аналитика – лишь некоторые из областей, в которых он используется. MySQL продолжает оставаться одной из ведущих систем управления базами данных, добавляя новые возможности, чтобы удовлетворить потребности современных проектов [9].

Одним из совместимых с MySQL веб-приложением является PHPMyAdmin, которое представляет из себя веб-интерфейс для администрирования СУБД MySQL.

В данной главе были рассмотрены теоретические аспекты о HTML, CSS, JS, PHP, MySQL, которые являются основными технологиями для создания web-приложений. Таким образом, HTML используется для структурирования контента, CSS - для визуального оформления, JS - для создания интерактивности на веб-странице, PHP - для обработки данных на сервере, а MySQL - для управления базами данных. В контексте разработки сайта-портфолио, эти технологии являются не только актуальными, но и практически обязательными. HTML позволяет создавать различные разделы и компоненты страницы, CSS придает им стиль и внешний вид, JS делает сайт интерактивным и динамическим, а PHP с MySQL позволяют хранить и получать данные о портфолио пользователя. Без этих технологий создание динамического и функционального сайта-портфолио будет затруднительным. HTML, CSS и JS - это основополагающие языки разработки веб-сайтов, а PHP и MySQL обеспечивают функциональность и возможность хранения информации. Их использование позволяет создавать интерактивные, адаптивные и практичные сайты-портфолио, привлекая посетителей и успешно демонстрируя достижения и навыки автора.

# Практическая часть

## **2.1 Требования к web-приложению и его характеристика**

К проекту, разрабатываемому web-приложению, были поставлены следующие требования:

1. Должна быть предусмотрена авторизация и регистрация пользователей.
2. Зарегистрированный пользователь имеет право редактировать свои данные и просматривать свои услуги.
3. Администраторы имеют право редактировать пользователей и добавлять услуги.
4. Неавторизированный пользователи имеют право просматривать список услуг
5. Сайт должен быть выдержан в определенном стиле и корректно открываться на любом разрешении.
6. Общий функционал может быть продуман самостоятельно.
7. Важно продемонстрировать не большой объем функционала, а минимальные требования к проекту. Проект может максимально простым, но при этом содержать все необходимые функции.
8. Отдельные бонусы предусмотрены за доп. функционал и красивый дизайн.

Также необходимо разработать базу данных для управления пользователями, используя СУБД MySQL. Реализовать добавление, редактирование, просмотр и удаление пользователей. Также предусмотреть функции авторизации и аутентификации. Более подробное описание реализации следующее:

* 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.

## **2.2 Создание базы данных**

Перед началом работы необходимо запустить приложение MAMP и подождать, пока автоматически не включатся необходимые серверы: apache Server и MySQL Server (рис.1).

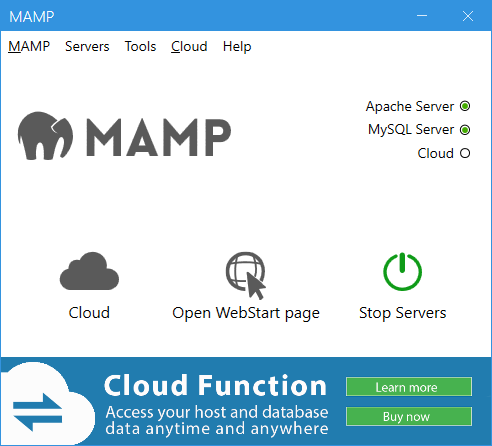


Рисунок 1 – запуск серверов через MAMP

Создание базы данных:

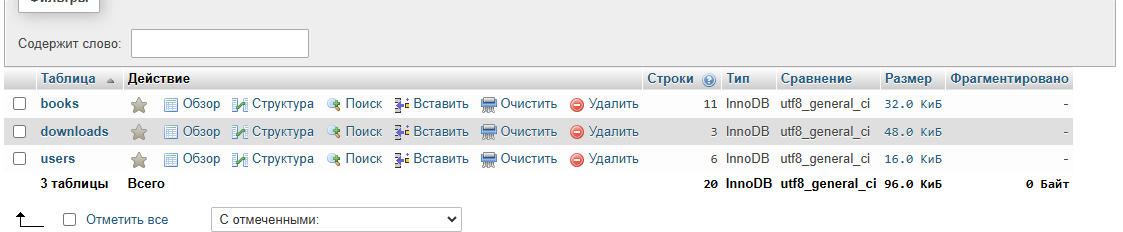
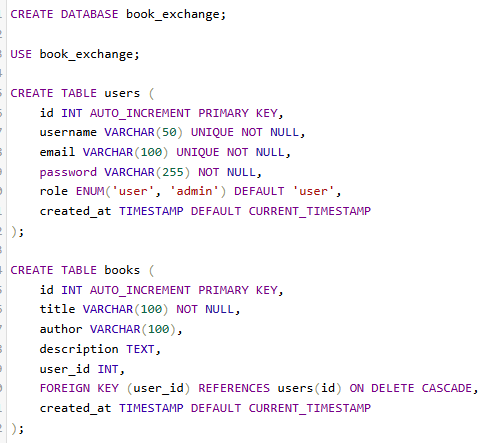


Рисунок 2 – Создание базы данных

Листинг:



База данных для проекта буккроссинга состоит из трех основных таблиц:

1. **users** – хранит данные пользователей:
   * id (INT, AUTO\_INCREMENT) — уникальный идентификатор пользователя.
   * username (VARCHAR) — имя пользователя.
   * email (VARCHAR) — электронная почта.
   * password (VARCHAR) — хэш пароля.
   * is\_admin (TINYINT) — флаг администратора (0 для обычного пользователя, 1 для администратора).
2. **books** – хранит данные о книгах, загруженных пользователями:
   * id (INT, AUTO\_INCREMENT) — уникальный идентификатор книги.
   * title (VARCHAR) — название книги.
   * author (VARCHAR) — автор книги.
   * genre (VARCHAR) — жанр книги.
   * owner\_id (INT) — идентификатор пользователя, загрузившего книгу, связанный с users.id.
   * file\_path (VARCHAR) — путь к файлу книги.
3. **downloads** – отслеживает скачивания книг пользователями:
   * id (INT, AUTO\_INCREMENT) — уникальный идентификатор записи.
   * user\_id (INT) — идентификатор пользователя, скачавшего книгу, связанный с users.id.
   * book\_id (INT) — идентификатор книги, связанный с books.id.
   * download\_date (TIMESTAMP) — дата и время скачивания.

## **2.3 Регистрация и авторизация**

На сайте согласно требованиям предусмотрена регистрация и авторизация пользователей. Страницы login.php для авторизации и register.php для регистрации представлены на рисунке 3 и 4.

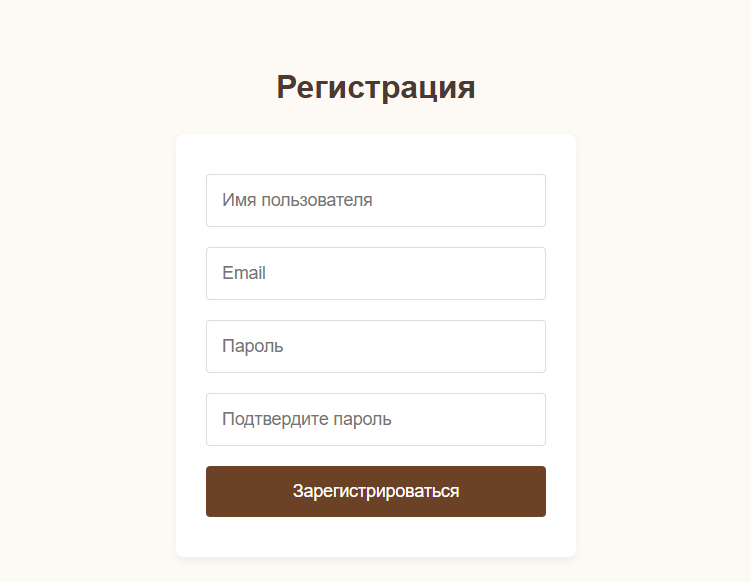


Рисунок 3 – Регистрация (страница register.php)

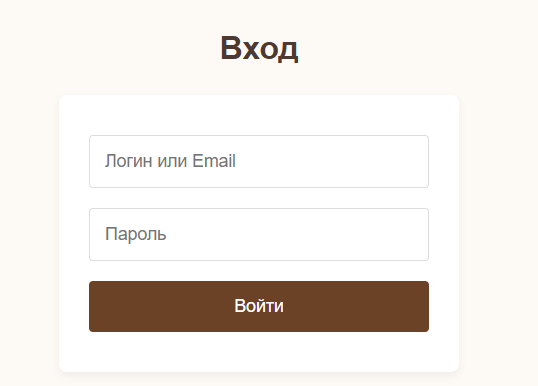


Рисунок 4 – Вход (страница login.php)

Регистрация пользователя в этом проекте включает следующие шаги

Форма регистрации (register.php): пользователь заполняет форму регистрации, вводя имя пользователя (username), адрес электронной почты (email) и пароль (password). Форма отправляется на сервер через метод POST для обработки.

Обработка данных на сервере: при получении данных от формы проверяется, что все поля заполнены. Пароль хэшируется с помощью password\_hash() для безопасного хранения.

Сохранение данных в базе: в таблицу users вставляется новая запись с username, email и хэшированным password. Уникальность username и email желательно проверять, чтобы предотвратить повторную регистрацию с одинаковыми данными.

Перенаправление на страницу входа: после успешной регистрации пользователь перенаправляется на страницу авторизации (login.php), чтобы войти в систему.

Проверка ошибок: если данные не прошли проверку (например, email уже зарегистрирован), пользователю показывается соответствующее сообщение об ошибке.

Листинг программы регистрации:

<?php

include 'db.php';

$success = "";

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

    $username = $\_POST['username'];

    $email = $\_POST['email'];

    $password = password\_hash($\_POST['password'], PASSWORD\_DEFAULT);

    $stmt = $pdo->prepare("INSERT INTO users (username, email, password) VALUES (?, ?, ?)");

    $stmt->execute([$username, $email, $password]);

    $success = "Регистрация успешна! Вы можете войти в систему.";

}

?>

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <title>Регистрация</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div class="container">

        <h2 style= "text-align:center">Регистрация</h2>

        <?php if ($success): ?>

            <p style="color:green;"><?php echo $success; ?></p>

        <?php endif; ?>

        <form id="registerForm" action="register.php" method="POST">

            <input type="text" name="username" placeholder="Имя пользователя" required>

            <input type="email" name="email" placeholder="Email" required>

            <input type="password" name="password" placeholder="Пароль" required>

            <input type="password" name="confirm\_password" placeholder="Подтвердите пароль" required>

            <button type="submit">Зарегистрироваться</button>

        </form>

    </div>

</body>

</html>

Авторизация в этом включает следующие шаги:

Форма авторизации (login.php): пользователь вводит свои логин или электронную почту и пароль в форму авторизации. Данные отправляются на сервер через метод POST для проверки.

Проверка данных на сервере: сервер принимает введенные данные и выполняет поиск пользователя в базе данных по логину или электронной почте. Если пользователь найден, введенный пароль проверяется с помощью функции password\_verify(), которая сравнивает введенный пароль с хэшированным паролем в базе данных.

Успешная авторизация: если проверка пройдена успешно, в сессии сохраняются данные пользователя (например, user\_id, username и статус is\_admin). Пользователь перенаправляется на страницу профиля (profile.php) или другую основную страницу приложения.

Обработка ошибок: если введены неверные логин/пароль или пользователь не найден, пользователю отображается сообщение об ошибке.

Листинг программы авторизации:

<?php

include 'db.php';

session\_start();

$error = "";

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

    $identifier = $\_POST['identifier']; // Логин или email

    $password = $\_POST['password'];

    // Запрос, проверяющий, есть ли такой пользователь по логину или email

    $stmt = $pdo->prepare("SELECT \* FROM users WHERE username = ? OR email = ?");

    $stmt->execute([$identifier, $identifier]);

    $user = $stmt->fetch();

    if ($user && password\_verify($password, $user['password'])) {

        $\_SESSION['user\_id'] = $user['id'];

        $\_SESSION['username'] = $user['username'];

        $\_SESSION['is\_admin'] = $user['is\_admin'];

        header("Location: profile.php");

        exit();

    } else {

        $error = "Неверный логин, email или пароль!";

    }

}

?>

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <title>Вход</title>

    <link rel="stylesheet" href="assets/style.css">

</head>

<body>

    <div class="container">

        <h2>Вход</h2>

        <?php if ($error): ?>

            <p style="color:red;"><?php echo $error; ?></p>

        <?php endif; ?>

        <form action="login.php" method="POST">

            <input type="text" name="identifier" placeholder="Логин или Email" required>

            <input type="password" name="password" placeholder="Пароль" required>

            <button type="submit">Войти</button>

        </form>

    </div>

</body>

</html>

## **2.4 Главная страница**

Главная страница (index.php): отображает приветствие и информацию о сайте. Пользователям, не вошедшим в систему, показывает ссылки на страницы регистрации и входа. Авторизованным пользователям предоставляет ссылки на профиль и список книг (рисунок 5).



Рисунок 5 – Главная страница для неавторизированных пользователей

Листинг главной страницы:

<?php

session\_start();

include 'db.php';

?>

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <title>Обмен книгами</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

<?php include 'sidebar.php'; ?>

    <div class="container">

        <h1 style= "text-align:center" >Добро пожаловать в библиотеку буккроссинга</h1>

        <p style="text-align:center; font-size:1.2rem; color:#6b4226;">

            Здесь вы можете скачать или поделиться любимыми книгами с другими читателями.

        </p>

        <?php if (isset($\_SESSION['user\_id'])): ?>

            <p>Здравствуйте, <?php echo $\_SESSION['username']; ?>!</p>

            <a href="profile.php">Профиль</a> | <a href="book\_list.php">Список книг</a> | <a href="logout.php">Выйти</a>

        <?php else: ?>

            <a href="register.php">Регистрация</a> | <a href="login.php">Вход</a> | <a href="book\_list.php">Список книг</a>

        <?php endif; ?>

    </div>

</body>

</html>

## **2.5 Дополнительные страницы**

### **2.5.1 Профиль пользователя**

Отображает информацию о текущем пользователе, такие как имя и email. Показывает список книг, загруженных пользователем, и список скачанных им книг. Включает форму для изменения данных пользователя (рисунок 6)

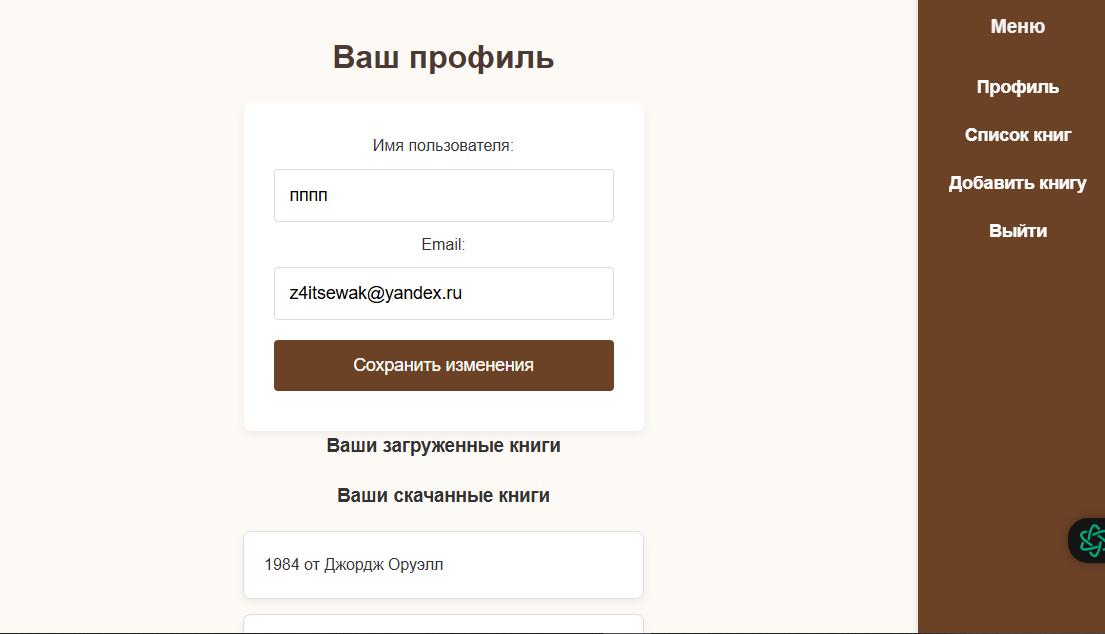


Рисунок 6 – Главная страница для авторизированных пользователей

Листинг страницы «Профиль пользователя»

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <title>Профиль пользователя</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <?php include 'sidebar.php'; ?>

    <div class="container">

        <h2>Ваш профиль</h2>

        <?php if ($success): ?>

            <p style="color:green;"><?php echo $success; ?></p>

        <?php endif; ?>

        <form method="POST" action="profile.php">

            <label>Имя пользователя:</label>

            <input type="text" name="username" value="<?php echo htmlspecialchars($user['username']); ?>" required>

            <label>Email:</label>

            <input type="email" name="email" value="<?php echo htmlspecialchars($user['email']); ?>" required>

            <button type="submit" name="update\_user">Сохранить изменения</button>

        </form>

        <h3>Ваши загруженные книги</h3>

        <ul>

            <?php foreach ($uploaded\_books as $book): ?>

                <li><?php echo htmlspecialchars($book['title']); ?> от <?php echo htmlspecialchars($book['author']); ?></li>

            <?php endforeach; ?>

        </ul>

        <h3>Ваши скачанные книги</h3>

        <ul>

            <?php foreach ($downloaded\_books as $book): ?>

                <li><?php echo htmlspecialchars($book['title']); ?> от <?php echo htmlspecialchars($book['author']); ?></li>

            <?php endforeach; ?>

        </ul>

    </div>

</body>

</html>

**2.5.2 Список книг**

Представляет каталог всех доступных книг с возможностью скачивания. Отображает книги в виде сетки, где каждая книга показана с обложкой, названием, автором и ссылкой для скачивания. Если пользователь скачивает книгу, система записывает это действие в базу данных (рисунок 7).

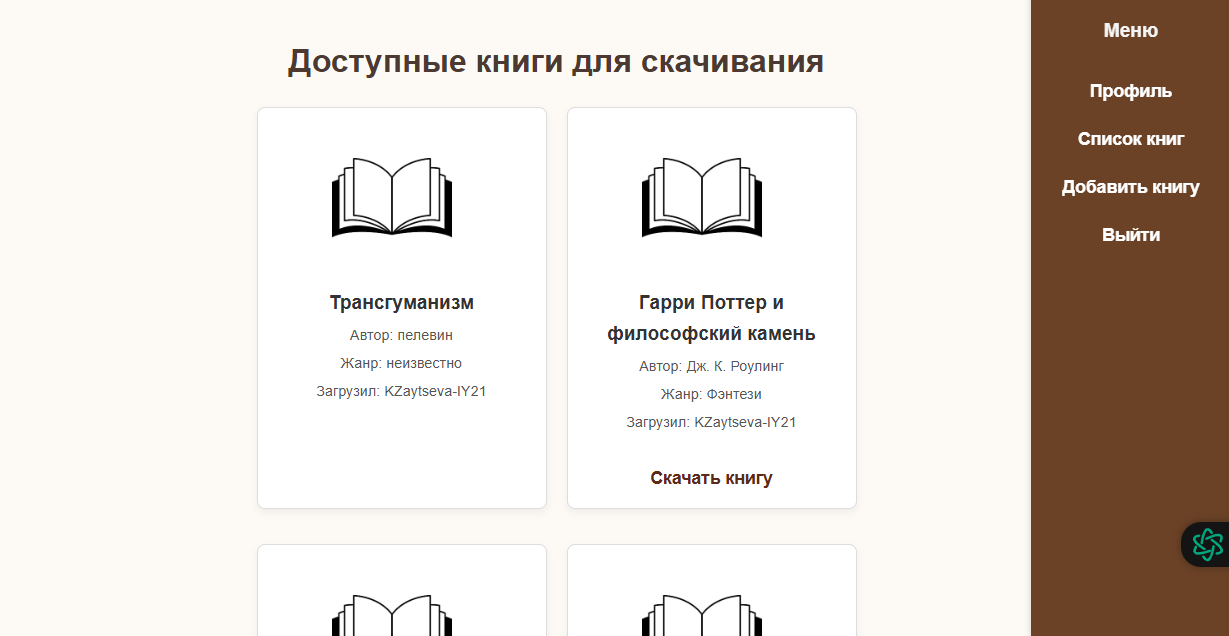


Рисунок 7 – Список книг (book\_list.php)

Листинг страницы «Список книг»:

<?php

session\_start();

include 'db.php';

// Обработка скачивания

if (isset($\_GET['download']) && isset($\_SESSION['user\_id'])) {

    $book\_id = $\_GET['download'];

    $user\_id = $\_SESSION['user\_id'];

    // Добавление записи о скачивании в таблицу downloads

    $stmt = $pdo->prepare("INSERT INTO downloads (user\_id, book\_id) VALUES (?, ?)");

    $stmt->execute([$user\_id, $book\_id]);

    // Перенаправление на файл для скачивания

    $stmt = $pdo->prepare("SELECT file\_path FROM books WHERE id = ?");

    $stmt->execute([$book\_id]);

    $book = $stmt->fetch();

    if ($book) {

        $file\_path = $book['file\_path'];

        header("Location: $file\_path");

        exit();

    }

}

// Получение списка книг

$stmt = $pdo->query("SELECT books.\*, users.username AS owner FROM books JOIN users ON books.owner\_id = users.id");

$books = $stmt->fetchAll();

?>

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <title>Список книг</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <?php include 'sidebar.php'; ?>

    <div class="container">

        <h2>Доступные книги для скачивания</h2>

        <ul class="book-grid">

            <?php foreach ($books as $book): ?>

                <li>

                    <img src="book.jpg" alt="Book Cover" class="book-cover">

                    <div class="book-info">

                        <strong><?php echo htmlspecialchars($book['title']); ?></strong><br>

                        <span>Автор: <?php echo htmlspecialchars($book['author']); ?></span>

                        <span>Жанр: <?php echo htmlspecialchars($book['genre']); ?></span>

                        <span>Загрузил: <?php echo htmlspecialchars($book['owner']); ?></span>

                        <?php if ($book['file\_path']): ?>

                            <br><a href="book\_list.php?download=<?php echo $book['id']; ?>" class="download-link">Скачать книгу</a>

                        <?php endif; ?>

                    </div>

                </li>

            <?php endforeach; ?>

        </ul>

    </div>

</body>

</html>

### **2.5.3 Добавление книги**

Содержит форму для загрузки новой книги с полями для названия, автора, жанра и файла книги. Обрабатывает загружаемый файл и сохраняет информацию о книге в базе данных. После успешной загрузки перенаправляет пользователя на страницу профиля.

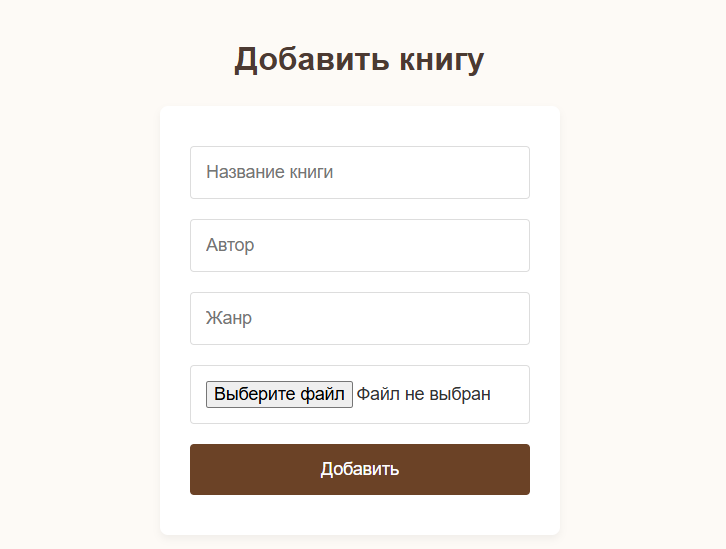


Рисунок 8 – Добавление книги (add\_book.php)

Листинг страницы «Добавление книги»:

<?php

session\_start();

include 'db.php';

if (!isset($\_SESSION['user\_id'])) {

    header("Location: login.php");

    exit();

}

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

    $title = $\_POST['title'];

    $author = $\_POST['author'];

    $genre = $\_POST['genre'];

    $owner\_id = $\_SESSION['user\_id'];

    // Обработка загрузки файла

    if (isset($\_FILES['book\_file']) && $\_FILES['book\_file']['error'] == UPLOAD\_ERR\_OK) {

        $uploadDir = 'uploads/';

        $fileName = basename($\_FILES['book\_file']['name']);

        $filePath = $uploadDir . $fileName;

        // Переместить загруженный файл в каталог 'uploads'

        if (move\_uploaded\_file($\_FILES['book\_file']['tmp\_name'], $filePath)) {

            // Сохранить информацию о книге в базе данных

            $stmt = $pdo->prepare("INSERT INTO books (title, author, genre, owner\_id, file\_path) VALUES (?, ?, ?, ?, ?)");

            $stmt->execute([$title, $author, $genre, $owner\_id, $filePath]);

            header("Location: profile.php");

            exit();

        } else {

            echo "Ошибка при загрузке файла.";

        }

    } else {

        echo "Пожалуйста, выберите файл для загрузки.";

    }

}

?>

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <title>Добавить книгу</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

<?php include 'sidebar.php'; ?>

    <div class="container">

        <h2>Добавить книгу</h2>

        <form action="add\_book.php" method="POST" enctype="multipart/form-data">

            <input type="text" name="title" placeholder="Название книги" required>

            <input type="text" name="author" placeholder="Автор" required>

            <input type="text" name="genre" placeholder="Жанр" required>

            <input type="file" name="book\_file" accept=".pdf,.epub" required>

            <button type="submit">Добавить</button>

        </form>

    </div>

</body>

</html>

## **2.6 Стилизация интерфейса с помощью CSS**

Стилизация интерфейса с помощью CSS. CSS-файл style.css отвечает за оформление пользовательского интерфейса веб-приложения буккроссинга, создавая легкий и привлекательный визуальный стиль, напоминающий библиотеку. Стилизация включает несколько основных элементов и концепций:

Основные стили: фоновые цвета, типографика и отступы были настроены так, чтобы создать удобную и комфортную для глаз палитру с акцентом на уютный библиотечный стиль. Основной цвет фона — светло-бежевый, что способствует теплой атмосфере.

Используется шрифт Nunito из Google Fonts для создания современного, но комфортного для чтения стиля.

Сетка для книг: список книг отображается с использованием CSS-сетки (display: grid), которая адаптируется под размер экрана. На больших экранах книги расположены в три столбца, на средних — в два, а на маленьких экранах — в один. Это улучшает восприятие и удобство пользования сайтом на различных устройствах.

Боковое меню: фиксированное боковое меню на правой стороне экрана позволяет пользователю легко перемещаться по основным разделам сайта. Меню оформлено в спокойных тонах с акцентами, выделяющими ссылки при наведении.

Стили форм и кнопок: формы регистрации, авторизации и добавления книг стилизованы с помощью рамок и мягких теней, чтобы выделить их на странице. Кнопки имеют стандартный стиль с акцентом на цвет и изменяются при наведении, улучшая интерактивность интерфейса.

Этот CSS-файл позволяет достичь интуитивности интерфейса, обеспечивая пользователю приятное взаимодействие с приложением.

Листинг СSS файла:

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body {

    font-family: 'Nunito', sans-serif;

    background-color: #fdfaf6;

    color: #333;

    line-height: 1.6;

    display: flex;

    justify-content: center;

    align-items: center;

    min-height: 100vh;

    flex-direction: column;

}

h1, h2 {

    text-align: center;

    color: #4c3a32;

    margin-top: 20px;

    font-size: 2.5rem;

}

h2 {

    font-size: 2rem;

    margin-bottom: 20px;

}

.container {

    max-width: 800px;

    margin: auto;

    padding: 20px;

    text-align: center;

}

ul {

    list-style-type: none;

    padding: 0;

    margin: 20px auto;

    max-width: 600px;

    text-align: center;

}

li {

    background-color: #fff;

    border: 1px solid #ddd;

    margin-bottom: 15px;

    padding: 20px;

    border-radius: 8px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.05);

    display: flex;

    align-items: center;

}

.book-cover {

    width: 60px;

    height: 90px;

    object-fit: cover;

    border-radius: 4px;

    margin-right: 20px;

}

.book-info {

    flex-grow: 1;

}

a {

    color: #5e2a18;

    text-decoration: none;

    font-weight: 700;

    font-size: 1.1rem;

}

a:hover {

    text-decoration: underline;

}

form {

    max-width: 400px;

    margin: auto;

    background: #fff;

    padding: 30px;

    border-radius: 8px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.05);

    text-align: center;

}

input[type="text"],

input[type="email"],

input[type="password"],

input[type="file"],

button {

    width: 100%;

    padding: 15px;

    margin: 10px 0;

    border: 1px solid #ddd;

    border-radius: 4px;

    font-size: 1.1rem;

}

button {

    background-color: #6b4226;

    color: #fff;

    border: none;

    cursor: pointer;

    font-size: 1.1rem;

}

button:hover {

    background-color: #5e2a18;

}

/\* Footer \*/

.footer {

    background-color: #4c3a32;

    color: #fff;

    text-align: center;

    padding: 10px;

    margin-top: 20px;

    font-size: 1rem;

}

/\* Боковое меню \*/

.sidebar {

    position: fixed;

    right: 0;

    top: 0;

    width: 200px;

    height: 100%;

    background-color: #6b4226;

    color: #fff;

    padding: 20px;

    text-align: center;

    box-shadow: -2px 0 5px rgba(0, 0, 0, 0.1);

    display: flex;

    flex-direction: column;

    align-items: center;

}

.sidebar h3 {

    font-size: 1.2rem;

    margin-bottom: 20px;

    color: #f4f4f4;

}

.sidebar a {

    color: #fff;

    text-decoration: none;

    margin: 10px 0;

    font-size: 1.1rem;

    display: block;

    transition: color 0.3s;

}

.sidebar a:hover {

    color: #ddd;

}

container {

    max-width: 1200px;

    margin: auto;

    padding: 20px;

    text-align: center;

}

/\* Сетка для книг \*/

.book-grid {

    display: grid;

    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));

    gap: 20px;

    padding: 0;

    list-style-type: none;

    margin: 20px auto;

}

.book-grid li {

    background-color: #fff;

    border: 1px solid #ddd;

    padding: 15px;

    border-radius: 8px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.05);

    display: flex;

    flex-direction: column;

    align-items: center;

}

.book-cover {

    width: 100%;

    max-width: 150px;

    height: auto;

    object-fit: cover;

    border-radius: 4px;

    margin-bottom: 15px;

}

.book-info {

    text-align: center;

}

.book-info strong {

    font-size: 1.2rem;

}

.book-info span {

    display: block;

    margin: 5px 0;

    font-size: 0.9rem;

    color: #555;

}

# ЗАКЛЮЧЕНИЕ

В результате работы над проектом была создана веб-платформа для интернет-буккроссинга, которая предоставляет пользователям все необходимые функции для обмена книгами. Приложение включает регистрацию и авторизацию, возможность добавления и скачивания книг, а также управление личными данными в профиле. Интуитивно понятный интерфейс, организованный в виде удобной сетки для просмотра книг, и встроенное боковое меню делают приложение простым в использовании и доступным для пользователей с разным уровнем цифровых навыков. Подход, использованный при проектировании, ориентирован на максимальное удобство, безопасность и доступность.

Работа над проектом позволила изучить и применить на практике современные веб-технологии и принципы разработки. Было уделено внимание безопасности данных пользователей, создана база данных, которая сохраняет всю необходимую информацию о книгах и пользователях, а также реализованы механизмы, обеспечивающие защиту данных, включая хэширование паролей. Проект позволяет пользователям не только обмениваться литературой, но и знакомиться с предпочтениями других участников, что способствует объединению единомышленников на основе общих интересов.

В заключение, данный проект не только реализует функции интернет-буккроссинга, но и способствует популяризации культуры чтения и обмена знаниями. В перспективе приложение может быть расширено: возможны улучшения интерфейса, добавление системы комментариев и оценок, а также внедрение дополнительных инструментов для общения между пользователями.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. PHP: сайт. – URL: https://www.php.net/manual/ru/intro-whatis.php (дата обращения: 25.10.2024)
2. Белунцов В. Новейший самоучитель по созданию Web-страниц / В. Белунцов. - М.: NT Press, 2012. - 185 с. (дата обращения: 10.10.2024)
3. Браун Брэдли Oracle Database. Создание Web-приложений: учебник / Брэдли Браун, Ричард Ниемик, Джозеф С. Треззо. - СПб.: Лори, 2011. - 722 с. (дата обращения: 11.10.2024)
4. Дуванов А. Web-конструирование. HTML / А. Дуванов. - СПб.: БХВ-Петербург, 2013. - 321 с. (дата обращения: 23.10.2024)
5. Кожемякин А. HTML и CSS в примерах. Создание Web-страниц: учебник / А. Кожемякин. - М.: Альтекс-А, 2011. - 415 с. (дата обращения: 11.10.2024)
6. Работа с MySQL в PHP: сайт. – URL: https://htmlacademy.ru/blog/php/mysql?ysclid=lrnhk4qz9b568571317 (дата обращения: 20.10.2024)
7. Руководства по использованию HTML, CSS, JavaScript: сайт. – URL: https://developer.mozilla.org/ (дата обращения: 05.10.2024)
8. Соединение с базой данных: сайт. – URL: https://lite.host/faq/hosting/soedinenie-bazoy-dannih?ysclid=lrdlj2xa3t541902095 (дата обращения: 12.10.2024)