

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

КУРСОВОЙ ПРОЕКТ
По курсу Проектирование веб-сервисов

ТЕМА
«Разработка веб-приложения для управления электронной библиотекой»

Выполнил Минчаков Аркадий Сергеевич
Группа 221-329
Проверил Кружалов Алексей Сергеевич

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

УТВЕРЖДАЮ
заведующая кафедрой
«Инфокогнитивные технологии»
_____ / Е. А. Пухова /
«__» _____ 2024 г.

ЗАДАНИЕ
на выполнение курсовой работы (проекта)
Минчакову Аркадию Сергеевичу,

обучающемуся группы 221–329,
направления подготовки 09.03.01 «Информатика и вычислительная техника»
по дисциплине «Разработка веб-приложений»
на тему «Разработка веб-приложения для управления электронной библиотекой»
1. Исходные данные к работе (проекту): информационные ресурсы в сети интернет, научные публикации в открытой печати.
2. Содержание задания по курсовой работе (проекту) – перечень вопросов, подлежащих разработке:

Разрабатываемый вопрос	Объем от всего задания, %	Срок выполнения	Примечание
Раздел 1. Анализ предметной области	10	07.10.2024	
Задача 1.1. Обзор существующих программных продуктов по теме работы	3	03.10.2024	
Задача 1.2. Анализ программных инструментов разработки веб-приложений	4	05.10.2024	
Задача 1.3. Формулировка цели и задач работы	3	07.10.2024	
Раздел 2. Проектирование веб-приложения	25	14.10.2024	
Задача 2.1. Анализ целевой аудитории	3	08.10.2024	
Задача 2.2. Описание функциональности приложения (диаграмма вариантов использования, user story и т. д.)	10	10.10.2024	
Задача 2.3. Проектирование модели данных (ER-диаграмма, логическая и физическая схемы БД)	4	12.10.2024	
Задача 2.4. Разработка макетов страниц (Wireframe)	8	14.10.2024	
Раздел 3. Разработка веб-приложения	50	25.10.2024	
Задача 3.1. Разработка базовой структуры приложения и верстка шаблонов страниц	10	16.10.2024	
Задача 3.2. Реализация аутентификации пользователей	10	19.10.2024	
Задача 3.3. Реализация разграничения прав пользователей	10	22.10.2024	
Задача 3.4. Реализация CRUD-интерфейса для книг и рецензий	10	24.10.2024	
Задача 3.5. Реализация модерации рецензий на книги	10	25.10.2024	
Раздел 4. Оформление итогов работы	15	28.10.2024	
Задача 4.1. Создание Git-репозитория с кодом проекта	5	15.10.2024	
Задача 4.2. Оформление отчёта о проделанной работе	10	28.10.2024	

Руководитель курсовой работы (проекта): преподаватель кафедры «Инфокогнитивные технологии»

«__» _____ 2024 г. _____

А. С. Кружалов

Дата выдачи задания

«14» марта 2024 г.

Дата сдачи выполненной работы (проекта)

«28» октября 2024 г.

Задание принял к исполнению

«1» октября 2024 г.

А. С. Минчаков



СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	3
ВВЕДЕНИЕ	4
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	5
1.1. Обзор существующих программных продуктов по теме работы.....	5
1.2. Анализ программных инструментов разработки.....	7
1.3. Формулировка цели и задач работы	8
2. ПРОЕКТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ	9
2.1. Анализ целевой аудитории.....	9
2.2. Описание функциональности приложения	9
2.3. Проектирование модели данных	10
2.4. Разработка макетов страниц	11
3. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ	17
3.1. Разработка базовой структуры приложения и вёрстка страниц	17
3.2. Реализация аутентификации пользователей.....	17
3.3. Реализация разграничения прав пользователей	18
3.4. Реализация CRUD-интерфейса для книг и рецензий	19
3.5. Реализация модерации рецензий на книги	22
4. ОФОРМЛЕНИЕ ИТОГОВ РАБОТЫ	24
4.1. Создание Git-репозитория с кодом проекта	24
4.2. Оформление отчёта о проделанной работе	24
ЗАКЛЮЧЕНИЕ.....	25
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	26

ВВЕДЕНИЕ

В современном мире цифровые технологии играют ключевую роль в улучшении и оптимизации различных сфер жизни, включая образование и доступ к знаниям. Библиотеки, как хранилища культурного и научного наследия, не остаются в стороне от этого процесса. Переход от традиционных бумажных форматов к электронным изданиям и ресурсам требует разработки эффективных и удобных инструментов для управления библиотечными фондами.

Электронные библиотеки предлагают уникальные возможности для пользователей, обеспечивая мгновенный доступ к широкому спектру литературы и ресурсов из любой точки мира. Однако, чтобы максимально раскрыть потенциал таких библиотек, необходимо создать функциональные и интуитивно понятные веб-приложения, способные удовлетворить потребности как обычных читателей, так и библиотекарей.

Основной целью данного курсового проекта является разработка веб-приложения для управления электронной библиотекой, которое будет включать в себя современные механизмы аутентификации пользователей, взаимодействия с библиотечными материалами, а также управления контентом и отзывами. Такое приложение должно стать мощным инструментом для систематизации и упрощения процесса работы с электронной библиотекой, способствуя более эффективному распространению знаний и повышению уровня образования.

Данный проект актуален не только с точки зрения технического выполнения, но и с позиции социальных и образовательных аспектов. Разработка подобного веб-приложения поможет улучшить доступ к образовательным ресурсам, сделав их более доступными для широкого круга пользователей. Это особенно важно в условиях роста потребности в дистанционном обучении и самообразовании.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Обзор существующих программных продуктов по теме работы

На российском рынке представлено несколько программных продуктов для управления электронными библиотеками, каждый из которых обладает уникальными особенностями и функциональными возможностями.

1.1.1. Электронная библиотека "Яндекс Книги"

Описание: "Яндекс Книги" — это сервис от компании Яндекс, предоставляющий доступ к обширному каталогу электронных книг и аудиокниг. Платформа ориентирована на широкий круг пользователей, включая студентов, любителей литературы и профессионалов.

Функциональность:

- Доступ к большому ассортименту книг, включая художественную, научную и учебную литературу.
- Возможность покупки и аренды книг, а также бесплатного чтения.
- Интеграция с другими сервисами Яндекса, такими как "Яндекс.Музыка" и "Яндекс.Диск".
- Персонализированные рекомендации на основе интересов пользователя и истории чтения.

Преимущества:

- Удобный и интуитивно понятный интерфейс для навигации и поиска книг.
- Регулярное обновление каталога с новинками и популярными произведениями.
- Возможность чтения на разных устройствах через приложение "Яндекс Книги" и веб-интерфейс.

1.1.2. Электронная библиотека "ЛитРес"

Описание: "ЛитРес" — одна из крупнейших электронных библиотек в России, предлагающая широкий ассортимент художественной и научно-популярной литературы. Платформа позволяет пользователям читать книги в различных форматах и предоставляет доступ к аудиокнигам.

Функциональность:

- Доступ к огромной библиотеке книг и аудиокниг.
- Возможность покупки, аренды и бесплатного чтения.
- Персонализированный кабинет пользователя для отслеживания прочитанных и сохраненных книг.
- Поддержка синхронизации между устройствами (смартфоны, планшеты, компьютеры).

Преимущества:

- Широкий выбор литературы различных жанров и форматов.
- Регулярное обновление каталога новыми изданиями.
- Пользовательский интерфейс, удобный для навигации и поиска.

1.1.3. Электронная библиотека "MyBook"

Описание: "MyBook" — популярная платформа, предоставляющая доступ к электронной библиотеке с большим количеством книг, включая новинки и бестселлеры. Она ориентирована как на любителей литературы, так и на студентов, предлагая разнообразный контент.

Функциональность:

- Подписка на доступ к большому количеству книг и аудиокниг.
- Возможность чтения онлайн или оффлайн через мобильное приложение.
- Персонализированные рекомендации на основе предпочтений пользователя.
- Инструменты для создания закладок и аннотирования текстов.

Преимущества:

- Удобный интерфейс с возможностью быстрого поиска и навигации по библиотеке.

- Регулярное добавление новых книг и жанров.
- Доступность для использования на различных устройствах.

1.2. Анализ программных инструментов разработки

Разработка веб-приложений на Python пользуется популярностью благодаря своей простоте, читаемости кода и большому количеству библиотек и фреймворков, которые облегчают процесс создания. В данном разделе рассматриваются основные программные инструменты и технологии, используемые при разработке веб-приложений на Python.

1.2.1. Фреймворки

Django: Один из самых популярных веб-фреймворков на Python, Django предлагает мощные инструменты для быстрой разработки. Он включает в себя ORM, админ-панель, систему аутентификации и множество других функций, что позволяет разработчикам сосредоточиться на логике приложения, а не на рутинных задачах.

Flask: Легковесный фреймворк, подходящий для создания простых и быстрых приложений. Flask предоставляет гибкость и возможность расширять функциональность через сторонние библиотеки, что делает его идеальным для микросервисов и небольших проектов.

FastAPI: Современный фреймворк для создания API с высокой производительностью. FastAPI поддерживает асинхронное программирование и автоматическую генерацию документации для API, что упрощает разработку и тестирование.

Pyramid: Гибкий фреймворк, который подходит как для небольших, так и для крупных приложений. Pyramid предоставляет возможность выбора компонентов, что позволяет разработчикам настраивать приложение под свои нужды.

1.2.2. ORM (Object-Relational Mapping)

SQLAlchemy: Библиотека для работы с базами данных, которая предоставляет инструменты для создания и управления схемами данных.

SQLAlchemy поддерживает множество СУБД и позволяет писать код, максимально приближенный к чистому SQL.

Django ORM: Встроенная ORM в Django, обеспечивающая простоту взаимодействия с базами данных. Django ORM позволяет разработчикам использовать объекты Python вместо SQL-запросов, что упрощает работу с данными.

1.2.3. Системы управления версиями

Git: Стандартная система управления версиями, позволяющая отслеживать изменения в коде, совместную работу над проектами и упрощение развертывания приложений. GitHub и GitLab являются популярными платформами для размещения и управления репозиториями.

1.2.4. Инструменты для работы с фронтом

Jinja2: Шаблонизатор, часто используемый в сочетании с Flask и другими фреймворками. Jinja2 позволяет динамически генерировать HTML-страницы, используя шаблоны и данные из приложения.

Bootstrap: CSS-фреймворк, который можно интегрировать с Python-приложениями для создания адаптивных и стильных пользовательских интерфейсов. Bootstrap упрощает процесс разработки фронта и ускоряет создание интерфейсов.

1.3. Формулировка цели и задач работы

Целью данной курсовой работы будет разработка веб-приложения для управления электронной библиотекой. Для ее выполнения можно сформулировать следующие задачи:

1. Анализ целевой аудитории.
2. Проектирование интерфейса.
3. Разработка функциональности.
4. Тестирование.
5. Оформление итогов.

2. ПРОЕКТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ

2.1. Анализ целевой аудитории

Для анализа целевой аудитории можно выделить несколько ключевых групп пользователей, но мы сконцентрируемся на трех из них. Каждая из них имеет свои потребности и интересы, которые приложение должно учитывать.

2.1.1. Любители чтения

Цели: Поиск новых книг и получение рекомендаций на основе рецензий других читателей.

Потребности: Удобный интерфейс для поиска книг, доступ к комментариям и обсуждениям других пользователей, возможность смотреть средний балл книги по рецензиям.

2.1.2. Книжные критики

Цели: Обмен мнениями о прочитанных книгах, создание и публикация рецензий, взаимодействие с аудиторией.

Потребности: Платформа для написания детальных рецензий и анализа книг.

2.1.2. Модераторы и администраторы

Цели: Обеспечение порядка и качества контента в приложении, контроль за рецензиями и взаимодействием пользователей.

Потребности: Эффективные инструменты для модерации рецензий, возможность редактировать и удалять неподобающий контент.

2.2. Описание функциональности приложения

Учитывая потребности перечисленных групп пользователей, можно составить список функций, которые будут выполняться приложением.

2.2.1. Аутентификация

Пользователи, не имеющие учетную запись, смогут просматривать рецензии и книги, имеющиеся на сайте. Аутентифицированные же пользователи получат вдобавок возможность оставлять собственные рецензии.

2.2.2. Роли

В системе будет три роли:

- Пользователь. Он сможет оставлять рецензии и просматривать чужие.
- Модератор. То же, что и пользователь, но еще он сможет редактировать описание книг и модерировать рецензии, оставленные пользователями.
- Администратор. То же, что и пользователь, но еще он сможет добавлять и удалять книги, а также редактировать их описание.

2.2.3. Манипуляция книгами

Будет возможность добавлять, удалять, редактировать и просматривать книги. На странице просмотра аутентифицированные пользователи смогут оставить рецензию.

2.2.4. Возможность оставления рецензий

Каждый аутентифицированный пользователь сможет поделиться своим мнением о той или иной книге. Каждая рецензия будет влиять на среднюю оценку книги. Будет возможность выбрать оценку от нуля до пяти и обосновать свое мнение в комментарии. Опубликованная рецензия сначала должна будет пройти модерацию.

2.2.5. Модерация рецензий

У каждой рецензии будет один из трех статусов: на рассмотрении, одобрено или отклонено.

Новые рецензии будут размещаться на отдельной странице, доступной только модераторам, со статусом “на рассмотрении”. Каждую из них модератор сможет проверить и одобрить или отклонить.

2.3. Проектирование модели данных

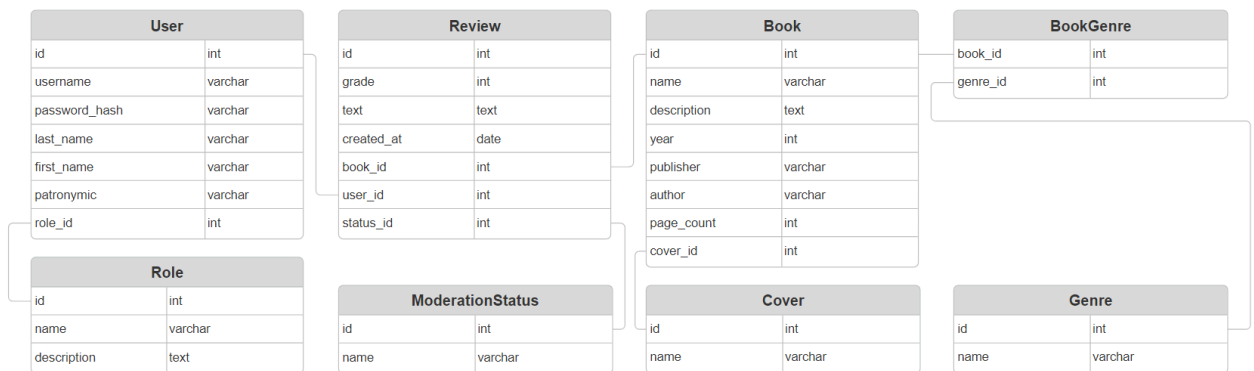


Рисунок 1 – ER-диаграмма и физическое определение полей

2.4. Разработка макетов страниц

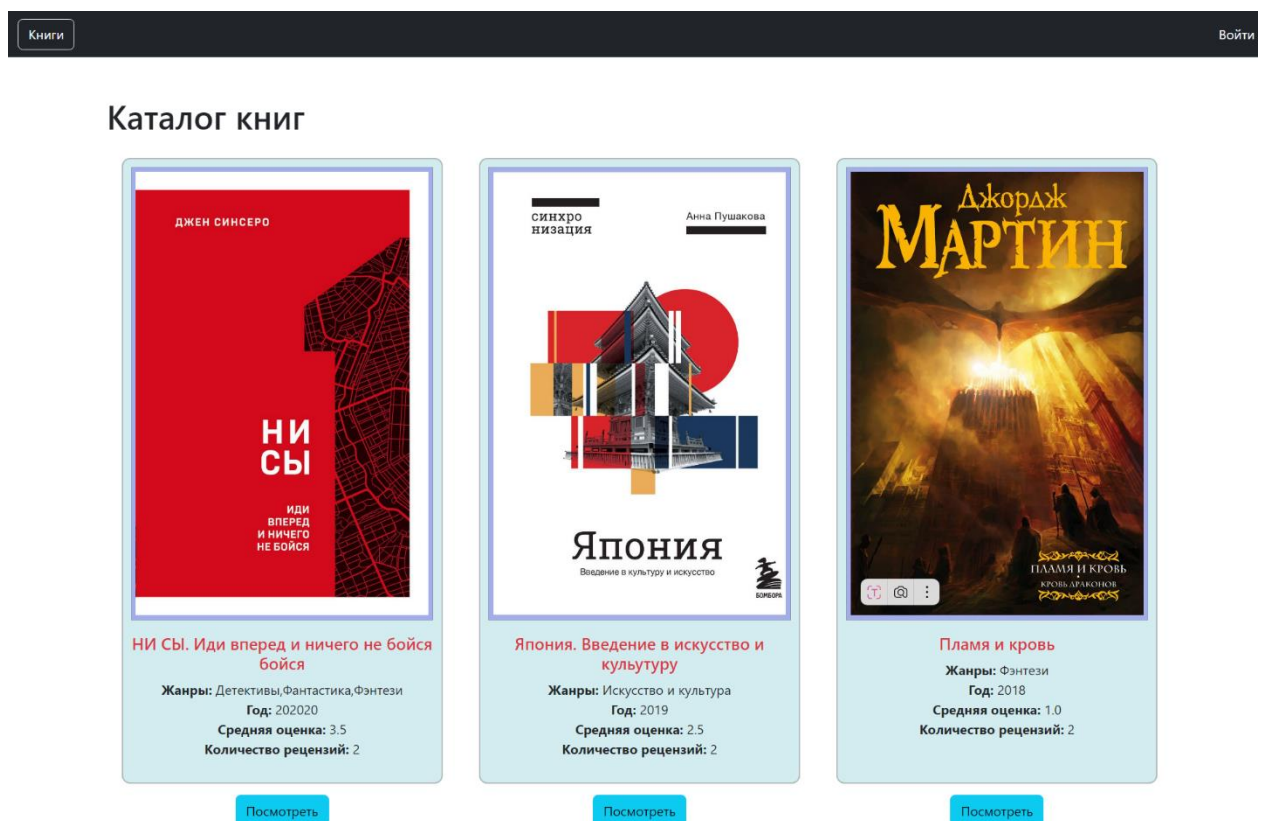


Рисунок 2 – макет главной страницы

Книги

Войти

Авторизация

Имя пользователя

Person

Пароль

☐ Запомнить меня

Войти

Минчаков Аркадий Сергеевич, группа 221-329

Рисунок 3 – макет страницы авторизации

Книги

Мои рецензии

Выйти

ДЖЕН СИНСЕРО

НИ СЫ

ИДИ ВПЕРЕД И НИЧЕГО НЕ БОЙСЯ

НИ СЫ. Иди вперед и ничего не бойся бойся

Автор: Джен Синсеро

Год издания: 202020

Издатель: Серебрянный Лист Лист

Количество страниц: 304000

Жанры: Фантастика, Фэнтези, Детективы

Краткое описание

Полка

Рецензии

MegaVanya228 - Оценка: отлично

Political one office day.

Person - Оценка: неудовлетворительно

Require southern boy store better.

Ваша рецензия

Оценка: неудовлетворительно

Статус: Одобрено

Require southern boy store better.

Минчаков Аркадий Сергеевич, группа 221-329

Рисунок 4 – макет страницы просмотра книги

Мои рецензии

Оценка: ужасно
Статус: Одобрено
Книга: Первому игроку приготовиться
Establish measure key.

Оценка: ужасно
Статус: Одобрено
Книга: Пламя и кровь
Reveal agreement play price example store.

Оценка: ужасно
Статус: Одобрено
Книга: Этюд в багровых тонах
President wrong discussion again end so.

Оценка: плохо
Статус: Одобрено
Книга: Собака Баскервилей
Contain election beat truth beautiful.

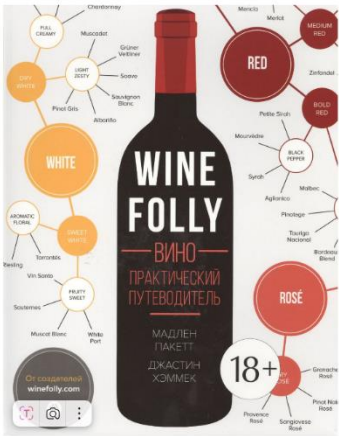
Оценка: удовлетворительно
Статус: На рассмотрении
Книга: Чингисхан
Exist person half national six.

Рисунок 5 – макет страницы со всеми рецензиями пользователя

Книги

Мои рецензии

Выйти



Вино. Практический путеводитель

Автор: Джастин Хэммек

Год издания: 2015

Издатель: Солнечная стрела

Количество страниц: 234

Жанры: Кулинарные книги

Краткое описание

Идеальный бумажный путеводитель по вину для эпохи цифровых технологий. Ясный, живой, без излишней заумности.

Написать рецензию

Минчаков Аркадий Сергеевич, группа 221-329

Рисунок 6 – макет страницы просмотра книги с кнопкой отправки рецензии

Книги
Мои рецензии
Выйти

Написать рецензию

Оценка

отлично

Текст рецензии

B I H
“ ”
≡ ≡
%

lines: 1 words: 0 1:1

Сохранить

Минчаков Аркадий Сергеевич, группа 221-329

Рисунок 7 – макет страницы написания рецензии

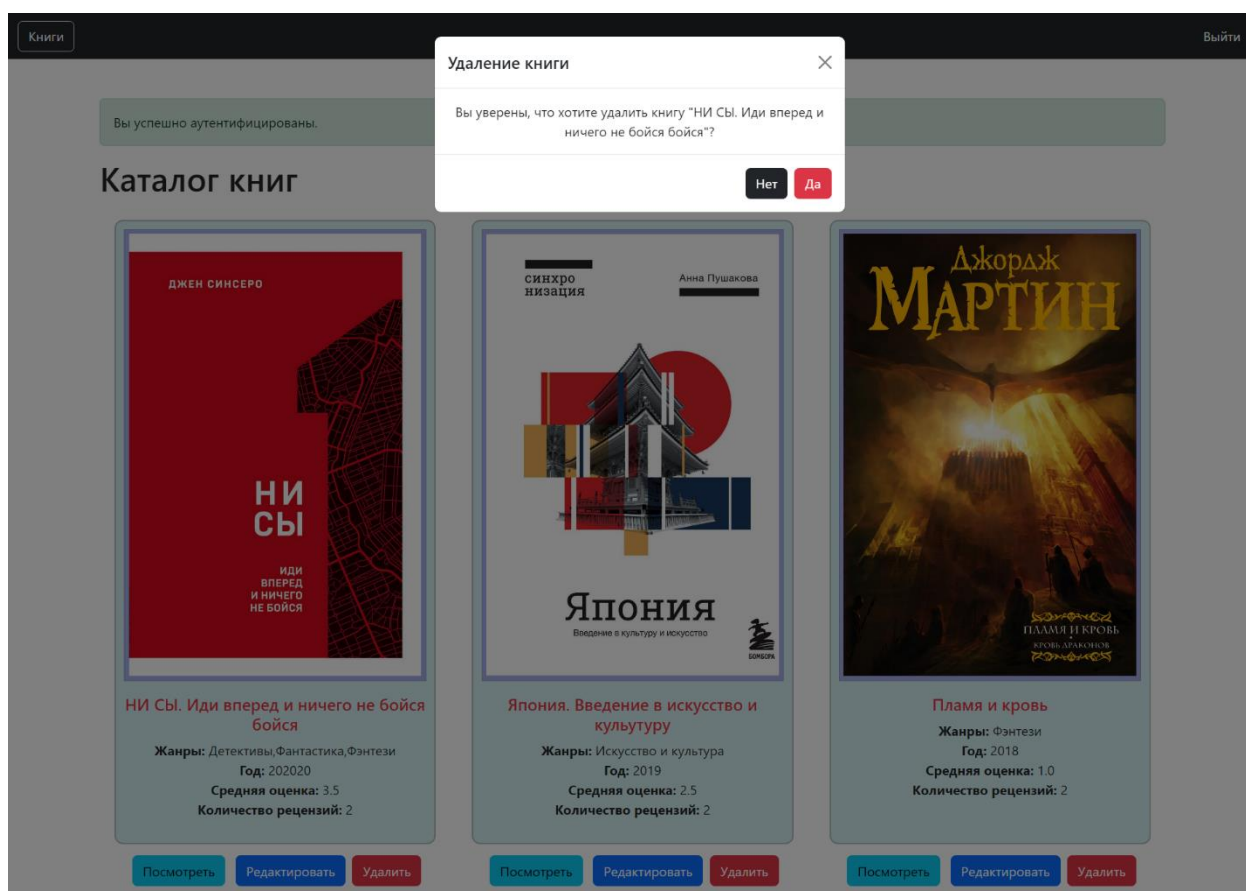


Рисунок 8 – макет главной страницы от имени администратора, удаление
КНИГИ

Книги

Выйти

Добавление книги

Название книги

Автор

Год

Издатель

Количество страниц

Жанры

Фантастика

Фэнтези

Детективы

Романы

Краткое описание

В I Н “ ≡ ≡ 🔗 🖼️ 👁 📄 ✕ ⓘ

lines: 1 words: 0 1:1

Обложка

Выберите файл

Файл не выбран

Сохранить

Минчаков Аркадий Сергеевич, группа 221-329

Рисунок 9 – макет страницы добавления книги

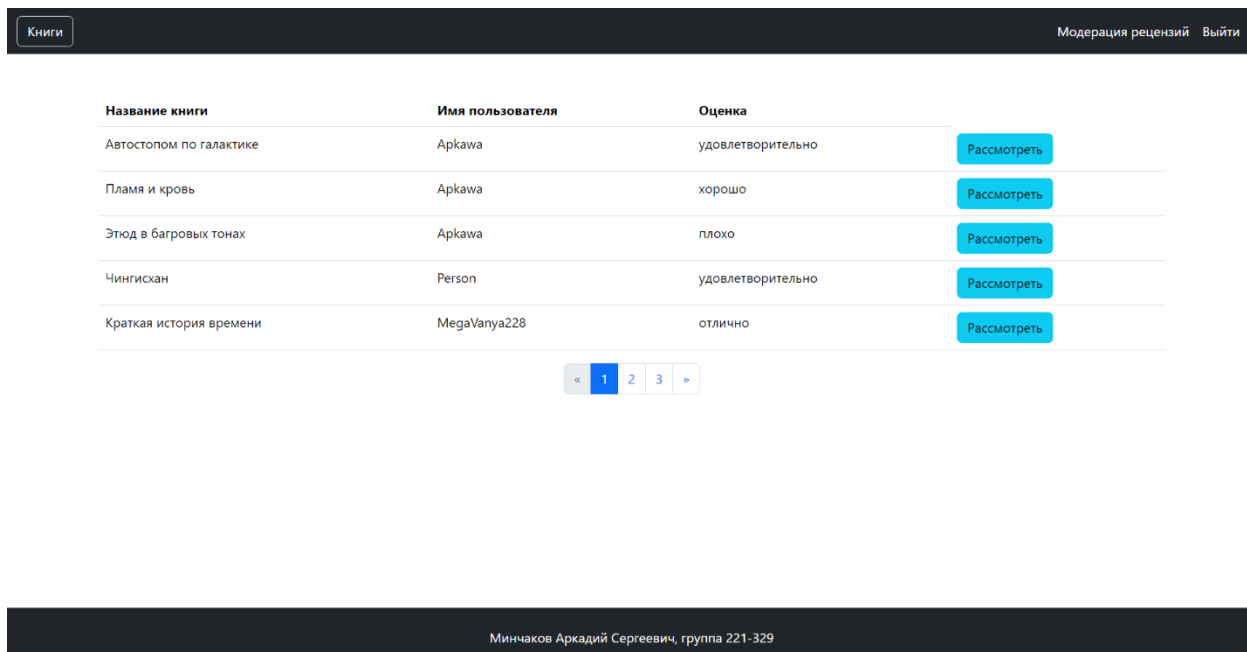


Рисунок 10 – макет страницы модерации рецензий

3. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ

3.1. Разработка базовой структуры приложения и вёрстка страниц

Сначала был реализован базовых шаблон, на котором прописан футер и хедер каждой из страниц. В футере указано мое имя и группа, а в хедере находится навигация по страницам.

Листинг 1 – хедер в базовом шаблоне

```
20 <header class="dark-background text-white">
21   <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
22     <div class="container-fluid">
23       <a class="btn btn-outline-light me-2" href="{{ url_for('index') }}">Книги</a>
24       <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
25         data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
26         aria-expanded="false" aria-label="Toggle navigation">
27         <span class="navbar-toggler-icon"></span>
28       </button>
29       <div class="collapse navbar-collapse" id="navbarSupportedContent">
30         <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
31           {% if current_user.is_authenticated %}
32             {% if current_user.role.name == 'user' %}
33               <li class="nav-item">
34                 <a class="nav-link active" aria-current="page"
35                   href="{{ url_for('reviews.my_reviews') }}">Мои рецензии</a>
36               </li>
37             {% elif current_user.role.name == 'moderator' %}
38               <li class="nav-item">
39                 <a class="nav-link active" aria-current="page"
40                   href="{{ url_for('reviews.moderate') }}">Модерация рецензий</a>
41               </li>
42             {% endif %}
43             <li class="nav-item">
44               <a class="nav-link active" aria-current="page"
45                 href="{{ url_for('auth.logout') }}">Выйти</a>
46             </li>
47           {% else %}
48             <li class="nav-item">
49               <a class="nav-link active" aria-current="page" href="{{ url_for('auth.login') }}">
50                 Войти </a>
51             </li>
52           {% endif %}
53         </ul>
54       </div>
55     </div>
56   </nav>
57 </header>
```

Листинг 2 – футер в базовом шаблоне

```
71 <footer class="py-4 text-center bg-dark text-white">
72   <p>Минчаков Аркадий Сергеевич, группа 221-329</p>
73 </footer>
```

3.2. Реализация аутентификации пользователей

При нажатии кнопки «Войти» на главной странице пользователь будет перенаправлен на страницу входа по адресу `/login`, где должен будет отправить форму с данными для входа, все это обрабатывается в данном куске кода:

Листинг 3 – обработка страницы входа и отправки формы

```

26 @bp.route(rule: '/login', methods=['GET', 'POST']) 1 usage  Arkadii Minchakov
27 <> def login():
28     if request.method == 'POST':
29         username = request.form.get('username')
30         password = request.form.get('password')
31         if login and password:
32             user = db.session.execute(db.select(User).filter_by(username=username)).scalar_one_or_none()
33             if user and user.check_password(password):
34                 login_user(user)
35                 flash(message: 'Вы успешно аутентифицированы.', category: 'success')
36                 next = request.args.get('next')
37                 return redirect(next or url_for('index'))
38             else:
39                 flash(message: 'Невозможно аутентифицироваться с указанными логином и паролем', category: 'danger')
40         else:
41             flash(message: 'Заполните все поля', category: 'danger')
42     return render_template('login.html')

```

После входа у пользователя в верхнем правом углу будет доступна кнопка выхода, которая обрабатывается в приложении так:

Листинг 4 – обработка выхода

```

46 @bp.route('/logout')  Arkadii Minchakov
47 @login_required
48 def logout():
49     logout_user()
50     return redirect(url_for('index'))
51

```

3.3. Реализация разграничения прав пользователей

В приложении присутствуют три роли. Я создал декоратор, который проверяет роль текущего залогиненного пользователя. Он используется на всех конечных точках, где требуются особые права. Например, при добавлении или редактировании книги, на странице просмотра рецензий для последующей модерации и так далее.

Листинг 5 – декоратор для разграничения прав доступа.

```

30 def check_role(roles): 8 usages  Arkadii Minchakov
31     def decorator(func):  Arkadii Minchakov
32         @wraps(func)  Arkadii Minchakov
33         def decorated(*args, **kwargs):
34             if current_user.role.name in roles:
35                 return func(*args, **kwargs)
36             flash(message: 'У вас недостаточно прав для выполнения данного действия', category: 'danger')
37             return redirect(url_for('index'))
38         return decorated
39     return decorator
40
41

```

Листинг 6 – использование декоратора при просмотре странице с модерацией рецензии

```

42 @bp.route('/moderate/<int:review_id>')  ⚡ Arkadii Minchakov
43 @login_required
44 @check_role(['moderator'])
45 <> def moderate_review(review_id):
46     review = db.session.query(Review).get_or_404(review_id)
47     review.text = markdown.markdown(review.text)
48     return render_template('reviews/moderate_review.html', review=review, grade_mapping=GRADE_MAPPING)
49

```

3.4. Реализация CRUD-интерфейса для книг и рецензий

3.4.1. CRUD для книг

Книгу можно добавить с правами администратора по кнопке в нижней части главной страницы.

Листинг 7 – обработка добавления книги

```

79 @bp.route(rule: '/new_book', methods=['GET', 'POST'])  ⚡ Arkadii Minchakov
80 @login_required
81 @check_role(['administrator'])
82 <> def new_book():
83     if request.method == 'POST':
84         try:
85             name = request.form['name']
86             year = request.form['year']
87             description = request.form['description']
88             publisher = request.form['publisher']
89             author = request.form['author']
90             page_count = request.form['page_count']
91             selected_genre_ids = request.form.getlist('genres')
92
93             cover_file = request.files['cover']
94             if cover_file and cover_file.filename != '':
95                 cover_id, cover_data, mime_type = save_cover_file(cover_file)
96             else:
97                 cover_id = None
98
99             description_html = bleach.clean(description)
100
101             new_book = Book(
102                 name=name,
103                 year=year,
104                 description=description_html,
105                 publisher=publisher,
106                 author=author,
107                 page_count=page_count,
108                 cover_id=cover_id
109             )
110             db.session.add(new_book)
111             db.session.commit()

```

```

112
113     for genre_id in selected_genre_ids:
114         connect = BookGenre(book_id=new_book.id, genre_id=int(genre_id))
115         db.session.add(connect)
116
117     db.session.commit()
118
119     if cover_data:
120         cover_path = os.path.join('static/images', f'{cover_id}.{mime_type}')
121         cover_file.save(cover_path)
122
123     flash(message: 'Книга успешно добавлена!', category: 'success')
124     return redirect(url_for(endpoint: 'books.view_book', book_id=new_book.id))
125 except SQLAlchemyError as e:
126     db.session.rollback()
127     flash(message: 'При сохранении данных возникла ошибка. Проверьте корректность введенных данных.', category: 'danger')
128 except ValueError as e:
129     flash(str(e), category: 'danger')
130
131 genres = db.session.query(Genre).all()
132 return render_template(template_name_or_list: 'books/new_book.html', genres=genres)

```

Прочитать данные о книге можно, перейдя на главной странице по кнопке «Посмотреть»

Листинг 8 – обработка просмотра данных книги и рецензий на нее

```

171 @bp.route('/view_book/<int:book_id>')  Arkadii Minchakov
172 <> def view_book(book_id):
173     book = db.session.query(Book).get_or_404(book_id)
174     book.description = markdown(book.description)
175
176     reviews = db.session.query(Review).filter_by(book_id=book_id, status_id=ACCEPTED_STATUS_ID).all()
177     for review in reviews:
178         review.text = markdown(review.text)
179
180     user_review = db.session.query(Review).filter_by(book_id=book_id, user_id=current_user.get_id()).first()
181     if user_review:
182         user_review.text = markdown(user_review.text)
183
184     genre_ids = db.session.query(BookGenre.genre_id).filter_by(book_id=book_id).all()
185     genres = db.session.query(Genre.name).filter(Genre.id.in_([id[0] for id in genre_ids])).all()
186     genres = ", ".join([genre.name for genre in genres])
187
188     cover_data = db.session.query(Cover.filename, Cover.mime_type).filter_by(id=book.cover_id).first()
189
190     return render_template(template_name_or_list: 'books/view_book.html', book=book, reviews=reviews, genres=genres, cover_data=cover_data,
191                           user_review=user_review, grade_mapping=GRADE_MAPPING)

```

Обновить данные книги можно с правами администратора и модератора по кнопке на главной странице.

Листинг 9 – обработка обновления данных книги

```

136 @bp.route(rule: '/edit_book/<int:book_id>', methods=['GET', 'POST'])  ⚡ Arkadii Minchakov
137 @login_required
138 @check_role(['administrator', 'moderator'])
139 <> def edit_book(book_id):
140     book = db.session.query(Book).get_or_404(book_id)
141
142     if request.method == 'POST':
143         try:
144             book.name = request.form['name']
145             book.year = request.form['year']
146             description = request.form['description']
147             book.description = bleach.clean(description)
148             book.publisher = request.form['publisher']
149             book.author = request.form['author']
150             book.page_count = request.form['page_count']
151
152             selected_genre_ids = request.form.getlist('genres')
153             db.session.query(BookGenre).filter(BookGenre.book_id == book_id).delete()
154             for genre_id in selected_genre_ids:
155                 connect = BookGenre(book_id=book_id, genre_id=int(genre_id))
156                 db.session.add(connect)
157
158             db.session.commit()
159             flash(message: 'Книга успешно обновлена!', category: 'success')
160             return redirect(url_for(endpoint: 'books.view_book', book_id=book_id))
161         except SQLAlchemyError as e:
162             db.session.rollback()
163             flash(message: 'При сохранении данных возникла ошибка. Проверьте корректность введенных данных.', category: 'danger')
164
165     genres = db.session.query(Genre).all()
166     book_genres = [book_genre.genre_id for book_genre in db.session.query(BookGenre).filter_by(book_id=book_id).all()]
167     return render_template(template_name_or_list: 'books/edit_book.html', book=book, genres=genres, book_genres=book_genres)

```

Удалить книгу можно по кнопке на главной странице с правами администратора.

Листинг 10 – обработка удаления книги

```

195 @bp.route(rule: '/<int:book_id>/delete_book', methods=['POST'])  ⚡ Arkadii Minchakov
196 @login_required
197 @check_role(['administrator', 'moderator'])
198 def delete_book(book_id):
199     book = db.session.query(Book).get_or_404(book_id)
200     cover_id = book.cover_id
201
202     try:
203         db.session.delete(book)
204         db.session.commit()
205
206         if cover_id:
207             cover = db.session.query(Cover).get(book.cover_id)
208             if cover and cover.filename:
209                 cover_path = os.path.join('static/images', f'{cover.filename}.{cover.mime_type}')
210                 if os.path.exists(cover_path):
211                     os.remove(cover_path)
212                     db.session.delete(cover)
213                     print("path***", cover_path)
214                 db.session.commit()
215
216         flash(message: 'Книга успешно удалена', category: 'success')
217     except SQLAlchemyError as e:
218         db.session.rollback()
219         flash(message: 'Ошибка при удалении книги', category: 'danger')
220         current_app.logger.error(f'Error deleting book: {str(e)}')
221     except ValueError as e:
222         flash(str(e), category: 'danger')
223
224     return redirect(url_for('index'))

```

3.4.2 CR для рецензий

Авторизованный пользователь может оставить рецензию на странице просмотра книги по кнопке «Добавить рецензию», если ранее он не оставлял рецензию. Рецензия создается по умолчанию со статусом «На рассмотрении».

Листинг 11 – обработка добавления рецензии

```
72 @bp.route(rule: '/write_review/<int:book_id>', methods=['GET', 'POST'])  Arkadii Minchakov
73 @login_required
74 <> def write_review(book_id):
75     book = db.session.query(Book).get_or_404(book_id)
76     existing_review = db.session.query(Review).filter_by(book_id=book_id, user_id=current_user.id).first()
77
78     if existing_review:
79         flash(message: 'Вы уже написали рецензию на эту книгу.', category: 'warning')
80         return redirect(url_for(endpoint: 'books.view_book', book_id=book_id))
81
82     if request.method == 'POST':
83         grade = request.form['grade']
84         text = request.form['text']
85
86         sanitized_text = bleach.clean(text)
87
88         review = Review(grade=grade, text=sanitized_text, book_id=book_id, user_id=current_user.id,
89                        status_id=PROCESSING_STATUS_ID)
90         db.session.add(review)
91         db.session.commit()
92
93         flash(message: 'Рецензия успешно добавлена!', category: 'success')
94         return redirect(url_for(endpoint: 'books.view_book', book_id=book_id))
95
96     return render_template(template_name_or_list: 'reviews/write_review.html', book=book, grade_mapping=GRADE_MAPPING)
```

Прочитать рецензии можно на странице просмотра книги, если они были одобрены модератором. Иначе рецензия будет отображаться только у автора в нижней части страницы, а также на странице со всеми рецензиями, доступной из навигации.

3.5. Реализация модерации рецензий на книги

Все рецензии, ожидающие модерации, отображены на странице */moderate*. Ее могут просматривать только модераторы.

Там можно просмотреть автора, текст и оценку рецензии, после чего принять решение об одобрении или отклонении рецензии, нажав соответствующую кнопку.

Листинг 13 – обработка страницы со всеми рецензиями, ожидающими модерации

```

27 @bp.route('/moderate')  ⚡ Arkadii Minchakov
28 @login_required
29 @check_role(['moderator'])
30 <> def moderate():
31     reviews_query = db.session.query(Review).filter(Review.status_id == PROCESSING_STATUS_ID).order_by(
32         Review.created_at.desc())
33
34     page = request.args.get('page', 1, type=int)
35     pagination = reviews_query.paginate(per_page=5, page=page)
36     reviews = pagination.items
37
38     return render_template(template_name_or_list='reviews/moderation.html', pagination=pagination, reviews=reviews,
39                           grade_mapping=GRADE_MAPPING)

```

Листинг 14 – обработка страницы модерации рецензии

```

42 @bp.route('/moderate/<int:review_id>')  ⚡ Arkadii Minchakov
43 @login_required
44 @check_role(['moderator'])
45 <> def moderate_review(review_id):
46     review = db.session.query(Review).get_or_404(review_id)
47     review.text = markdown.markdown(review.text)
48     return render_template(template_name_or_list='reviews/moderate_review.html', review=review, grade_mapping=GRADE_MAPPING)
49
50
51 << /reviews/accept/{review_id}
52 @bp.route(rule='/accept/<int:review_id>', methods=['POST'])  ⚡ Arkadii Minchakov
53 @login_required
54 @check_role(['moderator'])
55 def accept(review_id):
56     review = db.session.query(Review).get_or_404(review_id)
57     review.status_id = ACCEPTED_STATUS_ID
58     db.session.commit()
59     return redirect(url_for('reviews.moderate'))
60
61 << /reviews/decline/{review_id}
62 @bp.route(rule='/decline/<int:review_id>', methods=['POST'])  ⚡ Arkadii Minchakov
63 @login_required
64 @check_role(['moderator'])
65 def decline(review_id):
66     review = db.session.query(Review).get_or_404(review_id)
67     review.status_id = DECLINED_STATUS_ID
68     db.session.commit()
69     return redirect(url_for('reviews.moderate'))

```

4. ОФОРМЛЕНИЕ ИТОГОВ РАБОТЫ

4.1. Создание Git-репозитория с кодом проекта

Создан приватный репозиторий по адресу <https://github.com/ApkawaPopa/minchakov-221-329-web-dev-2024-2>, исходники проекта находятся по ссылке <https://github.com/ApkawaPopa/minchakov-221-329-web-dev-2024-2/tree/main/course-work-exam>

4.2. Оформление отчёта о проделанной работе

Отчет был оформлен в соответствии с методическими рекомендациями.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной курсовой работы была достигнута основная цель – разработка веб-приложения для управления электронной библиотекой. Данный проект сочетает в себе как теоретические, так и практические аспекты, что позволило комплексно подойти к решению поставленной задачи.

Теоретическая часть работы включала в себя анализ предметной области, что позволило выявить ключевые требования к функциональности веб-приложения и определить основные подходы к его реализации. Были изучены существующие программные продукты и инструменты разработки, что позволило выбрать наиболее подходящие технологии для данного проекта. Это включало использование Flask в качестве веб-фреймворка и SQLAlchemy для работы с базой данных, что обеспечило гибкость и масштабируемость приложения.

Практическая часть работы была посвящена непосредственной разработке веб-приложения. Были реализованы основные функции, такие как аутентификация пользователей, добавление и редактирование книг, управление рецензиями, а также модерация контента. Особое внимание было уделено обеспечению безопасности приложения и защите данных пользователей, что является критически важным аспектом при разработке современных веб-приложений.

Результаты проделанной работы показали, что разработанное приложение обладает высокой функциональностью и удобством использования, что делает его пригодным для реального применения в библиотечных системах. Разработанная система может быть использована как основа для дальнейшего развития и расширения функциональности, что открывает широкие возможности для её применения в различных образовательных и культурных учреждениях.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Документация Flask: <https://flask.palletsprojects.com/en/latest/>
2. Документация Flask-SQLAlchemy: <https://docs.sqlalchemy.org/en/14/>
3. Документация Flask-Login: <https://flasklogin.readthedocs.io/en/latest/>
4. Документация Python: <https://docs.python.org/3/index.html>
5. Документация Bootstrap: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
6. Курс «Разработка веб-приложений»: <https://online.mospolytech.ru/course/view.php?id=13940>
7. Консультация по курсовой работе: <https://www.youtube.com/watch?v=4MKQ7qwyd6E>
8. Методические рекомендации по написанию курсовых работ: https://online.mospolytech.ru/pluginfile.php/975381/mod_resource/content/1/M_R_po_KP_09.03.01.pdf