|  |  |  |
| --- | --- | --- |
|  |  | |
| Тема | Частное учреждение профессиональная образовательная организация | |
|  | ТЕХНИКУМ «БИЗНЕС И ПРАВО»      ЗАЩИЩЕНО С ОЦЕНКОЙ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись и ФИО руководителя)  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.  **КУРСОВАЯ РАБОТА**  РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ ВЕБ-ПРИЛОЖЕНИЯ  «УЧЕТ ВЫДАЧИ КНИГ В БИБЛИОТЕКЕ»  Проектирование и разработка веб-приложений    Реснянская А.А    4 курс, группа ИС-4    09.02.07 Информационные системы и программирование      Мартыненко В.А.          г. Белореченск 2023 г. |
| По МДК 09.01    Выполнила    Курс, группа    Специальность    Научный руководитель | |
|  | |  |

### СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc34394)

[1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 5](#_Toc34395)

[1.1 Анализ предметной области 5](#_Toc34396)

[1.2 Цели и задачи выполняемой системы 5](#_Toc34397)

[1.3 Метод реализации процесса проектирования 6](#_Toc34398)

[1.4 Обоснование проектных решений 8](#_Toc34399)

[2. ПРОЕКТНАЯ ЧАСТЬ 22](#_Toc34400)

[2.1 Техническое задание 22](#_Toc34401)

[2.2Установка и настройка среды разработки 24](#_Toc34402)

[2.3 Руководство пользователя 25](#_Toc34403)

[2.4 Руководство администратора 26](#_Toc34404)

[ЗАКЛЮЧЕНИЕ 27](#_Toc34405)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 29](#_Toc34406)

ПРИЛОЖЕНИЯ

# ВВЕДЕНИЕ

Актуальность исследования в современном мире успех компаний немыслим без автоматизированных систем накопления и обработки информации. Благодаря таким системам компании могут эффективно решать конкретные задачи в своей области и оптимизировать работу своего персонала. С продвижением технологий все больше библиотек принимают решение перейти на электронную систему учета книг и выдачу их читателям. Разработка веб-приложения, отвечающего за базу данных и учет книг, позволит автоматизировать процесс и упростить работу библиотекарей, тем самым уменьшить затраты времени.

Объект исследование – информационная система учет книг в библиотеке.

Предмет исследования - предметом исследования является учет выдачи книг в библиотеке.

Цель исследования направлена на разработку информационной системы для учета занесение только поступивших книг в библиотеку, организация хранения книг, добавление новых читателей, выдачи и возврат книг в библиотеке.

Задачи исследования:

* Изучение предметной области;
* Проведение анализа существующих систем учета выдачи книг в библиотеках и выявление их недостатков.
* Проведение анализа требований к разрабатываемому веб-приложению;
* Проектирование работы разрабатываемой системы;
* Разработка информационной системы;
* Тестирование разработанной информационной системы.

Методы исследования - для учета книг в библиотеке используются различные методы исследования, такие как:

− Статистический учёт — это составление количественной характеристики объёма, состава и движения фонда в процессе его формирования и функционирования.

− Учет библиотечного фонда – это процесс, который включает в себя сбор количественных данных о размере, составе и движении фонда в соответствии с определенными причинами. Он отражает приход и отток документов, а также общий размер библиотечного фонда и его подразделений. Кроме того, учет библиотечного фонда служит основой для составления статистических отчетов.

# АНАЛИТИЧЕСКАЯ ЧАСТЬ

## Анализ предметной области

Анализирование предметной области, связанной с учетом книг в библиотеке, позволяет выявить специфические особенности и требования. Этот анализ может быть полезен для определения размера и структуры базы данных, создания удобного интерфейса для библиотекарей и читателей, а также выявления потенциальных проблем и рисков, связанных с учетом и использованием книг. Эта система построена по трехзвенной клиент-серверной архитектуре.

## Цели и задачи выполняемой системы

Целями информационной системы является:

* удобство использования для персонала библиотеке
* авторизация и регистрация пользователя
* распределение ролей на пользователей и администраторов
* занесение авторов и книг в базу данных
* занесение новых читателей
* хранение данных
* изменение данных
* удаление данных

Основные задачи, решаемые информационной системой учета книг в библиотеке, включают:

1. Регистрация книг: система позволяет оперативно и точно учитывать весь книжный фонд библиотеки. Каждая книга получает уникальный идентификатор, а информация о ней, включая автора, название и другие характеристики, заносится в базу данных.
2. Контроль наличия книг: система позволяет в режиме реального времени отслеживать наличие каждой книги в библиотеке. Это позволяет предоставлять читателям актуальную информацию о доступных книгах, а также контролировать сроки возврата, чтобы предотвратить задержки.
3. Учет выдачи книг: система фиксирует все операции выдачи книг читателям, включая информацию о дате, времени и сроке возврата.
4. Статистический анализ: система позволяет проводить анализ данных, временем их использования, частотой выдачи и другими параметрами.

Это позволяет библиотеке оптимизировать управление книг.

## Метод реализации процесса проектирования

Проектирование информационной системы учета книг в библиотеке является комплексным и ответственным процессом, требующим профессиональных навыков, глубокого анализа и тщательной проработки каждого этапа. Для эффективной реализации этого процесса следует придерживаться определенного метода, который обеспечивает структурированный подход и минимизирует возможные риски и ошибки.

1. Осуществление процесса проектирования информационной системы учета книг в библиотеке начинается с детального анализа и понимания требований и потребностей библиотеки.
2. Проектирование архитектуры информационной системы, которая должна быть гибкой, масштабируемой и простой в использовании.
3. Разработка интерфейса пользователя и функциональных возможностей системы.
4. Тестирование и отладка системы.
5. Важно обеспечить поддержку и обучение пользователей системы, чтобы они смогли максимально эффективно использовать все ее функциональные возможности.

Таким образом, метод реализации процесса проектирования информационной системы учета книг в библиотеке предлагает систематический подход, который включает анализ требований, проектирование архитектуры и функциональности системы, тестирование и отладку, а также поддержку и обучение пользователей.

Система должная предполагать наличие следующих модулей, представленных на рисунке 1.1:

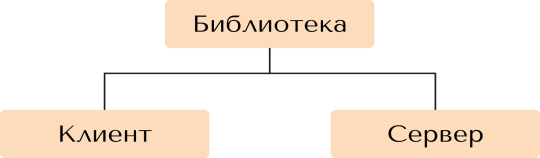


Рисунок 1.1 - Модули, которые должны быть в программе

Модули разрабатываемой системы, представленные на рисунке 1.2:

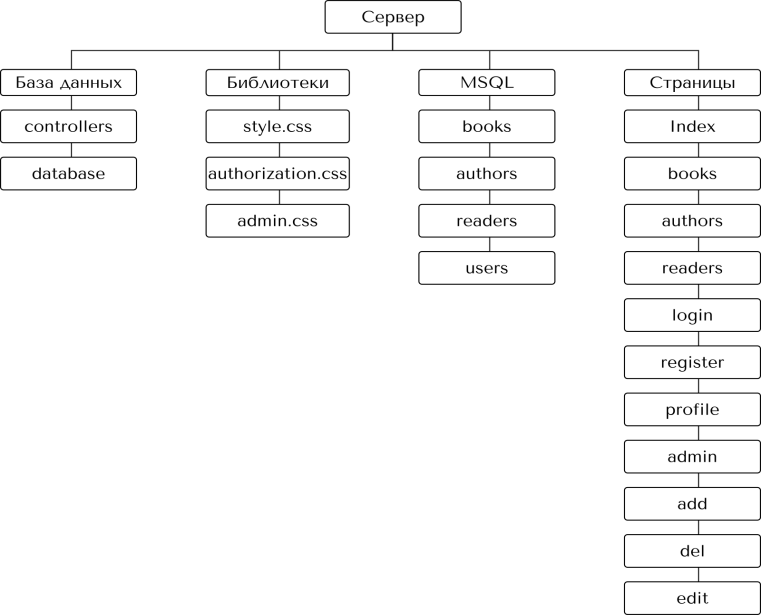


Рисунок 1.2 - Модули разрабатываемой системы

Описание модулей разрабатываемой системы:

* controllers - функции базы данных;
* database - подключение к базе данных;
* style.css – основные CSS стили;
* authorization.css – css стили для полей регистрации и авторизации;
* admin.css – css стили для админ-панели;
* books - название таблицы книг;
* authors - название таблицы авторов;
* readers - название таблицы читателей;
* users - название таблицы пользователей;
* Index - главная страница;
* books - страница с таблицей книг;
* authors - страница с таблицей авторов;
* readers - страница с таблицей читателей;
* login - страница авторизации;
* register - страница регистрации;
* profile - страница профиля;
* admin - страница админ-панели;
* add – добавление данных;
* del - удаление данных;
* edit - изменение данных;

## Обоснование проектных решений

Перед тем как создать программу мы рассмотрели следующие преимущества и недостатки:

Основные преимущества:

1. Автоматизация процесса учета книг: Информационная система позволит библиотекарям автоматизировать процесс учета книг, что сэкономит значительное количество времени и сил.
2. Оптимизация работы библиотекарей: Информационная система позволит библиотекарям избежать многочисленных рутинных операций, связанных с учетом и поиском книг. Они смогут уделять больше времени консультации читателей, проведению мероприятий и развитию библиотеки в целом.

Возможные недостатки:

1. Технические сложности: Реализация информационной системы учета книг может быть затруднена техническими проблемами или проблемами безопасности данных. Необходима тщательная подготовка и тестирование системы, чтобы минимизировать возможные риски.
2. Нехватка квалифицированного персонала: Управление информационной системой учета книг потребует наличия квалифицированных специалистов с навыками в области информационных технологий. В случае, если библиотека не сможет найти или обучить подобных сотрудников, реализация проекта может столкнуться с проблемами.
3. Расходы на реализацию и поддержку: Внедрение информационной системы учета книг потребует значительных затрат на покупку необходимого оборудования и программного обеспечения, а также на обучение персонала. Также необходима будет постоянная техническая поддержка системы, что может быть связано с дополнительными расходами.

Для того, чтобы сделать программу по учету книг в библиотеке, мы будем использовать клиент-серверную архитектуру.

В клиент-серверной архитектуре имеется три звена:

Представление данных - на стороне клиента.

Прикладной компонент - на выделенном сервере приложений, здесь происходит вся бизнес-логика.

Управление ресурсами - сервер базы данных, который и представляет запрашиваемые данные.

Преимущества данной архитектуры:

− Высокую степень гибкости и масштабируемости

− Высокую безопасность

− Высокую производительность

Схема клиент-серверной архитектуры представлен на рисунке 1.3:

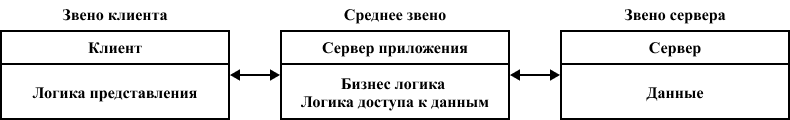


Рисунок 1.3 - Клиент-серверная архитектура

Требования к системе разработаны в соответствии со стандартом качества, описывающим многоуровневую модель характеристики качества и соответствующий им набор атрибутов.

Таблица 1.1 - Требования к системе

|  |  |  |
| --- | --- | --- |
| Атрибуты | Метрики | Требования |
| 1 | 2 | 3 |
|  | Удобство | |
| Простота использования | Среднее время, необходимое пользователю, чтобы найти  необходимый элемент интерфейса | Поиск отдельного элемента интерфейса не должен занимать более 1 секунды |
| Привлекательность | Соответствие интерфейса требованиям заказчика | 97% элементов интерфейса должны быть адаптированы для пользователя и 99% для  заказчика; |
| Обучаемость | Показатель, затрачиваемый пользователями на обучение работе с АИС | Пользователь должен изучить работу ПО за первые 3–5 минут использования |
|  | Производительность | |
| Временная эффективность | Время выполнения компонента программы | * Приложение должно   обслуживать любую страницу не дольше 300 мсек (не включая задержки в сети);   * Главная страница   сайта в среднем случае должна загружаться быстрее 3-x секунд. |
| Эффективность использования ресурсов | Объемы ресурсов требуемых для выполнения задач; | * Постоянная память, используемая программой – не более 10МБ; * Приложение должно   потреблять не более 4 МБ памяти на каждый неактивный сеанс с пользователем; |

Продолжение таблицы 1.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
|  | Переносимость |  |
| Удобство установки | Легкость установки | Необходима установка: xampp, mysql, любой веббраузер, admin php |
| Способность к сосуществованию | Способность ПО сосуществовать с другими программами в общем окружении | Программа не должна занимать большие объемы памяти устройства (максимум  4МБ) |
| Удобство замены другого ПО данным | Совместимость структур данных | - Система должна быть разработана по общим принципам и не требовать навыков от пользователя; - Должна поддерживать внедрение новых версий; - Система должна быть применима вместо других программных систем для решения тех же задач в определенном окружении |

Требования к численности и квалификации персонала системы и режиму его работы.

Для обеспечения бесперебойной работы информационной системы необходимо иметь достаточное число квалифицированных специалистов. Их количество должно быть определено исходя из объема работы и нагрузки на систему. Рекомендуется назначить ответственных лиц, координирующих работу персонала и осуществляющих контроль за корректностью работы системы.

Качество работы персонала системы также является важным аспектом. Каждый член команды должен обладать необходимыми знаниями и навыками, связанными с информационными технологиями

В системе есть две группы пользователей:

− пользователям предоставляется возможность авторизации,

регистрации, просмотра данных;

− администратор имеет возможность добавлять, изменять и удалять данные;

Пользователи системы должны обладать навыками работы с персональным компьютером на уровне пользователя.

Таблица 1.2 - Требования к надежности

|  |  |  |
| --- | --- | --- |
| Атрибуты | Метрики | Требования |
| Надежность | | |
| Устойчивость к отказу | Коэффициент аварийных отказов | * Не менее 90% ошибок в программе должны обрабатываться без экстренного завершения; * Обеспечение бесперебойного питание активного сетевого оборудования; |
| Способность к восстановлению | Отношение количества числа успешного восстановление работы системы к ее аварийному завершению за определенный за определенный промежуток времени | Восстановление  работоспособности системы не должно превышать 5 минут |
| Уровень зрелости | Среднее время работы без сбоев | * Система должна работать 24 часа в сутки; * Система должна оповещать пользователя при некорректных действиях в системе; * Данные о новых пользователях и товарах должны сохраняться. |

Средняя доступность АИС должна составлять не менее 99%

Среднее время между сбоями — это среднее время, за которое компонент или модуль может выполнять свои функции без перерыва. Измеряется от начала работы до момента следующего сбоя.

Среднее время работы без сбоев должно составлять не менее 1500 часов.

В таблице 1.3 представлен перечень и критерии отказов для каждой функции, по которой задаются требования по надежности.

Таблица 1.3 - Перечень и критерии отказов для каждой функции

|  |  |
| --- | --- |
| Функции | Критерии отказа |
| Отображение данных | Отсутствие соединения с базой данных |
| Просмотр списка книг, авторов, читателей | Отсутствие соединения с базой данных |
| Добавление | Отсутствие соединения с базой данных |
| Редактирование | Отсутствие соединения с базой данных |
| Удаление | Отсутствие соединения с базой данных |

Требования к функциям сайта, выполняемым программным обеспечением

Имя: Регистрация пользователя

Описание: Пользователь заходит на сайт впервые и для того, чтобы использовать все функции системы необходимо авторизироваться

Действующий субъект: пользователь Предусловие: пользователь зашел на сайт Сценарий:

|  |  |
| --- | --- |
| − | Перейдя на главную страницу нажать кнопку “Входа” |
| − | Затем нажав кнопку “Зарегистрироваться” |
| − | Пользователь вводит данные потом нажимает на кнопку |
| − | Система сохраняет информацию о пользователе в базе данных |
| − | Система переходит на Главную страницу |

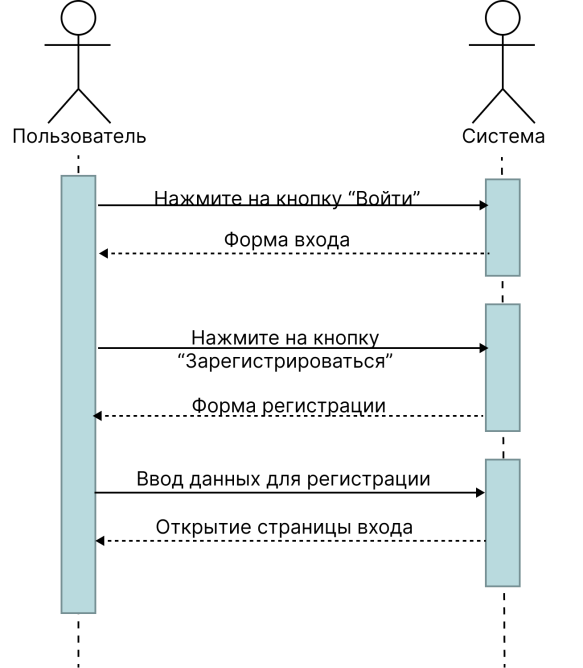


Рисунок 1.4 - Диаграмма последовательности сценария “Регистрация пользователя”

Расширения: в случае некорректного формата введенных данных, при нажатии на кнопку “Зарегистрироваться”, над формой выведется сообщения о неправильно введённых данных.

На рисунке 1.4 представлена диаграмма последовательности сценария

“Регистрация пользователя”

Имя: Авторизация

Описание: для того, чтобы начать работать с системой пользователю необходимо ввести свои данные для входа.

Действующий субъект: пользователь

Предусловие: пользователь должен быть уже зарегистрирован в системе и имеет корректные данные для входа Сценарий:

|  |  |
| --- | --- |
| − | Система отображает форму для входа в систему |
| − | Пользователь вводит необходимые данные для входа в систему |
| − | Пользователь нажимает на кнопку “Вход” |
| − | Система сверяет данные, введённые пользователем с данными в БД. |
| − | Система открывает главное окно |

Расширения: в случае неправильно введённых данных будет выведенная ошибка.

На рисунке 1.5 представлена диаграмма последовательности сценария " Авторизация ".

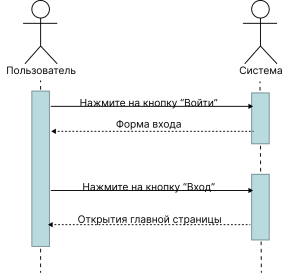


Рисунок 1.5 - Диаграмма последовательности сценария “Авторизация”

Имя: Добавление новых данных.

Описание: Пользователь имеет возможность добавить новые данные.

Действующий субъект: Пользователь.

Предусловие: Пользователь успешно зашел в систему по своим данным для входа.

Сценарий:

|  |  |
| --- | --- |
| − | Система открывает окно с формой и кнопкой отправки данных; |
| − | Пользователь указывает соответствующие данные; |
| − | Система заполняет введенные данные в соответствующие строки; |
| − | Пользователь нажимает на кнопку отправки; |
| − | Система обновляет систему, добавляет и выводит новые данные. |

Расширения: при отсутствии соединения с сервером при нажатии на кнопку «Добавить» выведется сообщение об ошибке.

На рисунке 1.6 представлена диаграмма последовательности сценария "Добавления".

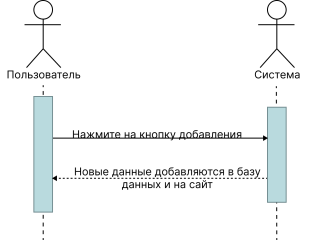


Рисунок 1.6 – Диаграмма последовательности сценария "Добавления" Имя: Редактирование данных.

Описание: Пользователь имеет возможность редактировать данные.

Действующий субъект: Пользователь.

Предусловие: Пользователь успешно зашел в систему по своим данным для входа.

Сценарий:

* Система открывает окно с какими-либо данными;
* Пользователь выбирает из них одну любую запись и нажимает кнопку для изменения;
* Система заполняет пустую строку выбранным значением;
* Пользователь редактирует данные;
* Система обновляет данные.

Расширения: при отсутствии соединения с сервером при нажатии на кнопку «Изменить» выведется сообщение об ошибке.

На рисунке 1.7 представлена диаграмма последовательности сценария "Редактирования".

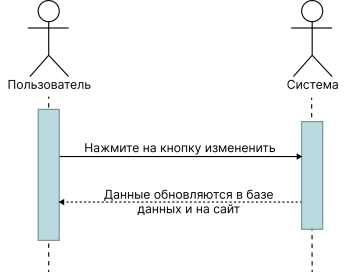


Рисунок 1.7 – Диаграмма последовательности сценария "Редактирования" Имя: Удаление данных.

Описание: Пользователь имеет возможность удалить данные.

Действующий субъект: Пользователь.

Предусловие: Пользователь успешно зашел в систему по своим данным для входа.

Сценарий:

* Система открывает окно с какими-либо данными;
* Пользователь выбирает строку, которую необходимо удалить;
* Система подсвечивает кнопку удаления;
* Пользователь нажимает на кнопку;
* Система удаляет выбранные данные и обновляет их

Расширения: при отсутствии соединения с сервером при нажатии на кнопку «Удаление» выведется сообщение об ошибке.

На рисунке 1.8 представлена диаграмма последовательности сценария "Удаления".

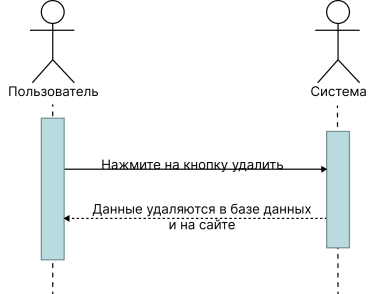


Рисунок 1.8 – Диаграмма последовательности сценария "Удаления"

Для функционирования системы требуется наличие сервера и клиентских компьютеров, на которых будет производиться работа с системой. Перед вводом системы в действие необходимо:

− Провести в библиотеке интернет-соединения;

− Обеспечить каждого библиотекаря персональным компьютером;

− Установить на каждый ПК любой браузер последней версии;

− Для работы также необходим локального веб-сервера, содержащая Apache и доступ подключения к базе данных;

При анализе предметной области разрабатываемой системы были выделены следующие сущности, на основе которых будет разрабатываться база данных системы:

− пользователи

− книги

− авторы

− читатели

Разрабатываемая система требует в своей архитектуре наличие БД-сервера со свободной реляционной системой управления.

База данных информационной системы должна состоять из следующих связанных таблиц:

1. books - таблица, содержащая информацию о книгах
2. authors - таблица, содержащая информацию об авторах
3. readers - таблица, содержащая информацию о читателях
4. users - таблица, содержащая информацию о пользователях

Данные о книгах представлены в таблице books 1.4:

Таблица 1.4 - Данные о книгах

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничения |
| id | bigint | 8 | Идентификатор книги | Первичный ключ |
| title | varchar | 255 | Название книги | NOT NULL |
| year | int | 4 | Год издания | NOT NULL |
| quantity | int | 20 | Город | NOT NULL |
| id\_author | int | 11 | Идентификатор таблицы authors | внешний ключ от таблицы authors |

Данные об авторах представлены в таблице authors 1.5:

Таблица 1.5 - Данные об авторах

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничения |
| id | int | 11 | Идентификатор пользователя | Первичный ключ |
| surname | varchar | 255 | Фамилия | NOT NULL |
| name | varchar | 255 | Имя | NOT NULL |
| lastname | varchar | 255 | Отчество | NOT NULL |

Данные о читателях представлены в таблице readers 1.6:

Таблица 1.6 - Данные о читателях

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничения |
| id | int | 11 | Идентификатор пользователя | Первичный ключ |
| surname | varchar | 255 | Фамилия | NOT NULL |
| name | varchar | 255 | Имя | NOT NULL |
| lastname | varchar | 255 | Отчество | NOT NULL |
| birthday | date | - | Дата рождения | NOT NULL |
| phone | decimal | 11,0 | Телефон | NOT NULL |
| email | varchar | 255 | Электронная почта | NOT NULL |

Данные о пользователях представлены в таблице users 1.7:

Таблица 1.7 - Данные о пользователях

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничения |
| id | int | 11 | Идентификатор пользователя | Первичный ключ |
| admin | tinynt | - | Фамилия | NOT NULL |
| username | varchar | 255 | Имя | NOT NULL |
| email | varchar | 255 | Электронная почта | NOT NULL |
| password | varchar | 255 | Пароль | NOT NULL |
| created | timestamp | - | Время и дата создания | NOT NULL |

На данном этапе разберем то, как будет вести себя система при выполнение выделенных модулей. Для этого мы будем использовать диаграммы состояний в нотации UML.

На рисунке 1.9 представлена общая диаграмма состояний. При переходе пользователем в программу пользователь будет находиться в том, состоянии, когда ему необходимо войти, чтобы продолжить работу с системой и осуществлять деятельность согласно выделенным вариантам использования.

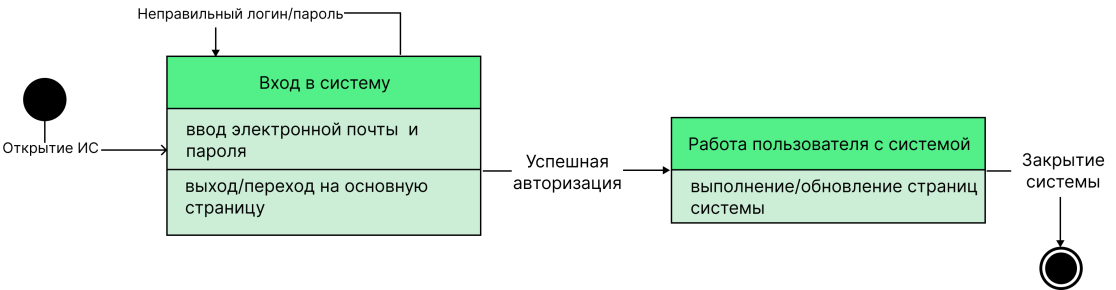


Рисунок 1.9 - Общая диаграмма состояний системы

На рисунке 1.10 показано детальное описание работы пользователя с системой. Сотрудник может просматривать список всех данных, добавлять, удалять, редактировать, производить поиск.

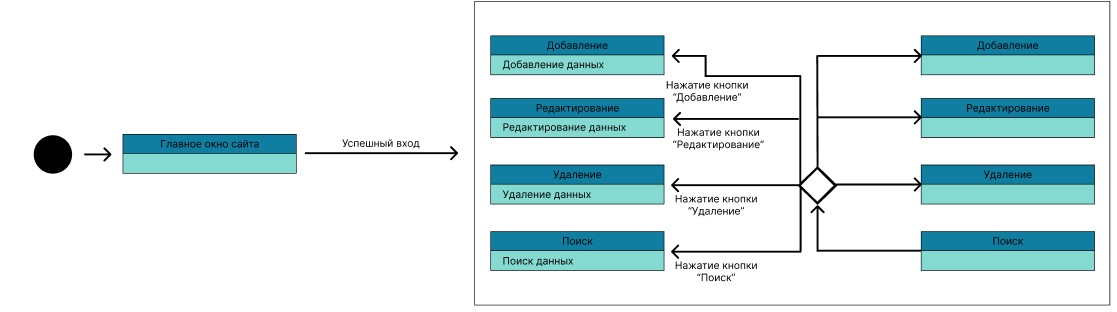


Рисунок 1.10 – Диаграмма состояний работы пользователя с системой

На рисунке 1.11 изображена декомпозиция состояния «Добавление». После того как пользователь успешно зашел в программу, ему необходимо выбрать добавление, которое ему необходимо. Далее он может заполнить поля данными, нажав на Кнопку «Добавить» добавятся данные в базу данных.

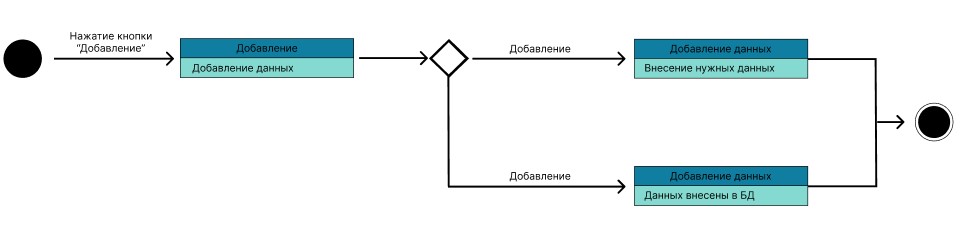


Рисунок 1.11 – Диаграмма состояний добавление данных

На рисунке 1.12 изображена декомпозиция состояния «Редактирования».

Пользователь должен нажать на кнопку «Редактировать». После этого приложение обновится и обновит данные в базе данных. Далее пользователь может просмотреть все изменённые данные.

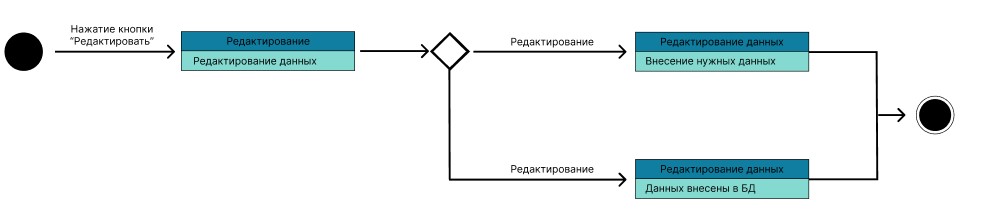


Рисунок 1.12 – Диаграмма состояний редактирования данных

На рисунке 1.13 изображена диаграмма состояния «Удаления». Пользователь может удалять все данные товара. Далее необходимо нажать на кнопку «Удалить», и программа выполнит запрос.

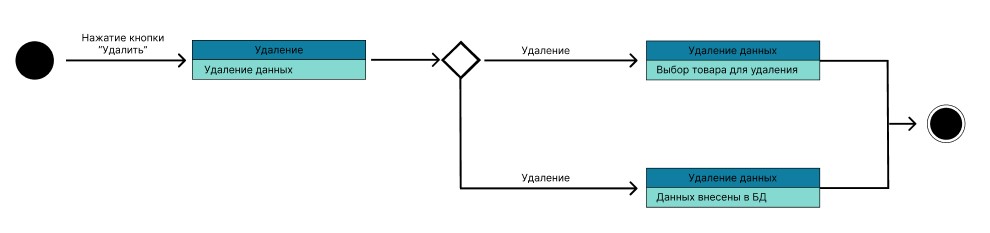


Рисунок 1.13 – Диаграмма состояний удаления данных

# ПРОЕКТНАЯ ЧАСТЬ

## Техническое задание

Информационное обеспечение является важной составляющей успешной работы информационной системы учета книг в библиотеке. Современные библиотеки сталкиваются со значительным увеличением количества книг, поступающих в их фонды, а также с растущим спросом со стороны читателей. Поэтому эффективная и надежная система учета книг становится неотъемлемой частью организации и управления библиотекой.

Информационная система учета книг в библиотеке отвечает за сбор, хранение, обработку и информацию о наличии книг, а также о доступности для читателей.

Все данные сайта должны храниться в структурированном виде под управлением реляционной СУБД - MYSQL. Наполнение различных сайтов, функционирование которых поддерживается одной и той же инсталляцией системы, должно храниться под управлением единой СУБД.

Система управления базами данных (СУБД) – это комплекс программно-языковых средств, позволяющих создать базы данных и управлять данными. Иными словами, СУБД — это набор программ, позволяющий организовывать, контролировать и администрировать базы данных. Большинство сайтов не могут функционировать без базы данных, поэтому СУБД используется практически повсеместно.

Требования по оформлению:

Дизайн и интерфейс должен быть интуитивно понятным и удобным в использовании для различных категорий пользователей. Веб-приложение должно обеспечивать защиту данных пользователей, а также предотвращать несанкционированный доступ к системе.

Разработка уникального и эффективного дизайна интерфейса для вебприложения, направленного на учет выдачи книг в библиотеки, требует глубокого понимания потребностей пользователей и их взаимодействия с системой.

Мы стремимся к минимализму и функциональности, чтобы пользователи могли сосредоточиться на самом процессе учета книг, а не на излишне сложном интерфейсе. Четкая структура, понятные и легко доступные функции, а также понятные инструкции помогают пользователям максимально быстро овладеть системой и начать использовать ее непосредственно для учета и выдачи книг.

Основные компоненты интерфейса включают в себя:

1. Главная страница, на которой пользователь может получать общую информацию о системе, а также информацию на выдачу и возврат книг.
2. Форму для добавления книг, включающую в себя поля для ввода информации о книге, ее авторе, количеству и прочих релевантных данных.
3. Страницу учета выдачи книг, где пользователь может создавать запросы на выдачу книг, указывая информацию о читателе, книге и сроке ее возврата.
4. Страницу администрирования, где администратор системы имеет доступ к полному функционалу системы, включая управление пользователями.

Функциональные требования:

1. Регистрация пользователей:

Пользователи должны иметь возможность зарегистрироваться в системе, указав свои персональные данные, включая имя, фамилию, адрес электронной почты и пароль.

При регистрации необходимо проводить проверку уникальности адреса электронной почты и пароля.

1. Авторизация пользователей:

После успешной регистрации пользователю должна быть предоставлена возможность войти в систему, используя свой адрес электронной почты и пароль.

1. Учет книг:

Пользователи должны иметь возможность просматривать список доступных книг в библиотеке.

Приложение должно отображать подробную информацию о каждой книге, включая название, автора, год издания и количество экземпляров.

1. Управление пользователями:

Библиотекари должны иметь возможность управлять учетными записями пользователей, включая создание, блокировку и удаление аккаунтов.

## Установка и настройка среды разработки

Для установки и настройки среды разработки для веб-приложения "Учет выдачи книг в библиотеки" было сделано несколько шагов:

1. Установка текстового редактора. В качестве текстового редактора можно использовать любой, но рекомендуется использовать популярные редакторы, такие как Visual Studio Code, PhpStorm, и другие.
2. Установка интерпретатора PHP. Имеет открытый исходный код. Изначально создавался для разработки веб-приложений, но в процессе обновлений стал языком общего назначения. Основная область применения — разработка скриптов, которые работают на стороне сервера. Также можно создавать скрипты командной строки и GUI-приложения (приложения с графическим интерфейсом пользователя).
3. Установка необходимых инструментов. При надобности следует установить дополнительные инструменты, такие как компиляторы и отладчики, в зависимости от того, какой язык программирования был выбран.
4. Настройка окружения. После установки редактора, интерпретатора PHP и других необходимых инструментов, нужно настроить среду разработки. Это может включать в себя установку дополнительных расширений или плагинов для текстового редактора, установку виртуальной среды PHP для изоляции проектов и установку дополнительных модулей и библиотек, необходимых для разработки веб-приложения.
5. Создание проекта. Последний шаг - создание самого проекта вебприложения “Учет выдачи книг в библиотеке”. Определить структуру проекта, создать необходимые файлы и начать разработку, используя выбранный текстовый редактор и язык программирования.

Для функционирования сайта необходимо следующее программное обеспечение:

− Операционная система: Windows 10/Linux/macOS;

− Сервер: MS SQL Server 2019;

− СУБД: MSSQL для MS SQL Server; − XAMPP или Open Server panel.

## Руководство пользователя

Первое, что является единым для всех пользователей – авторизация, без которой пользователь не сможет получить доступ к функционалу системы, он сможет только просмотреть информацию о сайте. Чтобы авторизоваться необходимо на главной странице в левом верхнем углу нажать на кнопку «Вход».

Для того чтобы авторизироваться необходимо пользователю в окне авторизации заполнить две строки: электронную почту и пароль, затем нажать на кнопку входа. Если же у пользователя нет свое учетной записи, он может создать ее. Для этого нужно в окне авторизации под кнопкой входа, нажать на ссылку «Регистрация» после чего пользователя перекидывает на новую страницу. Там ему необходимо ввести свои данные и также нажать соответствующую кнопку. Если все данные будут заполнены правильно, то пользователя перекинет на главную страницу, если же в заполнении данных была допущена ошибка, то пользователю над формой выведется соответствующие сообщения, после чего необходимо будет заполнить данные по-новому.

Главная страница включает в себя информацию о самой программе, ее функциях.

Незарегистрированному пользователю можно только просмотреть информацию о книгах, авторах и читателях. Если же пользователь зарегистрирован, то у него появляется возможность добавлять, изменять и удалять данные.

Также есть кнопка профиля наведя на которую можно выйти из своей учетной записи.

## Руководство администратора

Первые действия администратора на сайте идентичны с действиями обычного пользователя – авторизация.

При авторизации если нажать на кнопку войти, система проверит что вы являетесь администратором и вместо главной страницы перекинет на страницу админ панели, где предоставляется возможность редактировать данные о пользователях.

В панели администратора можно управлять информацией о пользователях, а именно создавать новых пользователей, измять информацию о существующих.

В остальном же администратору, как и обычному пользователю можно добавлять, изменять или удалять книги, авторов и читателей

# ЗАКЛЮЧЕНИЕ

Внедрение информационной системы учета книг в библиотеке предоставит множество преимуществ, таких как автоматизация процессов учета, удобство для читателей и оптимизация работы библиотекарей. Однако, необходимо учитывать возможные риски, такие как технические сложности, нехватка квалифицированного персонала и расходы на реализацию и поддержку системы. Все эти факторы требуют тщательного планирования и подготовки, чтобы обеспечить успешную реализацию данного проекта.

В результате выполнения было создано веб-приложение для учета выдачи книг в библиотеке, он позволяет внесения и хранения данных о книгах, авторах и читателях. было проведено обширное исследование и анализ современных технологий и подходов к созданию таких систем. При разработке этой работы удалось создать приложение удобное и функциональное для пользователей, способствующие оптимизации работы библиотеки обеспечению удобного взаимодействия с пользователями.

Одной из основных задач, решенных в ходе разработки, было создание надежного и безопасного веб-приложения, который обеспечивает аутентификацию и авторизацию пользователей, а также контролирует доступ к различным функциональным возможностям приложения.

Помимо этого, веб-приложение обеспечивает эффективное управление данными о книгах и пользователях. Была реализована система учета выдачи и возврата книг, которая позволяет отслеживать актуальное состояние каждой книги. Для оптимизации работы были использованы современные базы данных и методы их взаимодействия с сервером, что позволяет быстро и эффективно обрабатывать запросы пользователей.

Разработанный интерфейс обладает минималистичным дизайном и интуитивно понятной навигацией, что делает работу с приложением комфортной и доступной для всех пользователей.

В целом, разработка серверной части веб-приложения для учета выдачи книг в библиотеке является значимым шагом в современной автоматизации библиотечного дела. Созданный продукт обеспечивает эффективное и безопасное функционирование системы, оптимизирует работу библиотеки и обеспечивает удобство пользователей. Разработка данной системы имеет большой потенциал для применения в различных библиотечных учреждениях, способствуя совершенствованию и модернизации библиотечного сервиса.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бенкен Елена PHP, MySQL, XML. Программирование для

Интернета; БХВ-Петербург - М., 2019. - 336 c.

1. Вандюк, Джон К. CMS Drupal. Руководство по разработке системы управления сайтом / Вандюк, Джон К., Мэтт Вестгейт,. - М.: Вильямс, 2019. - 400 c.
2. Васильев, Алексей. Программирование на РНР в примерах и задачах / Алексей Ва-сильев. — Москва: Эксмо, 2021. — 352 с.
3. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, и др.. - М.: СПб: Питер, 2021. - 368 c.
4. Дакетт, Джон. РНР и MYSQL. Серверная веб-разработка / Джон Дакетт — Москва: Эксмо, 2023. — 688 с.
5. Зандстра Мэтт PHP. Объекты, шаблоны и методики программирования; Вильямс - М., 2021. - 560 c.
6. Колисниченко Д.Н. PHP и MySQL. Разработка веб-приложений; БХВ-Петербург - М., 2022. - 344 c.
7. Кузнецов PHP. Практика создания Web-сайтов / Кузнецов, М.В. и. - М.: БХВ-Петербург, 2020. - 895 c.
8. Леонтьев, Борис PHP 5.0 для начинающих, или как создать динамический WEB-сайт / Борис Леонтьев. - М.: Новый издательский дом, 2019.

- 176 c.

1. Маркин А. В. Программирование на SQL: учебное пособие для среднего профессионального образования/ А. В Меркин. - М.: издательство юрайт 2023. 435 c.
2. Никсон Р. Создаем динамические веб-сайты с помощью PHP,

MySQL, JavaScript и CSS; Питер - М., 2020. - 687 c.

1. Петюшкин, Алексей HTML в Web-дизайне / Алексей Петюшкин. - Москва: Машиностроение, 20. - 400 c.
2. Полуэктова, Н. Р. Разработка веб-приложений: учебное пособие для вузов / Н. Р. Полуэктова. — М.: Издательство Юрайт, 2023. — 204 с.
3. Решетникова Н.В Учет библиотечных фондов: методические материалы в помощь библиотекарю / составитель Н.В. Решетникова. – п.Мари-

Турек : МБУ Мари-Турекская МПЦБ, 2020. - 42с

1. Шкрыль, А. PHP - это просто. Программируем для Web-сайта / А.

Шкрыль. - М.: БХВ-Петербург, 2021. - 368 c.

**ПРИЛОЖЕНИЕ**

CSS

Используется для визуализации сайта.

style.css - Основные стили

/\* Подключение шрифтов \*/

@import url('https://fonts.googleapis.com/css2?family=Gabriela&family=Tenor+Sans&display=swap');

html {

    box-sizing: border-box;}

body::-webkit-scrollbar {

    width: 10px;}

body::-webkit-scrollbar-track {

    background: #FFD8BE;}

body::-webkit-scrollbar-thumb {

    background-color: #D8BEFF;

    border-radius: 20px;}

\*, \*::before, \*::after {

    margin: 0;

    padding: 0;

    box-sizing: inherit;}

body, section, header, footer {

    position: relative;}

a {

    color: inherit;

    text-decoration: none;}

li {

    list-style: none;}

img {

    max-width: 100%;}

b {

    font-family: 'Gabriela', serif;

    font-size: 24px;

    padding: 10px;}

body {

    background-color: #FFEEDD;

    font-family: 'Tenor Sans', sans-serif;

    font-size: 16px;

    line-height: 1.4;

    min-height: 100vh;}

.container {

    max-width: 1320px;

    padding: 0 10px;

    margin: 0 auto;}

.btn {

    display: inline-block;

    width: 140px;

    padding: 10px;

    border-radius: 5px;

    background: #FFD8BE;

    color: #000;

    text-align: center;

    font-family: 'Gabriela', serif;

    font-size: 20px;

    border: 2px solid #D8BEFF;

    cursor: pointer;}

.button-issuance {

    display: grid;

    justify-content: center;

    line-height: 10;}

.button-iss {

    width: 240px;

    background: #B584FF;

    color: white;

    border-radius: 10px;

    border: 3px solid #FFD8BE;}

/\*      HEADER   \*/

.header {

    height: 32.5em;

    background-color: #FFE3CD;}

.min-header {

    height: 5.6em;

    background-color: #FFE3CD;}

.content-header {

    height: 90px;}

.header-top {

    display: flex;

    justify-content: space-between;

    align-items: center;

    height: 90px;}

.logo {

    width: 137px;

    height: 80px;

    background-image: url("../images/logo.png");

    background-repeat: no-repeat;

    background-size: cover;}

.login {

    font-family: 'Gabriela', serif;

    font-size: 24px;}

.login:hover {

    font-size: 25px;

    color: #B584FF;}

header ul {

    margin: 0;

    padding: 0;

    list-style-type: none;}

header ul li {

    float: left;

    position: relative;}

header ul li a {

    display: block;

    font-size: 1.2rem;

    padding: 10px;}

header ul li a:hover {

    transition: 0.5s;

    color: #B584FF;}

header ul li ul {

    background-color: #FDDCBB;

    border-radius: 15px;

    border: 2px solid #B584FF;

    position: absolute;

    right: 0;

    width: 180px;

    display: none;

    z-index: 111;}

header ul li:hover ul {

    display: block;}

header ul li ul li {

    width: 100%;}

header ul li ul li a {

    padding: 10px;}

.m-sheet-1 {

    min-height: 430px;}

.m-layout-wrap {

    display: flex;

    margin: 0 auto;}

.m-layout-cell {

    border-radius: 20px;}

.m-layout-cell-1 {

    background-color: #FDDCBB;

    margin-left: auto;

    width: 50%;}

.m-image-1 {

    position: relative;

    top: 0;

    right: 0;

    width: 50%;

    height: 430px;}

.m-image {

    min-width: 700px;

    min-height: 430px;

    width: 100%;

    border-radius: 20px;

    background-image: url("../images/books1.jpg");

    background-repeat: round;

    background-size: cover;}

.m-text-cell {

    margin: 7rem 20px;

    height: 200px;

    width: 400px;}

.m-text {

    word-wrap: break-word;

    position: relative;

    padding: 20px;}

.m-text-1 {

    font-size: 3rem;

    font-family: 'Gabriela', serif;

    font-weight: 400;}

.m-text-2 {

    font-size: 25px;}

/\*       MENU   \*/

nav {

    margin: 0 auto;

    position: relative;

    width: 900px;

    height: 60px;

    background-color: #D8BEFF;

    border-radius: 20px;

    font-size: 0;}

.menu {

    display: flex;

    justify-content: space-between;

    align-items: center;

    padding: 0;}

.menu\_\_list {

    display: inline-block;

    padding: 10px 0;

    border-radius: 20px;}

.class\_\_link {

    font-size: 20px;

    padding: 18.2px 75px;

    border-radius: 20px;}

.menu :hover {

    background-color: #FFD8BE;}

.menu li.active {

    background-color: #B584FF;

    color: #fff;

    box-shadow: none;

    font-size: 24px;

    padding: 13px 0;}

.active :hover {

    background-color: #B584FF;}

/\*   MAIN    \*/

.main {

    position: relative;}

#modal\_window:target,

#modal\_window\_edit:target,

#modal\_window\_del:target {

    display: block;}

#modal\_window,

#modal\_window\_edit,

#modal\_window\_del {

    background: rgba(102, 102, 102, 0.68);

    width: 100%;

    height: 100%;

    position: fixed;

    top: 0;

    left: 0;

    display: none;

    z-index: 9999999999;}

window {

    position: absolute;

    top: 50%;

    left: 50%;

    width: 750px;

    border: 1px solid #cecece;

    padding: 50px;

    transform: translate(-50%, -50%);

    background-image: url(../images/watercolor-background.jpg);

    background-size: cover;

    background-repeat: no-repeat;}

.func {

    margin-top: 40px;}

div#main\_text a.close\_window {

    position: absolute;

    right: 10px;

    top: 4px;

    border: none;}

.btn-window {

    display: flex;

    justify-content: space-between;}

.a\_link {

    color: #000;

    text-align: center;

    font-size: 18px;

    font-style: normal;

    font-weight: 400;

    line-height: normal;

    padding: 0;}

.top {

    margin: auto;

    width: 100%;

    height: 60px;

    display: flex;

    align-items: center;}

.top\_\_link {

    margin: auto;

    margin-bottom: 1rem;}

.text\_book {

    position: absolute;

    padding-top: 10px;

    margin-right: -50%;

    left: 50%;

    transform: translate(-50%, -50%);}

.addInput {

    display: block;

    width: 100%;

    height: calc(3.25rem + 2px);

    padding: 0.375rem 0.75rem;

    font-size: 18px;

    font-weight: 400;

    line-height: 1.5;

    color: #B584FF;

    background-color: #fff;

    background-clip: padding-box;

    border: none;

    box-shadow: -5px -5px 10px 0 #FAFBFF inset, 5px 5px 10px 0px rgba(22, 27, 29, 0.23) inset;

    border-radius: 0.25rem;

    transition: border-color 0.15s ease-in-out, box-shadow 0.15s ease-in-out;}

.addInput::placeholder {

    color: #B584FF;}

.addOption {

    margin: 0 0 20px;}

.btn\_input {

    background: #B584FF;

    font-size: 20px;}

.center\_table {

    margin: auto;

    width: 94%;}

.allBook {

    width: 94%;

    margin: auto;

    font-size: 18px;

    border: 2px solid #B584FF;

    border-radius: 12px;

    border-spacing: 0;

    text-align: center;}

.allBook\_th {

    background: #B584FF;

    color: white;

    font-family: 'Gabriela', serif;

    font-style: normal;

    font-weight: 400;

    line-height: normal;

    padding: 10px 20px;}

.allBook\_th,

.allBook\_td {

    border: 1px solid #B584FF;}

th:first-child, td:first-child {

    text-align: left;}

th:first-child {

    border-top-left-radius: 10px;}

th:last-child {

    border-top-right-radius: 10px;

    border-right: none;}

td {

    padding: 10px 20px;}

tr:last-child td:first-child {

    border-radius: 0 0 0 10px;}

tr:last-child td:last-child {

    border-radius: 0 0 10px 0;}

tr td:last-child {

    border-right: none;}

.main\_content\_top {

    margin: 20px auto;

    width: 900px;

    display: flex;

    justify-content: space-between;

    align-items: center;}

.main\_content\_text {

    width: 440px;

    text-align: justify;}

.content\_text-1 {

    text-align: center;

    font-family: 'Gabriela', serif;

    font-size: 26px;

    font-style: normal;

    font-weight: 400;}

.content\_text-2 p {

    text-indent: 25px;

    padding-top: 7px;}

.main\_img img {

    border-radius: 50px;

    border: 7px solid #FDDCBB;

    width: 370px;

    height: 350px;}

.main\_rectangle {

    margin: 20px auto;

    border-radius: 50px;

    border: 15px solid #FDDCBB;

    background: #FFE3CD;

    width: 900px;

    height: 250px;}

.main\_rectangle\_content {

    margin: 0 auto;

    width: 500px;

    padding-top: 40px;}

.r\_text-1 {

    text-align: center;

    font-family: 'Gabriela', serif;

    font-size: 26px;

    font-style: normal;

    font-weight: 400;

    line-height: normal;

    letter-spacing: 0.36px;}

.r\_button-1 {

    padding-top: 15px;

    text-align: -webkit-center;}

.r\_button {

    width: 220px;

    height: 60px;

    padding: 10px 75px;

    display: flex;

    justify-content: center;

    align-items: center;

    border-radius: 20px;

    background: #B584FF;

    color: white;

    font-size: 20px;}

.content\_edit{

    height: 80vh;

    display: flex;

    font-family: 'Tenor Sans', sans-serif;

    flex-direction: column;

    justify-content: center;

    align-items: center;}

.editInput {

    display: block;

    width: 500px;

    height: 40px;

    padding-left: 20px;

    font-size: 18px;

    margin: 0 0 6px 0;

    color: #B584FF;

    background-color: #fff;

    background-clip: padding-box;

    border: none;

    box-shadow: -5px -5px 10px 0 #FAFBFF inset, 5px 5px 10px 0px rgba(22, 27, 29, 0.23) inset;

    border-radius: 0.25rem;

    transition: border-color 0.15s ease-in-out, box-shadow 0.15s ease-in-out;}

.editInput::placeholder {

    color: #B584FF;}

.btn\_edit{

    margin-top: 20px;}

/\*   FOOTER \*/

.footer {

    width: 100%;

    padding: 20px 50px 0;

    background-color: #FDDCBB;

    position: relative;

    display: flex;

    align-items: center;

    flex-direction: column;

    justify-content: flex-end;

    margin-top: 60px;}

.footer-content {

    display: flex;

    justify-content: space-between;}

.footer .footer-section {

    padding: 10px;}

.footer .about .contact span {

    display: block;

    font-size: 1.1em;

    margin-bottom: 10px;}

.footer .about .socials {

    position: relative;

    display: flex;

    justify-content: center;

    align-items: center;

    flex-wrap: wrap;}

.footer .about .socials a {

    width: 45px;

    height: 41px;

    padding-top: 5px;

    padding-right: 5px;

    text-align: center;

    display: inline-block;

    font-size: 1.3em;}

.footer .about .socials a:hover {

    color: #B584FF;

    transition: all .3s;}

.footer-menu li a:hover {

    opacity: 1;}

footer p {

    color: #a47a5d;

    text-align: center;

    font-size: 1.1em;}

admin.css - стили для панели администратора

@import url('https://fonts.googleapis.com/css2?family=Gabriela&family=Tenor+Sans&display=swap');

\*{font-family: 'Tenor Sans', sans-serif;}

html,h1,h2,h3,h4,h5,h6 {

    color: #5c5b5b;}

html::-webkit-scrollbar {

    display: none;}

body{

    background: #f6f6f6;}

a{

    text-decoration: none;

    color: inherit;}

.container {

    max-width: 1320px;

    padding: 0 10px;

    margin: 0 auto;}

.main\_container{

    min-height: calc(30em + 50px);}

/\*      HEADER   \*/

.header {

    height: 32.5em;

    background-color: #FFE3CD;}

.min-header{

    height: 5.6em;

    background-color: #FFE3CD;}

.content-header {

    height: 90px;}

.header-top {

    display: flex;

    justify-content: space-between;

    align-items: center;

    height: 90px;}

.logo {

    width: 137px;

    height: 80px;

    background-image: url("../images/logo.png");

    background-repeat: no-repeat;

    background-size: cover;}

.login {

    font-family: 'Gabriela', serif;

    font-size: 24px;}

.login:hover {

    font-size: 25px;

    color: #B584FF;}

header ul {

    margin: 0;

    padding: 0;

    list-style-type: none;}

header ul li {

    float: left;

    position: relative;}

header ul li a {

    display: block;

    font-size: 1.2rem;

    padding: 10px;}

header ul li a:hover {

    transition: 0.5s;

    color: #B584FF;}

header ul li ul {

    background-color: #FDDCBB;

    border-radius: 15px;

    border: 2px solid #B584FF;

    position: absolute;

    right: 0;

    width: 180px;

    display: none;

    z-index: 111;}

header ul li:hover ul {

    display: block;}

header ul li ul li {

    width: 100%;}

header ul li ul li a {

    padding: 10px;}

/\* FORMS \*/

.reg\_form h2{

    text-align: center;

    margin: 30px 0;}

.reg\_form input{

    height: 3.3rem;}

.reg\_form a {

    font-size: 1.2em;

    transition: all .3s;

    color: #5c5b5b;

    text-decoration: underline;

    margin-left: 60px;}

.reg\_form a:hover {

    margin-left: 55px;

    transition: all .3s;}

form div.err p {

    color: red;

    font-style: italic;

    font-size: 0.8rem;}

/\*  error msg  \*/

.add-post .err{

    padding: 0.5rem 0;

    color: #A52A2A;}

.add-post .err ul{

    list-style-type: none;

    color: #A52A2A;

    padding-left: 10px;

    font-weight: 500;}

/\*   FOOTER \*/

.footer{

    width: 100%;

    padding: 20px 50px 0 ;

    background-color: #FDDCBB;

    position: relative;

    display: flex;

    align-items: center;

    flex-direction: column;

    justify-content: flex-end;

    margin-top: 60px;}

.footer-content{

    display: flex;

    justify-content: space-between;}

.footer .footer-section{

    padding: 10px;}

.footer .about .contact span{

    display: block;

    font-size: 1.1em;

    margin-bottom: 10px;}

.footer .about .socials{

    position: relative;

    display: flex;

    justify-content: center;

    align-items: center;

    flex-wrap: wrap;}

.footer .about .socials a{

    width: 45px;

    height: 41px;

    padding-top: 5px;

    padding-right: 5px;

    text-align: center;

    display: inline-block;

    font-size: 1.3em;}

.footer .about .socials a:hover{

    color: #B584FF;

    transition: all .3s;}

.footer-menu li a:hover {

    opacity: 1;}

footer p {

    color: #a47a5d;

    text-align: center;

    font-size: 1.1em;}

authorization.css – css стили для полей регистрации и авторизации;

@import url('https://fonts.googleapis.com/css2?family=Gabriela&family=Tenor+Sans&display=swap');

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;}

body {

    background-color: #FFEEDD !important;

    font-size: 16px !important;

    line-height: 1.4;}

a {

    font-family: 'Gabriela', serif;

    color: #B584FF;

    font-weight: bold;

    text-decoration: none;}

p {

    margin: 10px 0;}

.logo {

    width: 137px;

    height: 80px;

    background-image: url("../images/logo.png");

    background-repeat: no-repeat;

    background-size: cover;}

.container {

    height: 80vh;

    display: flex;

    align-items: center;

    justify-content: center;

    font-family: 'Tenor Sans', sans-serif;

    max-width: 1320px;

    padding: 0 10px;

    margin: 0 auto;}

form {

    width: 400px;}

button {

    width: 100% ;

    background: #FFD8BE !important;

    color: #000 !important;

    font-family: 'Gabriela', serif !important;

    font-size: 20px !important;

    border: 2px solid #D8BEFF !important;}

.none {

    display: none;}

.reg\_form h2{

    font-family: 'Gabriela', serif;

    text-align: center;

    margin: 30px 0;}

.reg\_form a{

    text-decoration: none;

    font-size: 1.2em;

    margin-left: 0;}

.err{

    color: #ff0000;

    font-size: 20px;}

Index.php - код главной страницы

<!doctype html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel='stylesheet' href='css/style.css'>

<script src="https://kit.fontawesome.com/073633e180.js" crossorigin="anonymous"></script> <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM8ODewa9r" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.min.js"

integrity="sha384-BBtl+eGJRgqQAUMxJ7pMwbEyER4l1g+O15P+16Ep7Q9Q+zqX6gSbd85u4mG4QzX+" crossorigin="anonymous"></script>

<title>Libis</title>

</head>

<body>

<?php

//Подключение таблиц

require\_once("app/database/connect.php"); ?>

<!--HEADER--> <?php include("app/include/header.php"); ?>

<!--MENU-->

<nav class="nav">

<ul class="menu">

<li class="menu\_\_list active">

<a class="class\_\_link " href="Index.php">Главная</a>

</li>

<li class="menu\_\_list ">

<a class="class\_\_link " href="book.php">Книги</a>

</li>

<li class="menu\_\_list">

<a class="class\_\_link " href="authors.php">Авторы</a>

</li>

<li class="menu\_\_list">

<a class="class\_\_link " href="readers.php">Читатели</a>

</li>

</ul>

</nav>

<!--MAIN-->

<main>

<div class="container">

<div class="main\_content\_top">

<div class="main\_content\_text">

<div class="content\_text-1">Учет книг “ Libis ”</div>

<div class="content\_text-2"><p>Это онлайн программа для автоматизации учета книг в библиотеки позволяет упростить контроль выданных и возвращенных книг.</p>

<p>Для того чтобы начать работу необходимо зарегистрироваться, затем добавить автора и книгу, а затем ведите данные читателя.

После чего вы сможете выдать книгу читателю.</p></div> </div>

<div class="main\_img"><img src="./images/book2.jpg" alt="bookLib"/></div>

</div>

</div>

<?php if (!isset($\_SESSION['id'])): ?>

<div class="container">

<div class="main\_rectangle">

<div class="main\_rectangle\_content">

<div class="r\_text-1">Чтобы начать работу необходимо зарегистрироваться</div>

<div class="r\_button-1"><a class='r\_button' href='register.php'>Регистрация</a></div>

</div>

</div>

</div>

<?php endif; ?>

</main>

<!-- FOOTER -->

<?php include("app/include/footer.php"); ?>

</body>

</html>

header.php - вынесенная шапка сайта <!-- Подключение к БД -->

<?php include 'app/database/db.php'; ?>

<header class="header">

<div class="container">

<div class="content-header">

<div class="header-top">

<!-- Логотип -->

<a href="Index.php"><div class="logo"></div></a>

<!-- Login -->

<ul>

<li>

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

<a class="login" href='#'>

<i class="fa-solid fa-user"></i>

<span class="login">

<?php echo $\_SESSION['login'];?>

</span></a>

<ul>

<?php if ($\_SESSION['admin']): ?>

<li class="header-li"><a class=" " href="/../CWork/admin/admin.php">Админ панель</a></li> <?php endif; ?>

<li class="header-li"><a class=" " href="/../CWork/logout.php">Выход</a></li>

</ul>

<?php else: ?>

<a class="login" href='/../CWork/login.php'><i class="fa-regular fa-user"></i>

<span class="login">Вход</span></a>

<?php endif;?>

</li>

</ul>

</div>

</div>

</div

<div class="m-sheet-1">

<div class="m-layout-wrap">

<div class="m-layout-cell m-image-1">

<div class="m-image"></div>

</div>

<div class="m-layout-cell m-layout-cell-1">

<div class="m-text-cell">

<h1 class="m-text m-text-1">Libis</h1>

<p class="m-text m-text-2 ">Информационная система для библиотеки</p>

</div>

</div>

</div>

</div>

</header>

footer.php - вынесенный подвал сайта <footer class="footer container-fluid">

<div class="footer-content container">

<div class="row">

<div class="footer-section about footer-contact">

<div class="contact">

<span><i class="fas fa-phone"></i>&nbsp;8(888)88-88-888</span>

</div>

<div class="contact">

<span><i class="fas fa-envelope"></i>&nbsp;libis@email.com</span>

</div>

</div>

</div>

<div class="row">

<div class="footer-section about">

<div class="socials">

<a href="#"><i class="fa-brands fa-vk"></i></a>

<a href="#"><i class="fa-brands fa-telegram"></i></a>

<a href="#"><i class="fa-brands fa-whatsapp"></i></a>

</div>

</div>

</div>

</div>

<p> ©2023 Libis | Information system </p>

</footer>

users.php - авторизации и регистрации пользователей

<?php include "app/database/db.php";

$errMsg = ''; if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['button-reg'])) {

$admin = 0;

$login = trim($\_POST['login']);

$email = trim($\_POST['email']);

$passwordF = trim($\_POST['pass-first']);

$passwordS = trim($\_POST['pass-second']); if ($login === '' || $email === '' || $passwordF === '') {

$errMsg = 'Не все поля заполнены!';

} elseif (mb\_strlen($login, 'UTF-8') < 2) {

$errMsg = "Логин должен быть более 2-х символов";

} elseif ($passwordF !== $passwordS) {

$errMsg = "Пароли в обеих полях должны соответствовать!";

} else {

$existence = selectOne('users', ['email' => $email]); if ($existence['email'] === $email) {

$errMsg = "Пользователь с таким email уже зарегестрирован!";

} else {

$password = password\_hash($passwordF, PASSWORD\_DEFAULT);

$post = [

'admin' => $admin,

'username' => $login,

'email' => $email,

'password' => $password];

$id = insert('users', $post);

$user = selectOne('users', ['id' => $id]);

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

$\_SESSION['login'] = $user['username']; $\_SESSION['admin'] = $user['admin']; if ($\_SESSION['admin']) { header("Location: ../CWork/admin/admin.php");

} else { header("Location: ../CWork/Index.php");

} } }

} else {

$login = ' ';

$email = ' '; }

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['button-log'])) {

$email = trim($\_POST['email']); $pass = trim($\_POST['password']); if ($email === '' || $pass === '') {

$errMsg = 'Не все поля заполнены!'; } else {

$existence = selectOne('users', ['email' => $email]); if ($existence && password\_verify($pass, $existence['password'])) {

$\_SESSION['id'] = $existence['id'];

$\_SESSION['login'] = $existence['username']; $\_SESSION['admin'] = $existence['admin']; if ($\_SESSION['admin']) { header("Location: ../CWork/admin/admin.php");

} else { header("Location: ../CWork/Index.php");} } else {

$errMsg = "Почта либо пароль введены неверно!";}

}

} else {

$email = ' ';}

authors.php - страница со списком авторов

<!doctype html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel='stylesheet' href='css/style.css'>

<script src="https://kit.fontawesome.com/073633e180.js" crossorigin="anonymous"></script> <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM8ODewa9r" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.min.js"

integrity="sha384-BBtl+eGJRgqQAUMxJ7pMwbEyER4l1g+O15P+16Ep7Q9Q+zqX6gSbd85u4mG4QzX+" crossorigin="anonymous"></script>

<title>Libis</title>

</head>

<body>

<?php

//Подключение таблиц

$host = 'localhost';

$user = 'librarian';

$password = '12345';

$db\_name = 'library';

$link = mysqli\_connect($host, $user, $password, $db\_name); if ($link->connect\_error) { die("Ошибка: Невозможно подключиться к MySQL " . $link->connect\_error);

}

$queryTable = "SELECT \* FROM authors ORDER BY id DESC";

$resultTable = mysqli\_query($link, $queryTable) or die(mysqli\_error($link));

?>

<!--HEADER-->

<?php include("app/include/header.php"); ?>

<!--MENU-->

<nav class="nav">

<ul class="menu">

<li class="menu\_\_list">

<a class="class\_\_link start-home" href="Index.php">Главная</a>

</li>

<li class="menu\_\_list ">

<a class="class\_\_link start-books" href="book.php">Книги</a>

</li>

<li class="menu\_\_list active">

<a class="class\_\_link start-authors" href="authors.php">Авторы</a>

</li>

<li class="menu\_\_list">

<a class="class\_\_link start-readers" href="readers.php">Читатели</a>

</li>

<div class="animation start-home"></div>

</ul>

</nav>

<!--MAIN-->

<main class="main">

<!--MAIN\_TOP-->

<section class="top">

<div class="top\_\_link">

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

<a class="a\_link" href="#modal\_window">Добавить автора </a>

<?php endif; ?>

<div id="modal\_window">

<window>

<div class='func'>

<form action='add/addAuthors.php' method='post'>

<br><b class="text\_book">Добавление автора</b><br>

<input class='addInput' type='text' name='surname' placeholder='Фамилия'><br>

<input class='addInput' type='text' name='name' placeholder='Имя'><br>

<input class='addInput' type='text' name='lastname' placeholder='Отчество'><br>

<div class='btn-window'>

<a href='#close\_window' class='close\_window'><input type='button' class='btn btn\_input' value='Отменить'></a>

<a href='#close\_window' class='close\_window'><input type='submit' class='btn btn\_input' value='Сохранить'></a> </div>

</form>

</div>

</section>

<!--TABLE-->

<section class="center\_table">

<table id='authors' class='allBook'>

<thead>

<tr class='allBook-tr'>

<th class='allBook\_th'>Фамилия</th>

<th class='allBook\_th'>Имя</th>

<th class='allBook\_th'>Отчество</th>

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

<th class='allBook\_th'>Изменить</th>

<th class='allBook\_th'>Удалить</th>

<?php endif; ?>

</tr>

</thead>

<tbody>

<?php while ($row = mysqli\_fetch\_assoc($resultTable)) { echo "<tr>

<td class='allBook\_td'>" . $row['surname'] . "</td>

<td class='allBook\_td'>" . $row['name'] . "</td>

<td class='allBook\_td'>" . $row['lastname'] . "</td>" ?>

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

<?php echo "<td class='allBook\_td'><a class='btn' href='edit/editAuthors.php?id=" . $row["id"] . "'>Изменить</a></td> <td class='allBook\_td'><form action='del/delAuthors.php' method='post'>

<input type='hidden' name='id' value='" . $row['id'] . "'>

<input class='btn' type='submit' name='submit' value='Удалить'>" ?>

<?php endif; ?>

<?php echo "</form></td>

</tr>";} echo "</tbody>

</table>"; ?>

</section>

</main>

<!-- FOOTER -->

<?php include("app/include/footer.php"); ?>

</body>

</html>

book.php - страница со списком книг

<!doctype html>

<html lang="ru">

<head>

<meta charset="UTF-8"> <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link rel='stylesheet' href='css/style.css'>

<script src="https://kit.fontawesome.com/073633e180.js" crossorigin="anonymous"></script> <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM8ODewa9r" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.min.js"

integrity="sha384-BBtl+eGJRgqQAUMxJ7pMwbEyER4l1g+O15P+16Ep7Q9Q+zqX6gSbd85u4mG4QzX+" crossorigin="anonymous"></script>

<title>Libis</title>

</head>

<body>

<?php

//Подключение таблиц

require\_once("app/database/connect.php");

$host = 'localhost';

$user = 'librarian';

$password = '12345';

$db\_name = 'library';

$link = mysqli\_connect($host, $user, $password, $db\_name); if ($link->connect\_error) { die("Ошибка: Невозможно подключиться к MySQL " . $link->connect\_error);

}

$select = "SELECT \* FROM books WHERE id";

$resultTable = mysqli\_query($link, $select) or die(mysqli\_error($link));

$select\_2 = "SELECT \* FROM authors WHERE id";

//$authors = mysqli\_query($link, $select\_2) or die(mysqli\_error($link));

$authors = mysqli\_query($link, "SELECT id, name FROM authors");

?>

<!--HEADER-->

<?php include("app/include/header.php"); ?>

<!--MENU-->

<nav class="nav">

<ul class="menu">

<li class="menu\_\_list">

<a class="class\_\_link start-home" href="Index.php">Главная</a>

</li>

<li class="menu\_\_list active">

<a class="class\_\_link start-books" href="book.php">Книги</a>

</li>

<li class="menu\_\_list">

<a class="class\_\_link start-authors" href="authors.php">Авторы</a>

</li>

<li class="menu\_\_list">

<a class="class\_\_link start-readers" href="readers.php">Читатели</a>

</li>

<div class="animation start-home"></div>

</ul>

</nav>

<!--MAIN-->

<main class="main">

<!--MAIN\_TOP-->

<section class="main\_top">

<section class="top">

<div class="top\_\_link">

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

<a class="a\_link" href="#modal\_window">Добавить книгу </a>

<?php endif; ?>

<div id="modal\_window"> <window>

<b class="text\_book">Добавление книги</b>

<div class='func'>

<form action='add/addBook.php' method='post'>

<input class='addInput' type='text' name='title' placeholder='Название'><br>

<?php $authors = mysqli\_query($link, "SELECT id, name FROM authors"); ?>

<select class="addInput addOption" name="author">

<option value="">Выберите автора</option>

<?php while ($row = mysqli\_fetch\_array($authors)) { echo '<option value="' . $row["id"] . '">' . $row['surname'] . ' ' . $row["name"] . ' ' . $row["lastname"] . '</option>';

}

?>

</select>

<input class='addInput' type='number' name='year' placeholder='Год издания'><br>

<input class='addInput' type='number' name='quantity' placeholder='Количество'><br> <div class='btn-window'>

<a href='#close\_window' class='close\_window'><input type='button' class='btn btn\_input' value='Отменить'></a>

<a href='#close\_window' class='close\_window'><input type='submit' class='btn btn\_input' value='Сохранить'></a> </div>

</form>

</div>

</window>

</div>

</div>

</section>

</section>

<!--TABLE-->

<section class="center\_table">

<table id='books' class='allBook'>

<thead>

<tr class='allBook-tr'>

<th class='allBook\_th'>Название</th>

<th class='allBook\_th'>Автор</th>

<th class='allBook\_th'>Год издания</th>

<th class='allBook\_th'>Количество</th>

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

<th class='allBook\_th'>Изменить</th>

<th class='allBook\_th'>Удалить</th>

<?php endif; ?>

</tr>

</thead>

<tbody>

<?php while ($row = mysqli\_fetch\_assoc($resultTable)) { echo "<tr>

<td class='allBook\_td'>" . $row['title'] . "</td>

<td class='allBook\_td'>" . $row['id\_author'] . "</td>

<td class='allBook\_td'>" . $row['year'] . "</td>

<td class='allBook\_td'>" . $row['quantity'] . "</td> "?>

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

<?php echo"<td class='allBook\_td'><a class='btn' href='edit/edit.php?id=" . $row["id"] . "'>Изменить</a></td>

<td class='allBook\_td'><form action='del/delBook.php' method='post'> <input type='hidden' name='id' value='" . $row['id'] . "'>

<input class='btn' type='submit' name='submit' value='Удалить'>"?>

<?php endif; ?>

<?php echo "</form></td>

</tr>";} echo "</tbody>

</table>"; ?>

</section>

</main>

<!-- FOOTER --> <?php include("app/include/footer.php"); ?>

</body>

</html>

login.php - страница авторизации

<?php include"app/controllers/users.php"; ?>

<!doctype html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel='stylesheet' href='css/authorization.css'>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-

T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">

<title>Авторизация</title>

</head>

<body>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-

C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>

<a href="Index.php"><div class="logo"></div></a>

<div class="container reg\_form">

<form class="row" method="post" action="login.php">

<h2>Авторизация</h2>

<div class="mb-3 col-12 err">

<p><?=$errMsg?></p>

</div>

<div class="mb-3 col-12 ">

<label for="formGroupExampleInput" class="form-label">Эл. почта</label>

<input name="email" value="<?=$email?>" type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">

</div>

<div class="mb-3 col-12 ">

<label for="exampleInputPassword1" class="form-label">Пароль</label>

<input name="password" type="password" class="form-control" id="exampleInputPassword1">

</div>

<div class="mb-3 col-12 ">

<button type="submit" name="button-log" class="btn btn-primary">Войти</button> </div>

<div class="mb-3 col-12 ">

<p>Нет аккаунта? <a href="register.php">Зарегистрируйтесь</a></p> </div>

</form>

</div>

</body>

</html>

register.php - страница регистрации

<?php include"app/controllers/users.php"; ?>

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel='stylesheet' href='css/authorization.css'>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-

T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">

<title>Регистрация</title>

</head>

<body>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-

C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>

<a href="Index.php"><div class="logo"></div></a>

<div class="container reg\_form">

<form class="row" method="post" action="register.php">

<h2>Форма регистрации</h2>

<div class="mb-3 col-12 err">

<p><?=$errMsg?></p>

</div>

<div class="mb-3 col-12">

<label for="formGroupExampleInput" class="form-label">Ваш логин</label>

<input name="login" value="<?=$login?>" type="text" class="form-control" id="formGroupExampleInput">

</div>

<div class="mb-3 col-12 ">

<label for="exampleInputEmail1" class="form-label">Email</label>

<input name="email" value="<?=$email?>" type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">

</div>

<div class="mb-3 col-12 ">

<label for="exampleInputPassword1" class="form-label">Пароль</label>

<input name="pass-first" type="password" class="form-control" id="exampleInputPassword1"> </div>

<div class="mb-3 col-12 ">

<label for="exampleInputPassword2" class="form-label">Повторите пароль</label>

<input name="pass-second" type="password" class="form-control" id="exampleInputPassword2"> </div>

<div class="mb-3 col-12 ">

<button type="submit" class="btn btn-secondary" name="button-reg">Регистрация</button>

<div class="mb-3 col-12 ">

<p>У вас уже есть аккаунт? <a href="login.php">Войти</a></p>

</div>

</div>

</form>

</div>

</body>

</html>