

Частное учреждение
профессиональная образовательная организация
ТЕХНИКУМ «БИЗНЕС И ПРАВО»

ЗАЩИЩЕНО С ОЦЕНКОЙ

(подпись и ФИО руководителя)

«_____» _____

20____г.

КУРСОВАЯ РАБОТА

Тема	РАЗРАБОТКА WEB-УЧЕБНИКА «ПРОГРАММИРОВАНИЕ В СРЕДЕ JAVASCRIPT»
По МДК 09.01	Проектирование и разработка веб-приложений
Выполнил	Железко З.К.
Курс, группа	4 курс, группа ИС-4
Специальность	09.02.07 Информационные системы и программирование
Научный руководитель	Мартыненко В.А.

г. Белореченск
2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. АНАЛИТИЧЕСКАЯ ЧАСТЬ	4
1.1 Анализ предметной области	4
1.2 Цели и задачи выполняемой системы	4
1.3 Метод реализации процесса проектирования	5
1.4 Обоснование проектных решений	7
2. ПРОЕКТНАЯ ЧАСТЬ	18
2.1 Техническое задание	19
2.2 Установка и настройка среды разработки	20
2.3 Руководство пользователя	21
2.4 Руководство администратора	22
ЗАКЛЮЧЕНИЕ	24
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	25
ПРИЛОЖЕНИЕ	

ВВЕДЕНИЕ

Актуальность данной работы зависит от современных тенденций, под которые компании пытаются подстроиться, чтобы улучшить свой экономический статус. Можно выделить несколько популярных современных тенденций: автоматизация рутинных процессов, развитие бизнес-процессов с помощью цифровых технологий, улучшение качества обслуживания и т.д.

Разработка веб-приложения, отвечающего за сайт-учебник по языку программирования JavaScript, позволит автоматизировать процесс и упростить работу компании JsHelper, тем самым уменьшить затраты компании.

Объект исследования – веб приложение по изучению языка программирования JavaScript.

Предмет исследования - предметом исследования является учет отзывов и пользователей системы.

Цель исследования направлена на разработку веб-приложения учебника по языку программирования JavaScript. Приложение так же должно давать возможность вести учет пользователей и отзывов сайта, организация хранения, добавление, редактирование и удаление этих данных.

Задачи: провести анализ, как сейчас работает функционирует компания JsHelper, и найти способы улучшить продуктивность работы компании; на основе проведенного анализа разработать техническое задание для будущего веб-приложения; благодаря техническому заданию реализовать компоненты информационной системы; оформить руководство пользователя.

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Анализ предметной области

Компания «JsHelper», предоставляет информационные услуги для обучения пользователей языку программирования JavaScript.

Этот анализ может быть полезен для определения размера и структуры базы данных, создания удобного интерфейса для веб-учебника, а также выявления потенциальных проблем и рисков, связанных с учетом пользователей и использованием данных об отзывах. Данное веб-приложение построено по трехзвенной клиент-серверной архитектуре.

Компания «JsHelper» решила создать свою собственную информационную систему – веб-учебник для лучшего взаимодействия пользователей с их учебными пособиями. Веб-учебник будет предоставлять доступ к доступным урокам. Данное решение будет актуально для менеджеров(модераторов) компании, что позволит им эффективно выполнять свою работу.

1.2 Цели и задачи выполняемой системы

Целями веб-учебника являются:

- удобство использования сайта
- авторизация и регистрация пользователя
- распределение ролей с разными уровнями доступа
- занесение пользователей базу данных
- занесение отзывов пользователей в систему
- хранение данных
- изменение данных
- удаление данных

Основные задачи, решаемые веб-учебником, включают:

1. Информационная цель: веб учебник позволяет оперативно и удобно обращаться к урокам, которые доступны любым пользователям.

2. Контроль пользователей системы: система регистрирует всех пользователей в БД. Это позволяет предоставлять администрации данные обо всех пользователях, вести учет и регулирование доступа.

3. Контроль отзывов: система регистрирует все полученные отзывы в БД. Это позволяет предоставлять администрации данные обо всех отзывах, вести учет и статистику.

1.3 Метод реализации процесса проектирования

Проектирование веб-учебника является комплексным процессом, требующим навыков разработки информационных систем, 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 веб-учебника начинается с детального анализа и понимания требований компании «JsHelper».

2. Проектирование архитектуры веб-приложения, 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 веб-учебника по программированию с помощью языка JavaScript предлагает систематический подход, который включает анализ требований, проектирование архитектуры и функциональности системы, тестирование и отладку, а также поддержку и обучение пользователей.

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

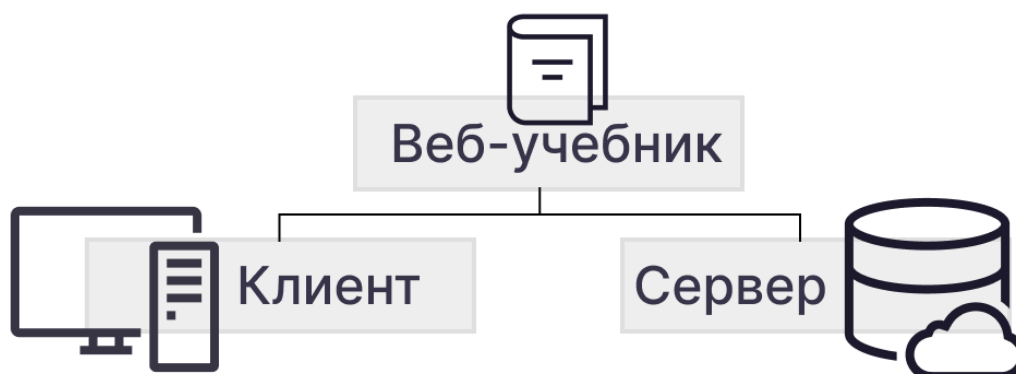


Рисунок 1.1 - Модули, которые должны быть в программе

Модули разрабатываемой системы, представленные на рисунке 1.2:

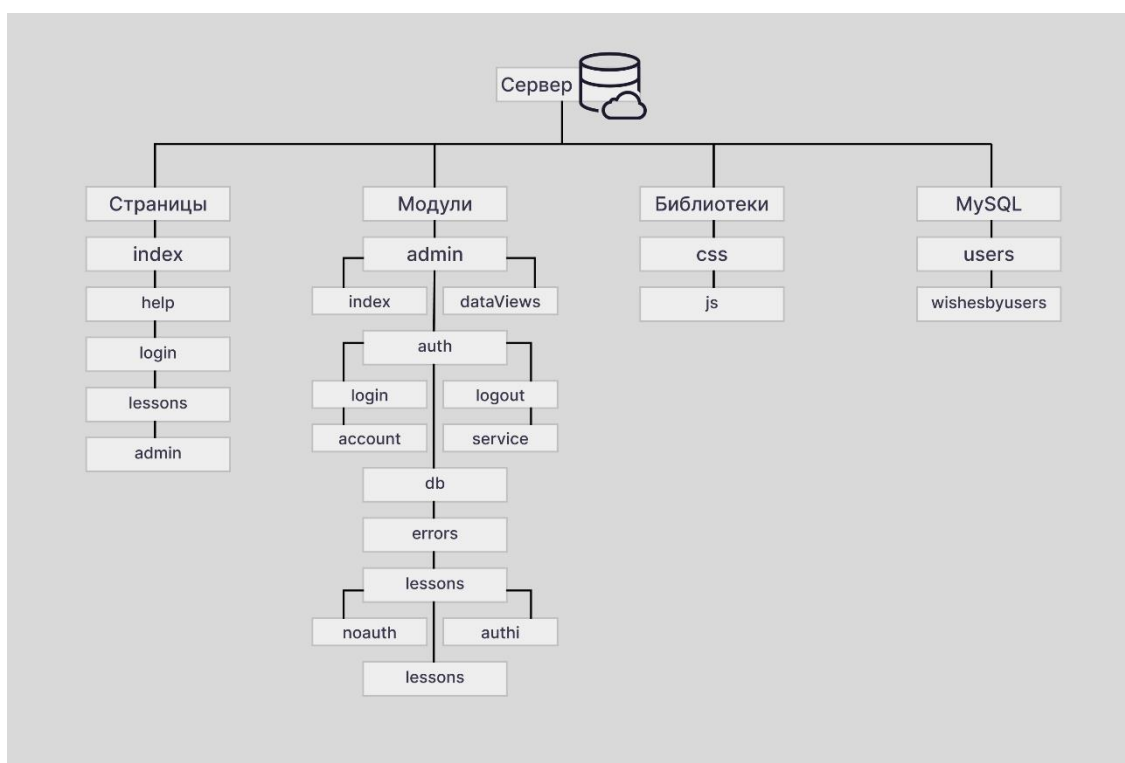


Рисунок 1.2 - Модули разрабатываемой системы

Описание модулей разрабатываемой системы:

- Страницы - статичное отображение данных:
 - Index – главная страница;
 - Help – страница помощи;

- Login – страница авторизации;
- Lessons – страница с уроками;
- Admin – страница для администрации;
- Модули – разделение логики на подсистемы:
 - Admin – подсистема администрирования:
 - Index – главная страница для администрации;
 - dataViews – подкаталог с логикой табличных данных;
 - auth – подсистема авторизации:
 - login – страница с авторизацией;
 - account – страница аккаунта;
 - logout – логика выхода из аккаунта;
 - service – подкаталог с логикой для аккаунта;
 - db – подсистема базы-данных;
 - errors – каталог ошибок;
 - lessons – подсистема уроков:
 - noauth – каталог уроков для неавторизованных пользователей;
 - authi – каталог уроков для продвинутых пользователей;
- MySQL – модуль базы-данных:
 - users – таблица с пользователями системы;
 - wishesbyusers – таблица с отзывами;

1.4 Обоснование проектных решений

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

Основные преимущества:

1. Автоматизация учебного процесса пользователей: веб-учебник позволит пользователям изучать информацию по языку программирования JavaScript используя веб-приложение.

2. Оптимизация работы библиотекарей: веб-учебник позволит администраторам автоматизировать процесс учета пользователей, что сэкономит значительное количество времени и сил.

Возможные недостатки:

1. Технические сложности: Реализация веб-приложения может быть затруднена техническими проблемами или проблемами безопасности данных.

2. Расходы на реализацию и поддержку: Внедрение веб-приложения потребует значительных затрат на покупку необходимого оборудования и программного обеспечения, а также на обучение персонала компании «JsHelper». Также необходима будет постоянная техническая поддержка веб-учебника, что может быть связано с дополнительными расходами.

Для того, чтобы сделать программу веб-учебник, 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:

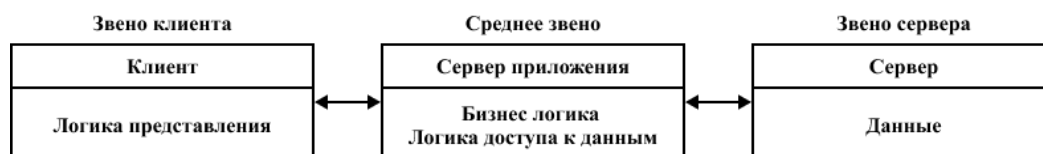


Рисунок 1.3 - Клиент-серверная архитектура

Требования к 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 веб-приложения.

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

В системе есть три группы пользователей:

- Пользователям предоставляется возможность авторизации, регистрации, просмотра базовых уроков;
- Продвинутым пользователям предоставляется возможность авторизации, просмотр базовых и продвинутых уроков;
- Администратор имеет возможность добавлять, редактировать, удалять данные и менять роли доступа у других пользователей.

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

Для того, чтобы понять, как будет работать система и каким образом она должна отвечать на действия пользователя, требуется указать сценарии использования.

Сценарий использования — это вариант использования, прецедент использования — в разработке программного обеспечения и системном проектировании это описание поведения системы, когда она взаимодействует с кем-то из внешней среды.

Система может отвечать на внешние запросы лица, а может сама выступать инициатором взаимодействия.

Другими словами, сценарий использования описывает, «кто» и «что» может сделать с рассматриваемой системой, или что система может сделать с «кем» или «чем».

Имя: Регистрация пользователя

Описание: Пользователь заходит на сайт впервые и для того, чтобы открыть доступ к продвинутому уроку ему необходимо авторизоваться

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
- Пользователь получает уведомление об успешной регистрации

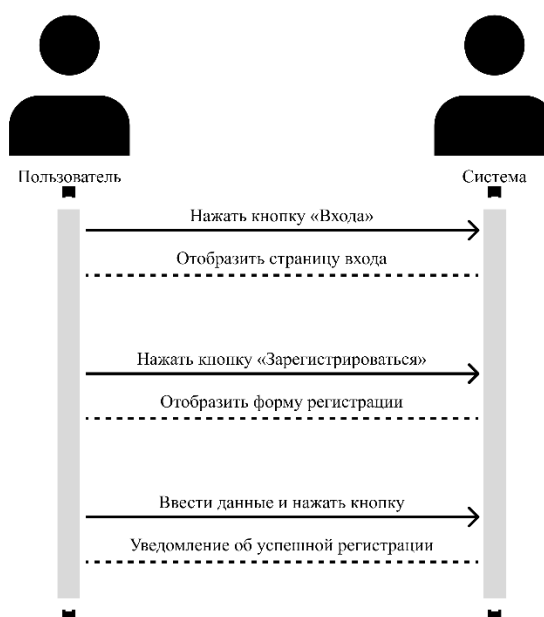


Рисунок 1.4 - Диаграмма последовательности сценария «Регистрация пользователя»

Расширения: в случае некорректного формата введенных данных, при нажатии на кнопку «Зарегистрироваться», над формой выведется сообщения-подсказка о неправильно введенных данных.

На рисунке 1.4 представлена диаграмма последовательности сценария «Регистрация пользователя»

Имя: Авторизация

Описание: для того, чтобы начать работать с продвинутыми уроками в веб-учебнике, 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:

- Перейдя на главную страницу нажать кнопку «Входа»
- Пользователь вводит данные потом нажимает на кнопку «Войти»
- Система проверяет введенную информацию о пользователе в базе данных
- Система переводит пользователя на Главную страницу

Расширения: в случае неправильно введенных данных будет выведено уведомление с ошибкой.

На рисунке 1.5 представлена диаграмма последовательности сценария «Авторизация».

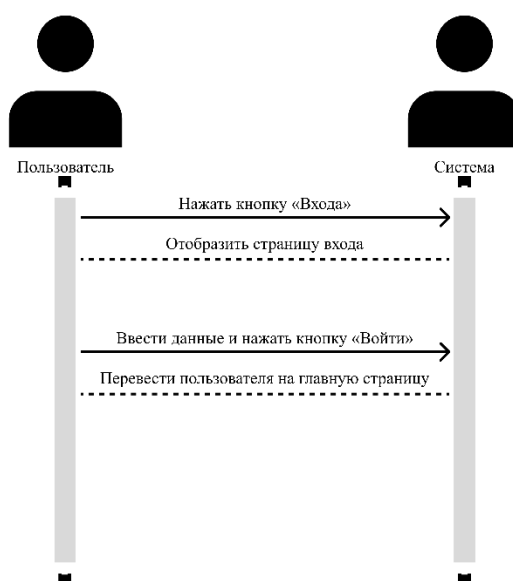


Рисунок 1.5 - Диаграмма последовательности сценария «Авторизация»

Имя: Подтверждение аккаунта пользователя

Описание: для того, чтобы активировать доступ к продвинутому урокам в веб-учебнике, пользователю необходимо подтвердить свои данные.

Действующий субъект: пользователь

Предусловие: пользователь должен быть уже зарегистрирован и авторизован в системе.

Сценарий:

- Перейдя на главную страницу нажать кнопку «Аккаунт»
- Пользователь окажется на странице своего аккаунта
- Пользователь должен нажать кнопку «Подтверждение аккаунта»
- Система отправляет форму с подтверждением
- Пользователь должен ввести данные и нажать на кнопку
- Система проверяет введенную информацию
- Система переводит пользователя на страницу его аккаунта

Расширения: в случае неправильно введенных данных будет выведено уведомление с ошибкой.

На рисунке 1.5 представлена диаграмма последовательности сценария «Подтверждение аккаунта».

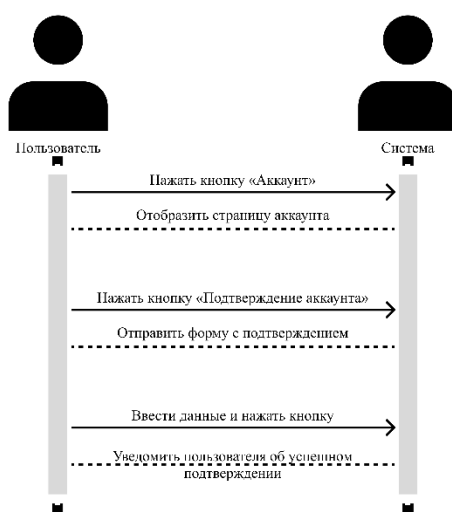


Рисунок 1.5 - Диаграмма последовательности сценария «Подтверждение аккаунта»

Имя: Добавление новых данных.

Описание: Администратор имеет возможность добавить новые данные.

Действующий субъект: Администратор.

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

Сценарий:

- Администратор выбрал нужную категорию данных на странице «Админ-панели» и нажал на кнопку «Добавить»
- Система открывает окно с формой и кнопкой отправки данных;
- Администратор указывает соответствующие данные;
- Система заполняет введенные данные в соответствующие строки;
- Администратор нажимает на кнопку отправки;
- Система обновляет систему, добавляет и выводит новые данные.

Расширения: в случае неправильно введенных данных будет выведено подсказка с ошибкой.

На рисунке 1.6 представлена диаграмма последовательности сценария «Добавления».

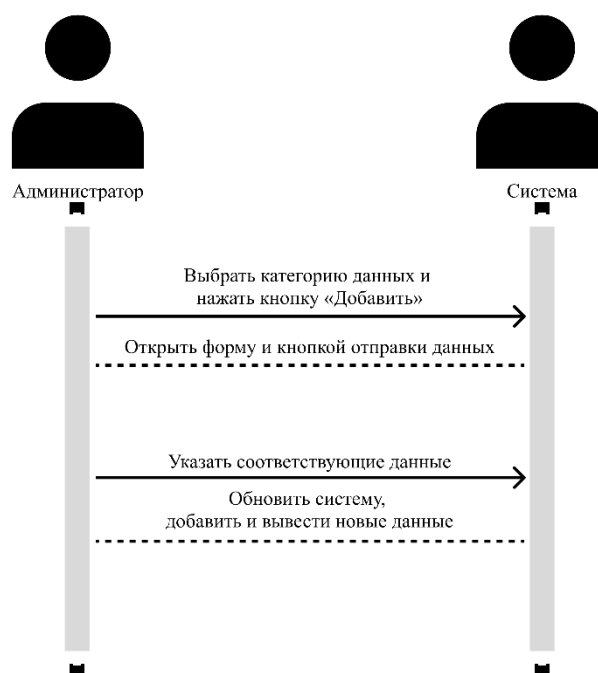


Рисунок 1.6 – Диаграмма последовательности сценария «Добавления»

Имя: Редактирование данных.

Описание: Администратор имеет возможность редактировать данные.

Действующий субъект: Администратор.

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

Сценарий:

- Администратор выбрал нужную категорию данных на странице «Админ-панели» и нажал на выбранную строку
- Система открывает окно с заполненными полями формы и кнопкой отправки данных;
- Администратор указывает соответствующие данные;
- Система заполняет введенные данные в соответствующие строки;
- Администратор нажимает на кнопку отправки;
- Система обновляет систему, добавляет и выводит новые данные.

Расширения: в случае неправильно введенных данных будет выведено подсказка с ошибкой. На рисунке 1.7 представлена диаграмма последовательности сценария «Редактирования».

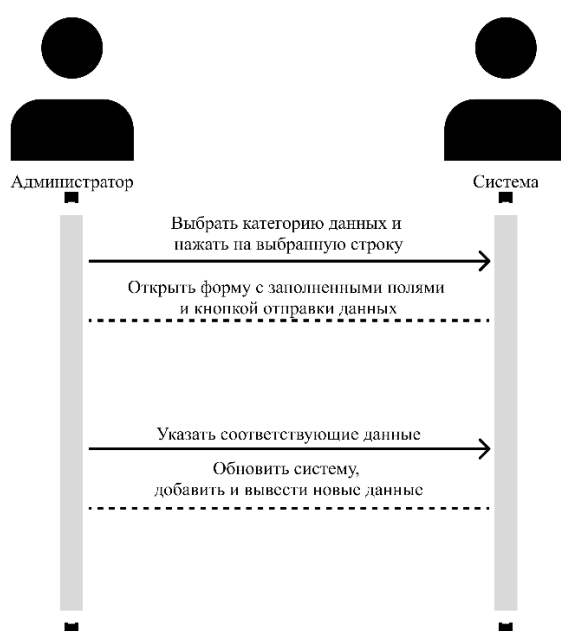


Рисунок 1.7 – Диаграмма последовательности сценария «Редактирования»

Имя: Удаление данных.

Описание: Администратор имеет возможность удалить данные.

Действующий субъект: Администратор.

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

Сценарий:

- Администратор выбрал нужную категорию данных на странице «Админ-панели» и нажал на кнопку «Удалить»
- Система обрабатывает запрос
- Система обновляет систему, удаляет и выводит обновленную таблицу.

На рисунке 1.8 представлена диаграмма последовательности сценария «Удаления».

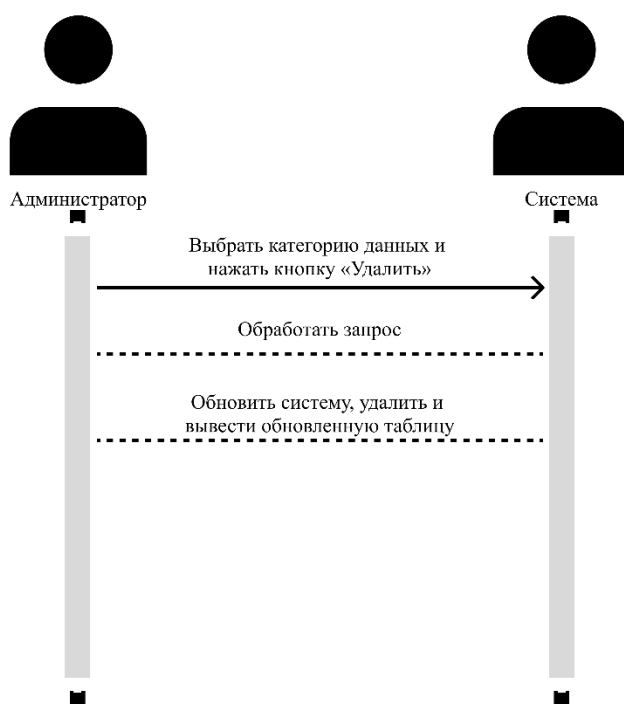


Рисунок 1.8 – Диаграмма последовательности сценария «Удаления»

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

- Обеспечить работника компании «JsHelper» персональным компьютером и дать ему доступ в интернет;
- Установить на каждый ПК любой браузер последней версии;
- Для работы так же необходим локальный веб-сервер, содержащий 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;
- отзывы;

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. users - таблица, содержащая информацию о пользователях
2. wishesbyusers - таблица, содержащая информацию об отзывах

Данные о пользователях представлены на рисунке 1.9 «Таблица данных users».

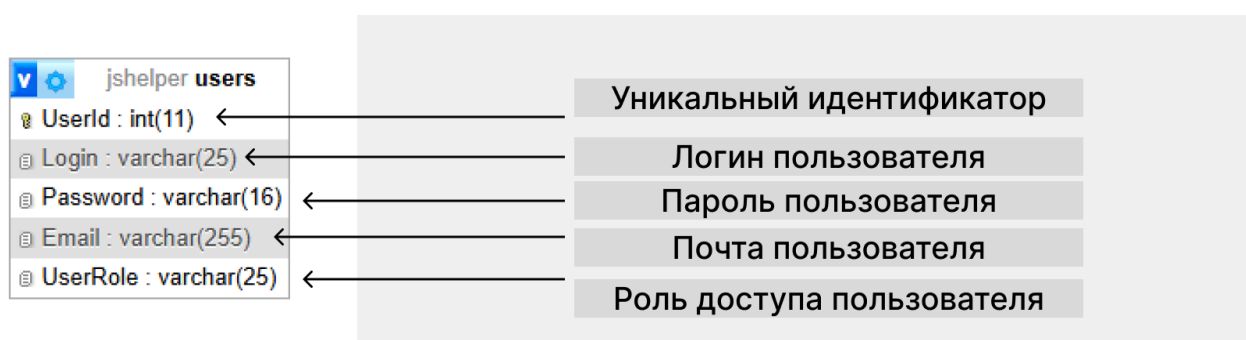


Рисунок 1.9 – Таблица данных users

Данные об отзывах представлены на рисунке 1.10 «Таблица данных wishesbyusers».



Рисунок 1.10 – Таблица данных wishesbyusers

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

На рисунке 1.11 представлена общая диаграмма состояний. При переходе пользователем в приложение пользователь будет находиться в том, состоянии, когда ему необходимо войти, чтобы начать работу с уроками и осуществлять деятельность согласно выделенным вариантам использования.



Рисунок 1.11 - Общая диаграмма состояний системы

На рисунке 1.12 показано детальное описание работы пользователя с веб-учебником. Администратор может просматривать список всех данных, добавлять, удалять, редактировать.

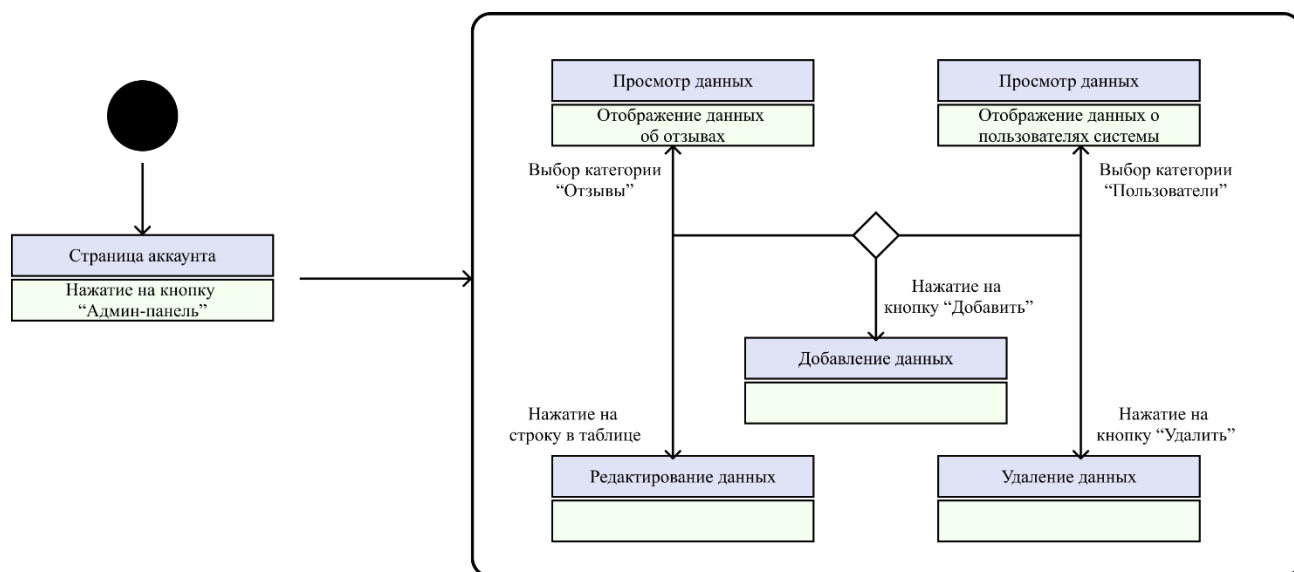


Рисунок 1.12 – Диаграмма состояний работы администратора с системой

На рисунке 1.13 изображена декомпозиция состояния «Добавление». После того как администратор успешно зашел в системы, ему необходимо выбрать категорию данных. Далее он может заполнить поля данными, нажав на Кнопку «Добавить» и после чего добавятся данные в базу данных. Кроме этого, администратору нужно правильно вводить данные, используя примеры, указанные под полями.

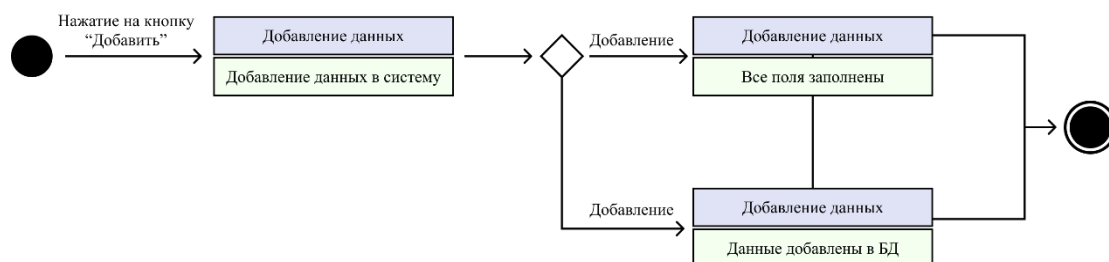


Рисунок 1.13 – Диаграмма состояний добавление данных

На рисунке 1.14 изображена декомпозиция состояния «Редактирования».

Администратор должен нажать на строку в таблице. После этого приложение откроет форму с данными. Администратору требуется обновить данные и нажать на кнопку. Далее администратор может просмотреть все изменённые данные.

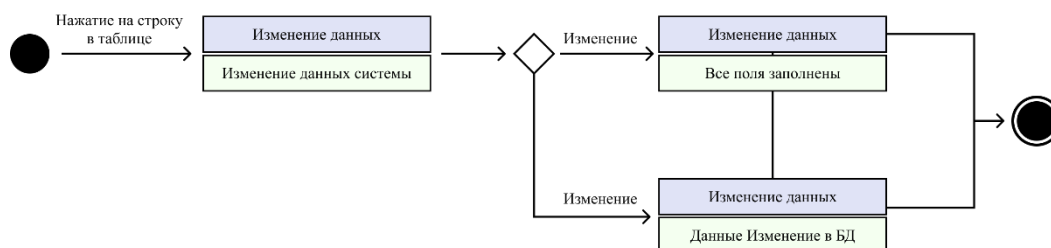


Рисунок 1.14 – Диаграмма состояний редактирования данных

На рисунке 1.15 изображена диаграмма состояния «Удаления». Администратор может удалять строки в таблице. Далее необходимо нажать на кнопку «Удалить», и программа выполнит запрос.

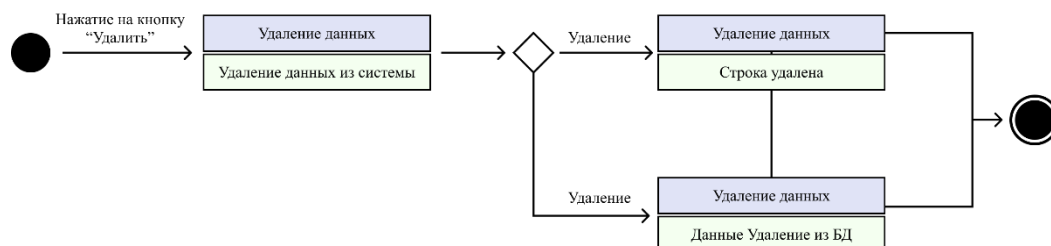


Рисунок 1.15 – Диаграмма состояний удаления данных

2 ПРОЕКТНАЯ ЧАСТЬ

2.1 Техническое задание

Информационное обеспечение является важной составляющей успешной работы веб-учебника.

Веб-приложение отвечает за сбор, хранение, обработку и информацию о наличии пользователей в системе, а также о доступные отзывы сайта. Кроме этого, веб-учебник отвечает за хранение различных уроков на языке программирования JavaScript.

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.

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-учебник должен обеспечивать защиту данных пользователей, а также предотвращать несанкционированный доступ к системе.

Разработка уникального и эффективного дизайна интерфейса для веб-приложения, требует глубокого понимания потребностей пользователей и их взаимодействия с сайтом.

Мы стремимся к минимализму и функциональности, чтобы пользователи могли сосредоточиться на самом процессе учения, а не на излишне сложном интерфейсе. Четкая структура, понятные и легко доступные функции, а также понятные инструкции помогают пользователям максимально быстро овладеть нашим веб-учебником и начать использовать.

Основные компоненты интерфейса включают в себя:

1. Главная страница, на которой пользователь может получать общую информацию о сайте.
2. Страницы уроков, включающих в себя различную информацию об языке программирования JavaScript.
3. 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 логин, пароль и адрес электронной.

2. Авторизация пользователей:

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

3. Учет пользователей:

Администраторы должны иметь возможность просматривать и взаимодействовать со списком всех пользователей системы.

4. Учет отзывов:

Администраторы должны иметь возможность просматривать и взаимодействовать со списком всех отзывов.

2.2 Установка и настройка среды разработки

Для установки и настройки среды разработки для веб-учебника по программированию на языке JavaScript 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 учебника. 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 или Windows 11/Linux;

Сервер: MS SQL Server 2022;

СУБД: MSSQL для MS SQL Server;

XAMPP или Open Server panel.

2.3 Руководство пользователя

Первое, что является единым для всех пользователей – авторизация. Чтобы авторизоваться необходимо на главной странице в левом верхнем углу нажать на кнопку «Вход».

Для того чтобы авторизоваться необходимо пользователю в окне авторизации заполнить две строки: логин и пароль, затем нажать на кнопку входа. Если же у пользователя нет своей учетной записи, он может создать ее. Для этого нужно в окне авторизации над кнопкой входа, нажать на ссылку «Регистрация» после чего форма поменяет свой вид. В новой форме необходимо ввести свои данные и также нажать соответствующую кнопку. Если все данные будут заполнены правильно, то пользователя уведомит об успешной регистрации, если же в заполнении данных была допущена ошибка, то пользователю над формой выведется соответствующие подсказки, после чего необходимо будет заполнить данные по-новому.

Главная страница включает в себя информацию о самом сайте.

Незарегистрированному пользователю можно только просмотреть информацию сайта, посмотреть остальные страницы или доступные открытые уроки. Доступ же к продвинутым урокам будет заблокирован. Если же пользователь зарегистрирован, то у него появляется возможность просмотреть свой аккаунт, подтвердить свой аккаунт, что приведет к выдаче доступа для продвинутых уроков.

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

2.4 Руководство администратора

Для того, чтобы получить данную роль доступа, требуется согласовать решение с создателями сайта, которые в свою очередь выдадут доступ пользователю.

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

При авторизации если нажать на кнопку войти, система проверит что вы являетесь администратором, после этого на странице аккаунта будет доступна

кнопка «Админ-панель», которая перекинет вас на страницу с категориями данных.

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

Так же администраторам доступна возможность управлять информацией об отзывах, а именно создавать новые отзывы, изменять их или удалять вовсе.

ЗАКЛЮЧЕНИЕ

В результате выполнения было создано веб-приложение, выполняющее роль веб-учебника, он позволяет вносить и хранить данных о пользователях и отзывах. Кроме этого, веб-учебник является целым учебным пособием по изучению языка программирования JavaScript.

Было проведено обширное исследование и анализ современных технологий и подходов к созданию таких приложений. При разработке этой работы удалось создать удобное и функциональное веб-приложение для пользователей, способствующие оптимизации работы компании «JsHelper» 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.

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

В целом, разработка веб-учебника по языку программирования JavaScript является значимым шагом в современной автоматизации данных. Созданный продукт обеспечивает эффективное и безопасное функционирование системы, оптимизирует работу веб-учебника и обеспечивает удобство пользователей. Разработка данной системы имеет большой потенциал для применения в различных учебных учреждениях, способствуя совершенствованию и модернизации учебного сервиса.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Васильев, Алексей. Программирование на PHP в примерах и задачах / Алексей Ва-сильев. — Москва: Эксмо, 2021. — 352 с.
- 2 Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, и др.. - М.: СПб: Питер, 2021. - 368 с.
- 3 Дакетт, Джон. PHP и MYSQL. Серверная веб-разработка / Джон Дакетт — Москва: Эксмо, 2023. — 688 с.
- 4 Колисниченко Д.Н. PHP и MySQL. Разработка веб-приложений; БХВ-Петербург - М., 2022. - 344 с.
- 5 Маркин А. В. Программирование на SQL: учебное пособие для среднего профессионального образования/ А. В Меркин. - М.: издательство юрайт 2023. 435 с.
- 6 Полуэктова, Н. Р. Разработка веб-приложений: учебное пособие для вузов / Н. Р. Полуэктова. — М.: Издательство Юрайт, 2023. — 204 с.
- 7 Васильев А.Н. Программирование на PHP в примерах и задачах. - 5 изд. - М: Эксмо, 2021. - 352 с.
- 8 Официальный мануал // Php URL: <https://www.php.net/manual/ru/langref.php> (дата обращения: 28.11.2024).

ПРИЛОЖЕНИЕ

КодCssстилей

```
/*Главныйшрифт*/
@importurl('https://fonts.googleapis.com/css2?family=Inter:ital,opsz,wght@0,14..32,100..900;1,14..32,100..900&display=swap');

/*Кириллицастилизированныйшрифт*/
@importurl('https://fonts.googleapis.com/css2?family=Inter:ital,opsz,wght@0,14..32,100..900;1,14..32,100..900&family=Press+Start+2P&display=swap');

html,body{
  scroll-behavior:smooth;
}
/*styles.css*/
body{
  margin:0;
  font-family:"Inter",serif;
  color:#1a1a1a;
}

.container{
  display:flex;
  flex-direction:column;
  min-height:100vh;
  background:#f7f7f7;
}

.top-bar{
  top:0;
  display:flex;
  justify-content:space-between;
  align-items:center;
  padding:10px20px;
  background:#f0f0f0;
  border-bottom:1pxsolid#ddd;
  position:sticky;
  z-index:2;
}

.small-text{
  font-size:12px;
  color:#888;
  font-family:"PressStart2P",serif;
  text-decoration:underline;
  text-decoration-color:#9B98B7;
}

.menu{
  display:flex;
  gap:10px;
}

.menu-btn{
  background:#fff;
  border:1pxsolid#ccc;
  border-radius:4px;
  padding:8px15px;
  margin-left:42px;
  font-size:14px;
  cursor:pointer;
  transition:background0.2s;
  border-color:#9B98B7;
  border-radius:21px;
  font-size:20px;
  border-width:2px;
}

.menu-btn:hover{
  background:#f0f0f0;
}

.content{
  display:flex;
  justify-content:center;
  align-items:center;
```

```

flex:1;
padding:20px;
background-image:url('/image/svg/backtriangle.svg');
background-repeat:no-repeat;
background-position:right;
}

.left{
padding-right:16%;
}

.topics{
list-style:none;
padding:0;
margin:0;
font-size:70px;
font-weight:300;
color:#080523;
}

.topicsli{padding:0px;}
.topicslia{margin:0px;display:block;width:100%;height:100%;text-decoration:none;color:unset;}

.topicsli{
margin-bottom:15px;
transition:all0.3sease;
}

.topicsli:hover{
cursor:pointer;
}
.topicsli:nth-child(1):hover{
background-color:rgb(246,213,251);
}
.topicsli:nth-child(2):hover{
background-color:rgb(251,213,217);
}
.topicsli:nth-child(3):hover{
background-color:rgb(213,251,228);
}
.topicsli:nth-child(4):hover{
background-color:rgb(213,216,251);
}

.right{
display:contents;
flex-direction:column;
align-items:flex-start;
justify-content:center;
flex:1;
position:relative;
padding-right:255px;
}

.curly-brace{
font-size:555px;
color:#201D3F;
padding-right:2%;
}

.logo{
font-size:70px;
font-weight:bold;
color:#201D3F;
margin:0;
font-family:"PressStart2P",serif;
}
.small-texta{
font-size:12px;
color:#888;
font-family:"PressStart2P",serif;
text-decoration:underline;
text-decoration-color:#9B98B7;
}
.cont-se-b{

```

```
background-image:url('/image/svg/2backreverttriangle.svg');
background-repeat:no-repeat;
background-position:right;
}
.container-second{
max-width:1200px;
margin:0auto;
padding:20px;
}

.top-section{
display:flex;
gap:20px;
margin-bottom:20px;
}

.card-one,.card-two{
background-color:#fff;
border:1pxsolid#ccc;
padding:20px;
flex:1;
border-radius:5px;
transition:all0.5sease;
}

.card-oneh2,.card-twoh2{
margin-top:0;
font-size:1.5rem;
color:#333;
}

.card-onep,.card-twop{
font-size:1rem;
color:#666;
}

.card-one:hover,.card-two:hover,.bottom-section:hover{
cursor:default;
background-color:rgba(117,117,117,0.065);
box-shadow:2px2px5pxrgba(0,0,0,0.1);
}

.bottom-section{
background-color:#fff;
border:1pxsolid#ccc;
padding:20px;
border-radius:5px;
transition:all0.5sease;
}

.bottom-sectionh2{
font-size:1.5rem;
margin-top:0;
}

.bottom-sectionp{
font-size:1rem;
color:#666;
}

.footer-container{
display:flex;
padding:50px;
background-color:#f2f2f2;
border-top:1pxsolid#ddd;
justify-content:center;
}

.footer-sidebar{
flex:00150px;
padding-right:30%;
}

.footer-sidebarula{
```

```
list-style:none;
text-decoration:none;
margin:0;
}

.footer-sidebarli:nth-child(1){
margin-top:0px;
}

.footer-sidebarli{
font-size:1rem;
color:#0a0a23;
margin-bottom:50px;
transition:all0.3sease;
}

.footer-sidebarli:hover{
background-color:#9B98B7;
color:white;
}

.footer-content{
flex:1;
border-radius:5px;
width:841px;
}

.b-f-sidebarp{
font-size:1.2rem;
color:#0a0a23;
font-weight:600;
margin-top:0px;
}

.form-input{
font-family:"Inter",sans-serif;
display:block;
width:100%;
padding:10px;
margin-bottom:10px;
font-size:1rem;
border:1pxsolid#ccc;
border-radius:4px;
box-sizing:border-box;
resize:none;
transition:all0.3sease;
}

.form-input:focus{
border:2pxsolid#0a0a23;
outline:none;
}

.form-input:hover{
border:2pxsolid#0a0a23;
outline:none;
}

.submit-button{
background-color:#0a0a23;
color:#fff;
border:none;
padding:10px20px;
font-size:1rem;
cursor:pointer;
border-radius:4px;
margin-right:10px;
width:100%;
transition:all0.4sease;
}

.submit-button:hover{
background-color:#0e0e63;
}

.reset-button{
```

```
background-color:#ccc;
color:#fff;
border:none;
padding:10px20px;
font-size:1rem;
cursor:pointer;
border-radius:4px;
width:100%;
margin-top:5px;
transition:all0.4sease;
}
```

```
.reset-button:hover{
background-color:#b3b3b3;
}
```

```
.footer-note{
text-align:center;
padding:10px;
background-color:#f2f2f2;
border-top:1pxsolid#ddd;
font-family:"PressStart2P",serif;
}
```

```
.footer-notep{
margin:0;
font-size:0.8rem;
color:#0a0a23;
letter-spacing:0.1rem;
text-transform:uppercase;
}
```

```
.notifi-stick{
position:sticky;
}
```

```
.notifi-account-reg-apply{
color:white;
```

```
position:fixed;
top:10%;
left:3%;
```

```
background:green;
padding:10px;
border-radius:5px;
z-index:1;
```

```
opacity:0;
transition:opacity1sease-in-out;
}
```

```
.notifi-account-reg-undo{
color:white;
top:30;
left:30;
background:rgb(128,0,15);
padding:10px;
border-radius:5px;
z-index:1;
```

```
position:sticky;
opacity:0;
transition:opacity1sease-in-out;
}
```

```
.notifi-account-reg-apply,.notifi-account-reg-undo{
opacity:0;
transition:opacity1sease-in-out;
}
```

```
.notifi-account-reg-apply.visible,.notifi-account-reg-undo.visible{
opacity:1;
}
```

Cssстилидлястраницы«Аккаунт»

/*Главныйшрифт*/

@importurl('https://fonts.googleapis.com/css2?family=Inter:ital,opsz,wght@0,14..32,100..900;1,14..32,100..900&display=swap');

/*Кириллицастилизованныйшрифт*/

@importurl('https://fonts.googleapis.com/css2?family=Inter:ital,opsz,wght@0,14..32,100..900;1,14..32,100..900&family=Press+Start+2P&display=swap');

```
html,body{
scroll-behavior:smooth;
}
/*styles.css*/
body{
margin:0;
font-family:"Inter",serif;
color:#1a1a1a;
}
.top-bar{
top:0;
display:flex;
justify-content:space-between;
align-items:center;
padding:10px20px;
background:#f0f0f0;
border-bottom:1pxsolid#ddd;
position:sticky;
z-index:2;
}

.small-text{
font-size:12px;
color:#888;
font-family:"PressStart2P",serif;
text-decoration:underline;
text-decoration-color:#9B98B7;
}

.menu{
display:flex;
gap:10px;
}
.small-texta{
font-size:12px;
color:#888;
font-family:"PressStart2P",serif;
text-decoration:underline;
text-decoration-color:#9B98B7;
}
.menu-btn{
background:#fff;
border:1pxsolid#ccc;
border-radius:4px;
padding:8px15px;
margin-left:42px;
font-size:14px;
cursor:pointer;
transition:background0.2s;
border-color:#9B98B7;
border-radius:21px;
font-size:20px;
border-width:2px;
}

.menu-btn:hover{
background:#f0f0f0;
}

.user-info{
background-color:#fff;
border-radius:8px;
box-shadow:0010pxrgba(0,0,0,0.1);
padding:20px;
width:400px;
text-align:left;
```



```

}
.user-infoh2{
margin-top:0;
color:#281864;
}
.user-infotable{
width:100%;
border-collapse:collapse;
margin-bottom:20px;
}
.user-infoth,.user-infotd{
border:1pxsolid#ddd;
padding:10px;
text-align:left;
}
.user-infoth{
background-color:#f2f2f2;
}
main{
display:flex;
justify-content:center;
}

.section-1,.section-2{
padding:10px;
}

```

Код главной страницы сайта

```

<?php
session_start();

?>
<!DOCTYPEhtml>
<htmllang="ru">
<head>
<metacharset="UTF-8">
<metaname="viewport"content="width=device-width,initial-scale=1.0">
<title>Помощь</title>
<linkrel="stylesheet"href="/jshelper/static/css/lessons/vedenie.css">
<style>
body{
font-family:Arial,sans-serif;
margin:0;
padding:0;
background-color:#f4f4f4;
color:#333;
}

.container{
max-width:800px;
margin:50pxauto;
background:#fff;
padding:20px;
border-radius:8px;
box-shadow:04px8pxrgba(0,0,0,0.1);
}

h1{
text-align:center;
color:#2c3e50;
}

.faq-item{
margin-bottom:20px;
border:1pxsolid#ddd;
border-radius:5px;
overflow:hidden;
}

.faq-header{
background:#3498db;
color:#fff;
padding:15px;
cursor:pointer;
font-size:18px;
}

```

```
display:flex;
justify-content:space-between;
align-items:center;
}
```

```
.faq-header:hover{
background:#2980b9;
}
```

```
.faq-content{
padding:15px;
font-size:16px;
background-color:#f9f9f9;
}
```

```
.toggle-icon{
font-size:20px;
}
```

```
</style>
```

```
</head>
```

```
<headerclass="top-bar">
```

```
<spanclass="small-text"><atarget="_blank"href="https://www.php.net/"></a></span>
```

```
<divclass="menu">
```

```
<buttonclass="menu-btn"onclick="location.href='http://localhost/jshelper/help.php'">Помощь</button>
```

```
<?php
```

```
if(!isset($_SESSION["auth-header-login"])){
```

```
?>
```

```
<buttonclass="menu-btn"onclick="location.href='http://localhost/jshelper/auth/login.php'">Войти</button>
```

```
<?php
```

```
}else{
```

```
?>
```

```
<buttonclass="menu-btn"onclick="location.href='http://localhost/jshelper/auth/account.php'">Аккаунт</button>
```

```
<buttonclass="menu-btn"onclick="location.href='http://localhost/jshelper/auth/logout.php'">Выход</button>
```

```
<?php
```

```
}
```

```
?>
```

```
<buttonclass="menu-btn"id="up"onclick="location.href='http://localhost/jshelper'">
```

```
<svgviewBox="001616"width="20px"fill="none"xmlns="http://www.w3.org/2000/svg"><gid="SVGRepo_bgCarrier"stroke-
```

```
width="0"></g><gid="SVGRepo_tracerCarrier"stroke-linecap="round"stroke-
```

```
linejoin="round"></g><gid="SVGRepo_iconCarrier"><pathd="M16V15H6V11C69.895436.89543989C9.104579109.895431011V15
```

```
H15V6L80L16Z"fill="#000000"></path></g></svg>
```

```
</button>
```

```
</div>
```

```
</header>
```

```
<body>
```

```
<divclass="container">
```

```
<h1>Часто задаваемые вопросы</h1>
```

```
<?php
```

```
if(isset($_SESSION["auth-header"])){
```

```
if($_SESSION["auth-header-role"]=="Продвинутый"OR$_SESSION["auth-header-role"]=="Админ"){?>
```

```
<divclass="faq-item">
```

```
<divclass="faq-header">
```

```
<span>Какдобавитьданные?</span>
```

```
<spanclass="toggle-icon">+</span>
```

```
</div>
```

```
<divclass="faq-content">
```

```
<p>Длядобавлениеданныхвыполнитеследующиешаги:</p>
```

```
<ol>
```

```
<li>Перейдитенастраницу"Аккаунт"инажмитенакнопку"Админ-панель".</li>
```

```
<li>Выберитенужнуюкатегорию</li>
```

```
<li>Нажмитенакнопку"Добавить"</li>
```

```
<li>Заполнитенеобходимыеполя(логин,email,пароль).</li>
```

```
<li>Нажмитенакнопку.</li>
```

```
</ol>
```

```
</div>
```

```
</div>
```

```
<divclass="faq-item">
```

```
<divclass="faq-header">
```

```
<span>Какредактироватьданные?</span>
```

```
<spanclass="toggle-icon">+</span>
```

```
</div>
```

```
<divclass="faq-content">
```

```
<p>Дляредактированияданныхвыполнитеследующиешаги:</p>
```

```
<ol>
```

```
<li>Перейдитенастраницу"Аккаунт"инажмитенакнопку"Админ-панель".</li>
```

```

<li>Выберите нужную категорию</li>
<li>Нажмите на строку в таблице</li>
<li>Заполните необходимые поля (логин, email, пароль).</li>
<li>Нажмите на кнопку.</li>
</ol>
</div>
<div class="faq-item">
<div class="faq-header">
<span>Как редактировать данные?</span>
<span class="toggle-icon">+</span>
</div>
<div class="faq-content">
<p>Для удаления данных выполните следующие шаги:</p>
<ol>
<li>Перейдите на страницу "Аккаунт" и нажмите на кнопку "Админ-панель".</li>
<li>Выберите нужную категорию</li>
<li>Нажмите на "Удалить" в таблице</li>
</ol>
</div>
</div>
<div class="faq-item">
<div class="faq-header">
<span>Как зарегистрироваться на сайте?</span>
<span class="toggle-icon">+</span>
</div>
<div class="faq-content">
<p>Для регистрации на сайте выполните следующие шаги:</p>
<ol>
<li>Перейдите на главную страницу и нажмите на кнопку "Вход".</li>
<li>Заполните необходимые поля (логин, email, пароль).</li>
<li>Нажмите на кнопку.</li>
</ol>
</div>
</div>
<div class="faq-item">
<div class="faq-header">
<span>Как активировать аккаунт?</span>
<span class="toggle-icon">+</span>
</div>
<div class="faq-content">
<p>Если вы хотите получить доступ к продвинутым урокам:</p>
<ol>
<li>На странице аккаунта нажмите "Подтвердить аккаунт".</li>
<li>Введите ваш email, спомощью капчи.</li>
<li>Нажмите на кнопку.</li>
</ol>
</div>
</div>
<div class="faq-item">
<div class="faq-header">
<span>Как посмотреть уроки?</span>
<span class="toggle-icon">+</span>
</div>
<div class="faq-content">
<p>Для просмотра уроков:</p>
<ul>
<li>Используйте кнопку "Уроки".</li>
<li>Или перейдите по адресу <a href="http://localhost/jshelper/lessons/lessons-hub.php">Ссылка</a>.</li>
</ul>
</div>
</div>
<div class="faq-item">
<div class="faq-header">
<span>Как связаться с поддержкой?</span>
<span class="toggle-icon">+</span>
</div>
<div class="faq-content">

```

```
<p>Вы можете связаться с нами через:</p>
<ul>
<li>Email:<a href="mailto:support@example.com">support@jshelper.com</a></li>
<li>Телефон:<a href="tel:+1234567890">+1(123)456-7890</a></li>
</ul>
</div>
</div>
</div>
</body>
</html>
```

Страница аккаунт

```
<?php
session_start();
require_once './service/userlogger.php';

if(isset($_SESSION["auth-header-login"])){ ?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Мой аккаунт</title>
<link rel="stylesheet" type="text/css" href="../static/css/account/account.css">
</head>
<body>
<header class="top-bar">
<span class="small-text"><a target="_blank" href="https://www.php.net/"></a></span>
<div class="menu">
<?php
if(strcasecmp($_SESSION["auth-header-role"],"Админ")==0){ ?>
<button class="menu-btn" onclick="location.href='http://localhost/jshelper/admin/'">Админ-панель</button>
<?php ?>
<button class="menu-btn" onclick="location.href='http://localhost/jshelper/lessons/lessons-hub.php'">Уроки</button>
<button class="menu-btn" onclick="location.href='logout.php'">Выход</button>
<button class="menu-btn" id="up" onclick="location.href='http://localhost/jshelper/'">
<svg viewbox="0 0 1616" width="20px" fill="none" xmlns="http://www.w3.org/2000/svg"><g id="SVGRepo_bgCarrier" stroke-
width="0"></g><g id="SVGRepo_tracerCarrier" stroke-linecap="round" stroke-
linejoin="round"></g><g id="SVGRepo_iconCarrier"><path d="M16V15H6V11C69.895436.89543989C9.104579109.895431011V15
H15V6L80L16Z" fill="#000000"></path></g></svg>
</button>
</div>
</header>
<main>
<div class="user-info">
<h2>Информация о пользователе</h2>
<table>
<tr>
<th colspan="2">Данные входа</th>
</tr>
<tr>
<td><strong>Последний вход:</strong></td>
<td><?=$_SESSION["auth-header-time"];?></td>
</tr>
</table>
<table>
<tr>
<th colspan="2">Данные аккаунта</th>
</tr>
<tr>
<td><strong>Логин:</strong></td>
<td><?=$_SESSION["auth-header-login"];?></td>
</tr>
<tr>
<td><strong>Почта:</strong></td>
<td><?=$_SESSION["auth-header-mail"];?></td>
</tr>
<tr>
<td><strong>Пароль:</strong></td>
<td><?=$_SESSION["auth-header-pass"];?></td>
</tr>
<tr>
<td><strong>Статус аккаунта:</strong></td>
<td><?=$_SESSION["auth-header-role"];?></td>
</tr>
```

```

</table>
<p>Действия с аккаунтом:</p>
<a href="service/updatepassword.php">Обновить пароль</a>
<br><br>
<?php
if(strcasecmp($_SESSION["auth-header-role"],"Пользователь")==0){?>
<a href="service/confirmaccount.php">Подтвердить аккаунт</a>
<?php}else{?>
<a href="service/confirmaccount.php" style="pointer-events:none;cursor:default;opacity:0.6;color:grey;text-decoration:none;">Подтвердить аккаунт</a>
<?php}?>
</div>
</main>
</body>
</html>
<?php}else{?>
<a href="http://localhost/jshelper/auth/login.php">Войти</a>
<a href="http://localhost/jshelper">Вернуться на главную</a>

<p>Вы не авторизованы!</p>
<?php}?>

```

Страница авторизации

```

<?php
session_start();
require_once '../db/db.php';
$message="";

if($_SERVER['REQUEST_METHOD']=='POST'){

    $login=trim($_POST['reglogin']);
    $password=trim($_POST['regpassword']);
    $email=trim($_POST['regemail']);

    try{
        $stmt=$pdo->prepare("SELECT * FROM Users WHERE Login=:login AND Password=:password");
        $stmt->execute(['login'=>$login,'password'=>$password]);
        $user=$stmt->fetch();

        if(empty($user)){
            $message="Ошибка: такой пользователь уже существует";
        }else{
            $stmt=$pdo->prepare("INSERT INTO Users(Login,Password,Email,UserRole) VALUES(:login,:password,:email,'Пользователь')");
            $stmt->bindParam(':login',$login,PDO::PARAM_STR);
            $stmt->bindParam(':password',$password,PDO::PARAM_STR);
            $stmt->bindParam(':email',$email,PDO::PARAM_STR);

            if($stmt->execute()){
                $message="Вы создали аккаунт";
            }else{
                $message="Ошибка регистрации";
            }
        }
    }catch(PDOException $e){
        $message="Ошибка: ".$e->getMessage();
    }
    elseif(isset($_GET['checkLogin'])){
        $login=trim($_GET['loginInput']);
        $password=trim($_GET['passInput']);
        try{
            $stmt=$pdo->prepare("SELECT * FROM Users WHERE Login=:login AND Password=:password");
            $stmt->execute(['login'=>$login,'password'=>$password]);
            $userlogin=$stmt->fetch();

            if(empty($userlogin)){
                $_SESSION["auth-header"]=end($userlogin);

                $_SESSION["auth-header-login"]=$userlogin['Login'];
                $_SESSION["auth-header-mail"]=$userlogin['Email'];
                $_SESSION["auth-header-pass"]=$userlogin['Password'];
                $_SESSION["auth-header-role"]=$userlogin['UserRole'];
                $_SESSION["auth-header-id"]=$userlogin['UserId'];

                $_SESSION["auth-header-time"]=date("Y.m.d\H:i:s");
            }
        }
    }
}

```

```

echo'<scripttype="text/javascript">';
echo'window.location.href="http://localhost/jshelper";';
echo'</script>';
}else{
$message="Ошибка:Неудалосьнайтипользователя";
}
}catch(PDOException$e){
$message="Ошибка:".$e->getMessage();
}
}
require_once'../db/duplicate.php';
?>
<!DOCTYPEhtml>
<html>
<head>
<metacharset="UTF-8">
<metaname="viewport"content="width=device-width,initial-scale=1.0">
<linkrel="stylesheet"type="text/css"href=" ../static/css/auth/login.css">
<linkhref="https://fonts.googleapis.com/css2?family=Jost:wght@500&display=swap"rel="stylesheet">
<title>Авторизация</title>
</head>
<body>
<?php
if($message){
if(str_contains($message,"Вы")){
echo"<p><strongclass='notifi-account-reg-apply'>$message</strong></p>";
}elseif(str_contains($message,"Ошибка")){
echo"<p><strongclass='notifi-account-reg-undo'>$message</strong></p>";
}
}
?>
<divclass="button-back">
<a href=" ../index.php"class="back-button">&larrhk;</a>
</div>
<divclass="main">
<inputtype="checkbox" id="chk" aria-hidden="true">
<divclass="signup">
<formmethod='post'>
<labelfor="chk" aria-hidden="true">Создать</label>
<inputtype="text" name="reglogin" placeholder="Логин" required>
<spanclass="form_span_help">
(англ.буквы,цифрыизнакподчеркивания)
</span>
<inputtype="email" name="regemail" placeholder="Email" required>
<inputtype="password" name="regpassword" placeholder="Пароль" required minlength=8>
<spanclass="form_span_help">
(минимум8символов)
</span>
<buttontype=submit>Зарегистрироваться</button>
</form>
</div>

<divclass="login">
<formmethod='GET'>
<labelfor="chk" aria-hidden="true">Войти</label>
<inputtype="text" name="loginInput" placeholder="Логин" required="">
<inputtype="password" name="passInput" placeholder="Пароль" required="">
<inputtype="submit" name='checkLogin' value='Войти'>
</form>
</div>
</div>
<scriptsrc="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$('.messagea').click(function(){
$(form).animate({height:"toggle",opacity:"toggle"},"slow");
});
</script>
</body>
</html>

```

Страница всех уроков

```

<?php
session_start();

```

```
?>
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Уроки - Учебник JavaScript</title>
  <link rel="stylesheet" href="/jshelper/static/css/lessons/vedenie.css">
  <style>
    /* Общие стили */
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f9f9f9;
      color: #333;
    }
    header {
      background-color: #2d3748;
      color: #fff;
      padding: 20px;
      text-align: center;
      font-size: 24px;
      font-weight: bold;
    }
    main {
      max-width: 1200px;
      margin: 30px auto;
      padding: 0 20px;
    }
    .lesson-grid {
      display: grid;
      grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
      gap: 20px;
    }
    .lesson-card {
      background-color: #fff;
      border-radius: 8px;
      box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
      overflow: hidden;
      transition: transform 0.2s, box-shadow 0.2s;
    }
    .lesson-card:hover {
      transform: translateY(-5px);
      box-shadow: 0 6px 10px rgba(0, 0, 0, 0.15);
    }
    .lesson-card h3 {
      font-size: 20px;
      margin: 0;
      padding: 20px;
      background-color: #4a5568;
      color: #fff;
    }
    .lesson-card p {
      padding: 15px 20px;
      font-size: 14px;
      color: #666;
      line-height: 1.5;
    }
    .lesson-card a {
      display: block;
      text-align: center;
      padding: 15px 20px;
      font-size: 16px;
      font-weight: bold;
      color: #fff;
      background-color: #2b6cb0;
      text-decoration: none;
      transition: background-color 0.2s;
    }
    .lesson-card a:hover {
      background-color: #2c5282;
    }
    footer {
      text-align: center;
```

```

        margin: 30px 0;
        font-size: 14px;
        color: #aaa;
    }
    .button-table-line {
        justify-content: space-between;
        display: flex;
    }
}
</style>
</head>
<body>
<header class="top-bar">
    <span class="small-text"><a target="_blank" href="https://www.php.net/"></a></span>
    <div class="menu">
        <button class="menu-btn" onclick="location.href='http://localhost/jshelper/help.php'">Помощь</button>
        <?php
        if(!isset($_SESSION["auth-header-login"])) {
            ?>
            <button class="menu-btn" onclick="location.href='http://localhost/jshelper/auth/login.php'">Войти</button>
        <?php
        } else {
            ?>
            <button class="menu-btn" onclick="location.href='http://localhost/jshelper/auth/account.php'">Аккаунт</button>
            <button class="menu-btn" onclick="location.href='http://localhost/jshelper/auth/logout.php'">Выход</button>
        <?php
        }
        ?>
        <button class="menu-btn" id="up" onclick="location.href='http://localhost/jshelper'">
        <svg viewBox="0 0 16 16" width="20px" fill="none" xmlns="http://www.w3.org/2000/svg"><g id="SVGRepo_bgCarrier" stroke-
width="0"></g><g id="SVGRepo_tracerCarrier" stroke-linecap="round" stroke-linejoin="round"></g><g
id="SVGRepo_iconCarrier"> <path d="M1 6V15H6V11C6 9.89543 6.89543 9 8 9C9.10457 9 10 9.89543 10 11V15H15V6L8 0L1 6Z"
fill="#000000"></path> </g></svg>
        </button>
    </div>
</header>
<main>
    <div class="lesson-grid">
        <!-- Карточка урока -->
        <div class="lesson-card">
            <h3>Введение</h3>
            <p>Изучите основы JavaScript: что это за язык, где он используется и как начать писать свой первый код.</p>
            <?php
            if (isset($_SESSION["auth-header-login"])) { ?>
            <div class="button-table-line">
                <a href="/jshelper/lessons/noauth/vedenie.php">Обычный курс</a>
                <a href="/jshelper/lessons/auth/prod-vedenie.php">Продвинутый курс</a>
            </div>
            <?php } else { ?>
                <a href="/jshelper/lessons/noauth/vedenie.php">Обычный курс</a>
            <?php } ?>
            </div>
            <div class="lesson-card">
                <h3>Основы JavaScript</h3>
                <p>Базовые понятия и термины, которые требуются для начала работы с JavaScript.</p>
                <a href="/jshelper/lessons/noauth/osnov-js.php">Обычный курс</a>
            </div>
            <div class="lesson-card">
                <h3>Переменные</h3>
                <p>Научитесь объявлять переменные, узнаете об их особенностях в JavaScript.</p>
                <?php
                if (isset($_SESSION["auth-header-login"])) { ?>
                <div class="button-table-line">
                    <a href="/jshelper/lessons/noauth/peremenie.php">Обычный курс</a>
                    <a href="/jshelper/lessons/auth/prod-peremenie.php">Продвинутый курс</a>
                </div>
                <?php } else { ?>
                    <a href="/jshelper/lessons/noauth/peremenie.php">Обычный курс</a>
                <?php } ?>
                </div>
            <div class="lesson-card">
                <h3>Функции</h3>
                <p>Погрузитесь в основы функций: что это такое, зачем они нужны и как их использовать.</p>
                <a href="/jshelper/errors/404.php">Обычный курс</a>
            </div>
            <div class="lesson-card">

```



```

        <h3>Циклы</h3>
        <p>Изучите, как использовать циклы для выполнения повторяющихся операций в вашем коде.</p>
        <a href="\jsHelper\errors\404.php">Обычный курс</a>
    </div>
    <div class="lesson-card">
        <h3>Объекты</h3>
        <p>Узнайте, как работать с объектами, их свойствами и методами.</p>
        <a href="\jsHelper\errors\404.php">Обычный курс</a>
    </div>
    <div class="lesson-card">
        <h3>Массивы</h3>
        <p>Научитесь эффективно использовать массивы для хранения и обработки данных.</p>
        <a href="\jsHelper\errors\404.php">Обычный курс</a>
    </div>
</div>
</main>
</body>
</html>

```

Страница Админ-панели

```

<style>
    .container {
        display: flex;
        margin: 55px;
        justify-content: space-between;
        padding: 20;
    }
    .cell {
        border: 1px solid #ccc;
        padding: 20px;
        width: 45%;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        text-align: center;
    }
    .cell h2 {
        margin-top: 0;
    }
    .cell p {
        margin: 10px 0;
    }
    .cell a {
        display: inline-block;
        padding: 10px 20px;
        background-color: #007BFF;
        color: white;
        text-decoration: none;
        border-radius: 5px;
    }
    .cell a:hover {
        background-color: #0056b3;
    }
</style>
<?php
session_start();

if (!isset($_SESSION["auth-header"])) {
    require_once("\xampp\htdocs\jsHelper\errors\403.php");
} else if (isset($_SESSION["auth-header"])) {
    if ($_SESSION["auth-header"] == "Пользователь") {
        require_once("\xampp\htdocs\jsHelper\errors\403.php");
    } else if ($_SESSION["auth-header"] == "Админ") { ?>
        <!DOCTYPE html>
        <html lang="en">
        <head>
            <meta charset="UTF-8">
            <meta name="viewport" content="width=device-width, initial-scale=1.0">
            <title>Админ панель</title>
            <link rel="stylesheet" type="text/css" href="../static/css/account/account.css">
        </head>
        <header class="top-bar">
            <span class="small-text"><a target="_blank" href="https://www.php.net/"></a></span>
            <div class="menu">
                <button class="menu-btn" onclick="location.href='http://localhost/jshelper/help.php'">Помощь</button>
                <button class="menu-btn" onclick="location.href='http://localhost/jshelper/auth/account.php'">Аккаунт</button>
            </div>
        </header>
    }
}

```

```

        <button class="menu-btn" onclick="location.href='http://localhost/jshelper/auth/logout.php'">Выход</button>
        <button class="menu-btn" id="up" onclick="location.href='http://localhost/jshelper'">
    <svg viewBox="0 0 16 16" width="20px" fill="none" xmlns="http://www.w3.org/2000/svg"><g id="SVGRepo_bgCarrier" stroke-
width="0"></g><g id="SVGRepo_tracerCarrier" stroke-linecap="round" stroke-linejoin="round"></g><g
id="SVGRepo_iconCarrier"> <path d="M1 6V15H6V11C6 9.89543 6.89543 9 8 9C9.10457 9 10 9.89543 10 11V15H15V6L8 0L1 6Z"
fill="#000000"></path> </g></svg>
    </button>
</div>
</header>
<div class="container">
    <div class="cell">
        <h2>Пользователи сайта</h2>
        <p>Таблица с пользователями, которые зарегистрировались на сайте</p>
        <a href="userView.php" class="btn">Подробнее</a>
    </div>
    <div class="cell">
        <h2>Пожелания сайта</h2>
        <p>Пожелания-сообщения из форма главной страницы</p>
        <a href="userMessage.php" class="btn">Подробнее</a>
    </div>
</div>
</body>
</html>
<?php
}
}

```

Страница отображения пользователей

```

<style>
    table {
        width: 100%;
        border-collapse: collapse;
        padding: 50;
    }

    table tbody tr:nth-child(even){
        background: #dbeef9;
    }

    thead {
        background: #d1d1d1;
    }
    .table-container {
        padding: 50;
    }
    th, td {
        border: 1px solid #ccc;
        padding: 20px;
        text-align: center;
        vertical-align: top;
    }
    .cell h2 {
        margin-top: 0;
    }
    .cell p {
        margin: 10px 0;
    }
    .cell a {
        display: inline-block;
        padding: 10px 20px;
        background-color: #007BFF;
        color: white;
        text-decoration: none;
        border-radius: 5px;
    }
    .cell a:hover {
        background-color: #0056b3;
    }

    .btn {
        background: #fff;
        border: 1px solid #ccc;
        border-radius: 4px;
        padding: 8px 15px;
    }

```

```

        font-size: 14px;
        cursor: pointer;
        transition: background 0.2s;
        border-color: #9B98B7;
        border-radius: 21px;
        font-size: 20px;
        border-width: 2px;
    }

    .btn:hover {
        background: #f0f0f0;
    }
    #toggleButton {
        margin-left: 3%;
        margin-top: 1%;
    }
    tr {
        transition: background 0.2s;
    }
    table tbody tr:hover {
        background: #bdfdd4;
        cursor: pointer;
    }
}

.slide-block {
    position: fixed;
    top: 0;
    right: -50%; /* Initial position off-screen */
    width: 300px;
    height: 100%;
    background-color: #f0f0f0;
    padding: 20px;
    box-shadow: 2px 0 5px rgba(0, 0, 0, 0.3);
    transition: all 0.5s ease;
}

.slide-block.active {
    right: 0; /* Slide in position */
}

#updateButton {
    display: none;
}
#updateButton.active {
    display: inline-block
}
#submitButton {
    display: none;
}
#submitButton.active {
    display: inline-block
}
}

#slideForm input {
    display: block;
    margin-bottom: 10px;
    margin-top: 10px;
    padding: 10px;
    width: 100%;
    box-sizing: border-box;
}

#slideForm button {
    padding: 10px 20px;
    margin-right: 10px;
}
</style>
<?php
session_start();
require_once('/xampp/htdocs/jshelper/db/db.php');

if (!isset($_SESSION["auth-header"])) {
    require_once('/xampp/htdocs/jshelper/errors/403.php');
} else if (isset($_SESSION["auth-header"])) {

```

```

if ($_SESSION["auth-header"] == "Пользователь") {
    require_once("../xampp/htdocs/jsHelper/errors/403.php");
} else if ($_SESSION["auth-header"] == "Админ") { ?>
    <!DOCTYPE html>
    <html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Пользователи</title>
        <link rel="stylesheet" type="text/css" href="../static/css/account/account.css">
    </head>
    <header class="top-bar">
        <span class="small-text"><a target="_blank" href="https://www.php.net/"></a></span>
        <div class="menu">
            <button class="menu-btn" onclick="location.href='http://localhost/jshelper/admin'">Назад</button>
            <button class="menu-btn" onclick="location.href='http://localhost/jshelper/auth/account.php'">Аккаунт</button>
            <button class="menu-btn" onclick="location.href='http://localhost/jshelper/auth/logout.php'">Выход</button>
            <button class="menu-btn" id="up" onclick="location.href='http://localhost/jshelper'">
                <svg viewBox="0 0 16 16" width="20px" fill="none" xmlns="http://www.w3.org/2000/svg"><g id="SVGRepo_bgCarrier" stroke-
width="0"></g><g id="SVGRepo_tracerCarrier" stroke-linecap="round" stroke-linejoin="round"></g><g
id="SVGRepo_iconCarrier"> <path d="M1 6V15H6V11C6 9.89543 6.89543 9 8 9C9.10457 9 10 9.89543 10 11V15H15V6L8 0L1 6Z"
fill="#000000"></path> </g></svg>
            </button>
        </div>
    </header>

    <button class="btn" id="toggleButton">Добавить новый аккаунт</button>

    <div id="slideBlock" class="slide-block">
        <form id="slideForm" method="POST">
            <div style="display:ruby-text">
                <label for="userId">Вы работаете с</label>
                <input type="text" id="userId" name="userId" readonly style="width: max-content;">

                <input type="text" placeholder="Логин" name="loginUser" id="loginUser" style="margin-bottom:0px;" maxlength="15"
required>
                <span style="opacity:0.75;font-size:15px">Пример: Login</span>

                <input type="text" placeholder="Пароль" name="password" id="password" style="margin-bottom:0px;" maxlength="25"
required>
                <span style="opacity:0.75;font-size:15px">Пример: qwerty123</span>

                <input type="email" placeholder="Почта" name="email" id="email" style="margin-bottom:0px;" maxlength="25" required>
                <span style="opacity:0.75;font-size:15px">Пример: mygmail@gmail.com</span>

                <input type="text" placeholder="Роль доступа" name="access" id="access" pattern="^(Пользователь|Модератор|Админ)$"
style="margin-bottom:0px;" maxlength="25" required>
                <span style="opacity:0.75;font-size:15px">Пример: Пользователь, Модератор, Админ</span>

                <br><br>
                <button type="submit" id="updateButton" name="updateRequest">Редактировать</button>
                <button type="submit" id="submitButton">Добавить</button>
                <button type="button" id="clearButton">Очистить</button>
            </form>
        </div>

    <?php
    try {
        // Пытаемся найти данные в базе
        $stmt = $pdo->prepare("SELECT * FROM users ORDER BY userId ASC");
        $stmt->execute();

        } catch (PDOException $e) {
            $message = "Ошибка: " . $e->getMessage();
        }
    }
    ?>
    <div class="table-container">

        <table class="data-table">
            <thead>
                <tr>
                    <th>Идентификатор</th>
                    <th>Логин</th>

```

```

        <th>Пароль</th>
        <th>Почта</th>
        <th>Роль доступа</th>
        <th>Действия</th>
    </tr>
</thead>
<tbody>
<?php
while ($row = $stmt->fetch(PDO::FETCH_LAZY)) {
    if ($row->Login == $_SESSION["auth-header-login"] AND $row->Password == $_SESSION["auth-header-pass"]) {
        echo '<tr class="list-items" style="pointer-events: none; background-color:#999999d4;">';
    } else {
        echo '<tr class="list-items">';
    }
    echo '<td>' . htmlspecialchars($row->UserId) . '</td>';
    echo '<td>' . htmlspecialchars($row->Login) . '</td>';
    echo '<td>' . htmlspecialchars($row->Password) . '</td>';
    echo '<td>' . htmlspecialchars($row->Email) . '</td>';
    echo '<td>' . htmlspecialchars($row->UserRole) . '</td>';
    echo '<td>
        <a href="?userId=' . htmlspecialchars($row->UserId) . '">удалить</a>
    </td>';
    echo '</tr>';
}
?>
</tbody>
</table>
</div>
</body>
</html>
<?php

```

```

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $access = trim($_POST['access']);
    $password = trim($_POST['password']);
    $email = trim($_POST['email']);
    $loginUser = trim($_POST['loginUser']);
    $userId = trim($_POST['userId'] ?? '');

    try {
        if (!empty($userId)) {
            $stmt = $pdo->prepare(
                "UPDATE users
                SET Login = :loginUser,
                    UserRole = :access,
                    Email = :Stockemail,
                    Password = :password
                WHERE userId = :userId"
            );
            $stmt->bindParam(':loginUser', $loginUser, PDO::PARAM_STR);
            $stmt->bindParam(':access', $access, PDO::PARAM_STR);
            $stmt->bindParam(':Stockemail', $email, PDO::PARAM_STR);
            $stmt->bindParam(':password', $password, PDO::PARAM_STR);
            $stmt->bindParam(':userId', $userId, PDO::PARAM_INT);

            if ($stmt->execute()) {
                $message = "Запись успешно обновлена.";
            } else {
                $message = "Ошибка: не удалось обновить запись.";
            }
        } else {
            $stmt = $pdo->prepare(
                "INSERT INTO users (Login, UserRole, Password, Email)
                VALUES (:loginUser, :access, :password, :Email)"
            );
            $stmt->bindParam(':loginUser', $loginUser, PDO::PARAM_STR);
            $stmt->bindParam(':access', $access, PDO::PARAM_STR);
            $stmt->bindParam(':Email', $email, PDO::PARAM_STR);
            $stmt->bindParam(':password', $password, PDO::PARAM_STR);

            if ($stmt->execute()) {
                $message = "Запись успешно добавлена.";
            } else {
                $message = "Ошибка: не удалось добавить запись.";
            }
        }
    }
}

```

```

    }

    // Перенаправление после успешного выполнения
    echo '<script type="text/javascript">';
    echo 'window.location.href = "http://localhost/jshelper/admin/userView.php"';
    echo '</script>';
} catch (PDOException $e) {
    $message = "Ошибка: " . $e->getMessage();
}
}

if (isset($_GET['userId'])) {
    $Id = $_GET['userId'];

    try {
        // Подготовка запроса с использованием параметра
        $stmt = $pdo->prepare("DELETE FROM `users` WHERE `userId` = :userId");
        $stmt->bindParam(':userId', $Id, PDO::PARAM_INT); // Привязка параметра

        // Выполнение запроса
        if ($stmt->execute()) {
            echo '<script type="text/javascript">';
            echo 'window.location.href = "http://localhost/jshelper/admin/userView.php"';
            echo '</script>';
        } else {
            echo "Ошибка при удалении записи.";
        }
    } catch (PDOException $e) {
        echo "Ошибка: " . $e->getMessage();
    }
}

require_once 'xampp/htdocs/jshelper/db/duplicate.php';

}
}
?>
<script src="/jshelper/static/js/add.js"></script>

```

Страница отображения отзывов

```

<style>
    table {
        width: 100%;
        border-collapse: collapse;
        padding: 50;
    }

    table tbody tr:nth-child(even){
        background: #dbeef9;
    }

    thead {
        background: #d1d1d1;
    }
    .table-container {
        padding: 50;
    }
    th, td {
        border: 1px solid #ccc;
        padding: 20px;
        text-align: center;
        vertical-align: top;
    }
    .cell h2 {
        margin-top: 0;
    }
    .cell p {
        margin: 10px 0;
    }
    .cell a {
        display: inline-block;
        padding: 10px 20px;
        background-color: #007BFF;
        color: white;
        text-decoration: none;
    }

```

```

        border-radius: 5px;
    }
    .cell a:hover {
        background-color: #0056b3;
    }

    .btn {
        background: #fff;
        border: 1px solid #ccc;
        border-radius: 4px;
        padding: 8px 15px;
        font-size: 14px;
        cursor: pointer;
        transition: background 0.2s;
        border-color: #9B98B7;
        border-radius: 21px;
        font-size: 20px;
        border-width: 2px;
    }

    .btn:hover {
        background: #f0f0f0;
    }
    #toggleButton {
        margin-left: 3%;
        margin-top: 1%;
    }
    tr {
        transition: background 0.2s;
    }
    table tbody tr:hover {
        background: #bdfdd4;
        cursor: pointer;
    }

    .slide-block {
        position: fixed;
        top: 0;
        right: -50%; /* Initial position off-screen */
        width: 300px;
        height: 100%;
        background-color: #f0f0f0;
        padding: 20px;
        box-shadow: 2px 0 5px rgba(0, 0, 0, 0.3);
        transition: all 0.5s ease;
    }

    .slide-block.active {
        right: 0; /* Slide in position */
    }

    #updateButton {
        display: none;
    }
    #updateButton.active {
        display: inline-block
    }
    #submitButton {
        display: none;
    }
    #submitButton.active {
        display: inline-block
    }

    #slideForm input {
        display: block;
        margin-bottom: 10px;
        margin-top: 10px;
        padding: 10px;
        width: 100%;
        box-sizing: border-box;
    }

    #slideForm {

```

```

    top: 10%;
    position: relative;
}

#slideForm button {
    padding: 10px 20px;
    margin-right: 10px;
}
</style>
<?php
session_start();
require_once('/xampp/htdocs/jshelper/db/db.php');

if (!isset($_SESSION["auth-header"])) {
    require_once('xampp/htdocs/jsHelper/errors/403.php');
} else if (isset($_SESSION["auth-header"])) {
    if ($_SESSION["auth-header"] == "Пользователь") {
        require_once('xampp/htdocs/jsHelper/errors/403.php');
    } else if ($_SESSION["auth-header"] == "Админ") { ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Отзывы</title>
    <link rel="stylesheet" type="text/css" href="../static/css/account/account.css">
</head>
<header class="top-bar">
    <span class="small-text"><a target="_blank" href="https://www.php.net/"></a></span>
    <div class="menu">
        <button class="menu-btn" onclick="location.href='http://localhost/jshelper/admin'">Назад</button>
        <button class="menu-btn">Помощь</button>
        <button class="menu-btn" onclick="location.href='http://localhost/jshelper/auth/account.php'">Аккаунт</button>
        <button class="menu-btn" onclick="location.href='http://localhost/jshelper/auth/logout.php'">Выход</button>
        <button class="menu-btn" id="up" onclick="location.href='http://localhost/jshelper'">
            <svg viewBox="0 0 16 16" width="20px" fill="none" xmlns="http://www.w3.org/2000/svg"><g id="SVGRepo_bgCarrier" stroke-
width="0"></g><g id="SVGRepo_tracerCarrier" stroke-linecap="round" stroke-linejoin="round"></g><g
id="SVGRepo_iconCarrier"> <path d="M1 6V15H6V11C6 9.89543 6.89543 9 8 9C9.10457 9 10 9.89543 10 11V15H15V6L8 0L1 6Z"
fill="#000000"></path> </g></svg>
        </button>
    </div>
</header>

<button class="btn" id="toggleButton">Добавить новый отзыв</button>

<div id="slideBlock" class="slide-block">
    <form id="slideForm" method="POST">
        <div style="display: ruby-text">
            <label for="userId">Вы работаете с</label>
            <input type="text" id="userId" name="userId" readonly style="width: max-content;">
        </div>

        <textarea placeholder="Отзыв" name='message' id='message' class="form-input" rows="10" cols="33" required
maxlength=550"></textarea>

        <input type="text" placeholder="Юзернейм" name="whosend" id="whosend" style="margin-bottom: 0px;" maxlength="25"
required>
        <span style="opacity: 0.75; font-size: 15px">Пример: Захар</span>

        <br><br>
        <button type="submit" id="updateButton" name="updateRequest">Редактировать</button>
        <button type="submit" id="submitButton">Добавить</button>
        <button type="button" id="clearButton">Очистить</button>
    </form>
</div>

<?php
try {
    // Пытаемся найти данные в базе
    $stmt = $pdo->prepare("SELECT * FROM wishesbyusers ORDER BY wishesId ASC");
    $stmt->execute();

} catch (PDOException $e) {
    $message = "Ошибка: " . $e->getMessage();
}

```



```

?>
<div class="table-container">

<table class="data-table">
  <thead>
    <tr>
      <th>Идентификатор</th>
      <th>Отзыв</th>
      <th>От кого</th>
      <th>Действия</th>
    </tr>
  </thead>
  <tbody>
<?php
while ($row = $stmt->fetch(PDO::FETCH_LAZY)) {
  echo '<tr class="list-items">';
  echo '<td>' . htmlspecialchars($row->WishesId) . '</td>';
  echo '<td>' . htmlspecialchars($row->Message) . '</td>';
  echo '<td>' . htmlspecialchars($row->Whosend) . '</td>';
  echo '<td>
    <a href="?WishesId=' . htmlspecialchars($row->WishesId) . '">удалить</a>
  </td>';
  echo '</tr>';
}
?>
</tbody>
</table>
</div>
</body>
</html>
<?php

```

```

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
  $whosend = trim($_POST['whosend']);
  $message = trim($_POST['message']);
  $WishesId = trim($_POST['userId'] ?? '');

  try {
    if (!empty($WishesId)) {
      $stmt = $pdo->prepare(
        "UPDATE wishesbyusers
        SET Message = :Message,
        Whosend = :Whosend,
        WHERE WishesId = :WishesId"
      );
      $stmt->bindParam(':Message', $message, PDO::PARAM_STR);
      $stmt->bindParam(':Whosend', $whosend, PDO::PARAM_STR);
      $stmt->bindParam(':WishesId', $WishesId, PDO::PARAM_INT);

      if ($stmt->execute()) {
        $message = "Запись успешно обновлена.";
      } else {
        $message = "Ошибка: не удалось обновить запись.";
      }
    } else {
      $stmt = $pdo->prepare(
        "INSERT INTO wishesbyusers (Message, Whosend)
        VALUES (:Message, :Whosend)"
      );
      $stmt->bindParam(':Message', $message, PDO::PARAM_STR);
      $stmt->bindParam(':Whosend', $whosend, PDO::PARAM_STR);

      if ($stmt->execute()) {
        $message = "Запись успешно добавлена.";
      } else {
        $message = "Ошибка: не удалось добавить запись.";
      }
    }
  }

  // Перенаправление после успешного выполнения
  echo '<script type="text/javascript">';
  echo 'window.location.href = "http://localhost/jshelper/admin/userMessage.php";';
  echo '</script>';
} catch (PDOException $e) {
  $message = "Ошибка: " . $e->getMessage();
}

```

```

    }
}

if (isset($_GET['WishesId'])) {
    $Id = $_GET['WishesId'];

    try {
        // Подготовка запроса с использованием параметра
        $stmt = $pdo->prepare("DELETE FROM `wishesbyusers` WHERE `WishesId` = :WishesId");
        $stmt->bindParam(':WishesId', $Id, PDO::PARAM_INT); // Привязка параметра

        // Выполнение запроса
        if ($stmt->execute()) {
            echo '<script type="text/javascript">';
            echo 'window.location.href = "http://localhost/jshelper/admin/userMessage.php";';
            echo '</script>';
        } else {
            echo "Ошибка при удалении записи.";
        }
    } catch (PDOException $e) {
        echo "Ошибка: " . $e->getMessage();
    }
}

require_once '/xampp/htdocs/jshelper/db/duplicate.php';

}
}
?>
<script src="/jshelper/static/js/add.js"></script>

```