Частное учреждение

профессиональная образовательная организация

ТЕХНИКУМ «БИЗНЕС И ПРАВО»

ЗАЩИЩЕНО С ОЦЕНКОЙ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и ФИО руководителя)

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.

**КУРСОВАЯ РАБОТА**

|  |  |
| --- | --- |
| Тема | РАЗРАБОТКА WEB-УЧЕБНИКА ПО ДИСЦИПЛИНЕ «ОПЕРАЦИОННЫЕ СИСТЕМЫ И СРЕДЫ». |
| По МДК 09.01 | Проектирование и разработка веб-приложений |
| Выполнила | Корчагина Е.В. |
| Курс, группа | 4 курс, группа ИС-4 |
| Специальность | 09.02.07 Информационные системы и программирование |
| Научный руководитель | Мартыненко В.А. |

г. Белореченск

2023 г.

СОДЕРЖАНИЕ

|  |  |
| --- | --- |
| ВВЕДЕНИЕ | 3 |
| 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ | 5 |
| 1.1 Анализ предметной области | 5 |
| 1.2 Цели и задачи выполняемой системы | 5 |
| 1.3 Метод реализации процесса проектирования | 6 |
| 1.4 Обоснование проектных решений | 7 |
| 2. ПРОЕКТНАЯ ЧАСТЬ | 18 |
| 2.1 Техническое задание | 18 |
| 2.2 Установка и настройка среды разработки | 19 |
| 2.3 Руководство пользователя | 21 |
| 2.4 Руководство администратора | 22 |
| ЗАКЛЮЧЕНИЕ | 23 |
| СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ | 24 |
| ПРИЛОЖЕНИЕ |  |

ВВЕДЕНИЕ

Процесс вхождения учебной организации в мировое образовательное пространство требует совершенствования, а также серьёзной переориентации компьютерно-информационной составляющей. Вторая половина ХХ века стала периодом перехода к информационному обществу.

Информационный взрыв породил множество проблем, важнейшей из которых является проблема обучения. Особый интерес представляют вопросы, связанные с автоматизацией обучения, поскольку «ручные методы» без использования технических средств давно исчерпали свои возможности. Наиболее доступной формой автоматизации обучения является использование компьютеров, то есть использование компьютерного времени для обучения и обработки результатов тестовой проверки знаний учащихся. Все более широкое использование компьютеров позволяет автоматизировать и тем самым упростить сложную процедуру, которую также используют учителя при создании учебных пособий. Таким образом, представление различного рода «электронных учебников» и учебных пособий на компьютере имеет ряд важных преимуществ.

Объект исследование – информационная система электронного учебника “Операционные системы и среды”.

Предмет исследования – предметом исследования являются процессы создания, размещения тем и заданий учебника “Операционные системы и среды”.

Цель исследование: рассмотреть и установить повышение эффективности процессов электронного учебника “Операционные системы и среды” за счет разработки и внедрения автоматизированной информационной системы.

Задачи исследования:

1. Изучить предметную область;
2. Рассмотреть ряд существующих разработок для решения поставленной цели;
3. Определить способ реализации информационной системы;
4. Разработка системы электронного учебника “Операционные системы и среды”;
5. Тестирование разработанного электронного учебника “Операционные системы и среды”.

Методы исследования – изучение и анализ информационных источников, обобщение и реализация разрабатываемой информационной системы.

Электронный учебник “Операционные системы и среды” подразумевает под собой возможность создание тем и их наполнение подходящей информацией.

Также должен быть разделен доступ к определенным возможностям разных групп пользователей. Первая это сами пользователи, которые имеют доступ только к прочтению тем, представленных на сайте. Вторая группа – модераторы(преподаватели), которые имеют доступ уже к созданию заданий и их изменению. Ну и самым главным в этой системе будет администратор, которые имеет доступ не только ко всем действиям, но также может работать с зарегистрированными в системе пользователями, изменять их данные или вовсе удалять из системы.

1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Анализ предметной области

Разрабатываемая система предназначена для дистанционного обучения или прохождения нового материала для студентов, что облегчит работу преподавателю. Система построена по трехзвенной клиент-серверной архитектуре. Она также должна быть удобной, простой и доступной в использовании и понимании для всех категорий пользователей.

Модератор сможет легко изменять содержимое тем, добавлять новые и удалять старые, в то время как администратор сможет легко и быстро вносить изменения о пользователях.

Сама система необходима для облегчения работы для преподавателей и обучения для студентов.

1.2 Цели и задачи выполняемой системы

Основными целями создания системы будут являться:

* Облегчение работы для преподавателей;
* Простота использования для обучающих;
* Создание тем и упражнений после них на проверку знаний;
* Работа с пользователями и распределение ролей;
* Изменение данных;
* Удаление данных.

Задачами являются:

* Достичь поставленных целей;
* Разработать электронный учебник “Операционные системы и среды”;
* Сделать возможным разделение пользователей на категории;
* Связать данные БД с программой и вывести их;
* Освоить полученные навыки;
* Оформить документацию;
* Разработать итоговую структурированную программу.

1.3 Метод реализации процесса проектирования

При разработке информационной системы должны автоматизироваться процессы добавления, редактирования и удаления тем, упражнений к ним и пользователей. При внедрении системы должны быть достигнуты ранее описанные этапы.

Объекты автоматизации характеризуются:

* обеспечивать максимальную работоспособность;
* быть надежной и технологичной;
* обслуживание должно проводиться минимальным количеством персонала;

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

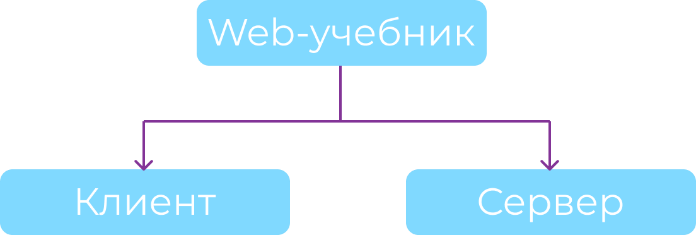


Рисунок 1.1 – Модули, которые должны быть в программе

Модули разрабатываемой системы, представленные на рисунке 1.2:

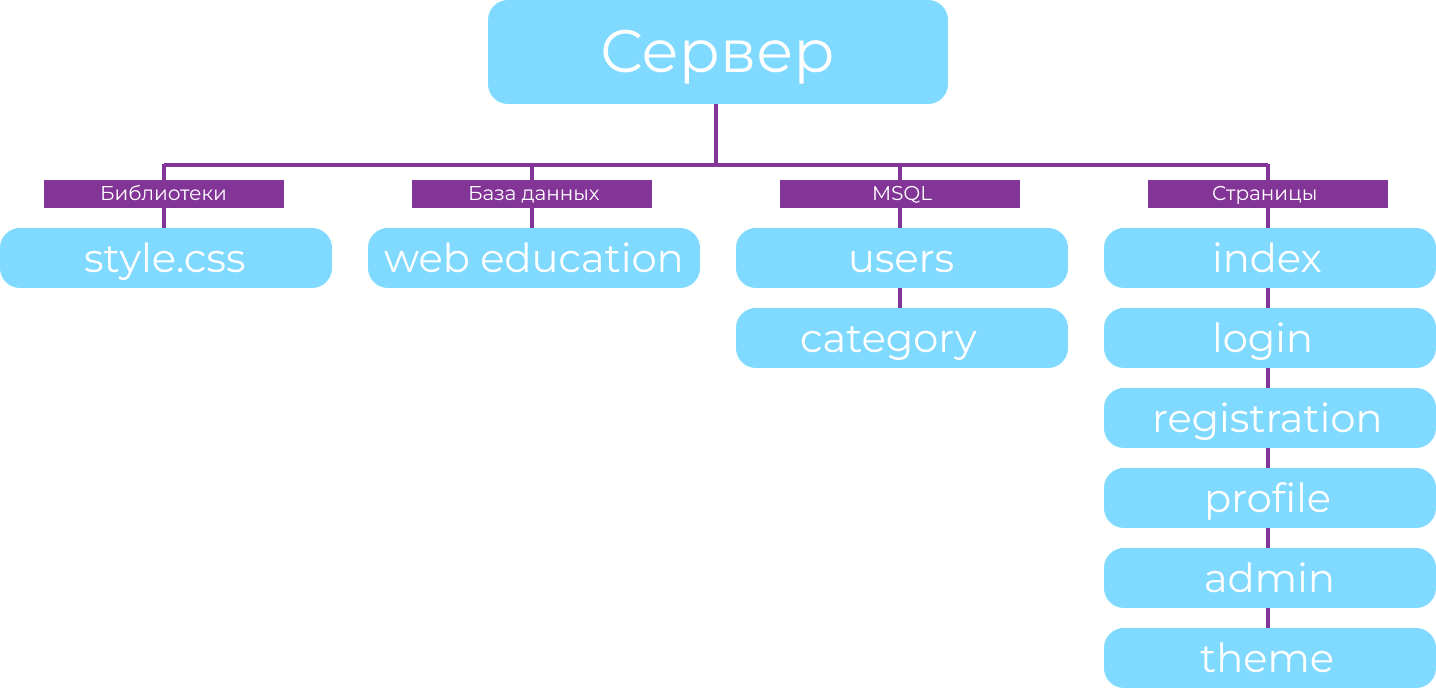


Рисунок 1.2 – Модули разрабатываемой системы

Описание модулей разрабатываемой системы:

* style.css – CSS стили;
* web education – подключение к базе данных и имя таблицы;
* users– имя таблицы и зарегистрированные пользователи;
* category – имя таблицы и категории пользователей;
* index - основная страница;
* login - страница авторизации пользователя;
* registration - страница регистрации пользователя;
* profile - профиль пользователя;
* admin - админ-панель с возможностью редактирования и просмотра информации о пользователях;
* theme - тема учебника.

1.4 Обоснование проектных решений

Для создания программы будет использоваться трехзвенная клиент-серверная архитектура. Трехуровневая архитектура приложений — это модульная клиент-серверная архитектура, которая состоит из уровня представления, уровня приложения и уровня данных. Уровень данных обеспечивает хранение информации, уровень приложений обрабатывает логику, а уровень представления являет собой графический интерфейс пользователя (GUI), который взаимодействует с двумя другими уровнями.

К достоинствам системы клиент/сервер следует отнести:

* сильную централизованную защиту;
* центральное хранилище файлов;
* возможность совместного использования серверами доступного технического и программного обеспечения;
* простую управляемость при большом числе пользователей;
* централизованную организацию, предотвращающую потерю данных на компьютерах.

MySQL — это система управления базами данных (СУБД), позволяющая хранить, организовывать большие объемы данных, и манипулировать ими.

MySQL сочетает в себе хорошую производительность, высокую надежность и простоту, что делает рассматриваемую СУБД популярным выбором у создателей и администраторов веб-сайтов, требующих обработки данных.

На рисунке 1.3 представлена структура клиент–серверной архитектуры:

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание

Рисунок 1.3 – Клиент–серверная архитектура

Требования к системе разработаны в соответствии со стандартом качества программного обеспечения ISO 9126:2001, описывающим многоуровневую модель характеристик качества и соответствующий им набор атрибутов.

Требования к системе представлены в таблице 1.1.

Таблица 1.1 – Требования к системе

|  |  |  |
| --- | --- | --- |
| Атрибуты | Метрики | Требования |
| 1 | 2 | 3 |
| Удобство | | |
| Простота использования | Среднее время, необходимое  пользователю, чтобы найти  необходимый элемент в программе | Поиск отдельного элемента интерфейса не должен занимать более 1 секунды |
| Привлекательность | Соответствие интерфейса  требованиям | 99% элементов должны быть работоспособны для  пользователя |
| Обучаемость | Показатель, затрачиваемый  пользователями на обучение  работе | Пользователь должен изучить работу ПО за первые 3-5 минут  использования |
| Производительность | | |
| Временная эффективность | Время выполнения компонента программы | - Приложение должно  обслуживать любую функцию не дольше 1 сек; |
| Эффективность использования ресурсов | Объемы ресурсов требуемых для выполнения задач | - Постоянная память,  используемая программой – не более 10МБ;  - Приложение должно потреблять не более 3МБ памяти на каждый неактивный сеанс с пользователем; |

Продолжение таблицы 1.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Переносимость | | |
| Удобство установки | Легкость установки | Необходима установка:  xampp, mysql, admin php |
| Способность к сосуществованию | Способность ПО сосуществовать с другими  программами в общем окружении | Программа не должна  занимать большие объемы памяти устройства (максимум 3МБ) |
| 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, квалификации и обучению персонала системы, ее документацию и руководство:

Численность персонала должна быть достаточной для обеспечения эффективного функционирования системы. Оптимальное количество зависит от размера и сложности системы, а также от потребностей пользователя.

Квалификация персонала: персонал, работающий с системой, должен обладать необходимыми знаниями и навыками для выполнения своих обязанностей. Это может включать знание специфических технических навыков, знание программного обеспечения, опыт работы и возможно даже сертификацию.

Обучение персонала: персонал должен быть обучен использованию системы, ее функциональности и возможностям. Обучение может быть проведено внутри компании.

Документация и руководства: система должна содержать подробную документацию и руководства для персонала, которые помогут им эффективно работать с системой. Это может включать инструкции по установке, настройке и использованию системы.

В системе существуют 3 группы пользователей:

1. Пользователь выполняет функцию: просмотр, изучение, авторизация, регистрация;
2. Модератор имеет доступ к занесению, редактированию и удалению данных;
3. Администратор имеет все функции двух предыдущих групп и к ним добавляется работа с пользователями и доступ к функциям БД.

Пользователи системы должны иметь опыт работы с персональным компьютером на базе операционных систем Microsoft Windows на уровне пользователя.

В таблице 1.2 представлены требования к надежности.

Таблица 1.2 – Требования к надёжности

|  |  |  |
| --- | --- | --- |
| Атрибуты | Метрики | Требования |
| Надёжность | | |
| 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 определённый 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;  - Система должна оповещать пользователя при некорректных действиях в системе;  - Данные, которые вводит пользователь, должны сохраняться. |

Средняя доступность приложения 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 темы | 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: Авторизация.

Описание: Один из пользователей заходит в систему.

Действующий субъект: Пользователь.

Предусловие: Пользователь успешно зашел в систему.

Сценарий:

* Пользователь имеет персональный компьютер;
* Пользователь заходит на сайт;
* Пользователь успешно зашёл на сайт.

Расширения: в случае некорректного формата введенных данных в БД, при попытки зайти по уникальному адресу, выведется сообщение с описанием ошибки «Ошибка подключения».

На рисунке 1.4 представлена диаграмма последовательности сценария "Авторизация".

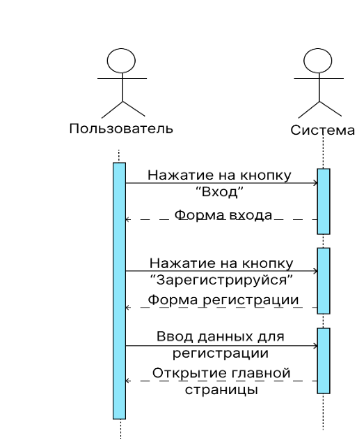


Рисунок 1.4 – Диаграмма последовательности сценария "Авторизация".

Имя: Регистрация.

Описание: Пользователь имеет возможность создать учетную запись для дальнейшего пользования системой.

Предусловие: Пользователь не может зайти в систему, не имея учетной записи.

Сценарий:

* Пользователь пытается зайти в систему;
* Система просит данные для входа;
* Пользователь регистрирует свои данные;
* Система открывает доступ.

Расширения: при отсутствии соединения с сервером выведется сообщение об ошибке.

На рисунке 1.5 представлена диаграмма последовательности сценария "Регистрация".

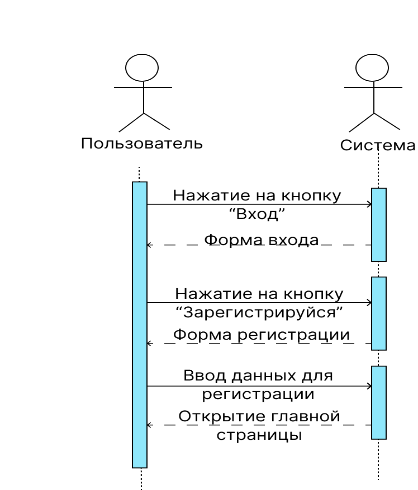


Рисунок 1.5 – Диаграмма последовательности сценария "Регистрация"

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".

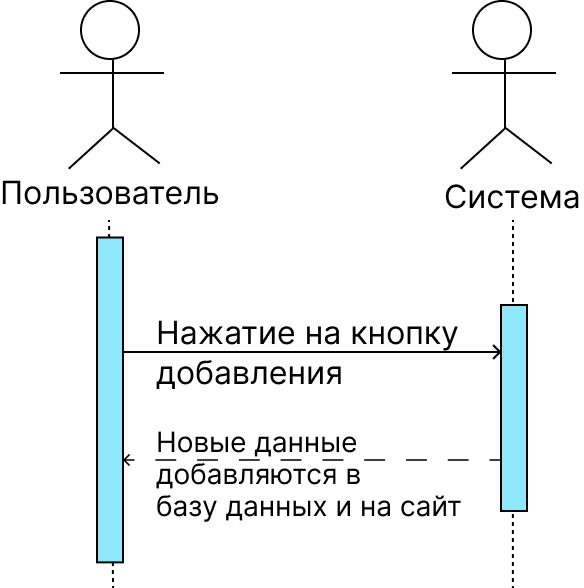


Рисунок 1.6 – Диаграмма последовательности сценария "Добавления"

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".

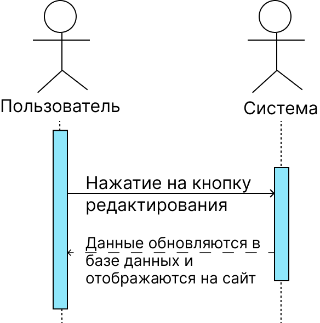


Рисунок 1.7 – Диаграмма последовательности сценария "Редактирования"

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".

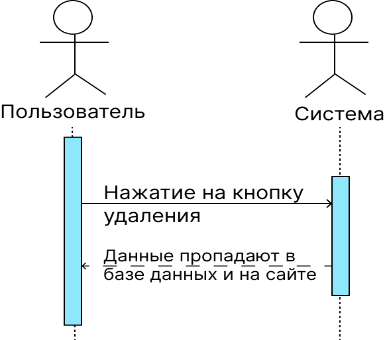


Рисунок 1.8 – Диаграмма последовательности сценария "Удаления"

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

* интернет-соединение;
* обеспечить каждого преподавателя персональным компьютером;
* установить на каждый ПК данную программу;
* иметь программу xampp;
* база данных MySQL.

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

* Пользователи;
* Категории;

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

База данных информационной системы должна состоять из следующих связанных таблиц:

* users – таблица содержащая информацию о зарегистрированных.
* category – таблица, содержащая информацию о категориях пользователей.

Данные о пользователях представлены в таблице users 1.4.

Таблица 1.4 – Данные о пользователях

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | int | 11 | Идентификатор | Первичный ключ |
| name | varchar | 255 | Имя пользователя | NOT NULL |
| email | varchar | 60 | Электронная почта | NOT NULL |
| password | int | 64 | Пароль пользователя | NOT NULL |
| status | varchar | 30 | Статус пользователя | NOT NULL |
| date | timestamp | - | Дата регистрации | NOT NULL |
| role | varchar | 20 | Роль пользователя | NOT NULL |

Данные о категориях пользователей должны размещаться в таблице category 1.5.

Таблица 1.5 – Данные о категориях пользователей

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | int | 11 | Идентификатор | Первичный ключ |
| name | text | - | Название роли | NOT NULL |
| order | int | 11 | Сортировка | NOT NULL |
| url | varchar | 255 | Адрес роли | NOT NULL |
| parent\_id | int | 11 | Связывающая таблиц | NOT NULL |
| date | timestamp | - | Когда был создана роль | NOT NULL |
| email | varchar | 60 | Кем был создана роль | NOT NULL |
| status | varchar | 20 | Статус | NOT NULL |

На данном этапе разберем, то, как будем 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.

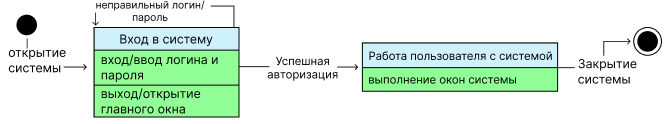


Рисунок 1.9 – Общая диаграмма состояний системы

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

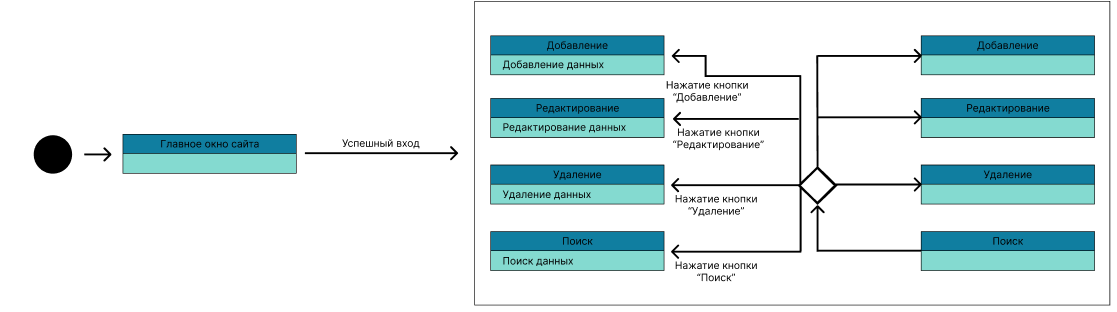


Рисунок 1.10 – Диаграмма состояний работы пользователя с системой

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.

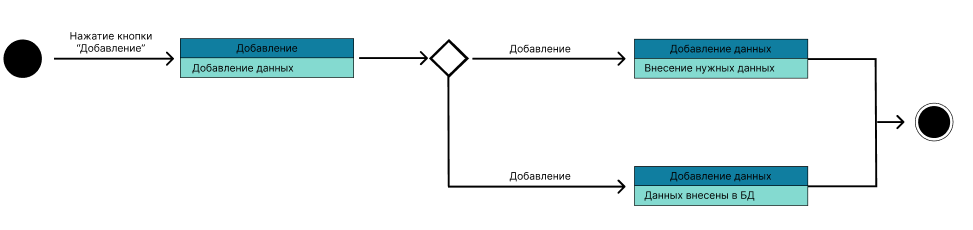


Рисунок 1.11 – Диаграмма состояний добавление данных

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.

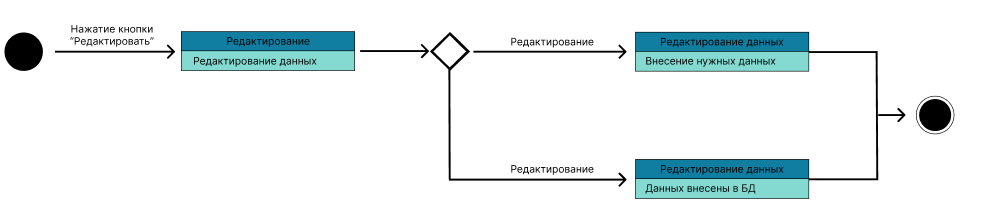


Рисунок 1.12 – Диаграмма состояний редактирования данных

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

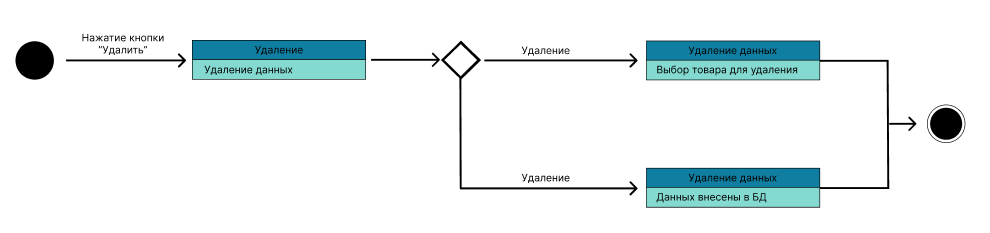


Рисунок 1.14 – Диаграмма состояний удаления данных

2. ПРОЕКТНАЯ ЧАСТЬ

2.1 Техническое задание

Информационное обеспечение информационной системы (ИОИС) включает в себя процессы, методы и технологии, которые используются для сбора, обработки, хранения и передачи информации в рамках данной системы. Оно обеспечивает доступность, достоверность, целостность, конфиденциальность и актуальность информации, а также эффективность и надежность функционирования всей системы.

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.

1. Дизайн интерфейса

Визуальное оформление сайта должно быть привлекательным и удобным для использования. Цветовая гамма может быть основана на светлых и пастельных тонах, чтобы создать приятную атмосферу для обучения. В нашем случае были выбраны фиолетовые оттенки фона и голубые для элементов, таких как кнопки.

Расположение и оформление элементов управления: Главное меню сайта размещено вверху страницы в виде горизонтальной навигационной панели, которая состоит из кнопки в левом углу и логотипе, который при нажатии ведет пользователя на главную страницу, он находится в противоположном углу. Оформление ссылок и кнопок должно быть понятным и привлекательным для пользователя.

Каждая страница сайта может иметь четкую структуру, начиная с заголовка урока или раздела, затем содержать основное содержимое, такое как текст, изображения, видеоматериалы и интерактивные элементы. Внизу страницы размещен тест, который направлен на закрепление прочитанного материала и навигационная панель со ссылками на предыдущую и следующую страницы.

Выбор шрифтов и типографики играют важную роль в создании приятного визуального впечатления и удобного чтения. Рекомендуется использовать шрифты, которые четко читаемы и хорошо сочетаются друг с другом. В нашем случае выбраны шрифты - Jura и Montesserat, который также используется в виде курсива. Цвет шрифта - белый, размеры используются - 25 и 32 пикселя.

1. Функциональные возможности системы:

* Предоставление информации о базовых понятиях и основных принципах операционных систем и сред.
* Описание архитектуры и компонентов операционных систем.
* Рассмотрение основных типов операционных систем (многозадачные, однопользовательские и т. д.).
* Предоставление информации о процессах и потоках выполнения.
* Обзор системного вызова и его роли в операционных системах.
* Рассмотрение вопросов безопасности и защиты в операционных системах.
* Представление примеров из реального мира и отраслевых решений.
* Интерактивные элементы, такие как тесты, задачи, практические упражнения и визуализации.
* Возможность добавления дополнительных материалов, ссылок и литературы.

2.2 Установка и настройка среды разработки

Для установки и настройки среды разработки для электронного учебника "Операционные системы и среды" 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, Sublime Text или Atom.

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 учебника.

5. Создание проекта. Последний шаг - создание самого проекта электронного учебника “Операционные системы и среды”. Определить структуру проекта, создать необходимые файлы и начать разрабатывать содержание учебника, используя выбранный текстовый редактор и язык программирования.

Эти шаги являются общими рекомендациями, и в зависимости от конкретных потребностей и требований учебника, некоторые шаги могут отличаться. Рекомендуется ознакомиться с документацией и руководствами к выбранным инструментам для получения более подробной информации о настройке их использования для разработки электронного учебника "Операционные системы и среды".

2.3 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 «Войти».

Окно авторизации состоит из двух строчек, в которые пользователю необходимо ввести свой логин - электронную почту и пароль и нажать кнопку входа. Если пользователь не имеет учетной записи, он может создать ее. Для этого необходимо в окне авторизации нажать кнопку регистрации, после чего пользователя перекидывает на новую страницу. Там ему необходимо ввести свое имя, электронную почту и пароль и также нажать соответствующую кнопку. Ему выдается строка с текстом об успешной регистрации и ссылкой для перехода на главную страницу.

Главная страница включает в себя информацию о самой программе, ее функциях, далее идет описание нашего электронного учебника «Операционные системы и среды» и его содержание со ссылкой на первую тему.

При переходе на страницу с темой занятия пользователь имеет доступ к прочтению и в конце для закрепления полученных знаний, пройти тестирование, которое выдаст количество правильных ответов и если результат больше 80%, то нас пустит на следующую страницу, но также всегда можно вернуться к уже пройденному материалу.

Также у пользователя есть кнопка профиля, и здесь функционал меняется, в зависимости от роли авторизованного пользователя. Если он имеет самую низкую роль, то ему доступна лишь кнопка выхода из системы на странице профиля. Если пользователь имеет роль модератора, то ему открывается новая функция – управление занятиями. Под модератором подразумевается преподаватель, которые имеет возможность добавлять материал и редактировать его. Ну и последняя роль – администратор, его возможности будут описаны далее.

2.4 Руководство администратора

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

Новые возможности можно будет увидеть на странице профиля, где появляются новые кнопки «Админ панель» и «Создание ролей».

Создание ролей говорит само за себя, здесь администратор имеет возможность добавлять новые роли пользователей, например, для дальнейшего разделения функционала, также редактировать существующие или и вовсе удалять их. Для создания новой роли необходимо заполнить пустое поле, ввести туда название и нажать на кнопку добавления. После чего под формой с заполнением отобразится созданные данные, рядом с которыми будут две кнопки, изменения и удаления. При нажатии на первую (“Изменить”) в поле над таблицей появится выбранная строка, которую можно отредактировать, нажать на ту же кнопку, что и при создании и данные в таблице изменятся. Кнопка удаление удаляет строку и на сайте, и в базе данных.

Перейдем к панели для администратора - здесь отображается список всех зарегистрированных пользователей, их имя, электронную почту, дата регистрации, роль и кнопку для их удаления. Роль пользователя отображается в виде выпадающего списка, чтобы ее изменить на сайте и в базе данных необходимо просто открыть этот список и выбрать необходимое значение, после чего страница обновляется вместе с ролью. Удаление работает, как и везде, администратор выбирает строку, нажимает кнопку и запись о пользователе удаляется и на сайте, и в БД.

ЗАКЛЮЧЕНИЕ

При разработке электронного учебника «Операционные системы и среды» была достигнута цель курсовой работы – создание приложения для работы с базой данных - и решены задачи для ее реализации. Было также проделана работа с ролями, то есть разделение зарегистрированных пользователей на группы, такие как пользователь, администратор и модератор, и доступ каждой к разным возможностям.

В результате выполнения курсовой, был создан динамический сайт для внесения и хранения там данных о темах учебника и пользователях. Для его разработки применялось актуальное ПО, без помощи которого не было бы возможности добиться поставленной цели. Это программное обеспечение включает в себя:

* язык программирования PHP;
* веб-сервер XAMPP;
* реляционная система управления БД MySQL;
* приложение PHPMyAdmin.

С помощью этих инструментов значительно сокращена ручная работа и повторяющийся код.

Таким образом, затрачено меньше времени на разработку сайта. Всё информационное наполнение сайта находится в БД.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дронов, В. JavaScript в Web-дизайне / В. Дронов. - М.: СПб: БХВ, 2019. - 880 c.
2. Дронов, В. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов / В. Дронов. - М.: БХВ-Петербург, 2019. - 480 c.
3. Дронов, В.А. PHP 5/6, MySQL 5/6 и Dreamweaver CS4. Разработка интерактивных Web-сайтов / В.А. Дронов. - М.: БХВ-Петербург, 2019. - 820 c.
4. Дронов, Владимир JavaScript и AJAX в Web-дизайне / Владимир Дронов. - Москва: Высшая школа, 2019. - 736 c.
5. Дронов, Владимир PHP, MySQL и Dreamweaver MX 2004. Разработка интерактивных Web-сайтов / Владимир Дронов. - М.: "БХВ-Петербург", 2019. - 448 c.
6. Колисниченко, Д. PHP 5/6 и MySQL 6. Разработка Web-приложений / Д. Колисниченко. - М.: БХВ-Петербург, 2021. - 560 c.
7. Колисниченко, Д.Н. PHP 5/6 и MySQL 6. Разработка Web-приложений / Д.Н. Колисниченко. - М.: БХВ-Петербург, 2021. - 364 c.
8. Кузнецов PHP. Практика создания Web-сайтов / Кузнецов, М.В. и. - М.: БХВ-Петербург, 2020. - 895 c.
9. Кузнецов, М. PHP 5. Практика создания Web-сайтов / М. Кузнецов, И. Симдянов, С. Голышев. - М.: БХВ-Петербург, 2020. - 960 c.
10. Леонтьев, Борис PHP 5.0 для начинающих, или как создать динамический WEB-сайт / Борис Леонтьев. - М.: Новый издательский дом, 2019. - 176 c.
11. Петюшкин, Алексей HTML в Web-дизайне / Алексей Петюшкин. - Москва: Машиностроение, 20. - 400 c.
12. Шкрыль, А. PHP - это просто. Программируем для Web-сайта / А. Шкрыль. - М.: БХВ-Петербург, 2021. - 368 c.

ПРИЛОЖЕНИЕ

Style.css

Используется для визуализации сайта.

/\*УБИРАЕМ ПРОКРУТКУ\*/

html::-webkit-scrollbar {

display: none;}

body {

margin: auto;

background-color: #89A1EF;

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

font-weight: 100;

font-size: 32px;

color: white;}

.blog {

margin: 30px;}

.header {

padding: 20px;

background-color: #7489CC;

display: flex;}

.footer {

display: flex;

padding: 20px;

background-color: #7489CC;

color: white;

font-size: 20px;

font-family: 'Montserrat', sans-serif;}

.year {

margin: auto;}

.city {

margin: auto 0;}

.logo {

margin-left: auto;}

#logotype {

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

font-style: italic;

font-weight: 600;

font-size: 48px;

color: #7489CC;

text-shadow: 0px 0 0px #32CBFF,

/\*ОБВОДКА ТЕКСТА\*/

0 2px 5px #32CBFF,

-2px 0 5px #32CBFF,

0 -2px 5px #32CBFF;}

#btn {

display: inline-block;

text-decoration: none;

text-align: center;

font-family: inherit;

font-size: 36px;

width: 185px;

background-color: #32CBFF;

border: none;

color: white;

border-radius: 10px;

box-shadow: 5px 5px 5px #20708b;

padding: 5px;}

.title {

margin: inherit;

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

font-style: italic;

font-weight: 600;

font-size: 48px;

text-align: center;

border-bottom: 1px solid #32cbff;

padding-top: 65px;

padding-bottom: 30px;}

.title\_content {

text-align: center;

font-size: 45px;

padding-top: 3%;

padding-bottom: 50px;}

li {

font-size: 32px;

padding-bottom: 35px;}

.bloks {

display: flex;

margin: 10px;}

.text\_bloks {

width: 40%;

padding: 30px;

background-color: #00a4e08f;}

.text\_bloks\_two {

width: 45%;

padding: 30px;

margin-left: auto;

background-color: #00a4e08f;}

.educate {

padding: 30px;

text-align: center;}

.text {

font-size: 45px;

padding: 20px;}

label {

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

font-size: 25px;

color: white;

display: flex;}

input {

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

color: white;

font-weight: 300;

margin-bottom: 30px;

border: none;

border-bottom: 2px solid #7489CC;

background-color: #89A1EF;

width: -webkit-fill-available;

height: 2.25em;

font-size: 25px;}

a {

text-decoration: none;}

.body-knopochki {

display: flex;

padding: 20px 0px;}

#btn-registr {

display: block;

padding: 5px;

border-radius: 5px;

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

font-style: italic;

color: white;

box-shadow: none;

background: none;

font-size: 30px;

font-weight: 300;}

#btn-registr:hover {

box-shadow: 5px 5px 5px #7489CC;

font-weight: 400;

width: 100%;}

.btn-login {

margin-left: auto;}

#btn-two {

padding: 5px 25px;

margin-left: auto;

font-size: 30px;

display: inline-block;

text-decoration: none;

text-align: center;

font-family: inherit;

width: auto;

background-color: #32CBFF;

border: none;

color: white;

border-radius: 10px;

box-shadow: 5px 5px 5px #20708b;}

/\* свойства для блока, содержащего контент модального окна \*/

.login-content {

position: relative;

display: flex;

-webkit-box-orient: vertical;

-webkit-box-direction: normal;

flex-direction: column;

background-color: #89A1EF;

background-clip: padding-box;}

/\* свойства для заголовка модального окна \*/

.login-header {

display: flex;

-webkit-box-align: center;

align-items: center;

-webkit-box-pack: justify;

justify-content: space-between;

padding: 5% 10%;}

.modal-title {

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

font-style: italic;

color: #ffffff;

margin-top: 0;

margin-bottom: 0;

font-size: 2rem;

font-weight: 600;}

/\* свойства для блока, содержащего основное содержимое окна \*/

.login-body {

position: relative;

padding: 2% 10%;

overflow: auto;}

/\*-------------РЕГИСТРАЦИЯ-----------------\*/

.container-regist {

margin: 5%;}

.title-name {

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

font-style: italic;

color: #ffffff;

margin-bottom: 2%;

padding-bottom: 2%;

font-size: 3rem;

font-weight: 600;

border-bottom: 1px solid #32CBFF;}

.main {

padding-bottom: 20px;}

#main-btn {

color: inherit;}

.regist-info {

padding-bottom: 20px;

font-size: 25px;}

.regist-body {

margin: 2%;

border: 1px solid #7c90d1;

padding: 5%;}

.saveUser {

text-align: right;}

.web-info {

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

font-style: italic;

margin: auto 60px;

padding: 0px 35px;

font-size: 32px;

border: 1px solid #32CBFF;}

.os-info {

display: flex;

margin: auto 60px;}

.os-text {

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

font-style: italic;}

#profile {

font-weight: 500;

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

background-color: #89A1EF;

padding-top: 7px;

margin: 0px 20px;

border-radius: 5px;

text-align: center;

width: 200px;

color: white;

box-shadow: 5px 5px 5px #32CBFF;}

.welcome\_name {

font-weight: 300;

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

font-size: 25px;

text-align: center;

width: 350px;}

#action\_btn {

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

font-weight: 300;

display: flex;

justify-content: center;

background-color: #7489CC;

padding: 20px;

margin: 20px 20px;

border-radius: 5px;

color: white;}

.user\_action {

/\*height: 78.25vh;\*/

width: 55%;

margin: 60px auto;}

#logout {

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

font-size: inherit;

color: white;

padding: 20px;

height: 80px;

margin: 0px 20px;

font-weight: 300;

border: none;

border-radius: 5px;

background-color: rgb(58, 30, 30);}

/\*test\*/

.test {

margin: 60px;}

#rad {

width: 50px;

font-size: 10px;}

/\*РОЛИ\*/

.role\_table {

border: 1px solid #32CBFF;

padding: 30px;

text-align: center;

width: fit-content;

margin: 0 auto;}

#del\_btn {

margin: 10px 0px;

width: 200px;

border-radius: 5px;

background-color: rgb(90, 18, 18);}

th{

padding: 5px 10px 35px 20px;}

td{

padding: 10px 15px 15px 15px;}

/\*АДМИНКА\*/

select{

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

font-size: inherit;

font-weight: 100;

border: 1px solid white;

color: white;

padding: 20px;

background-color: #89A1EF;

font-size: 25px;

}

Index.php

Код главной страницы

<?php

session\_start();

// Проверка, авторизован ли пользователь

if (!isset($\_SESSION["email"])) {

header("Location: login.php"); // Перенаправление на страницу входа, если пользователь не авторизован

exit();

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>ОС</title>

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

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Jura:wght@300;500;700&family=Montserrat:ital,wght@0,100;0,300;0,400;0,500;0,600;1,100;1,300;1,400;1,500;1,600&display=swap" rel="stylesheet">

</head>

<body>

<div class="container">

<div class="header">

<?php

if (isset($\_SESSION["email"])) {

?>

<a href="admin/profile.php" id="profile" type="s">Профиль</a>

<?php

} ?>

<div class="logo"><a href="index.php" id="logotype" type="s">Web education</a></div>

</div>

<div class="blog">

<div class="web-info">

<p align="justify"><a href="index.php" id="logotype" type="s">Web education</a> - это современное средство обучения, в котором системно излагается

материал в определённой области знаний, на современном уровне достижений науки и техники. Концепция электронных учебников

состоит в том, чтобы сделать их не просто заменителями бумажных пособий, а инструментом обучения с расширенными возможностями

по сравнению с традиционными учебниками. Основное преимущество электронного пособия — интерактивность.</p>

</div>

<div class="title">ОПЕРАЦИОННЫЕ СИСТЕМЫ И СРЕДЫ</div>

<div class="os-info">

<div class="img" style="padding-top: 50px; height: 300px;">

<img src="img/os.png" style="width: 300px;" alt="Операционные системы">

</div>

<div class="os-text" style="padding-left: 45px;">

<p align="justify"><b>Операционная система, сокр. ОС (англ. operating system, OS)</b> — комплекс управляющих и обрабатывающих программ,

которые, с одной стороны, выступают как интерфейс между устройствами вычислительной системы и прикладными программами,

а с другой стороны — предназначены для управления устройствами, управления вычислительными процессами, эффективного

распределения вычислительных ресурсов между вычислительными процессами и организации надёжных вычислений.</p>

</div>

</div>

<div class="content">

<div class="title\_content">Содержание курса</div>

<div class="bloks">

<div class="text\_bloks">

<ul>

<li><a href="theme.php">Теоретическое введение</a></li>

<li>Файловые системы</li>

<li>Управление памятью в ОС</li>

<li>Процессы</li>

<li>Пользователи системы</li>

</ul>

</div>

<div class="text\_bloks\_two">

<ul>

<li>Файлы пользователей</li>

<li>Управление пользователями</li>

<li>Прикладное программирование</li>

<li>Межпроцессорное взаимодействие</li>

</ul>

</div>

</div>

</div>

<div class="educate">

<div class="text">Приступить к обучению!</div>

<a href="#" id="btn" type="submit">Начать</a>

</div>

</div>

</div>

<div class="footer">

<div class="contacts">

<div class="contact-info">Связь с нами<br></div>

<div class="email"> [pochta@mail.ru<br></div](mailto:pochta@mail.ru%3cbr%3e%3c/div)>

<div class="phoneNum"> +7(999)-000-09-90</div>

</div>

<div class="year">

<?php $today = date("F j, Y, H:i");

echo $today; ?>

</div>

<div class="city">Белореченск</div>

</div>

</body>

</html>

Login.php

Код для страницы авторизации

<?php

session\_start();

// Подключение к базе данных (замените параметры подключения своими)

$servername = "localhost";

$username = "web";

$password = "123";

$dbname = "webeducation";

$conn = new mysqli($servername, $username, $password, $dbname);

// Проверка соединения

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

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

if (strlen($\_POST['password']) < 6) {

$errmsg = "Ваш пароль меньше 6 символов.";

} else {

$email = $\_POST['email'];

$password = $\_POST['password'];

// Вставка данных в базу данных

$sql = "INSERT INTO users (email, password, name) VALUES ('{$\_POST['email']}','{$\_POST['password']}','{$\_POST['name']}')";

$sql2 = "SELECT \* FROM users WHERE email = '$email' AND password = '$password'";

$result = mysqli\_query($conn, $sql2);

if ($conn->query($sql) === TRUE) {

// Успешная регистрация

$\_SESSION['email'] = $\_POST['email'];

$okmsg = "Вы успешно вошли. <a href='index.php'>Главная</a>";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

}

}

$conn->close();

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Регистрация</title>

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

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Jura:wght@300;500;700&family=Montserrat:ital,wght@0,100;0,300;0,400;0,500;0,600;1,100;1,300;1,400;1,500;1,600&display=swap" rel="stylesheet">

</head>

<body>

<div class="container">

<div class="header">

<div class="logo"><a href="index.php" id="logotype" type="s">Web education</a></div>

</div>

<div class="container-regist">

<div class="title-regist">

<div class="title-name">Регистрация</div>

<!--<div class="main">

<a href="index.php" id="main-btn" type="s"> Вернуться на главную страницу</a>

</div>-->

<div class="regist-info">

Чтобы получить полный доступ к систему, вам необходимо завести учетную запись.

</div>

</div>

<div class='regist-body'>

<?php if (isset($errmsg)) { ?>

<span class='errmsg'>Opss... <?= $errmsg ?></span>"

<?php } ?>

<?php if (isset($okmsg)) { ?>

<span class='errmsg'> <?= $okmsg ?></span>"

<?php } ?>

<form method='post'>

<label for='name'>Имя </label>

<input type='text' id='' required name='name' placeholder='Иван'>

<label for='email'>Элекронная почта</label>

<input type='email' id='' required name='email' [placeholder='email@mail.ru](mailto:placeholder='email@mail.ru)'>

<label for='password'>Пароль</label>

<input type='password' id='' required name='password' placeholder='\*\*\*\*\*\*\*\*\*'>

<div class='regist-btn'>

<div class='captcha'>

</div>

<div class='saveUser'>

<input id='btn' type='submit' value='Создать'>

</div>

</div>

</form>

</div>

</div>

<div class="footer">

<div class="contacts">

<div class="contact-info">Связь с нами<br></div>

<div class="email"> [pochta@mail.ru<br></div](mailto:pochta@mail.ru%3cbr%3e%3c/div)>

<div class="phoneNum"> +7(999)-000-09-90</div>

</div>

<div class="year">

<?php $today = date("F j, Y, H:i");

echo $today; ?>

</div>

<div class="city">Белореченск</div>

</div>

</div>

</body>

</html>

Registration.php

Код для страницы регситрации

<?php

session\_start();

// Подключение к базе данных (замените параметры подключения своими)

$servername = "localhost";

$username = "web";

$password = "123";

$dbname = "webeducation";

$conn = new mysqli($servername, $username, $password, $dbname);

// Проверка соединения

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

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

if (strlen($\_POST['password']) < 6) {

$errmsg = "Ваш пароль меньше 6 символов.";

} else {

$email = $\_POST['email'];

$password = $\_POST['password'];

// Вставка данных в базу данных

$sql = "INSERT INTO users (email, password, name) VALUES ('{$\_POST['email']}','{$\_POST['password']}','{$\_POST['name']}')";

$sql2 = "SELECT \* FROM users WHERE email = '$email' AND password = '$password'";

$result = mysqli\_query($conn, $sql2);

if ($conn->query($sql) === TRUE) {

// Успешная регистрация

$\_SESSION['email'] = $\_POST['email'];

$okmsg = "Вы успешно вошли. <a href='index.php'>Главная</a>";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

}

}

$conn->close();

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Регистрация</title>

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

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Jura:wght@300;500;700&family=Montserrat:ital,wght@0,100;0,300;0,400;0,500;0,600;1,100;1,300;1,400;1,500;1,600&display=swap" rel="stylesheet">

</head>

<body>

<div class="container">

<div class="header">

<div class="logo"><a href="index.php" id="logotype" type="s">Web education</a></div>

</div>

<div class="container-regist">

<div class="title-regist">

<div class="title-name">Регистрация</div>

<!--<div class="main">

<a href="index.php" id="main-btn" type="s"> Вернуться на главную страницу</a>

</div>-->

<div class="regist-info">

Чтобы получить полный доступ к систему, вам необходимо завести учетную запись.

</div>

</div>

<div class='regist-body'>

<?php if (isset($errmsg)) { ?>

<span class='errmsg'>Opss... <?= $errmsg ?></span>"

<?php } ?>

<?php if (isset($okmsg)) { ?>

<span class='errmsg'> <?= $okmsg ?></span>"

<?php } ?>

<form method='post'>

<label for='name'>Имя </label>

<input type='text' id='' required name='name' placeholder='Иван'>

<label for='email'>Элекронная почта</label>

<input type='email' id='' required name='email' [placeholder='email@mail.ru](mailto:placeholder='email@mail.ru)'>

<label for='password'>Пароль</label>

<input type='password' id='' required name='password' placeholder='\*\*\*\*\*\*\*\*\*'>

<div class='regist-btn'>

<div class='captcha'>

</div>

<div class='saveUser'>

<input id='btn' type='submit' value='Создать'>

</div>

</div>

</form>

</div>

</div>

<div class="footer">

<div class="contacts">

<div class="contact-info">Связь с нами<br></div>

<div class="email"> [pochta@mail.ru<br></div](mailto:pochta@mail.ru%3cbr%3e%3c/div)>

<div class="phoneNum"> +7(999)-000-09-90</div>

</div>

<div class="year">

<?php $today = date("F j, Y, H:i");

echo $today; ?>

</div>

<div class="city">Белореченск</div>

</div>

</div>

</body>

</html>

Theme.php

Код для страницы задания

<?php

session\_start();

// Проверка, авторизован ли пользователь

if (!isset($\_SESSION["email"])) {

header("Location: login.php"); // Перенаправление на страницу входа, если пользователь не авторизован

exit();

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>ОС</title>

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

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Jura:wght@300;500;700&family=Montserrat:ital,wght@0,100;0,300;0,400;0,500;0,600;1,100;1,300;1,400;1,500;1,600&display=swap" rel="stylesheet">

</head>

<body>

<div class="container">

<div class="header">

<?php

if (isset($\_SESSION["email"])) {

?>

<a href="admin/profile.php" id="profile" type="s">Профиль</a>

<?php

} ?>

<div class="logo"><a href="index.php" id="logotype" type="s">Web education</a></div>

</div>

<div class="blog">

<div class="title">Основные понятия</div>

<div class="content" style="margin:60px;font-size:25px;">

<p align="justify">&nbsp;&nbsp;&nbsp;При решении своих задач в среде операционной системы пользователь должен определить данные и инструментальное (программное) средство для их обработки, В большинстве случаев решение задачи пользователя сводится к последовательном применению нескольких инструментов (например, ввода данных, сортировки, слияния, вывода].

</p>

<p align="justify">&nbsp;&nbsp;&nbsp;Операционная система предоставляет пользователю базовый набор инструментов и среду для хранения данных, а также средства управления последовательностью использования инструментов Интервал времени, в течение которого пользователь решает одну или несколько последовательных задач, пользуясь при этом средствами, предоставляемыми ОС, называется сеансом.

В начале любого сеанса пользователь идентифицирует себя, в конце указывает, что сеанс закончен, Описание последовательности использования программных инструментов, записанное па некотором формальном языке, называется заданием, а сам язык — языком управления заданиями.

Выполнение заданий в большинстве операционных систем производится командным интерпретатором, более подробное определение которого будет дано далее.

</p>

<p align="justify">&nbsp;&nbsp;&nbsp;Обычно пользователю предоставляется некоторый интерфейс общения с командным интерпретатором, при использовании которого команды вводятся с клавиатуры, а результат их выполнения выводится на экран. Такой интерфейс ассоциируется с логическим понятием терминала — совокупности устройства ввода (обычно клавиатуры} и устройства вывода (дисплея, выводящего текстовую информацию). В настоящее время более употребительным является графический интерфейс пользователя (С), рассмотрение которого выходит за рамки данного учебника.

</p>

<div class="img" style="display:flex;text-align: center;">

<img src="img/po.png" style="width: 300px;margin:0px 100px;" alt="Ой">

<p align="justify">&nbsp;&nbsp;&nbsp;Программа — набор инструкций процессора,хранящийся на диске [или другом накопителе информации). Для того чтобы программа могла быть запущена на выполнение, оле-рационная система должна создать среду выполнения — информационное окружение решаемой задачи. После этого операционная система перемещает исполняемый код и данные программы в оперативную память и инициирует выполнение программы.

</p>

</div>

<p align="justify">&nbsp;&nbsp;&nbsp;Операционная система выполняет функции управления аппаратными ресурсами, их распределения между выполняемыми программами пользователя и формирует среду исполнения, которая содержит все данные, необходимые для программы. Такая среда в дальнейшем и будет называться информационным окружением. В информационное окружение входят данные и объекты, обрабатываемые операционной системой, которые влияют на выполнение программы, т.е. на решение задачи пользователя, В ходе дальнейшего изложения будут приведены примеры информационного окружения различного характера.

</p>

<p align="justify">&nbsp;&nbsp;&nbsp;Используя понятия программы, данных и информационного окружения, можно определить понятие задачи в среде ОС как совокупность программы и данных, являющихся частью информационного окружения.

</p>

<p align="justify">&nbsp;&nbsp;&nbsp;Выполняемая программа образует процесс. Процесс представляет собой совокупность информационного окружения и области памяти, содержащей исполняемый код и данные программы. Обычно в памяти, контролируемой операционной системой, может одновременно работать большое число процессов.

</p>

<p align="justify"> &nbsp;&nbsp;&nbsp;Вполне естественно, что на однопроцессорных компьютерах возможно одновременное выполнение программного кола только одного процесса, поэтому часть процессов находятся в режиме ожидания, а один из процессов — в режиме выполнения. Процессы при этом образуют очередь, операционная система передает управление первому процессу в очереди, затем следующему ит.л,

</p>

<p align="justify"> &nbsp;&nbsp;&nbsp;Процесс, имеющий потенциальную возможность получить входные данные от пользователя с клавиатуры и вывести результаты своей работы на экран, называется процессом переднего плана; процесс, выполняемый без непосредственного взаимодействия с пользователем, — фоновым процессом.

</p>

<p align="justify">&nbsp;&nbsp;&nbsp; В ходе своей работы процессы используют вычислительную мощность процессора, оперативную память, обращаются к внеш ним файлам, внутренним данным ядра операционной системы. Все эти объекты входят в информационное окружение процесса и называются ресурсами.

</p>

<p align="justify">&nbsp;&nbsp;&nbsp;Ресурсом может быть как физический объект, к которому ОС предоставляет доступ, — процессор, оперативная память, дисковые накопители, так и логический объект, который существует только в пределах самой ОС, например таблица выполняемых процессов или сетевых подключений. Необходимость в управлении ресурсами со стороны ОС вызвана в первую очередь тем, что ресурсы ограничены (по объему, времени использования, количеству обслуживаемых пользователей и т. п.). Операционная система в данной ситуации либо управляет ограничениями ресурсов, предотвращая их исчерпание, либо предоставляет средства обработки ситуаций, связанных с исчерпанием ресурсов. Лимиты многих ресурсов, заданные в ОС по умолчанию, могут изменяться затем администратором системы. Примером такого ресурса может служить максимальное количество файлов, одновременно открытых пользователем.

</p>

</div>

<div class="title">Тестирование</div>

<div class="test">

<p style="font-size:32px;text-align:center;">Вопрос 1: Что такое HTML?</p>

<input type="radio" id="rad" name="question1" value="1"> 1. Гиперссылка<br>

<input type="radio" id="rad" name="question1" value="2"> 2. Язык программирования<br>

<input type="radio" id="rad" name="question1" value="3"> 3. Язык разметки гипертекста<br>

<p style="font-size:32px;text-align:center;">Вопрос 2: Что такое CSS?</p>

<input type="radio" id="rad" name="question2" value="1"> 1. Computer Style Sheets<br>

<input type="radio" id="rad" name="question2" value="2"> 2. Cascading Style Sheets<br>

<input type="radio" id="rad" name="question2" value="3"> 3. Common Style Sheets<br>

<p style="font-size:32px;text-align:center;">Вопрос 3: Что такое JavaScript?</p>

<input type="radio" id="rad" name="question3" value="1"> 1. Мультипарадигменный язык программирования <br>

<input type="radio" id="rad" name="question3" value="2"> 2. Язык разметки гипертекста<br>

<input type="radio" id="rad" name="question3" value="3"> 3. Язык программирования Python<br>

<button id="btn" style=" margin-top: 40px;width: auto;" onclick="submitTest()">Отправить</button>

</div>

<script>

function submitTest() {

// Получаем выбранные значения из радиокнопок

var answer1 = document.querySelector('input[name="question1"]:checked').value;

var answer2 = document.querySelector('input[name="question2"]:checked').value;

var answer3 = document.querySelector('input[name="question3"]:checked').value;

// Проверяем правильность ответов

var score = 0;

if (answer1 === "3") {

score += 1;

}

if (answer2 === "2") {

score += 1;

}

if (answer3 === "1") {

score += 1;

}

// Выводим результат теста

alert("Ваш результат: " + score + " из 3");

}

</script>

</div>

<div class="footer">

<div class="contacts">

<div class="contact-info">Связь с нами<br></div>

<div class="email"> [pochta@mail.ru<br></div](mailto:pochta@mail.ru%3cbr%3e%3c/div)>

<div class="phoneNum"> +7(999)-000-09-90</div>

</div>

<div class="year">

<?php $today = date("F j, Y, H:i");

echo $today; ?>

</div>

<div class="city">Белореченск</div>

</div>

</body>

</html>

404.php

Код для страницы с ошибкой

<?php

$servername = "localhost";

$username = "web";

$password = "123";

$dbname = "webeducation";

$conn = new mysqli($servername, $username, $password, $dbname);

$query1 = "SELECT \* FROM users";

$result1 = mysqli\_query($conn, $query1) or die(mysqli\_error($conn));

session\_start();

// Проверка, авторизован ли пользователь

if (!isset($\_SESSION["email"])) {

header("Location: ../login.php"); // Перенаправление на страницу входа, если пользователь не авторизован

exit();

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Профиль</title>

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

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Jura:wght@300;500;700&family=Montserrat:ital,wght@0,100;0,300;0,400;0,500;0,600;1,100;1,300;1,400;1,500;1,600&display=swap" rel="stylesheet">

</head>

<body>

<div class="container">

<div class="header">

<?php

if (isset($\_SESSION["email"])) {

?>

<a href="admin/profile.php" id="profile" type="s">Профиль</a>

<?php

} ?>

<div class="logo"><a href="index.php" id="logotype" type="s">Web education</a></div>

</div>

<div class="404-body">

<div class="404\_text" style="text-align: center; margin:60px;">

ОЙ.. Кажется такой страницы больше не существует :с

</div>

<div class="img" style="text-align: center; margin:60px;">

<img src="img/404.png" style="width: 600px;" alt="Ой">

</div>

<div class="404\_btn" style="text-align: center; margin:60px; ">

<a href="index.php" style="color:white; border:1px solid white;padding:20px;">Вернуться на главную</a>

</div>

</div>

<div class="footer">

<div class="contacts">

<div class="contact-info">Связь с нами<br></div>

<div class="email"> [pochta@mail.ru<br></div](mailto:pochta@mail.ru%3cbr%3e%3c/div)>

<div class="phoneNum"> +7(999)-000-09-90</div>

</div>

<div class="year">

<?php $today = date("F j, Y, H:i");

echo $today; ?>

</div>

<div class="city">Белореченск</div>

</div>

</body>

</html>

Admin-panel.php

Код для страницы админ панели

<?php

$servername = "localhost";

$username = "web";

$password = "123";

$dbname = "webeducation";

$conn = new mysqli($servername, $username, $password, $dbname);

session\_start();

// Проверка, авторизован ли пользователь

if (!isset($\_SESSION["email"]) || $\_SESSION['role'] != "Администратор") {

header("Location: ../login.php"); // Перенаправление на страницу входа, если пользователь не авторизован

exit();

}

?>

<!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 href="../css/style.css" rel="stylesheet">

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Jura:wght@300;500;700&family=Montserrat:ital,wght@0,100;0,300;0,400;0,500;0,600;1,100;1,300;1,400;1,500;1,600&display=swap" rel="stylesheet">

</head>

<body>

<div class="container">

<div class="header">

<?php

if (isset($\_SESSION["email"])) {

?>

<a href="profile.php" id="profile" type="s">Профиль</a>

<?php

} ?>

<?php

if (!isset($\_SESSION["email"])) { ?>

<a href="../login.php" id="btn" type="s">Войти</a>

<?php

} ?>

<div class="logo"><a href="../index.php" id="logotype" type="s">Web education</a></div>

</div>

<div class="list\_body" style="margin: 50px;">

<div class="role\_table">

<table>

<tr>

<th>№</th>

<th>Имя пользователя</th>

<th>Электронная почта</th>

<th>Роль</th>

<th>Дата регистрации</th>

<th>Удаление</th>

</tr>

<?php

if ($\_POST['user\_role\_id'] > 0) {

if (isset($\_POST['role\_s']))

$user\_role\_update = "UPDATE users set role='{$\_POST['role\_s']}' WHERE id='{$\_POST['user\_role\_id']}'";

$result\_role\_users = mysqli\_query($conn, $user\_role\_update) or die(mysqli\_error($conn));

}

$user\_role = "SELECT \* FROM category WHERE url='list-role.php'";

$result\_role\_category = mysqli\_query($conn, $user\_role) or die(mysqli\_error($conn));

$all\_roles\_array = array();

while ($role = mysqli\_fetch\_assoc($result\_role\_category)) {

array\_push($all\_roles\_array, $role['name']); //массив с запоминанием всех существующих ролей

}

$num = 0;

$all\_user = "SELECT \* FROM users";

$result = mysqli\_query($conn, $all\_user) or die(mysqli\_error($conn));

while ($user = mysqli\_fetch\_assoc($result)) {

?>

<tr>

<td><?= ++$num ?></td>

<td><?= $user['name'] ?></td>

<td><?= $user['email'] ?></td>

<td>

<form method='POST'>

<input type='hidden' name='user\_role\_id' value='<?= $user['id'] ?>'>

<select name='role\_s' onchange='this.form.submit()'>

<?php

foreach ($all\_roles\_array as $role) {

$selected = '';

if ($role == $user['role']) {

$selected = 'selected';

}

?>

<option <?= $selected ?>><?= $role ?></option>

<?php } ?>

</select>

</form>

</td>

<td><?= $user['date'] ?></td>

<td>

<form action='delete-user.php' method='post'>

<input type='hidden' name='id' value='<?= $user['id'] ?>'>

<input type='submit' id="del\_btn" class='btn' value='Удалить'>

</form>

</td>

</tr>

<?php }

?>

</table>

</div>

</div>

<div class="footer">

<div class="contacts">

<div class="contact-info">Связь с нами<br></div>

<div class="email"> [pochta@mail.ru<br></div](mailto:pochta@mail.ru%3cbr%3e%3c/div)>

<div class="phoneNum"> +7(999)-000-09-90</div>

</div>

<div class="year">

<?php $today = date("F j, Y, H:i");

echo $today; ?>

</div>

<div class="city">Белореченск</div>

</div>

</body>

</html>

List-role.php

Код со списком ролей

<?php

$servername = "localhost";

$username = "web";

$password = "123";

$dbname = "webeducation";

$conn = new mysqli($servername, $username, $password, $dbname);

session\_start();

// Проверка, авторизован ли пользователь

if (!isset($\_SESSION["email"]) || $\_SESSION['role'] != "Администратор") {

header("Location: ../login.php"); // Перенаправление на страницу входа, если пользователь не авторизован

exit();

}

if (isset($\_POST['name'])) {

$url\_full = basename($\_SERVER['REQUEST\_URI']);

$url\_array = explode("?", $url\_full);

$url = $url\_array[0];

if (isset($\_POST['name'])) {

if (isset($\_GET['red\_id'])) {

$sql\_up = "UPDATE category SET name = '{$\_POST['name']}' WHERE id={$\_GET['red\_id']}";

$result\_up = mysqli\_query($conn, $sql\_up);

} else {

$sql\_in = "INSERT INTO category (name, parent\_id, url, email) VALUES ('{$\_POST['name']}','0', '{$url}','{$\_SESSION['email']}')";

$result\_in = mysqli\_query($conn, $sql\_in);

header("Location: list-role.php");

}

}

}

//Если передана переменная red\_id, то надо обновлять данные. Для начала достанем их из БД

if (isset($\_GET['red\_id'])) {

$sql\_re = "SELECT id, name FROM category WHERE id={$\_GET['red\_id']}";

$result\_re = mysqli\_query($conn, $sql\_re);

$product = mysqli\_fetch\_array($result\_re);

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Создание ролей</title>

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

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Jura:wght@300;500;700&family=Montserrat:ital,wght@0,100;0,300;0,400;0,500;0,600;1,100;1,300;1,400;1,500;1,600&display=swap" rel="stylesheet">

</head>

<body>

<?

include "includes/list.php";

?>

<div class="header">

<?php

if (isset($\_SESSION["email"])) {

?>

<a href="profile.php" id="profile" type="s">Профиль</a>

<?php

} ?>

<?php

if (!isset($\_SESSION["email"])) { ?>

<a href="login.php" id="btn" type="s">Войти</a>

<?php

} ?>

<div class="logo"><a href="../index.php" id="logotype" type="s">Web education</a></div>

</div>

<div class="list\_body" style="margin: 50px;">

<div class="title"> Создание списка </div>

<form method="post" style="margin: 50px;">

<label for='name'>Имя </label>

<input type='text' id='' value="<?= isset($\_GET['red\_id']) ? $product['name'] : ''; ?>" required name='name' placeholder='Введите имя элемента'>

<div class='role-btn'>

<input id='btn' type='submit' value='Добавить'>

</div>

</form>

<div class="role\_table" >

<table>

<?php

$num = 0;

$task = "SELECT \* FROM category";

$resultTask = mysqli\_query($conn, $task) or die(mysqli\_error($conn));

echo "<thead> <tr>

<th>Номер</th>

<th>Имя</th>

<th>Редактирование</th>

<th>Удаление</th>

</tr>

</thead>

<tbody>

<tr>";

while ($item = mysqli\_fetch\_assoc($resultTask)) {

echo "<td>" . ++$num . "</td>

<td>" . $item['name'] . "</td>

<td><a href='?red\_id={$item['id']}'>Изменить</a></td>

<td><form action='delete-role.php' method='post'>

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

<input type='submit' id='del\_btn' value='Удалить'>

</form></td>

</tr></tbody>";

}

?>

</table>

</div>

</div>

<div class="footer">

<div class="contacts">

<div class="contact-info">Связь с нами<br></div>

<div class="email"> [pochta@mail.ru<br></div](mailto:pochta@mail.ru%3cbr%3e%3c/div)>

<div class="phoneNum"> +7(999)-000-09-90</div>

</div>

<div class="year">

<?php $today = date("F j, Y, H:i");

echo $today; ?>

</div>

<div class="city">Белореченск</div>

</div>

</body>

</html>

Profile.php

Профиль пользователя

ф<?php

$servername = "localhost";

$username = "web";

$password = "123";

$dbname = "webeducation";

$conn = new mysqli($servername, $username, $password, $dbname);

$query1 = "SELECT \* FROM users";

$result1 = mysqli\_query($conn, $query1) or die(mysqli\_error($conn));

session\_start();

// Проверка, авторизован ли пользователь

if (!isset($\_SESSION["email"])) {

header("Location: ../login.php"); // Перенаправление на страницу входа, если пользователь не авторизован

exit();

}

// Если пользователь авторизован и отправил запрос на выход

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

// Разрушение сессии

session\_unset();

session\_destroy();

header("Location: ../login.php"); // Перенаправление на страницу входа

exit();

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Профиль</title>

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

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Jura:wght@300;500;700&family=Montserrat:ital,wght@0,100;0,300;0,400;0,500;0,600;1,100;1,300;1,400;1,500;1,600&display=swap" rel="stylesheet">

</head>

<body>

<div class="container">

<div class="header">

<?php

if (isset($\_SESSION["email"])) {

?>

<div class="welcome\_name">

<?php

$username = $\_SESSION['email']; // получаем имя авторизованного пользователя

echo "Добро пожаловать, <u><i>" . $username . "</i></u>"; // выводим сообщение

} ?>

</div>

<div class="logo"><a href="../index.php" id="logotype" type="s">Web education</a></div>

</div>

<div class="user\_info">

<?php

// Выводим информацию об авторизованном пользователе

?>

</div>

<div class="user\_action">

<?php

if (isset($\_SESSION['email']) && $\_SESSION['role'] == "Администратор") {

?>

<a href="../404.php" id="action\_btn"> Управление занятиями</a>

<a href="list-role.php" id="action\_btn">Роли пользователей</a>

<a href="admin-panel.php" id="action\_btn">Админ панель</a><?php } ?>

<?php

if (isset($\_SESSION['email']) && $\_SESSION['role'] == "Модератор") {

?>

<a href="../404.php" id="action\_btn"> Управление занятиями</a>

<?php }

if (isset($\_SESSION["email"])) {

?>

<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>">

<input type="submit" id="logout" value="Выйти из профиля">

</form>

<?php

} ?>

</div>

<div class="footer">

<div class="contacts">

<div class="contact-info">Связь с нами<br></div>

<div class="email"> [pochta@mail.ru<br></div](mailto:pochta@mail.ru%3cbr%3e%3c/div)>

<div class="phoneNum"> +7(999)-000-09-90</div>

</div>

<div class="year">

<?php $today = date("F j, Y, H:i");

echo $today; ?>

</div>

<div class="city">Белореченск</div>

</div>

</body>

</html>

Delete-role.php

Удаление роли

<?php

if(isset($\_POST["id"]))

{

$conn = new mysqli("localhost", "web", "123", "webeducation");

if($conn->connect\_error){

die("Ошибка: " . $conn->connect\_error);

}

$userid = $conn->real\_escape\_string($\_POST["id"]);

$sql = "DELETE FROM category WHERE id = '$userid'";

if($conn->query($sql)){

header("Location: list-role.php");

}

else{

echo "Ошибка: " . $conn->error;

}

$conn->close();

}

?>

Delete-user.php

Удаление пользователя

<?php

if(isset($\_POST["id"]))

{

$conn = new mysqli("localhost", "web", "123", "webeducation");

if($conn->connect\_error){

die("Ошибка: " . $conn->connect\_error);

}

$userid = $conn->real\_escape\_string($\_POST["id"]);

$sql = "DELETE FROM users WHERE id = '$userid'";

if($conn->query($sql)){

header("Location: admin-panel.php");

}

else{

echo "Ошибка: " . $conn->error;

}

$conn->close();

}

?>