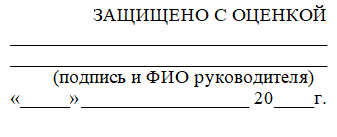
Частное учреждение

профессиональная образовательная организация

ТЕХНИКУМ «БИЗНЕС И ПРАВО»



**Курсовая работа**

|  |  |
| --- | --- |
| По МДК 09.01 | Проектирование и разработка информационных систем |
| Выполнил | Логвиненко О.В. |
| Курс, группа | 4 курс, группа ИС-4 |
| Специальность | 09.02.07 Информационные системы и программирование |
| Научный руководитель | Мартыненко В.А. |

г. Белореченск

2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ 4

1.1 Анализ предметной области 4

1.2 Цели и задачи выполняемой работы 5

1.3 Метод реализации процесса проектирования 6

1.4 Обоснование проектных решений 7

2 ПРОЕКТНАЯ ЧАСТЬ 9

2.1 Техническое задание 9

2.2 Установка и настройка среды разработки 20

2.3 Руководство пользователя 20

2.4 Руководство администратора 21

ЗАКЛЮЧЕНИЕ 23

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 24

ПРИЛОЖЕНИЕ

ВВЕДЕНИЕ

Портфолио студента - сборник работ, проектов, исследований и других материалов, которые демонстрируют навыки, знания и достижения студента в определенной области. Портфолио может включать различные формы представления информации, такие как тексты, графики, фотографии, видео и т.д.

В настоящее время портфолио студента становится все более актуальным, поскольку оно позволяет студентам демонстрировать свои навыки и знания в различных областях, а также служит инструментом для оценки их успехов и прогресса. Кроме того, портфолио студента может быть использовано для поиска работы или продолжения образования.

В данной курсовой работе мы будем изучать предметную область портфолио студента, выявляя ее основные характеристики, структуру, закономерности и проблемы. Кроме того, мы будем проводить анализ литературы, связанной с темой портфолио студента, и выявлять основные теоретические подходы и концепции, которые могут быть использованы для анализа данного вопроса.

В рамках курсовой работы также будут рассмотрены различные методы создания и оценки портфолио студента, а также их преимущества и недостатки. Будут проанализированы существующие практики использования портфолио в образовательных учреждениях и на рабочих местах, а также их влияние на процесс обучения и профессиональное развитие студентов. В конце работы будут предложены рекомендации по использованию портфолио студента в образовании и на работе, а также возможные направления дальнейших исследований в данной области.

1. АНАЛИТИЧЕСКАЯ ЧАСТЬ
   1. Анализ предметной области

Портфолио студента - сборник работ, проектов, исследований и других материалов, которые демонстрируют навыки, знания и достижения студента в определенной области. Портфолио может включать различные формы представления информации, такие как тексты, графики, фотографии, видео и т.д.

Основными характеристиками портфолио студента являются:

Целевая направленность: портфолио должно быть направлено на демонстрацию определенных навыков и знаний студента, связанных с его учебным планом и профессиональными целями.

Структура: портфолио должно быть структурировано и организовано таким образом, чтобы было легко найти и оценить каждую работу.

Выразительность: портфолио должно быть наглядным и информативным, чтобы было легко понять, какие навыки и знания демонстрирует студент.

Достоверность: портфолио должно содержать только достоверную информацию, которая может быть проверена и подтверждена.

Уникальность: портфолио должно быть уникальным и отражать индивидуальные качества и особенности студента.

Основными проблемами, связанными с портфолио студента, являются:

Недостаточная структурированность: не все студенты могут правильно организовать свои работы и проекты в портфолио, что может сделать его трудночитаемым и непонятным.

Недостаточная достоверность: некоторые студенты могут включать в портфолио недостоверную информацию или украшать свои достижения, что может исказить представление о их навыках и знаниях.

Недостаточная уникальность: некоторые студенты могут использовать шаблоны и готовые решения, что может сделать их портфолио слишком похожим на другие.

Недостаточная актуальность: некоторые студенты могут не обновлять свои портфолио, что может исказить представление о их последних достижениях и навыках.

В целом, анализ предметной области портфолио студента позволяет выявить основные характеристики и проблемы, связанные с этой темой, и разработать рекомендации для улучшения процесса создания и оценки портфолио студентов.

* 1. Цели и задачи

Цели выполняемой работы:

Изучить предметную область портфолио студента, выявив ее основные характеристики, структуру, закономерности и проблемы.

Провести анализ литературы, связанной с темой портфолио студента, и выявить основные теоретические подходы и концепции, которые могут быть использованы для анализа данного вопроса.

Определить основные требования к содержанию и структуре портфолио студента, а также к методам его создания и оценки.

Разработать рекомендации для улучшения процесса создания и оценки портфолио студентов.

Задачи выполняемой работы:

Изучить предметную область портфолио студента, выявив ее основные характеристики, структуру, закономерности и проблемы.

Провести анализ литературы, связанной с темой портфолио студента, и выявить основные теоретические подходы и концепции, которые могут быть использованы для анализа данного вопроса.

Определить основные требования к содержанию и структуре портфолио студента, а также к методам его создания и оценки.

Разработать рекомендации для улучшения процесса создания и оценки портфолио студентов.

Провести экспериментальное исследование, связанное с темой портфолио студента, и проанализировать полученные результаты.

Сформулировать выводы и рекомендации на основе проведенного анализа и экспериментального исследования.

В целом, цели и задачи выполняемой работы должны быть сформулированы четко и конкретно, чтобы дать читателю представление о том, что будет изучено и решено в рамках данной курсовой работы.

* 1. Метод реализации процесса проектирования

Исследование и анализ: первый этап метода реализации процесса проектирования - это исследование и анализ. На данном этапе мы будем изучать предметную область портфолио студента, выявляя ее основные характеристики, структуру, закономерности и проблемы. Кроме того, мы будем проводить анализ литературы, связанной с темой портфолио студента, и выявлять основные теоретические подходы и концепции, которые могут быть использованы для анализа данного вопроса.

Определение требований: на втором этапе мы определим основные требования к содержанию и структуре портфолио студента, а также к методам его создания и оценки. Это будет сделано на основе результатов анализа и исследования предметной области.

Разработка концепции: на третьем этапе мы разработаем концепцию портфолио студента, которая будет соответствовать определенным требованиям и характеристикам. Концепция будет включать в себя описание структуры портфолио, его содержания, а также методов его создания и оценки.

Проектирование: на четвертом этапе мы будем проектировать портфолио студента, используя определенные инструменты и технологии. Это может включать в себя создание макетов, прототипов и других визуальных материалов, которые будут использоваться для демонстрации и оценки портфолио.

Тестирование и оценка: на заключительном этапе мы будем проводить тестирование и оценку портфолио студента, чтобы определить его эффективность и соответствие определенным требованиям. Это может включать в себя получение обратной связи от преподавателей, студентов и других заинтересованных сторон, а также проведение экспериментального исследования.

В целом, метод реализации процесса проектирования портфолио студента должен быть основан на исследовании и анализе предметной области, определении требований, разработке концепции, проектировании и тестировании и оценке. Это позволит создать эффективное и соответствующее определенным требованиям портфолио студента.

1.4 Обоснование проектных решений

Использование PHP: PHP - это язык программирования, который широко используется для создания веб-приложений. Он позволяет создавать динамические веб-страницы, обрабатывать данные пользователей и взаимодействовать с базой данных. Использование PHP для создания сайта позволит нам создать функциональное и гибкое веб-приложение.

Использование HTML, CSS и JavaScript: HTML, CSS и JavaScript - это основные технологии, которые используются для создания веб-страниц. HTML определяет структуру страницы, CSS - ее внешний вид, а JavaScript - поведение страницы. Использование этих технологий позволит нам создать привлекательный и функциональный интерфейс для нашего сайта.

Использование Bootstrap: Bootstrap - это популярная библиотека CSS и JavaScript, которая позволяет создавать адаптивные и мобильные-френдли веб-страницы. Использование Bootstrap позволит нам создать сайт, который будет выглядеть хорошо на различных устройствах и браузерах.

Использование SQL базы данных: SQL - это язык запросов, который используется для управления базами данных. Использование SQL базы данных позволит нам хранить и обрабатывать данные пользователей, такие как их имена, адреса электронной почты и другие данные. Это позволит нам создать функциональное и гибкое веб-приложение, которое может быть легко расширено и изменено в будущем.

1. ПРОЕКТНАЯ ЧАСТЬ

2.1 Техническое задание

Полное наименование системы и ее условное обозначение Полное наименование системы: Портфолио студента для ЧУПО техникум Бизнес и Право.

Краткое наименование “ПС”.

Номер договор: №1/12-34-56-001 от 15.11.2023.

Наименование предприятий разработчика и заказчика систем, их реквизиты

Полное наименование заказчика: ЧУПО техникум Бизнес и Право

Сокращённое название: Бизнес и Право.

Адрес ул. Чапаева 46 г. Белореченск

Контактный телефон: +7(918)954-41-21.

ИНН: 3444049170

КПП: 343502001.

БИК: 041806001

ОГРН: 1023403440818

Расчетный счет: 40501810403492000005.

Полное наименование разработчика: Логвиненко Олег Валерьевич

Краткое наименование: “ИП”

Адрес ул. (напиши свой) г. Белореченск

Контактный телефон: (свой номер)

ИНН:

КПП:

Расчетный счет:

Порядок оформления и предъявления заказчику результатов работ по созданию системы, ее частей и отдельных средств

Система передается в виде полностью функционирующего комплекса на базе средств вычислительной техники заказчика и исполнителя в сроки, установленные договором. Приемка системы и осуществляется комиссией в составе уполномоченных представителей заказчика и исполнителя.

Весь комплект документации на Портфолио студента, разработанный исполнителем, передается заказчику на бумажных (в двух экземплярах), один экземпляр после подписания передается исполнителю. Текстовые документы, передаваемые на машинных носителях, должны быть представлены в форматах PDF.

Назначение и цели создания (развития) системы

Назначение системы

Разработанная система представляет электронный сайт, на котором будут храниться достижения студентов, которые принимали участие в спортивных или иных соревнованиях от техникума.

Цели создания системы

Основными целями создания Портфолио студента являются:

* Сохранение достижений студентов
* Сохранение проектов для дальнейшего трудоустройства
* Хранение информации о студенте

Характеристика объектов автоматизации

В ходе работы проведения работ по разработке Портфолио студента автоматизируются процессы по сохранению достижений.

Автоматизируемым процессом будет добавления достижения. При заполнении формы будет идти автоматическое сохранение и привязка к студенту.

Так же можно будет просмотреть информацию об самом студенте, для дальнейшего трудоустройства это может очень сильно помочь, так будет видно, где участвовал студент и будут отображаться его личные проекты.

Требования к системе

Требования к системе в целом

Требования к структуре и функционированию системе

Система должна предполагать наличие следующих модулей, представленных на рисунке 2.1.:

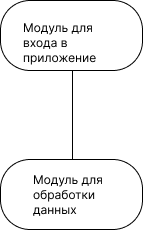


Рисунок 2.1. – Модули разрабатываемой системы

1. Модуль для входа в приложение. Благодаря ему, администратор сможет начать пользоваться им.
2. Модуль для обработки данных. Благодаря ему, администратор сможет получать нужные ему данные, либо внести новые.

На данном этапе необходимо описать примерную архитектуру разрабатываемой системы.

Для того, чтобы сделать приложение, мы будем использовать клиент-серверную архитектуру. Чтобы убедится, что она нам точно подходит рассмотрим ее подробнее. В клиент-серверной архитектуре имеется три звена:

- Представляет данных – на стороне клиента.

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

- Управление ресурсами – сервер БД, который и представляет запрашиваемые данные.

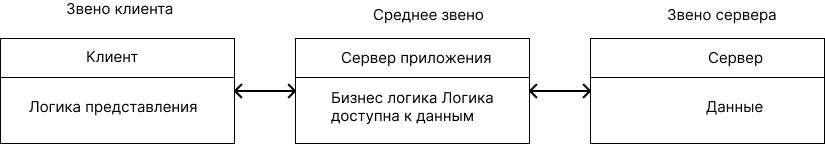
Мы выбрали эту архитектуру, так как у нее есть ряд преимуществ перед другими архитектурами:

- Высокую степень гибкости и масштабируемости.

- Высокую безопасность.

- Высокую производительность (т.к задачи распределены между серверами).

На рисунке 2.2. представлена структура клиент-серверной архитектуры.

  
Рисунок 2.2. - Клиент-серверная трехзвенная 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 Портфолио студента представления в таблице 2.1.

Таблица 2.1. – Требования к Портфолио студента

|  |  |  |
| --- | --- | --- |
| Атрибуты | Метрики | Требования |
| 1 | 2 | 3 |
| Удобство | | |
| Простота использования | Среднее время, необходимое пользователю, чтобы найти необходимо элемент интерфейса | Поиск отдельного элемента интерфейса не должен занимать более 2 секунды |
| Привлекательность | Соответствие интерфейса требования заказчика | 97% элементов интерфейса должны быть адаптированы для пользователя и 99% для заказчика; |
| Обучаемость | Показатель, затрачиваемый пользователями на обучение работе с ПС | Пользователь должен изучить работу ПО за первые 3-5 минут использования |
| Производительность | | |
| Временная эффективность | Время выполнения компонента программы | - Приложение должно выполнять любую функцию не дольше 2 сек;  - Главная страница приложения в среднем случае должна |

Продолжение таблицы 2.1 - Требования к Портфолио студента

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Эффективность использования ресурсов | Объемы ресурсов требуемых для выполнения задач; | - Постоянна  я память, используемая программой – не более 10МБ; - Приложение должно потреблять не более 4 Кбайт памяти на каждый неактивный сеанс с пользователем; - Нагрузка на CPU и используемый объем жесткого диска на сервере баз данных не должны превышать 70%, а время обработки запросов не должно превышать 2 секунд |

Средняя доступность ПС 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 2.3. представлен перечень и критерии отказов для каждой функции, по которой задаются требования по надежности.

Таблица 2.3. – Перечень и критерии отказов для каждой функции

|  |  |
| --- | --- |
| Функция | Критерии отказа |
| 1 | 2 |
| Загрузка достижения студента | Студент/пользователь не может самостоятельно добавлять достижение |
| Оставление комментария | Комментарий могут оставлять только авторизированные или зарегистрированные пользователи |
| Добавление собственных проектов | Администратор не может добавлять проекты пользователям, эта функция доступна только студентам |

Требования к функции (ВИ), Выполняемым информационная система

Имя: Добавление информации в портфолио студента

Описание: Оператор запрашивает у программы выполнение функции “Загрузки достижения”

Действующий субъект: администратор

Предусловие: администратор заходит на сайт, для загрузки новой информации о студенте.

Сценарий:

1. Администратор нажимает на студента, после появляется форма заполнения

2. Клиент отправляет запрос на добавление

3. Сервер обрабатывает полученную информацию (если она соответствует требованиям, то все проходит успешно).

4. В случае неверно введенных данных система будет выдавать ошибку.  
5. В случае успешного добавления будет выводиться сообщение об успешном добавлении

Расширение: в случае некорректных данных или пустых полях система будет выдавать ошибку.

Примечание:  
Все запросы выполнены по схожему типу, то есть в случае некорректно введённых данных система будет выдавать сообщение об ошибке.

Требования к видам обеспечения

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

Провести в компании Интернет-соединение;

Обеспечить каждого специалиста персональным компьютером;

Установить на каждый ПК клиента;

Аренда хостинга, размещение на нем web-сервера или СУБД;

Изучение персоналом документа “Руководства пользователя”;

Эксплуатация системы.

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

- docs

- project

- reviews

- role

- users

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:

Таблица docs - представляет собой загруженную информацию.

Таблица project - представляет собой проекты студента

Таблица rewiews - представляет комментарии

Таблица role - представляет перечень ролей для распределения прав в web-приложении

Таблица users – таблица, содержащая всех зарегистрированных пользователей на сайте

Данные о docs должны располагаться в таблице docs:

Таблица 2.3. - Таблица docs в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ключ |
| id | int | 11 | Идентификатор | NOT NULL |
| name | varchar | 255 | Имя | NOT NULL |
| text | varchar | 255 | Текст | NOT NULL |
| img | varchar | 255 | Картинка | NOT NULL |

Данные о проектах должны располагаться в таблице projects:

Таблица 2.4. - Таблица projects в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ключ |
| 1 | 2 | 3 | 4 | 5 |
| id | int | 11 | Идентификатор | NOT NULL |

Продолжение таблицы 2.4. - Таблица projects в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| link | varchar | 255 | Ссылка | NOT NULL |
| img | varchar | 255 | Картинка | NOT NULL |
| disc | varchar | 255 | Диск | NOT NULL |

Данные о комментариях должны располагаться в таблице reviews:

Таблица 2.5. - Таблица reviews в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ключ |
| id | int | 11 | Идентификатор | NOT NULL |
| User\_id | int | 11 | ID\_пользователя | Внешний ключ к таблице users |
| Reviews\_text | text | 255 | Текст | NOT NULL |
| Created\_at | timestamp | 255 | Время комментария | NOT NULL |

Данные о ролях должны располагаться в таблице role:

Таблица 2.6. - Таблица role в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ключ |
| id | int | 11 | Идентификатор | NOT NULL |
| name | varchar | 255 | Название роли | NOT NULL |

Данные о пользователях должны располагаться в таблице users:

Таблица 2.7. - Таблица users в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ключ |
| 1 | 2 | 3 | 4 | 5 |
| id | int | 11 | Идентификатор | NOT NULL |
| name | varchar | 11 | Имя | NOT NULL |

Продолжение таблицы 2.7 - Таблица users в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| email | varchar | 255 | Почта пользователя | NOT NULL |
| password | varchar | 255 | Пароль | NOT NULL |

Таблица 2.8. - Таблица role в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ключ |
| Id\_role | int | 11 | ID\_role | Внешний ключ к таблице role |
| date | timestamp | 255 | Дата | NOT NULL |
| avatar | varchar | 255 | Аватарка пользователя | NOT NULL |

Требования к эргономике и технической эстетике.

Требования к размещению элементов управления на экранных формах

Общие требования:

- Фоновый цвет белый

- Справа с верху будет располагаться кнопка “Вход” и “Регистрация”.

Требования к видам обеспечения

Для функционирования системы требуется сервер и клиентских компьютеры, на которых будет производиться запуск системы.

Требования для сервера БД:

1. Процессор 2 ядра (4 потока), частота от 2 и выше
2. Оперативная память 4 Гб и выше
3. Свободное место на диске не менее 100 Мб
4. HDD (жёсткий диск), тип SATA2

Требования к клиентским устройствам:

Персональные компьютеры должны быть под управлением Windows 7 или выше.

Монитор на клиентских компьютерах должен быть с разрешением не ниже 1280х720 пикселей.

В качестве ОС клиентских устройств может использоваться:

1. Microsoft Windows 7 и выше.

Эскиз основного окна системы

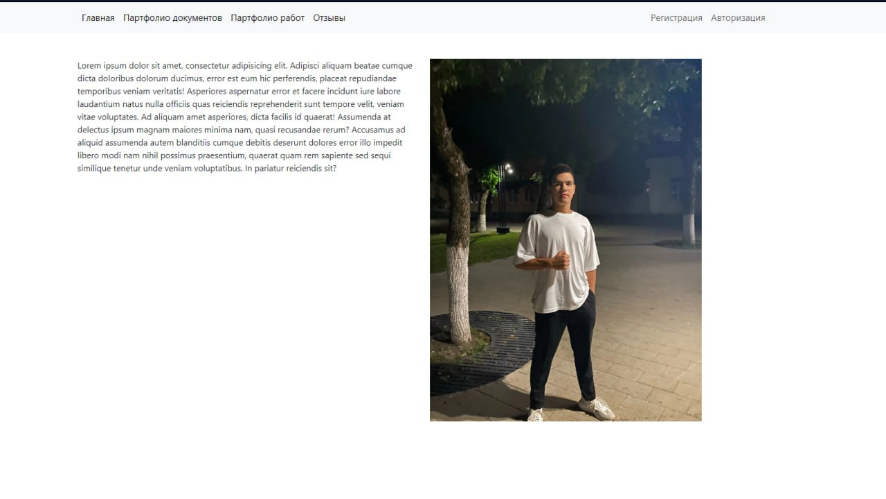


Рисунок 2.2 - Окно из портфолио с информацией

Состав и содержание работ по созданию системы

Комплекс работ по созданию «ПС» должен осуществляться в несколько этапов. Этапы приведены в таблице 2.9:

Таблица 2.9 - Этапы работ по созданию «ПС»

|  |  |  |  |
| --- | --- | --- | --- |
| Стадия работ | Выполняемые работы | Сроки | Итоги выполнения работ |
| 1 | 2 | 3 | 4 |
| Формирование требований | Обследование объектов автоматизации | выполнено | Отчет о результатах обследования; |
| Разработка технического задания на создание системы | Утверждение заказчиком ТЗ на создание системы; |
| Проектирование | Разработка технического проекта на систему  Разработка прототипа системы | 10.11.2023  -  12.11.2023 | Технический проект на систему;  Спецификации программно-аппаратных средств системы; |
| Разработка проектов организационно-распорядительной, программной и эксплуатационной документации на систему |

Продолжение таблицы 2.9 - Этапы работ по созданию «ПС»

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Поставка программно-технических средств для эксплуатации | Поставка программно-технических средств для опытной эксплуатации на объектах автоматизации, входящих в состав опытной зоны | 15.11.2023  -  18.11.2023 | Акты; |
| Разработка программных средств | Разработка, отладка и тестирование программных средств Подсистемы | 19.11.2023  -  30.12.2023 | Программные средства на цифровых |

Система передается в виде полностью функционирующего комплекса на базе средств вычислительной техники заказчика и исполнителя в сроки, установленные договором. Приемка системы осуществляется комиссией в составе уполномоченных представителей заказчика и исполнителя.

Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

Заказчик обязан предоставить сервер и персональные компьютеры, требования к которым указаны в разделе «Требования к видам обеспечения».

Необходимое программное обеспечение для сервера:

Операционная система: Windows 10

СУБД MySQL

Необходимое программное обеспечение для персонального компьютера:

Операционная система: Windows 10

Перечень подлежащих разработке комплектов и видов документов:

- Описание бизнес

-процессов предметной области

- Техническое задание

- Эскизный проект

Источники разработки

Источниками разработки настоящего Технического Задания являются:

- ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы;

- ГОСТ 24.601-86 Автоматизированные системы. Стадии создания;

2.2 Установка и настройка средств разработки

Необходимо скачать проект

Установить его в папку htdocs (E:\PO\htdocs приер пути)

Открыть xampp

Перейти в панель phpMyAdmin

Экспортировать базу данных

После по пути, открыть проект: http://localhost/pudge/kurs/user/login.php

После установки всех необходимых инструментов и библиотек, необходимо настроить их для работы вместе. Это включает в себя настройку Apache-сервера, MySQL-сервера и PHP-библиотеки, а также настройку связи между PHP и MySQL.

В целом, установка и настройка среды разработки - это важный процесс, который позволяет нам создавать и тестировать веб-приложения на локальном компьютере перед развертыванием их на веб-сервере.

* 1. Руководство пользователя

Загрузка веб-сайта: Чтобы загрузить веб-сайт, необходимо открыть браузер и ввести адрес сайта в адресной строке.

Регистрация: Если вы еще не зарегистрированы на сайте, необходимо зарегистрироваться, чтобы получить доступ к всем функциям сайта. Для регистрации необходимо нажать на кнопку "Регистрация" и заполнить форму регистрации.

Вход: Если вы уже зарегистрированы на сайте, необходимо войти в свой аккаунт, нажав на кнопку "Вход" и введя свои учетные данные.

Навигация: На сайте есть меню навигации, которое позволяет перемещаться между разными страницами сайта. Меню навигации находится вверху страницы.

Поиск: Если вы ищете определенную информацию на сайте, можно использовать поисковую функцию. Для этого необходимо ввести ключевое слово в поле поиска и нажать на кнопку "Найти".

Создание контента: Если вы зарегистрированы на сайте, вы можете создавать контент, такой как статьи, комментарии и т.д. Для этого необходимо нажать на кнопку "Создать контент" и выбрать тип контента, который вы хотите создать.

Управление аккаунтом: Если вы зарегистрированы на сайте, вы можете управлять своим аккаунтом, изменяя свои учетные данные, профиль и т.д. Для этого необходимо нажать на кнопку "Управление аккаунтом" и выбрать нужный пункт меню.

Выход: Если вы зарегистрированы на сайте и хотите выйти из своего аккаунта, необходимо нажать на кнопку "Выход" и подтвердить выход.

В целом, веб-сайт был разработан с учетом простоты использования и интуитивно понятного интерфейса. Если у вас возникнут какие-либо вопросы или проблемы, вы можете обратиться к нашей поддержке, нажав на кнопку "Поддержка" в меню навигации.

2.4 Руководство администратора

Вход в административную панель: Чтобы войти в административную панель, необходимо нажать на кнопку "Администрирование" в меню навигации и ввести свои учетные данные.

Управление пользователями: В административной панели можно управлять пользователями сайта, добавляя новых пользователей, удаляя существующих и изменяя их права доступа. Для этого необходимо нажать на пункт меню "Пользователи" и выбрать нужный пункт меню.

Управление контентом: В административной панели можно управлять контентом сайта, добавляя новые статьи, удаляя существующие и изменяя их содержание. Для этого необходимо нажать на пункт меню "Контент" и выбрать нужный пункт меню.

Управление категориями: В административной панели можно управлять категориями контента, добавляя новые категории, удаляя существующие и изменяя их содержание. Для этого необходимо нажать на пункт меню "Категории" и выбрать нужный пункт меню.

Управление комментариями: В административной панели можно управлять комментариями к статьям, добавляя новые комментарии, удаляя существующие и отвечая на них. Для этого необходимо нажать на пункт меню "Комментарии" и выбрать нужный пункт меню.

Управление настройками: В административной панели можно управлять настройками сайта, такими как название сайта, адрес сайта, язык сайта и т.д. Для этого необходимо нажать на пункт меню "Настройки" и выбрать нужный пункт меню.

Управление правами доступа: В административной панели можно управлять правами доступа пользователей, добавляя новые роли, удаляя существующие и изменяя их права доступа. Для этого необходимо нажать на пункт меню "Права доступа" и выбрать нужный пункт меню.

В целом, административную панель была разработана с учетом простоты использования и интуитивно понятного интерфейса. Если у вас возникнут какие-либо вопросы или проблемы, вы можете обратиться к нашей поддержке, нажав на кнопку "Поддержка" в меню навигации.

ЗАКЛЮЧЕНИЕ

В заключение, веб-сайт был разработан с учетом потребностей пользователей и администраторов, обеспечивая простоту использования и интуитивно понятный интерфейс. Для пользователей доступны функции регистрации, входа, поиска, создания контента и управления аккаунтом. Для администраторов доступны функции управления пользователями, контентом, категориями, комментариями, настройками и правами доступа.

Была поставлена цель создать веб-сайт, который был бы полезен для сообщества и позволял бы его участникам делиться информацией и идеями. Также была задача обеспечить безопасность и конфиденциальность данных пользователей, чтобы они могли использовать сайт с уверенностью.

Созданный веб-сайт в будущем поможет сообществу обмениваться информацией и идеями, а также будет полезен для всех, кто использует его. Также планируется работа над дополнением функциональности и безопасности сайта, чтобы обеспечить лучший опыт для пользователей и администраторов.

В рамках курсовой работы также были рассмотрены различные методы создания и оценки портфолио студента, а также их преимущества и недостатки. Были проанализированы существующие практики использования портфолио в образовательных учреждениях и на рабочих местах, а также их влияние на процесс обучения и профессиональное развитие студентов. В конце работы были предложены рекомендации по использованию портфолио студента в образовании и на работе, а также возможные направления дальнейших исследований в данной области.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Гаевский, А.Ю. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. - М.: Триумф, 2020. - 464 c.
2. Дакетт, Джон HTML и CSS. Разработка и дизайн веб-сайтов / Джон Дакетт. - Москва: Наука, 2019. - 480 c.
3. Дронов, В. JavaScript в Web-дизайне / В. Дронов. - М.: СПб: БХВ, 2019. - 880 c.
4. Дронов, В. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов / В. Дронов. - М.: БХВ-Петербург, 2019. - 480 c.
5. 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.
6. Дронов, Владимир JavaScript и AJAX в Web-дизайне / Владимир Дронов. - Москва: Высшая школа, 2019. - 736 c.
7. Дронов, Владимир PHP, MySQL и Dreamweaver MX 2004. Разработка интерактивных Web-сайтов / Владимир Дронов. - М.: "БХВ-Петербург", 2019. - 448 c.
8. Колисниченко, Д. PHP 5/6 и MySQL 6. Разработка Web-приложений / Д. Колисниченко. - М.: БХВ-Петербург, 2021. - 560 c.
9. Колисниченко, Д.Н. PHP 5/6 и MySQL 6. Разработка Web-приложений / Д.Н. Колисниченко. - М.: БХВ-Петербург, 2021. - 364 c.
10. Кузнецов PHP. Практика создания Web-сайтов / Кузнецов, М.В. и. - М.: БХВ-Петербург, 2020. - 895 c.

ПРИЛОЖЕНИЕ

index.php -Страница доступная пользователям, которые не авторизовались

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

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

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

    <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=Poppins:wght@100;200;300;400;500;600;700;800;900&display=swap"

        rel="stylesheet">

    <title>Посадочная</title>

</head>

<body>

    <img class="index-decor" src="../images/gr-1.svg" alt="decor">

    <section class="section\_top">

        <div class="container">

            <div class="section\_top-wrapper">

                <div class="section\_top-left">

                    <h1 class="section\_top-title">Образовательная <li>платформа<li></h1>

                    <p class="section\_top-description">Вычислительные системы, сети и телекоммуникации

                        <br>

                        - Читайте темы от преподавателей

                        <br>

                        - Отмечайте прочитанным выполненные задания

                        <br>

                        - Развивайтесь вместе с нашей платформой</p>

                    <form action="#" class="section\_top-form">

                        <a href="reg-teacher.php" class="section\_top-btn">Я преподаватель</a>

                        <a href="reg-student.php" class="section\_top-btn">Я студент</a>

                    </form>

                </div>

                <div class="section\_top-right">

                    <img src="../images/logo-big.svg" alt="logo">

                </div>

            </div>

        </div>

    </section>

    <section class="section\_bottom">

        <div class="container">

            <div class="section\_blocks">

                <div class="block">

                    <img src="../images/main-1.jpg">

                    <h1 class="block\_title">Познавательные темы</h1>

                    <p class="block\_description">Читайте, отмечайте прочитанное

                        Делайте многое другое крутое на нашей платформе</p>

                </div>

                <div class="block">

                    <img src="../images/main-2.jpg">

                    <h1 class="block\_title">Весь мир</h1>

                    <p class="block\_description">У нас сидят студенты и преподы с самых разных стран начиная от России

                        заканчивая Атлантидой</p>

                </div>

                <div class="block">

                    <img src="../images/main-3.jpg">

                    <h1 class="block\_title">Познавательные темы</h1>

                    <p class="block\_description">Читайте, отмечайте прочитанное

                        Делайте многое другое крутое на нашей платформе</p>

                </div>

            </div>

        </div>

    </section>

</body>

</html>

notification.html- страница уведомлений

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

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

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

    <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=Poppins:wght@100;200;300;400;500;600;700;800;900&display=swap"

        rel="stylesheet">

    <title>Уведомления</title>

</head>

<body>

    <header class="header">

        <div class="header-link-top">

            <ul>

                <li>

                    <a href="profile-for-student.php">

                        <img src="../images/header-avatar.svg" alt="avatar">

                    </a>

                </li>

                <li>

                    <a href="notification.html">

                        <img src="../images/header-notification.svg" alt="avatar">

                    </a>

                </li>

                <li><a href="tasks-for-student.php">

                        <img src="../images/header-tasks.svg" alt="avatar">

                    </a>

                </li>

            </ul>

        </div>

        <div class="header-link-bottom">

            <ul>

                <li><a href="index.php">

                      <img src="../images/header-logout.svg" alt="avatar">

                    </a>

                </li>

                <li><a href="setting.php">

                        <img src="../images/header-setting.svg" alt="avatar">

                    </a>

                </li>

            </ul>

        </div>

    </header>

    <section class="user-themes userContainer section-notification">

        <div class="notification-wrapper">

            <img src="../images/icon-notification.svg" alt="icon-notification">

            <h1 class="notification-title">Новых уведомлений нет</h1>

        </div>

    </section>

</body>

</html>

profile-for-student.php-профиль студента

<?php

session\_start();

// Проверяем, авторизован ли пользователь

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

    header("Location: login.php"); // Перенаправляем на страницу авторизации, если не авторизован

    exit();

}

// Подключение к базе данных (замените данными вашей базы)

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "sem";

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

// Проверка соединения

if ($conn->connect\_error) {

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

}

// Получение данных о прочитанных лекциях для текущего пользователя

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

$sql = "SELECT lectures.\*, reading.is\_read FROM lectures LEFT JOIN lecture\_readings AS reading ON lectures.lecture\_id = reading.lecture\_id AND reading.user\_id = $user\_id";

$result = $conn->query($sql);

// Закрытие соединения с базой данных

$conn->close();

?>

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

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

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

    <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=Poppins:wght@100;200;300;400;500;600;700;800;900&display=swap"

          rel="stylesheet">

    <title>Профиль студента</title>

</head>

<body>

<?php include "header.php"?>

<section class="user-data userContainer">

    <div class="user-data-name">

        <!-- Выводим аватар пользователя -->

        <img src="<?php echo $\_SESSION['user\_avatar']; ?>" alt="ava" width="130" height="130">

        <div class="user-data-name-text">

            <!-- Выводим имя пользователя -->

            <h1><?php echo $\_SESSION['user\_name']; ?></h1>

            <!-- Выводим email пользователя -->

            <p><?php echo $\_SESSION['user\_login']; ?></p>

            <a href="setting.php">Редактировать</a>

        </div>

    </div>

    <div class="user-data-statistic">

        <!-- Ваши блоки статистики... -->

    </div>

</section>

<section class="user-themes userContainer">

    <h1 class="user-themes-title"> Доступные темы</h1>

    <div class="user-themes-wrapper">

        <?php

        if ($result->num\_rows > 0) {

            while ($row = $result->fetch\_assoc()) {

                $lecture\_title = $row["lecture\_title"];

                $lecture\_content = $row["lecture\_content"];

                $is\_read = $row["is\_read"];

                $lecture\_id= $row['lecture\_id'];

                ?>

                <div class="user-themes-inner">

                    <a href="theme-details.php?lecture\_id= <?=$lecture\_id?>">

                    <img src="uploads/<?=$row['img']?>" alt="img-theme">

                    <div class="user-themes-inner-title">

                        <a href="theme-details.php?lecture\_id=<?=$row['lecture\_id']?>"><?php echo $lecture\_title; ?></a>

                        <span><?php echo $is\_read ? 'Прочитано' : 'Не прочитано'; ?></span>

                    </div>

                    </a>

                </div>

                <?php

            }

        } else {

            echo "Нет данных о прочитанных темах.";

        }

        ?>

    </div>

</section>

</body>

</html>

profile-for-teacher.php-профиль преподавателя

<?php

session\_start();

// Проверяем, авторизован ли пользователь

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

    header("Location: login.php"); // Перенаправляем на страницу авторизации, если не авторизован

    exit();

}

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "sem";

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

// Запрос для выборки лекций пользователя

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

$sql = "SELECT \* FROM lectures WHERE user\_id = ?";

$stmt = $conn->prepare($sql);

// Проверка на успешность подготовленного запроса

if (!$stmt) {

    die("Ошибка в запросе: " . $conn->error);

}

$stmt->bind\_param("i", $user\_id);

// Выполнение запроса

$stmt->execute();

// Получение результата запроса

$result = $stmt->get\_result();

?>

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

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

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

    <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=Poppins:wght@100;200;300;400;500;600;700;800;900&display=swap"

          rel="stylesheet">

    <title>Профиль преподавателя</title>

</head>

<body>

<?php include "header.php"?>

<section class="user-data userContainer">

    <div class="user-data-name admin-profile-wrapper">

        <div class="user-data-name-text admin-profile">

            <h1><?php echo $\_SESSION['user\_name']; ?></h1>

            <img src="<?php echo $\_SESSION['user\_avatar']; ?>" alt="ava" width="130" height="130">

            <p><?php echo $\_SESSION['user\_login']; ?></p>

        </div>

    </div>

</section>

<section class="user-themes userContainer">

    <h1 class="user-themes-title">Созданные темы</h1>

    <?php

    // Вывод лекций пользователя

    while ($row = $result->fetch\_assoc()) {

        echo '<div class="user-themes-inner">';

        echo ' <a href="theme-details.php?lecture\_id='.$row['lecture\_id'].'">';

        echo '<img src="uploads/'.$row['img'].'" alt="img-theme">';

        echo '<div class="user-themes-inner-title">';

        echo '<h2>' . htmlspecialchars($row['lecture\_title']) . '</h2>';

        echo '</div>';

        echo '<div class="user-themes-inner-starsAndRepeat admin-tools">';

        echo ' <a href="del.php?id='.$row['lecture\_id'].'">удалить</a>';

        echo ' <a href="edit.php?id='.$row['lecture\_id'].'">редактировать</a>';

        echo '</div>';

        echo '</div>';

    }  ?>

    <a  href="create-theme.php" class="final-test create-theme">Создать тест</a>

</section>

</body>

</html>

reg-student.php-регистрация студента

<?php

session\_start();

// Подключение к базе данных (замените данными вашей базы)

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "sem";

$characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';

$charactersLength = strlen($characters);

$randomString = '';

for ($i = 0; $i < 5; $i++) {

    $randomString .= $characters[rand(0, $charactersLength - 1)];

}

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

// Проверка соединения

if ($conn->connect\_error) {

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

}

// Функция для регистрации пользователя с аватаркой

function registerUser($conn, $name, $login, $password, $avatar) {

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

    $role = "student";

    $status = "active";

    // Подготовка запроса вставки данных пользователя

    // Подготовка запроса вставки данных пользователя

    $stmt = $conn->prepare("INSERT INTO users (name, login, password, avatar, role, status) VALUES (?, ?, ?, ?, ?, ?)");

    // Привязка параметров

    $stmt->bind\_param("ssssss", $name, $login, $hashedPassword, $avatar, $role, $status);

// Значения роли и статуса

    // Выполнение запроса

    $stmt->execute()

    // Закрытие запроса

    $stmt->close();

    header("location:auth.php");

}

// Проверка, что данные приходят из формы методом POST

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

    // Проверка наличия и непустоты обязательных полей

    if (isset($\_POST['name']) && isset($\_POST['login']) && isset($\_POST['password']) &&

        !empty($\_POST['name']) && !empty($\_POST['login']) && !empty($\_POST['password'])) {

        // Проверка капчи

        $captchaInput = isset($\_POST['captcha']) ? $\_POST['captcha'] : '';

            // Обработка остальных полей и регистрация пользователя

            $name = $\_POST['name'];

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

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

            // Обработка загрузки аватарки

            $avatarDir = "uploads/";

            $avatarPath = $avatarDir . basename($\_FILES["avatar"]["name"]);

            move\_uploaded\_file($\_FILES["avatar"]["tmp\_name"], $avatarPath);

            // Регистрация пользователя с аватаркой

            registerUser($conn, $name, $login, $password, $avatarPath);

   } else {

        echo "Пожалуйста, заполните все обязательные поля формы.";

    }

}

// Закрытие соединения с базой данных

$conn->close();

?>

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

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

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

    <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=Poppins:wght@100;200;300;400;500;600;700;800;900&display=swap"

        rel="stylesheet">

    <title>Регистрация студента</title>

</head>

<body>

    <img class="index-decor" src="../images/gr-1.svg" alt="decor">

    <div class="container">

        <div class="section-wrapper">

            <form class="reg-form" action="" method="post" enctype="multipart/form-data">

                <svg width="251" height="35" viewBox="0 0 251 35" fill="none" xmlns="http://www.w3.org/2000/svg">

                    <path

                        fill="white" />

                </svg>

                <h1 class="reg-form-title">Регистрация</h1>

                <div class="reg-form-input">

                    <label for="name">Ваше имя</label>

                    <input type="text" class="input" name="name" id="name" placeholder="Введите свое имя">

                </div>

                <div class="reg-form-input">

                    <label for="login">Логин</label>

                    <input type="login" class="input" name="login" id="login" placeholder="Придумайте логин">

                </div>

                <div class="reg-form-input">

                    <label for="password">Пароль</label>

                    <input type="password" class="input" name="password" id="password" placeholder="Придумайте пароль">

                </div>

                <div class="reg-form-input">

                    <label for="avatar">Выберите аватар</label>

                    <input type="file" class="input" name="avatar" id="avatar" accept="image/\*">

                </div>

                <p class="acc\_have">Уже имеете аккаунт?<a href="auth-student.html"> Авторизоваться?</a></p>

                <label>

                    <input required type="checkbox" name="happy" value="yes">Подтвердите что вы не робот

                </label>

<!--<h1 style="background-color: blue;color: red;user-select: none; text-align: center"> --><?php //= $randomString?><!-- </h1>-->

<!--<input class="form-control mt-1" type="text" placeholder="Введите текст выше" required>-->

                <button class="reg-form-btn" type="submit">Зарегистрироваться</button>

            </form>

            <img src="../images/logo-big.svg" alt="logo">

        </div>

    </div>

</body>

</html>

reg-teacher.php-регистрация преподавателя

<?php

session\_start();

// Подключение к базе данных (замените данными вашей базы)

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "sem";

$characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';

$charactersLength = strlen($characters);

$randomString = '';

for ($i = 0; $i < 5; $i++) {

    $randomString .= $characters[rand(0, $charactersLength - 1)];

}

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

// Проверка соединения

if ($conn->connect\_error) {

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

}

// Функция для регистрации пользователя с аватаркой

function registerUser($conn, $name, $login, $password, $avatar) {

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

    $role = "teacher";

    $status = "active";

    // Подготовка запроса вставки данных пользователя

    // Подготовка запроса вставки данных пользователя

    $stmt = $conn->prepare("INSERT INTO users (name, login, password, avatar, role, status) VALUES (?, ?, ?, ?, ?, ?)");

    // Привязка параметров

    $stmt->bind\_param("ssssss", $name, $login, $hashedPassword, $avatar, $role, $status);

// Значения роли и статуса

    // Выполнение запроса

    $stmt->execute();

    // Закрытие запроса

    $stmt->close();

    header("location:auth.php");

}

// Проверка, что данные приходят из формы методом POST

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

    // Проверка наличия и непустоты обязательных полей

    if (isset($\_POST['name']) && isset($\_POST['login']) && isset($\_POST['password']) &&

        !empty($\_POST['name']) && !empty($\_POST['login']) && !empty($\_POST['password'])) {

        // Проверка капчи

        $captchaInput = isset($\_POST['captcha']) ? $\_POST['captcha'] : '';

        // Обработка остальных полей и регистрация пользователя

        $name = $\_POST['name'];

        $login = $\_POST['login'];

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

        // Обработка загрузки аватарки

        $avatarDir = "uploads/";

        $avatarPath = $avatarDir . basename($\_FILES["avatar"]["name"]);

        move\_uploaded\_file($\_FILES["avatar"]["tmp\_name"], $avatarPath);

        // Регистрация пользователя с аватаркой

        registerUser($conn, $name, $login, $password, $avatarPath);

    } else {

        echo "Пожалуйста, заполните все обязательные поля формы.";

    }

}

// Закрытие соединения с базой данных

$conn->close();

?>

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

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

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

    <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=Poppins:wght@100;200;300;400;500;600;700;800;900&display=swap"

        rel="stylesheet">

    <title>Регистрация студента</title>

</head>

<body>

    <img class="index-decor" src="../images/gr-2.svg" alt="decor">

    <div class="container">

        <div class="section-wrapper">

            <form class="reg-form" method="post" enctype="multipart/form-data" action="">

                <svg width="251" height="35" viewBox="0 0 251 35" fill="none" xmlns="http://www.w3.org/2000/svg">

                    <path

                        fill="white" />

                </svg>

                <h1 class="reg-form-title">Регистрация</h1>

                <div class="reg-form-input">

                    <label for="name">Ваше имя</label>

                    <input type="text" class="input" name="name" id="name" placeholder="Введите свое имя">

                </div>

                <div class="reg-form-input">

                    <label for="login">Логин</label>

                    <input type="login" class="input" name="login" id="login" placeholder="Придумайте логин">

                </div>

                <div class="reg-form-input">

                    <label for="password">Пароль</label>

                    <input type="password" class="input" name="password" id="password" placeholder="Придумайте пароль">

                </div>

                <div class="reg-form-input">

                    <label for="avatar">Выберите аватар</label>

                    <input type="file" class="input" name="avatar" id="avatar" accept="image/\*">

                </div>

                <p class="acc\_have">Уже имеете аккаунт?<a href="auth-student.html"> Авторизоваться?</a></p>

                <label>

                    <input required type="checkbox" name="happy" value="yes">Подтвердите что вы не робот

                </label>

                <button class="reg-form-btn for-teacher" type="submit">Зарегистрироваться</button>

            </form>

            <img src="../images/logo-big.svg" alt="logo">

        </div>

    </div>

</body>

</html>

auth.php-авторизация

<?php

session\_start();

// Проверяем, была ли отправлена форма

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

    // Здесь должен быть ваш код подключения к базе данных

    // Например, вы можете использовать mysqli

// Подключение к базе данных (замените данными вашей базы)

    $servername = "localhost";

    $username = "root";

    $password = "";

    $dbname = "sem";

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

    // Проверяем соединение

    if ($conn->connect\_error) {

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

    }

    // Получаем данные пользователя из формы

    $login = $\_POST["login"];

    $password = $\_POST["password"];

    // Валидируем входные данные (в реальном проекте нужна более тщательная валидация)

    if (empty($login) || empty($password)) {

        echo "Необходимо ввести логин и пароль.";

    } else {

        // Хешируем пароль (в реальном проекте следует использовать password\_hash())

        $hashedPassword = md5($password);

        // Проверяем, существует ли пользователь в базе данных

        $sql = "SELECT \* FROM users WHERE login='$login'";

        $result = $conn->query($sql);

        if ($result->num\_rows > 0) {

            // Пользователь найден, сохраняем информацию в сессии

            $user = $result->fetch\_assoc();

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

     {

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

         $\_SESSION['user\_login'] = $user['login'];

         $\_SESSION['user\_name'] = $user['name'];

         $\_SESSION['user\_avatar'] = $user['avatar'];

         $\_SESSION['user\_role'] = $user['role'];

         if ($user['role']=="teacher")

         {

             header("Location: profile-for-teacher.php");

         }else{

             header("Location: profile-for-student.php");

         }

     }

            echo "Неверный логин или пароль. Пожалуйста, попробуйте снова.";

        }

    }

    // Закрываем соединение с базой данных

    $conn->close();

}

?>

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

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

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

    <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=Poppins:wght@100;200;300;400;500;600;700;800;900&display=swap"

          rel="stylesheet">

    <title>Регистрация студента</title>

</head>

<body>

<img class="index-decor" src="../images/gr-1.svg" alt="decor">

<div class="container">

    <div class="section-wrapper">

        <form class="reg-form" action="" method="post" enctype="multipart/form-data">

            <svg width="251" height="35" viewBox="0 0 251 35" fill="none" xmlns="http://www.w3.org/2000/svg">

                <path

                    fill="white" />

            </svg>

            <h1 class="reg-form-title">авторизация</h1>

            <div class="reg-form-input">

                <label for="login">Логин</label>

                <input type="login" class="input" name="login" id="login" placeholder="логин">

            </div>

            <div class="reg-form-input">

                <label for="password">Пароль</label>

                <input type="password" class="input" name="password" id="password" placeholder="пароль">

            </div>

            <p class="acc\_have">нет аккаунта?<a href="reg-student.php"> регистрация студента</a></p>

            <p class="acc\_have">нет аккаунта?<a href="reg-teacher.php"> регистрация учителя</a></p>

            <button class="reg-form-btn" type="submit">Авторизироваться</button>

        </form>

        <img src="../images/logo-big.svg" alt="logo">

    </div>

</div>

</body>

</html>

create-test.html-создание теста

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

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

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

    <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=Poppins:wght@100;200;300;400;500;600;700;800;900&display=swap"

        rel="stylesheet">

    <title>Профиль студента</title>

</head>

<body>

    <header class="header">

        <div class="header-link-top">

            <ul>

                <li>

                    <a href="profile-for-teacher.php">

                        <img src="../images/header-avatar.svg" alt="avatar">

                    </a>

                </li>

                <li>

                    <a href="notification.html">

                        <img src="../images/header-notification.svg" alt="avatar">

                    </a>

                </li>

                <li><a href="tasks-for-teacher.php">

                        <img src="../images/header-tasks.svg" alt="avatar">

                    </a>

                </li>

            </ul>

        </div>

        <div class="header-link-bottom">

            <ul>

                <li><a href="index.php">

                        <img src="../images/header-logout.svg" alt="avatar">

                    </a>

                </li>

                <li><a href="#">

                        <img src="../images/header-setting.svg" alt="avatar">

                    </a>

                </li>

            </ul>

        </div>

    </header>

    <section class="userContainer" style="margin-top: 20px; overflow: auto; max-height: 850px;">

        <form action="api-tests">

            <h1 class="user-themes-title">Создание теста к теме “Название темы”</h1>

            <div class="test\_wrapper-all">

                <div class="test-img-wrapper">

                    <img src="../images/test.png" alt="test-img" style="filter: blur(0);">

                </div>

                <div class="test\_wrapper">

                    <div class="one-que">

                        <input type="text" placeholder="1 Вопрос">

                        <input type="text" maxlength="255" placeholder="Правильный ответ к 1 вопросу">

                    </div>

                    <div class="one-que">

                        <input type="text" placeholder="2 Вопрос">

                        <input type="text" maxlength="255" placeholder="Правильный ответ к 2 вопросу">

                    </div>

                </div>

            </div>

            <button class="final-test" style="background: #C00D83;">Создать тест</button>

        </form>

    </section>

</body>

</html>

createTheme.css-создание темы

.test\_wrapper {}

.test\_wrapper textarea {

display: flex;

width: 100%;

height: 439px;

padding: 40px 350px 20px 40px;

border-radius: 15px;

border: 1px solid #B8B8B8;

background: rgba(184, 184, 184, 0.06);

backdrop-filter: blur(15px);

color: #FFF;

font-family: Poppins;

font-size: 21px;

font-weight: 500;

line-height: 150%;

outline: none;

}

.test\_wrapper-inputs input::placeholder,

.test\_wrapper textarea::placeholder {

color: white;

opacity: .6;

}

.test\_wrapper-inputs {

display: flex;

align-items: center;

justify-content: space-between;

gap: 10px;

}

.file {

width: 100%;

display: contents;

}

.test\_wrapper-inputs input,

.file label {

width: 100%;

padding: 20px 40px;

border-radius: 15px;

border: 1px solid #B8B8B8;

background: rgba(184, 184, 184, 0.06);

backdrop-filter: blur(15px);

color: #FFF;

font-family: Poppins;

font-size: 21px;

font-weight: 500;

line-height: 140%;

cursor: pointer;

outline: none;

}

.file input {

display: none;

}global.css

@import 'reset.css';

@import 'style.css';

@import 'reg-student.css';

@import 'reg-teacher.css';

@import 'profile.css';

@import 'notification.css';

@import 'tasks-for-student.css';

@import 'theme-details.css';

@import 'setting.css';

@import 'test.css';

@import 'tasks-for-teacher.css';

@import 'createTheme.css';

notification.css

.section-notification {

min-height: 851px;

position: relative;

}

.notification-wrapper {

position: absolute;

top: 50%;

left: 50%;

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

text-align: center;

}

.notification-title {

margin-top: 15px;

color: #FFF;

font-family: Poppins;

font-size: 28px;

font-weight: 600;

line-height: 140%;

}

profile.css

|  |  |
| --- | --- |
| .header {  position: fixed;  top: 0;  left: 0;  background: #2E2E3D;  border-radius: 0px 30px 30px 0px;  padding: 17px 25px;  display: flex;  flex-direction: column;  justify-content: space-between;  height: 100%;  }  .header-link-bottom ul,  .header-link-top ul {  display: flex;  flex-direction: column;  gap: 11px;  }  .header-link-bottom ul li a,  .header-link-top ul li a {  display: block;  border-radius: 50%;  border: 1px solid transparent;  transition: all .3s;  opacity: .8;  }  .header-link-bottom ul li a:hover,  .header-link-top ul li a:hover {  border: 1px solid #244BFF;  opacity: 1;  }  .userContainer {  padding: 50px 80px;  max-width: 1753px;  margin-left: 153px;  margin-right: 78px;  background: #2E2E3D;  box-shadow: 13.07088px 13.07088px 104.56705px 0px rgba(36, 75, 255, 0.10);  backdrop-filter: blur(13.070880889892578px);  border-radius: 30px;  }  .user-data {  display: flex;  align-items: center;  justify-content: space-between;  margin-top: 65px  }  .user-data-name {  display: flex;  align-items: center;  gap: 30px;  }  .user-data-name-text {  display: flex;  flex-direction: column;  gap: 5px;  }  .user-data-name-text h1 {  color: #FFF;  font-family: Poppins;  font-size: 22px;  font-weight: 600;  line-height: 140%;  }  .user-data-name-text p {  color: #DCDCDD;  font-family: Poppins;  font-size: 18.299px;  font-weight: 400;  line-height: 140%;  }  .user-data-name-text a {  margin-top: 5px;  color: #FFF;  font-family: Poppins;  text-align: center;  font-size: 14px;  font-weight: 600;  line-height: 140%;  padding: 11px 18px;  border-radius: 13px;  background: #244BFF;  width: fit-content;  transition: all .3s;  }  .user-data-name-text a:hover {  background: #1737c2;  }  .user-data-statistic {  display: flex;  align-items: center;  gap: 15px;  text-align: center; | user-data-statistic div {  display: flex;  flex-direction: column;  gap: 5px;  }  .user-data-statistic span {  color: #DCDCDD;  font-family: Poppins;  font-size: 15px;  font-weight: 400;  line-height: 140%;  }  .user-data-statistic p {  color: #FFF;  font-family: Poppins;  font-size: 28px;  font-weight: 700;  line-height: 140%;  }  .line {  width: 76px;  height: 1px;  background: #DCDCDD;  transform: rotate(90deg);  .user-themes {  margin-top: 20px;  max-height: 555px;  overflow: auto;  }  .user-themes-title {  color: #FFF;  font-family: Poppins;  font-size: 28px;  font-weight: 600;  line-height: 140%;  }  .user-themes-wrapper {  display: grid;  grid-template-columns: repeat(4, 1fr);  gap: 20px;  margin-top: 30px;  max-width: 1356px;  }  .user-themes-inner {  position: relative;  width: 324px;  padding: 138px 20px 20px 20px;  border-radius: 15px;  border: 1px solid #B8B8B8;  background: rgba(184, 184, 184, 0.06);  backdrop-filter: blur(5px);  overflow: hidden;  .user-themes-inner img {  position: absolute;  left: 0;  top: 0;  width: 324px;  height: 137px;  object-fit: cover;  }  .user-themes-inner-title {  display: flex;  justify-content: space-between;  margin-top: 15px;  align-items: center;  }  .user-themes-inner-title h2 {  color: #FFF;  font-family: Poppins;  font-size: 17px;  font-weight: 600;  line-height: 140%;  }  .user-themes-inner-title span {  color: #FFF;  font-family: Poppins;  font-size: 14px;  font-weight: 500;  line-height: 140%;  }  .user-themes-inner-starsAndRepeat {  display: flex;  align-items: center;  justify-content: space-between;  margin-top: 15px;  }  .user-themes-inner-starsAndRepeat span {  color: #FFF;  font-family: Poppins;  font-size: 13px;  font-weight: 400;  line-height: 140%;  }  .user-themes-inner-starsAndRepeat a {  color: #FFF;  font-family: Poppins;  font-size: 13px;  font-weight: 500;  line-height: 140%;  padding: 5px 15px;  border-radius: 90px;  border: 1px solid #1A2DFF;  background: rgba(36, 75, 255, 0.10);  transition: all .3s;  }  .user-themes-inner-starsAndRepeat a:hover {  border: 1px solid transparent;  background: #1A2DFF;  } |

reg-student.css

|  |  |
| --- | --- |
| .section-wrapper {  display: flex;  align-items: center;  justify-content: space-around;  position: absolute;  top: 50%;  left: 50%;  transform: translate(-50%, -50%);  max-width: 1440px;  width: 100%;  }  .reg-form {  display: flex;  flex-direction: column;  gap: 22px;  width: 450px;  }  .logo-img {  width: 256px;  height: 60px;  object-fit: cover;  display: block;  }  .reg-form-title {  color: #FFF;  font-family: Poppins;  font-size: 24px;  font-weight: 500;  line-height: 140%;  }  .reg-form-input {  display: flex;  gap: 7px;  flex-direction: column;  }  .reg-form-input label {  color: #FFF;  font-family: Poppins;  font-size: 18px;  font-weight: 100;  line-height: 140%;  }  .input {  color: #7F7F7F;  font-family: Poppins;  font-size: 18px;  font-weight: 300;  line-height: 140%;  padding: 22px 18px;  transition: all .3s;  background: transparent; | border-radius: 7px;  border: 1px solid #808080;  outline: none;  }  .input:focus {  color: #FFF;  border: 1px solid #FFF;  }  .acc\_have {  color: #FFF;  font-family: Poppins;  font-size: 15px;  font-weight: 300;  line-height: 140%;  }  .acc\_have a {  transition: all .3s;  }  .acc\_have a:hover {  color: #323fd3;  }  .reg-form-btn {  color: #FFF;  font-family: Poppins;  font-size: 18px;  font-weight: 300;  line-height: 140%;  padding: 17px 30px;  gap: 10px;  border-radius: 7px;  background: #1A2DFF;  border: 1px solid transparent;  transition: all .5s;  cursor: pointer;  .reg-form-btn:hover {  border: 1px solid #1A2DFF;  background: transparent;  color: #FFF;  } |

reg-teacher.css

.for-teacher {

background: #C00D83;

.for-teacher:hover {

background: transparent;

border: 1px solid #C00D83;

reset.css

|  |  |
| --- | --- |
| @charset "utf-8";  body,  h1,  h2,  h3,  h4,  h5,  h6 {  font: 14px/18px Verdana, sans-serif;  }  code,  kbd,  pre,  samp,  tt,  var {  font: 14px/18px Consolas, "Courier New", monospace;  }  body,  optgroup {  color: #000;  }  body,  code,  dl,  dd,  form,  pre {  margin: 0;  }  a:link {  color: white;  }  a:link,  a:visited,  ins {  text-decoration: none;  }  a:hover {  text-decoration: none;  }  a:link img,  a:visited img,  object,  fieldset,  abbr,  acronym {  border: none;  }  sub,  sup {  position: relative;  display: inline-block;  font-size: 11px;  line-height: 11px;  }  sub {  top: 5px;  vertical-align: middle;  }  sup {  top: -2px;  margin-top: -11px; | }  abbr,  acronym {  cursor: help;  border-bottom: 1px dotted #999;  }  address,  cite,  dfn {  font-style: normal;  }  code,  pre {  white-space: pre-wrap;  }  img {  vertical-align: bottom;  }  td,  th,  textarea {  vertical-align: top;  }  input,  select,  button {  overflow: visible;  vertical-align: middle;  }  legend {  color: #000;  margin: 0 0 -2px;  padding: 0;  }  li {  display: block;  list-style: none;  }  optgroup option {  padding-left: 22px;  }  table {  border-collapse: collapse;  }  caption,  th {  text-align: left;  }  td,  th {  padding: 0;  }  address,  blockquote,  h1,  h2,  h3,  h4,  h5,  h6,  fieldset,  p,  ol,  ul {  margin: 0;  padding: 0;  }  th,  h1,  h2,  h3,  h4,  h5,  h6 {  font-weight: normal;  }  q {  quotes: "\00ab" "\00bb" "\201e" "\201c";  }  q:after {  content: "\00bb";  }  q:before {  content: "\00ab";  }  q>q:after {  content: "\201c";  }  q>q:before {  content: "\201e";} |

setting.css

|  |  |
| --- | --- |
| .container-setting {  max-width: 707px;  width: 100%;  position: absolute;  top: 50%;  left: 50%;  transform: translate(-50%, -50%);  }  .setting-actions {  display: flex;  gap: 20px;  }  .setting-ava {  position: relative;  }  .setting-ava input {  display: none;  }  .setting-ava label img {  position: relative;  width: 237px;  height: 237px;  object-fit: cover;  cursor: pointer;  opacity: .6;  }  .setting-ava label svg {  position: absolute;  top: 50%;  left: 50%;  transform: translate(-50%, -50%);  cursor: pointer;  }  .setting-inputs {  display: flex;  flex-direction: column;  gap: 15px;  width: 100%;  }  .setting-inputs input {  background: transparent;  outline: none;  border: 1px solid #FFF;  border-radius: 7px;  padding: 22px; | text-align: center;  color: white;  font-family: Poppins;  font-size: 18px;  font-weight: 300;  line-height: 140%;  transition: all .3s;  opacity: 0.5;  }  .setting-inputs input:focus {  opacity: 1;  }  .setting-btns {  margin-top: 30px;  display: flex;  align-items: center;  gap: 20px;  }  .setting-btns a {  justify-content: center;  color: #FFF;  font-family: Poppins;  font-size: 16px;  font-weight: 600;  line-height: 140%;  display: flex;  width: 100%;  padding: 15px 18px;  border-radius: 5px;  background: #E02547;  transition: all .3s;  opacity: .8;  }  .setting-btns a:hover {  opacity: 1;  }  .setting-btns a:nth-child(2) {  background: #244BFF; |

}style.css

|  |  |
| --- | --- |
| \*,  \*::before,  \*::after {  box-sizing: border-box;  }  html {  scroll-behavior: smooth;  }  body {  font-family: 'Poppins', sans-serif;  font-weight: 500;  font-size: 16px;  line-height: 23px;  color: #fff;  background: #212121;  overflow: hidden;  }  a {  text-decoration: none;  color: #000000;  }  .container {  max-width: 1440px;  padding: 0 15px;  margin: 0 auto;  }  .index-decor {  position: absolute;  top: -178px;  right: -952px;  }  .section\_bottom,  .section\_top {  margin-top: 80px;  }  .section\_top-wrapper {  display: flex;  align-items: center;  justify-content: space-between;  }  .section\_top-left {  display: flex;  flex-direction: column;  gap: 30px;  .section\_top-title {  width: 720px;  color: #FFF;  font-family: Poppins;  font-size: 60px;  font-weight: 600;  line-height: 140%;  }  .section\_top-title li {  color: #1A2DFF;  }  .section\_top-description {  font-family: Poppins;  max-width: 502px;  color: #FFF;  font-size: 19px;  font-weight: 500;  line-height: 140%;  }  .section\_top-form {  display: flex;  align-items: center;  gap: 10px;  padding: 10px;  border-radius: 5px;  border: 1px solid #1A2DFF;  background: rgba(26, 45, 255, 0.17);  max-width: 426px;  }  .section\_top-btn {  border-radius: 5px;  background: #C00D83;  padding: 17px 37px;  gap: 10px;  text-align: center;  border: 1px solid transparent;  outline: none;  cursor: pointer;  color: #FFF;  font-family: Poppins;  font-size: 19px;  font-weight: 500;  line-height: 140%;  transition: all .4s;  }  .section\_top-btn:hover {  border: 1px solid #C00D83;  color: #C00D83; | background: transparent;  }  .section\_top-btn:nth-child(2) {  background: #1A2DFF;  }  .section\_top-btn:nth-child(2):hover {  border: 1px solid #1A2DFF;  color: #1A2DFF;  background: transparent;  }  .section\_blocks {  display: flex;  align-items: center;  justify-content: space-between;  }  .block {  position: relative;  display: block;  width: 450px;  height: 257px;  background: linear-gradient(0deg, rgba(0, 0, 0, 0.50) 0%, rgba(0, 0, 0, 0.50) 100%), lightgray 50% / cover no-repeat;  border-radius: 20px;  }  .block img {  filter: blur(0.5px);  border-radius: 20px;  object-fit: cover;  }  .block\_title {  position: absolute;  top: 30px;  left: 25px;  color: #FFF;  font-family: Poppins;  font-size: 24px;  font-weight: 700;  line-height: 140%;  }  .block\_description {  position: absolute;  top: 80px;  left: 25px;  color: #FFF;  font-family: Poppins;  font-size: 19px;  font-weight: 500;  line-height: 140%;  } |