Частное учреждение

профессиональная образовательная организация

ТЕХНИКУМ «БИЗНЕС И ПРАВО»

ЗАЩИЩЕНО С ОЦЕНКОЙ

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

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

(подпись и ФИО руководителя)

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

**КУРСОВАЯ РАБОТА**

|  |  |
| --- | --- |
| Тема | Создание Web-сайта учебного курса «Вычислительные системы, сети и телекоммуникации» |
| По МДК 09.01 | Проектирование и разработка веб-приложений |
| Выполнил | Семернин А.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](#_Toc153265751)

[1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 5](#_Toc153265752)

[1.1 Анализ предметной области 5](#_Toc153265753)

[1.2 Цели и задачи выполняемой системы 5](#_Toc153265754)

[1.3 Метод реализации процесса проектирования 6](#_Toc153265755)

[1.4 Обоснование проектных решений 7](#_Toc153265756)

[2. ПРОЕКТНАЯ ЧАСТЬ 19](#_Toc153265758)

[2.1 Техническое задание 19](#_Toc153265759)

[2.2 Программное обеспечение информационной системы 21](#_Toc153265760)

[2.3 Руководство пользователя 22](#_Toc153265761)

[2.4 Руководство администратора 23](#_Toc153265762)

[ЗАКЛЮЧЕНИЕ 25](#_Toc153265763)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 26](#_Toc153265764)

[ПРИЛОЖЕНИЕ](#_Toc153265765)

# ВВЕДЕНИЕ

Актуальность исследования – успехи в учебе студентов напрямую связаны с системой обучения. В зависимости от того, как реализовано обучение, изучение информации и ее усвоение зависит успех в изучении продукта, предоставленных преподавателями. Создание учебного курса подразумевает создание сайта, способного помочь студентам усвоить нужные ему темы предметов. Актуальность темы заключается в том, что учебный курс поможет закрепить усвоенную информацию на занятиях или же с нуля.

Объект исследование – информационная система учебного курса.

Предмет исследования – предметом исследования являются процессы создания, изучения и прохождение тестирования по изученным темам.

Цель исследования – рассмотреть и установить повышение эффективности процессов обучения, создания тем для изучения и внедрение информационной системы учебного курса

Задачи исследования:

– изучить предметную область;

– рассмотреть ряд существующих разработок для решения поставленной цели;

– выбрать метод реализации работы учебного курса;

– спроектировать работу разрабатываемой системы учебного курса;

– разработать информационную систему;

– протестировать разработанную информационную систему учебного курса.

Методы исследования:

* Анализ лучших практик в создании веб-учебников.
* Эксперимент с различными методами обучения, чтобы выбрать наилучшие для целевой аудитории.

Учебный курс – подразумевает постоянную возможность изучения нужных студенту тем или создания их преподавателем.

Данный учебный курс может быть полезен студентам программистского направления и всем, кто желает освоить основы программирования и связанных с программированием тем. Благодаря понятному материалу, практическим заданиям и интерактивным элементам, он предоставляет возможность эффективного изучения деятельности программиста и связанных с ними представителей, что делает его интересным и полезным ресурсом.

# 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

## Анализ предметной области

Разрабатываемая система предназначена для более удобного и легкого изучения тем, с помощью учебного курса.

Система будет очень удобной, лёгкой в своём использовании. Для преподавателей и студентов будет всё понятно и доступно.

Преподаватель сможет очень легко и быстро создавать темы и тесты для изучения их студентами. Тем самым будет очень легко и просто давать студентам лекции с возможностью улучшения их знаний. Работа учебного курса не будет заторможенной, соответственно будет хорошая оптимизация, понятный интерфейс, доступный преподавателям и студентам.

## 1.2 Цели и задачи выполняемой системы

Основные цели системы:

* Разработать удобный интерфейс для обучения.
* Реализовать функциональность регистрации и авторизации пользователей.
* Структурировать материалы учебника для последовательного и логического изучения.
* Обеспечить систему автоматического обновления обучающего контента.

Задачами являются:

* Достичь поставленных целей;
* Автоматическое обновление данных;
* Освоить навыки, полученные на учении;
* Связать учебник с базой данных для эффективного хранения информации.
* Создать отчет о выполненной работе и результатах обучения.

## 1.3 Метод реализации процесса проектирования

При разработке учебного курса внедряются инновационные методы, направленные на обеспечение эффективности обучения и управления контентом. Процесс проектирования системы включает в себя автоматизацию проверки знаний, и управление пользователями, обеспечивая легкий доступ к образовательному материалу.

В ходе внедрения системы будут достигнуты следующие цели:

* Предоставление администраторам возможности удаления устаревших данных.
* Обеспечение удобной и правильной работы системы, способствующей повышению интереса пользователей и формированию положительного рейтинга.

Объекты автоматизации включают в себя:

* Обеспечение максимальной интерактивности и учебной эффективности системы.
* Создание технологичной и надежной платформы для обучения.
* Минимизация необходимости человеческого вмешательства в процессы обновления и поддержки системы.
* Регулярное обновление и совершенствование материалов учебника.

В результате внедрения данной системы ожидается создание инновационного образовательного ресурса, способствующего развитию студентов, изучающих материал и удовлетворяющего потребности пользователей в современных методах обучения.

## 1.4 Обоснование проектных решений

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

**Изображение выглядит как текст, визитная карточка, снимок экрана, Прямоугольник

Автоматически созданное описание**

Рисунок 1.1 – Модули которые должны быть в программе

На данном этапе необходимо описать примерную архитектуру разрабатываемой системы.

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

– Представление данных — на стороне клиента.

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

– Управление ресурсами — сервер БД (базу данных), который и представляет запрашиваемые данные.

Структура трехзвенной клиент-серверной архитектуры представлена на рисунке 1.2:

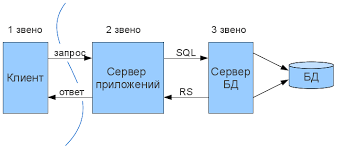


Рисунок 1.2 – 3-х звенная клиент серверная архитектура

Для данного проекта выбрана эта архитектура, 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).

MySQL – это система управления базами данных. База данных представляет собой структурированную совокупность данных. Эти данные могут быть любыми - от простого списка предстоящих покупок до перечня экспонатов картинной галереи или огромного количества информации в корпоративной сети.

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 (учебный курс) представлены в таблице 1.1.

Таблица 1.1 – Требования к системе учебного курса

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибуты | Метки | | 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  (не включая задержки в сети); | |

Продолжение таблицы 1.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | | 3 |
|  | Производительность |  | |
| 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; | |
| 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 | - Должна поддерживать  внедрение новых версий;  - Система должна быть применима вместо других  программных систем для решения тех же задач в определенном окружении. | |

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

В системе существуют 2 группы пользователей: Преподаватели и Студенты. Студент выполняет функцию: просмотра, обновления, поиск.

Преподаватель выполняет функцию: просмотра, занесение данных, редактирование, удаление данных, обновления, поиск, 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 | |
| 1 | | 2 | | 3 |
| Надёжность | | | | |
| Устойчивость к отказу | Коэффициент аварийных | | - Не менее 90% ошибок | |

Продолжение таблицы 1.2

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
|  | отказов | 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 временя 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 |
| 1 | 2 |
| Отображение данных | Отсутствие соединения с базой данных |
| Поиск | Отсутствие соединения с базой данных |
| Добавление | Отсутствие соединения с базой данных |
| Редактирование | Отсутствие соединения с базой данных |

Продолжение таблицы 1.3

|  |  |
| --- | --- |
| 1 | 2 |
| Удаление | Отсутствие соединения с базой данных |

Требования к функциям учебного курса, выполняемым программным обеспечением

Имя: Вход студента

Описание: Один из студентов мог зайти по уникальному адресу

Действующий субъект: Студент

Предусловие: специалист сотрудник зашел в программу учебного курса Сценарий:

– сотруднику предоставляется персональный компьютер

– пользователь заходит на сайт по уникальному адресу

– 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 1.3 представлена диаграмма последовательности сценария "Входа студента".

Изображение выглядит как линия, текст, диаграмма, График

Автоматически созданное описание

Рисунок 1.3 – Диаграмма последовательности сценария "Входа студента"

Имя: поиск товаров на складе

Описание: сотрудник имеет возможность найти нужный ему товаров на складе

Действующий субъект: преподаватель

Предусловие: преподаватель успешно зашел в систему, используя уникальный адрес

Сценарий:

– система отображает весь список доступных тем

– сотрудник в поле поиска пишет то, какая тема ему нужна и он хочет найти именно ее

–студент нажимает на нужный ему проект по изучению

– система входит в выбранную студентом тему и дает возможность для изучения необходимых данных

Имя: добавление

Описание: Преподаватель имеет возможность добавить новые данные

Действующий субъект: сотрудник

Предусловие: сотрудник успешно зашел в систему, используя уникальный адрес

Сценарий:

– система добавляет новые данные

– сотрудник в полях указывает данные

– сотрудник нажимает на кнопку «Добавить»

– система обновляет приложение и выводит добавленные данные. Те, что ввел преподаватель

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

На рисунке 1.6 представлена диаграмма последовательности сценария "Добавления".Изображение выглядит как текст, линия, График, диаграмма

Автоматически созданное описание

Рисунок 1.6 - последовательности сценария "Добавления"

Имя: редактирование данных

Описание: сотрудник имеет возможность редактировать данные

Действующий субъект: Сотрудник

Предусловие: пользователь успешно зашел в систему, используя уникальный адрес

Сценарий:

– система редактирует темы

– преподаватель вносит в поля редактирования ту информацию, на какую хочет заменить

– преподаватель нажимает на кнопку «Редактировать»

– система обновляет сайт и выводит данные с заменёнными данными

На рисунке 1.7 представлена диаграмма последовательности сценария "Редактирования".

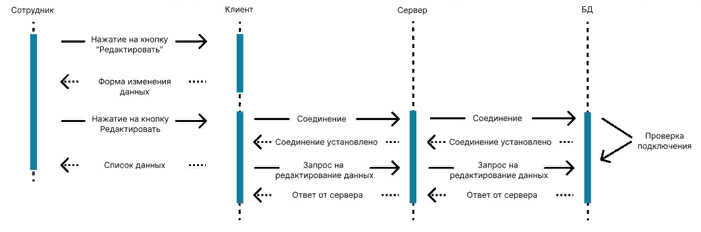


Рисунок 1.7 – Диаграмма последовательности сценария "Редактирования"

Имя: удаление данных

Описание: преподаватель имеет возможность удалить данные

Действующий субъект:

Преподаватель

Предусловие:

преподаватель успешно зашел в систему, используя уникальный адрес

Сценарий:

– система удаления тем

– сотрудник выбирает тему который хочет удалить

– сотрудник нажимает на кнопку «Удалить»

– система обновляет данные и удаляет выбранную тему

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

На рисунке 1.7 представлена диаграмма последовательности сценария "Удаления".

Изображение выглядит как линия, текст, диаграмма, График

Автоматически созданное описание

Рисунок 1.7 – Диаграмма последовательности сценария "Удаления"

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

– интернет соединение;

– обеспечить каждого специалиста персональным компьютером;

– установить на каждый ПК данную программу;

– иметь программу xampp;

– база данных MySQL.

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:

–lectures– таблица содержащая данные лекций.

–users– таблица содержащая информацию о пользователях.

Данные о темах представлены в таблице lectures 1.4.

Таблица 1.4 – Данные о темах

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| Reading\_id | int | 11 | Идентификатор категории | Первичный ключ |
| User\_id | int | 11 | Название категории | NULL |

Продолжение таблицы 1.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lecture\_id | int | 11 | Уникальный номер лекции | NULL |
| Is\_read | tinyint | 1 | Когда была прочитана тема | NOT NULL |

Данные о пользователе должны размещаться в таблице users 1.5.

Таблица 1.5 – Данные о пользователе

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | Int | 11 | Идентификатор категории | Первичный ключ |
| Login | varchar | 50 | Наименование логина | NOT NULL |
| Name | varchar | 100 | Имя пользователя | NOT NULL |
| Password | varchar | 255 | Значение пароля | NOT NULL |
| Avatar | varchar | 255 | Сохранение аватарки | NOT NULL |
| Role | Enum | - | Роль пользователя | NOT NULL |
| status | enum | - | Статус пользователя | NOT NULL |

В данном окне указывается название темы, ее содержимое, картинка,

На рисунке 1.8 представлено окно добавления темы.

Изображение выглядит как текст, снимок экрана, мультимедиа, программное обеспечение

Автоматически созданное описание

Рисунок 1.8 – Окно добавление темы для изучения

На рисунке 1.9 представлено окно просмотра всех товаров на складе.

Изображение выглядит как снимок экрана, текст, Мультимедийное программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

Рисунок 1.9 – Список тем доступных для изучения

На рисунке 2.1 – представлено окно просмотра выбранной темы.

Изображение выглядит как снимок экрана, текст

Автоматически созданное описание

Рисунок 2.1 – Окно показа выбранной темы

На рисунке 2.2 – представлено окно удаления темы.

Изображение выглядит как снимок экрана, текст

Автоматически созданное описание

Рисунок 2.3 – Окно удаления теста

На рисунке 2.4 – представлено окно редактирования темы.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2.4 – Окно редактирования теста

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 2.5 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.

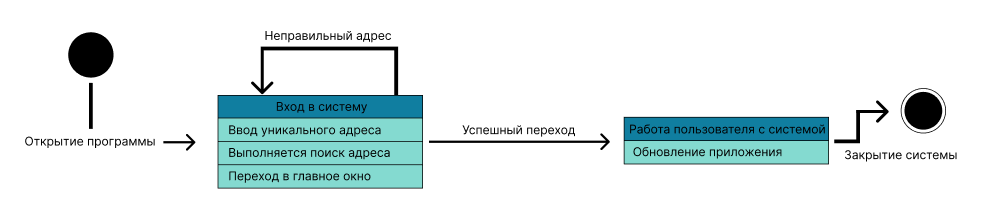


Рисунок 2.5 – Общая диаграмма состояний системы

На рисунке 2.6 показано детальное описание работы пользователя с системой. Сотрудник может просматривать список всех данных, добавлять, удалять, редактировать, производить поиск.

Изображение выглядит как снимок экрана, диаграмма, текст, дизайн

Автоматически созданное описание

Рисунок 2.6 – Диаграмма состояний работы пользователя с системой

На рисунке 2.7 изображена декомпозиция состояния «Добавление». После того как сотрудник 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.

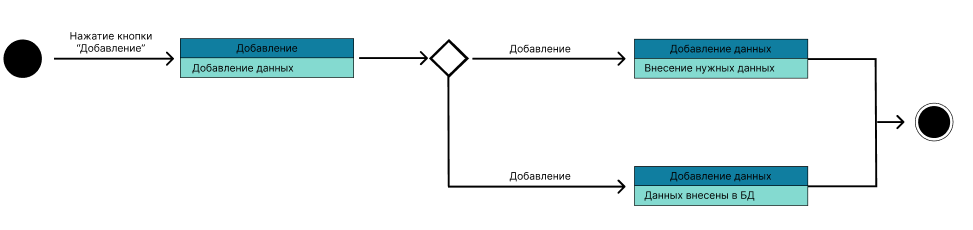


Рисунок 2.7 – Диаграмма состояний добавление данных

На рисунке 2.8 изображена декомпозиция состояния «Редактирования».

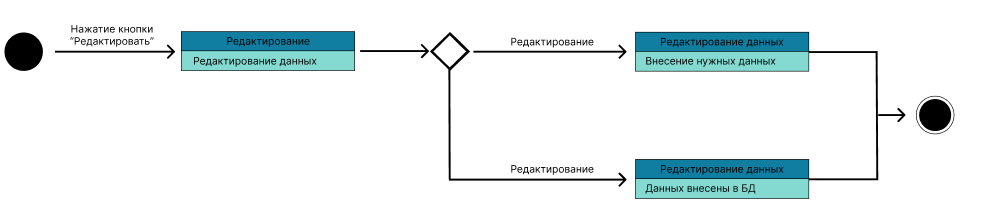
Сотрудник должен нажать на кнопку «Редактировать». После этого приложение обновится и обновит данные в базе данных. Далее сотрудник может просмотреть все изменённые данные.

Рисунок 2.8 – Диаграмма состояний редактирования данных

На рисунке 2.9 изображена диаграмма состояния «Удаления». Сотрудник может удалять все данные товара. Далее необходимо нажать на кнопку «Удалить» и программа выполнит запрос.

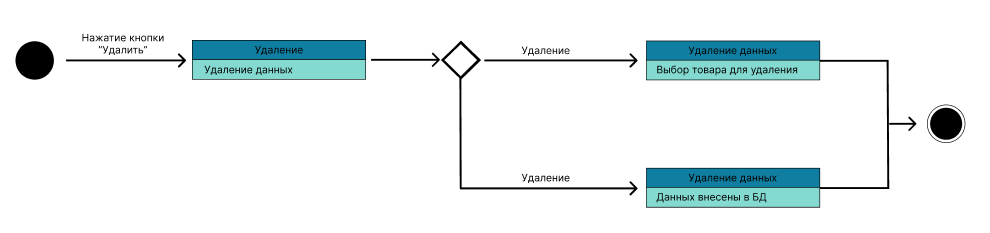


Рисунок 2.9 – Диаграмма состояний удаления данных

На рисунке 3.1 изображена диаграмма состояния «Поиска». Сотрудник может делать поиск по всем типам данных. Далее необходимо нажать на кнопку «Поиск» и программа выполнит запрос поиск.

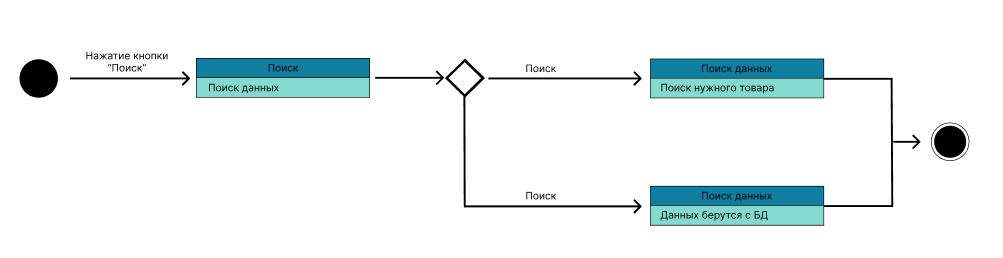


Рисунок 3.1 – Диаграмма состояний поиска данных

# 2. ПРОЕКТНАЯ ЧАСТЬ

## 2.1 Техническое задание

Все данные сайта должны храниться в структурированном виде под управлением реляционной СУБД. Исключения составляют файлы данных, предназначенные для просмотра и скачивания (изображения, видео, документы и т.п.). Такие файлы сохраняются в файловой системе, а в БД размещаются ссылки на них.

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.

Основные функции СУБД:

– управление данными во внешней памяти (на дисках);

– управление данными в оперативной памяти с использованием дискового кэша;

– журнализация изменений (сохранение истории), резервное копирование и восстановление базы данных после сбоев;

– поддержка языков БД (язык определения данных, язык манипулирования данными).

Для реализации статических страниц и шаблонов должны использоваться языки HTML 4.0 и CSS.

HTML (от английского HyperText Markup Language) — это язык гипертекстовой разметки текста. Он нужен, чтобы размещать на веб-странице элементы: текст, картинки, таблицы и видео.

HTML состоит из тегов — команд, которые указывают браузеру, как отображать помещённый в них текст. Это и есть элементы веб-страницы. У каждого тега есть имя, которое заключается в угловые скобки < и >.

Для реализации интерактивных элементов сервероной части должны использоваться языки php, javascript.

Для реализации динамических страниц должен использоваться язык PHP.

Объем одной стандартной загружаемой страницы сайта в среднем не должен превышать 170 kb.

Объем flash-заставки не должен превышать 300 Kb.

CSS (Cascading Style Sheets, каскадные таблицы стилей) — язык описания внешнего вида HTML-документа. Это одна из базовых технологий в современном интернете. Практически ни один сайт не обходится без CSS, поэтому HTML и CSS действуют в единой связке.

Каскадные таблицы стилей работают с HTML, но это совершенно другой язык. HTML структурирует документ и упорядочивает информацию, а CSS взаимодействует с браузером, чтобы придать документу оформление.

## 2.2 Программное обеспечение информационной системы

Для функционирования сайта необходимо следующее программное обеспечение:

– Операционная система: Windows 10/Linux/macOS;

– Сервер: MS SQL Server 2019;

– СУБД: MSSQL для MS SQL Server;

– XAMPP.

SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире. Данная СУБД подходит для самых различных проектов: от небольших приложений до больших высоконагруженных проектов.

SQL Server был создан компанией Microsoft. Первая версия вышла в 1987 году. А текущей версией является версия 2022, которая вышла в ноябре 2022 году и которая будет использоваться в текущем руководстве.

SQL Server долгое время был исключительно системой управления базами данных для Windows, однако начиная с версии 16 эта система доступна и на Linux.

Необходимое программное обеспечение для персонального компьютера:

- Операционная система: Windows 10/Linux/macOS

Сайт должен быть доступен для полнофункционального просмотра с помощью следующих браузеров:

– Opera 6.0 и выше;

– Google;

– Yandex.

JavaScript - предназначен для написания сценариев для активных HTML-страниц. Язык JavaScript не имеет никакого отношения к языку Java. Java разработан фирмой SUN. JavaScript - фирмой Netscape Communication Corporation. Первоначальное название - LiveScript. После завоевания языком Java всемирной известности LiveScript из коммерческих соображений переименовали в JavaScript.

2.3 Руководство пользователя

Авторизация и Регистрация: Для доступа ко всем урокам необходимо авторизоваться. Нажмите на кнопку «Авторизация» в левом верхнем углу.

Если у вас нет аккаунта, вы можете создать его, нажав «Создать аккаунт» и заполнив необходимую информацию, включая имя, логин, пароль и прохождение капчи.

После успешной авторизации или регистрации вы попадете на страницу уроков.

Структура Уроков:

Уроки представлены в виде глав, внутри которых находятся темы, а темы подразделены на параграфы.

Навигация между главами, темами и параграфами осуществляется интуитивно понятным образом.

Чтение и Ответы на Задания

Авторизированные пользователи могут читать уроки и отвечать на задания в конце каждого урока.

Профиль и Настройки:

В личном кабинете (левый верхний угол, где указан логин) вы можете выбрать «Настройки», чтобы изменить информацию в вашем профиле.

Также доступна опция «Удалить аккаунт» для удаления профиля.

FAQ и Помощь:

Перейдите на страницу FAQ, чтобы получить дополнительную информацию и ответы на часто задаваемые вопросы.

Обратная Связь:

На всем сайте размещены формы обратной связи. Вы можете написать свою почту и вопрос, затем нажать «Отправить».

Выход из Аккаунта:

В левой нижней части сайта вы можете выбрать «Выйти» для безопасного завершения сеанса, кнопка «Выйти» обозначена специальным понятным знаком виде двери.

2.4 Руководство администратора

Авторизация:

Как администратор, вы авторизуетесь так же, как и обычный пользователь, нажимая на «Авторизация» и вводя свой логин и пароль.

Роль администратора определена в базе данных как «admin».

Доступ к Админ-Панели:

После создания темы администратор увидит поля редактирования и удаления лекции.

Нажмите на этот пункт, чтобы перейти в административную панель.

Обзор Пользователей:

В админ-панели отображается таблица со всеми пользователями, включая их данные.

Каждая строка содержит кнопки «Изменить» и «Удалить» для управления данными пользователя.

Функции Администрирования:

Изменить: Нажмите на кнопку «Изменить» рядом с пользователем для внесения изменений в его профиль.

Удалить: Кнопка «Удалить» удаляет профиль пользователя. Подтвердите действие, чтобы завершить удаление.

Создать:На странице админ-панели есть кнопка «Создать», которая позволяет администратору добавлять новые темы.

Безопасность:

Пожалуйста, будьте осторожны при удалении тем лекций, так как это действие необратимо.

Выход из Админ-Панели:

Для завершения сеанса в админ-панели выберите «Выйти» из вашего личного кабинета.

# ЗАКЛЮЧЕНИЕ

При разработке приложения были достигнуты цели курсовой работы, а именно: Разработан удобный интерфейс для обучения, Реализована функциональность регистрации и авторизации пользователей, Структурированы материалы учебника для последовательного и логического изучения, Обеспечена система автоматического обновления обучающего контента.

Удалось создать трёхуровневую архитектуру приложения. Цель работы достигнута, а именно создано приложение для работы с базой данных.

Особое внимание уделено удобству ввода и редактирования данных.

Теперь куда проще создавать темы для изучения и хранить их на сайте.

Задачи решены, а именно создать трёхуровневую архитектуру приложения, быстрый доступ к базе данных, добавлению темы, просмотра, редактирования и удаления любой темы.

В результате выполнения курсовой, был создан сайт для внесения и хранения там данных о темах. Сайт дает преподавателям быстро и эффективно создавать темы для своих любимых студентов, чтобы те в последствии могли получить удовольствие от обучения, путем изучения новых тем, по лекциям связанных, с темами вычислительных систем, сетей и телекоммуникаций. Основная задача сайта — обеспечить преподавателям возможность быстрого и эффективного создания учебных тем, предназначенных для их студентов. Этот инструмент позволяет студентам получать удовольствие от обучения. Для его разработки применялось актуальное программное обеспечение, 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.

Всё информационное наполнение сайта находится в БД.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вандюк, Джон К. CMS Drupal. Руководство по разработке системы управления сайтом / Вандюк, Джон К., Мэтт Вестгейт,. - М.: Вильямс, 2019. - 400 c.
2. 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.
3. Дакетт, Джон HTML и CSS. Разработка и дизайн веб-сайтов / Джон Дакетт. - Москва: Наука, 2019. - 480 c.
4. Дронов, В. JavaScript в Web-дизайне / В. Дронов. - М.: СПб: БХВ, 2019. - 880 c.
5. Дронов, В. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов / В. Дронов. - М.: БХВ-Петербург, 2019. - 480 c.
6. 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.
7. Дронов, Владимир JavaScript и AJAX в Web-дизайне / Владимир Дронов. - Москва: Высшая школа, 2019. - 736 c.
8. Дронов, Владимир PHP, MySQL и Dreamweaver MX 2004. Разработка интерактивных Web-сайтов / Владимир Дронов. - М.: "БХВ-Петербург", 2019. - 448 c.
9. Колисниченко, Д. PHP 5/6 и MySQL 6. Разработка Web-приложений / Д. Колисниченко. - М.: БХВ-Петербург, 2021. - 560 c.
10. Колисниченко, Д.Н. PHP 5/6 и MySQL 6. Разработка Web-приложений / Д.Н. Колисниченко. - М.: БХВ-Петербург, 2021. - 364 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%;  } |