Содержание

ВВЕДЕНИЕ..... 4
Глава 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ..... 6
1.1. Обзор портала..... 6
1.2. Обоснование разработки модуля портала..... 6
1.3 Организационная структура образовательного учреждения. Документопоток..... 7
1.4 Анализ IT-структуры..... 10
1.5 Структура электронной информационно-образовательной среды (ЭИОС) образовательного учреждения..... 12
1.6 Вывод по главе..... 13
Глава 2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ..... 14
2.1 Требования к дизайну сайта..... 14
2.2 Требования к функциональности сайта..... 15
2.2.1 Требования к содержимому сайта..... 16
2.2.2 Детальное описание страниц сайта..... 16
2.3 Вывод по главе..... 17
Глава 3. ОБЗОР ПРОГРАММНЫХ, АППАРАТНЫХ И/ИЛИ ИНЖЕНЕРНО-ТЕХНИЧЕСКИХ СРЕДСТВ ДЛЯ РЕАЛИЗАЦИИ ПРАКТИЧЕСКОЙ ЧАСТИ ВКР. РЕАЛИЗАЦИЯ ТЕХНИЧЕСКОГО ЗАДАНИЯ..... 18
3.1 Обзор средств для реализации практической части выпускной квалификационной работы..... 18
3.2 Реализация технического задания..... 18
3.2.1. Создание базы данных..... 18
3.2.2. Написание программного кода..... 23
3.2.3 Выполнение отладки тестирования..... 34
3.2.4 Оформление документации на программного обеспечения..... 36
3.3 Экономическая часть..... 39
3.4 Вывод по главе..... 40
ЗАКЛЮЧЕНИЕ..... 42
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ..... 43

ВВЕДЕНИЕ

Век информационных технологий затронул практически все сферы общества, включая образование.

Автоматизация рабочего процесса становится необходимостью для учебного заведения, так как данный процесс позволяет повысить эффективность работы преподавателей.

Разработка веб-сайта включает в себя следующие этапы:

- проектирование сайта (разработка технического задания, проектирование интерфейса);
- создание дизайн - концепции сайта;
- создание макетов страниц;
- создание мультимедиа и FLASH-элементов;
- вёрстка страниц и шаблонов;
- программирование (разработка функциональных инструментов) или интеграция в систему управления содержимым;
- оптимизация и размещение материалов сайта;
- тестирование и внесение корректировок;
- публикация проекта на хостинге;
- обслуживание работающего сайта или его программной основы.

В зависимости от задачи, какие-то из этапов могут отсутствовать.

Распространенными автоматизированными информационными системами на данный момент являются «Университет» и «1С Вуз». Вышеупомянутые системы обеспечивают автоматизацию работы сотрудников учебных заведений, однако, данные системы невозможно применить к учету успеваемости студентов. В настоящее время ни одна из существующих систем не решает задачу вычисления рейтинга и формирования отчетов на основе успеваемости студентов учебных заведений. В связи с этим, в учебных заведениях необходим продукт, обеспечивающий автоматизацию процесса подсчета рейтинга и формирования отчета по каждому студенту. В этом заключается актуальность работы.

Разработка проекта автоматизации для клиентской части системы успеваемости студентов учебного заведения позволит упростить процесс проставления оценок, а также предоставит доступ к просмотру успеваемости студенту в онлайн режиме.

В связи с увеличением количество студентов, большое значение приобрела проблема контроля за большим количеством студентов в учебном процессе. Разработанный модуль решит данную проблему.

Целью выпускной квалификационной работы является разработка проекта автоматизации для клиентской части системы успеваемости студентов образовательного учреждения на таких языках, как: HTML, CSS, JavaScript, PHP, SQL.

Для разработки системы требуется выполнить следующие задачи:

- провести анализ предметной области;
- спроектировать и разработать систему успеваемости студентов.

Объектом исследования является проект автоматизации для клиентской части системы успеваемости студентов образовательного учреждения.

Предмет исследования – клиентская часть системы успеваемости студентов.

Выпускная квалификационная работа состоит из введения, трех глав, заключения, списка использованных источников.

В первой главе проводится анализ предметной области.

Во второй главе проводится описание технического задания.

В третьей главе производится проектирование, разработка, а также описывается экономическая часть ВКР.

В заключении приведены результаты и выводы по выполненной работе.

Глава 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Обзор портала

Современное образовательное учреждение – это в том числе и высокий уровень информатизации и проникновения интернет-технологий во внутренние бизнес-процессы и в учебный процесс.

У любого учебного заведения, в том числе и у колледжа, имеется большой объем данных об успеваемости, посещаемости студентов и другой информации. Портал позволяет полностью контролировать всю деятельность учебного заведения. Таким образом, портал, внедренный в учреждение, способствует повышению эффективности работы, за счет того, что данные обрабатывается автоматически.

Разработка портала (его модуля) производится на кросс-платформенном редакторе исходного кода, например Visual Studio Code. Данный редактор обладает доступным набором инструментов для редактирования и отладки. Редактор легко интегрируется с другими сервисами. Его собственные свойства также легко расширить.

1.2. Обоснование разработки модуля портала

Недостатки ведения бумажных форм отчетности, отсутствие возможности осуществления контроля студентов прозрачными для администрации учебного заведения привели к необходимости создания и интегрирования в существующую систему управления учебным заведением разрабатываемого модуля.

Разработка проекта автоматизации для клиентской части системы успеваемости студентов будет являться частью комплексного решения для образовательного учреждения, которое позволит построить закрытую социально-образовательную сеть для учащихся и преподавателей, интегрированную в учебный процесс.

В портале студент может просмотреть только оценки за экзамен, зачёт. Созданный модуль позволит предоставить студентам полную и оперативную

информацию об их успеваемости, а также вести преподавателям учёт успеваемости по занятиям.

Использование разработанного модуля позволит легко вычислять средний балл на основе выставленных оценок, упростит процедуру формирования оценок по дисциплинам, повысит доступность информации о результатах учебного процесса.

Одним из популярных редакторов исходного кода является VS Code. Он разработан компанией Microsoft. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. В нём реализован функционал необходимый для разработки модуля по успеваемости студентов образовательного учреждения.

1.3 Организационная структура образовательного учреждения. Документопоток

На рисунке 1 изображена типовая организационная структура образовательного учреждения. Проректор по учебной работе координирует составление кафедрами учебных программ, обеспечивает подготовку расписаний профессорско-преподавательского состава и др. Под руководством проректора по учебной работе входят: деканат и кафедры.

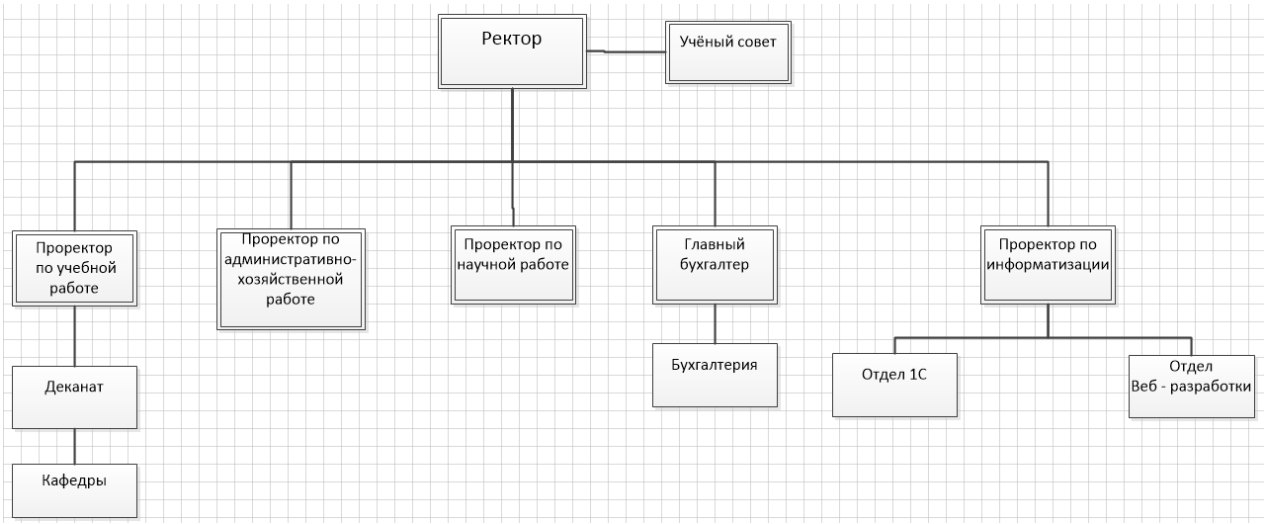


Рисунок 1 - Организационная структура

Деканат выполняет руководящую, координирующую и контролирующую функции. В перечень его основных задач входит:

- контроль над деятельностью преподавателей (выполнение учебного плана, ведение документации и пр.);
- оценка успеваемости студентов;
- совместная работа с приемной комиссией в части приема и зачисления абитуриентов.

Кафедры занимаются разработкой и предоставлением на утверждение рабочих программ дисциплин, проводят контроль успеваемости, промежуточной аттестации обучающихся, также проводят учебные занятия, предусмотренные соответствующими учебными планами и пр.

Проректор по административно-хозяйственной работе руководит капитальным ремонтом, строительством, обеспечивает структурные подразделения необходимым оборудованием и материалами и пр.

Проректор по научной работе обеспечивает научно-методическую и консультационную поддержку преподавателей, студентов, магистрантов, проводит работу с кафедрами в области организации системы подготовки научных кадров и пр.

Главный бухгалтер осуществляет организацию бухгалтерского учета хозяйственно-финансовой деятельности и контроль за экономным использованием ресурсов.

Бухгалтерия занимается учётом всех хозяйственных операций учреждения, также занимаются расчётом по заработной плате с сотрудниками учреждения и пр.

Проректор по информатизации координирует освоение и внедрение новых информационных технологий, осуществляет контроль за организацией работы по информатизации и пр.

Отдел веб-разработки занимается:

- настройкой серверной стороны сайта, портала;
- установкой на сервере необходимого программного обеспечения;

- разработкой макета веб-сайта;
- постраничной вёрсткой веб-сайта;
- установкой CMS системы на сервер;
- настройкой CMS системы.

Отдел 1С занимается:

- Установкой, настройкой и обслуживанием сервера SQL;
- Контролем производительности, оптимизация работы баз;
- резервированием и восстановлением баз;
- поиском и исправлением ошибок в данных;
- перепроведением информационной базы.

Учёный совет принимает решение о создании или упразднения кафедры или факультета, рассматривает вопросы экономического развития, устанавливать объем и структуру приема студентов на первый курс для обучения за счет средств федерального бюджета и пр.

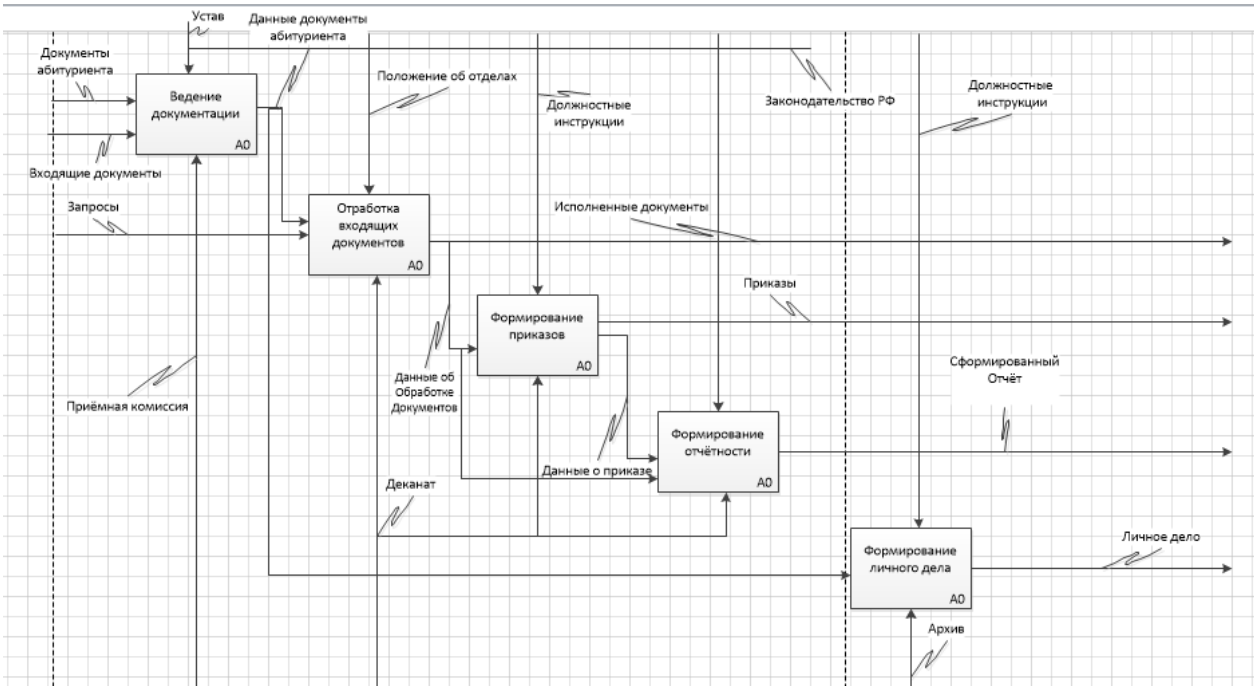


Рисунок 2 - Схема документопотока

В приёмной комиссии заполняют документы об абитуриенте. Затем отправляют документы в деканат на обработку. Деканат, пользуясь должностными инструкциями, обрабатывает эти документы и пишет приказ о

зачислении на учёбу. Затем оформляют отчётность и отправляют ректору. После того как отчёт был составлен, архив получает документы об абитуриенте и составляют его личное дело (см. рис.2).

1.4 Анализ IT-структуры

IT-инфраструктура учебного учреждения представляет собой совокупность сетевых, информационных и мультимедийных решений для обеспечения быстрого доступа, хранения, отображения и обработки информации при высоком уровне безопасности.

В центре обработки данных используются серверные мощности на базе блейд-систем HPE Synergy и HPE C7000 с суммарным объемом оперативной памяти 11 Тб и система хранения данных корпоративного класса HPE 3PAR с объемом хранения 242 Тб, из них на быстрых твердотельных дисках SSD – 35 Тб.

Материально-техническая база состоит из:

- 672 автоматизированных рабочих мест для педагогического состава;
- 1550 компьютеров с выходом в информационно-телекоммуникационную сеть «Интернет», к которым имеют доступ обучающиеся
- 320 единиц телекоммуникационного оборудования;
- 13 единиц серверного оборудования и систем хранения данных.

Учебные помещения оснащены современным презентационным, демонстрационным и мультимедийным оборудованием.

В учебном учреждении функционирует веб-портал. Портал имеет модульную структуру и в настоящий момент включает в себя 6 базовых и 14 дополнительных модулей. В базовые модули входят: Преподаватели, Сотрудники, Родители, Студенты, Абитуриент, Электронная приёмная комиссия. В дополнительные модули входят: Успеваемость, Практика, Рабочая программа, Сессия, Проверка КР, Проверка ВКР, Расписание, Балльно-рейтинговая система, Договоры на обучение, Финансы, Библиотека, Почта, Заявления и справки, Техподдержка.

Для всех участников образовательного процесса реализованы личные кабинеты.

Для всех участников образовательного процесса доступны облачные сервисы:

- платёжный шлюз;
- Head Hunter;
- E-Library;
- IPRBooks.

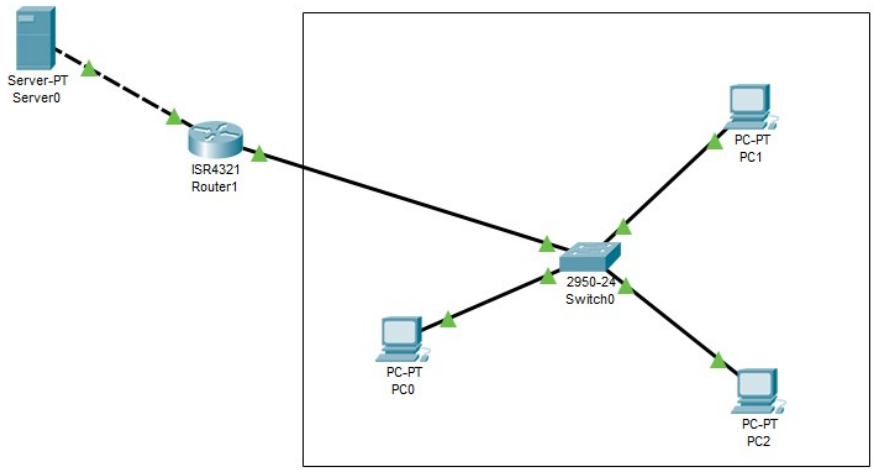


Рисунок 3 - Схема сети веб-отдела.

На рисунке 3 изображена схема сети веб-отдела, данная схема была создана в программе Cisco Packet Tracer. В кабинете находятся три компьютера и коммутатор, а маршрутизатор и сервер находятся в соседнем кабинете. Три компьютера соединяются с коммутатором internet – кабелями, таким же проводом коммутатор подсоединён к маршрутизатору. Маршрутизатор подключён к серверу кросс – кабелем.

Локальная сеть создана для повышения производительности труда и обеспечения доступа к интернету.

Все компьютеры веб-отдела имеют одинаковую конфигурацию:

- Процессор: Intel core i5-4460 3.20GHz
- Видеокарта: Intel HD Graphics 4600
- ОЗУ: 8 гб

- Hdd: 150 гб

В веб-отделе используются следующие программы:

- ОС Windows 7;
- ОС Windows 10;
- 1С: Предприятие 1.83;
- Adobe photoshop CS6;
- Google Chrome;
- Ccleaner.

1.5 Структура электронной информационно-образовательной среды (ЭИОС) образовательного учреждения

Электронная информационно-образовательная среда (ЭИОС) включает в себя:

Электронные информационные ресурсы:

- официальный сайт в сети «Интернет».
- официальные сайты структурных подразделений образовательного учреждения в сети «Интернет», находящиеся в домене;
- электронные библиотечные системы (далее – ЭБС), обеспечивающие доступ пользователям ЭИОС (в том числе авторизованный к полнотекстовым документам) к информационным ресурсам. В порядке регистрации в ЭБС читателю в библиотеке выдается логин/пароль под роспись, либо он их получает в личном кабинете обучающегося на портале.

Портал:

- обеспечивает доступ к учебным планам, рабочим программам дисциплин (модулей), практик, и к изданиям электронных библиотечных систем и электронным образовательным ресурсам, указанным в рабочих программах;
- фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения образовательных программ;
- формирование электронного портфолио обучающегося;

- взаимодействие между участниками образовательного процесса, в том числе синхронное и/или асинхронное взаимодействия посредством сети "Интернет".

На рисунке 4 изображена часть ЭИОС образовательного учреждения.

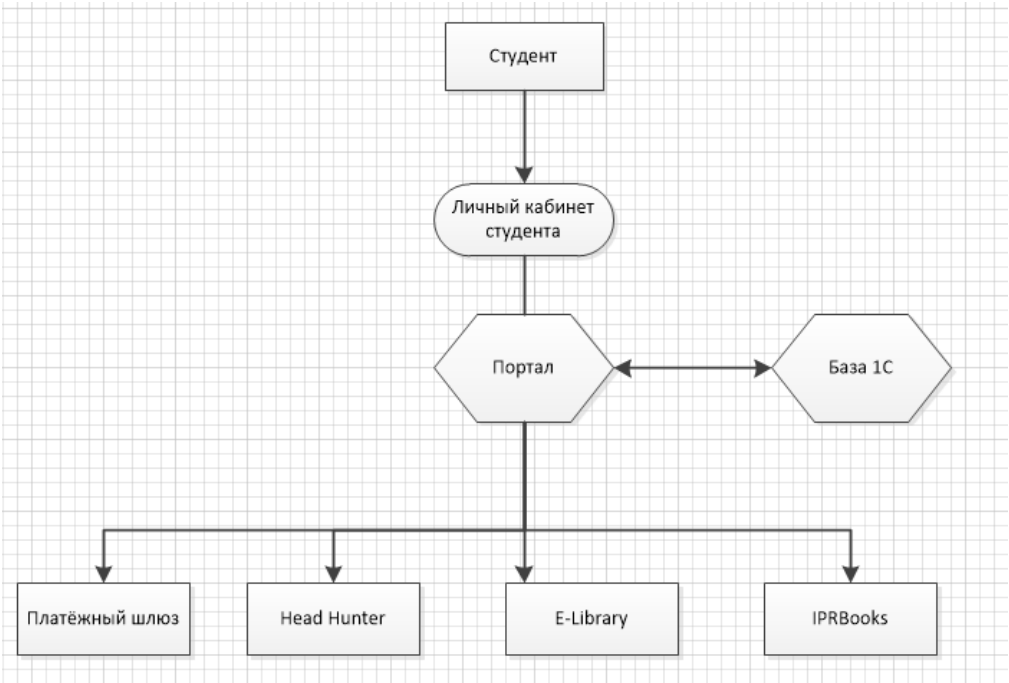


Рисунок 4 - ЭИОС образовательного учреждения

Студент заходит в свой личный кабинет, который подключен к порталу. Портал подключён к базе 1С. Также к порталу подключены сервисы: платёжный шлюз, Head Hunter, E-Library, IPRBooks.

На основании проведённого анализа можно сделать вывод, что образовательное учреждение хорошо структурировано. Также оно использует портал для контроля своей деятельности. Для повышения доступности информации о результатах учебного процесса необходим разработанный модуль. Это, в свою очередь, повысит эффективность работы портала учреждения.

1.6 Вывод по главе

Таким образом, для повышения эффективности портала и снижение нагрузки на деканат необходим модуль по успеваемости студентов.

Разработка проводилась на Viusal Studio Code.

Глава 2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

2.1 Требования к дизайну сайта

Дизайн должен быть выдержан в строгих и мягких тонах. Использовать сине-голубые оттенки. Дизайн сайта должен быть выполнен с использованием языка HTML и CSS, при необходимости для создания отдельных графических элементов допустимо использование JavaScript.

Сайт должен корректно отображаться в браузерах Яндекс версия 18.0 или позднее; Google Chrome версия 79.0 или позднее.

Необходимо создать структуру (шаблон) страницы авторизации, состоящую из одного блока. В этом блоке должен быть заголовок “Авторизация”, логотип и название учреждения, поле с логином, поле с паролем и переключатель между студентом и преподавателем. Блок должен находиться чуть выше середине страницы.

На рисунке 5 показан шаблон страницы авторизации.

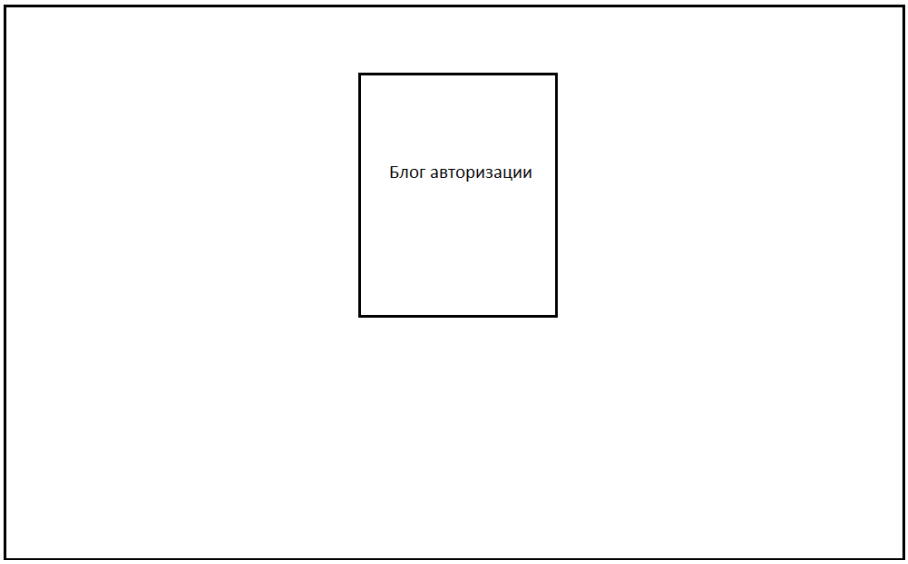


Рисунок 5 - Графическая схема шаблона страницы авторизации

Также необходимо создать структуру (шаблон) страницы авторизованного пользователя, состоящую из следующих элементов:

1. “Шапка”. В данном блоке необходимо расположить приветствие авторизованного пользователя. Под приветствием должна находиться кнопка

“Выход” для выхода с аккаунта. Данный блок должен располагаться слева на сайте;

2. Основной блок. В нём должно быть отображения списка групп (для преподавателя) и отображения успеваемости (для студентов). Требования к блоку будут изложены далее. Блок должен иметь список групп, при выборе которого отобразиться список со студентами конкретной группы и их отметки (для преподавателя). Для студента должна сразу отобразиться таблица с предметами и оценками. Для администратора блок должен иметь поля для ввода информации о новом пользователе и возможность его зарегистрировать, а также возможность делать отчёты по успеваемости. Блок должно располагаться справа от “шапки”.

На рисунке 6 представлена графическая схема шаблона сайта.

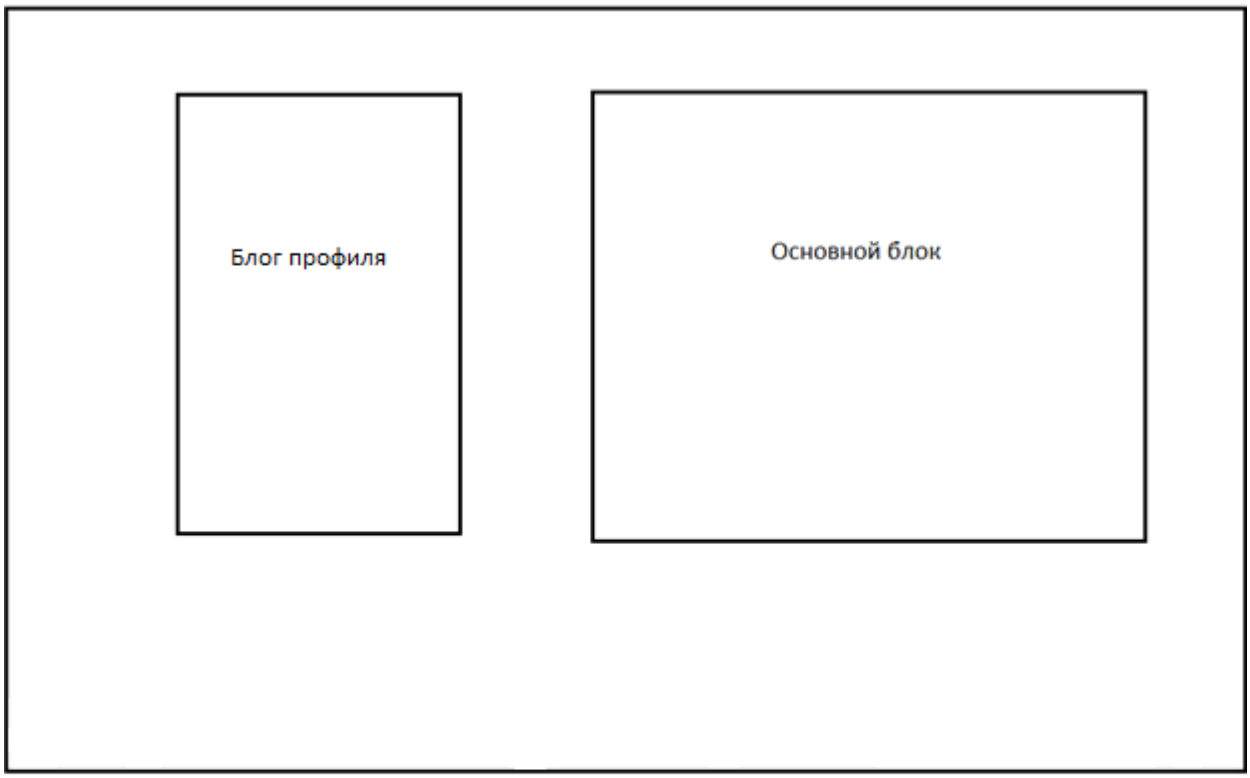


Рисунок 6 - Графическая схема шаблона основной страницы сайта

2.2 Требования к функциональности сайта

Необходимо обеспечить возможность предоставлять пользователям информацию на русском языке.

Сайт должен позволять пользователям выполнять вход на сайт как зарегистрированный пользователь для возможности просмотра успеваемости для студента и/или добавления/редактирования содержимого сайта для преподавателя.

Система управления сайтом должна позволять:

- управлять страницами сайта (добавлять, удалять, изменять их содержимое);
- создавать отчёты по данным из таблиц;
- регистрировать новых пользователей.

Таблицы в сайте должны быть соединены с базой данных MySQL.

2.2.1 Требования к содержимому сайта

Необходимо создать следующие страницы сайта:

- Страница авторизации;
- Страница студента;
- Страница преподавателя
- Страница администратора.

2.2.2 Детальное описание страниц сайта

Страница авторизации

Чуть выше середины страницы должен стоять блок с заголовком “Авторизация”. Под заголовком должен находиться логотип и название учреждения. Потом идут два поле с логином и паролем пользователя, и затем переключатель между студентом и преподавателем. Внизу блока находится кнопка “Войти”.

Страница студента

Данная страница должна состоять из двух блоков:

- блок с приветствием авторизованного пользователя. Под приветствием должна находиться кнопка “Выход”. Находится слева на странице;
- блок с таблицей успеваемости. Находится справа на странице;

Таблица с успеваемостью должна состоять из таких данных, как:

- предмет;
- дата пары;
- оценка.

Страница преподавателя

Данная страница должно состоять из двух блоков:

- блок с приветствием авторизованного пользователя. Под приветствием должна находиться кнопка “Выход”. Находится слева на странице;
- блок со списком групп студентов. При выборе группы должна сразу отобразиться таблица студентов для проставления оценок. Находится справа на странице;

Таблица должна содержать фио студента, дата пары и возможность поставить оценку.

Страница администратора

Данная страница должно состоять из трёх блоков:

- блок с приветствием авторизованного пользователя. Под приветствием должна находиться кнопка “Выход”. Находится слева на странице;
- блок, где администратор может регистрировать новых пользователей. Находится справа на странице;
- блок с отчётом об успеваемости студентов. Находится внизу страницы.

В блоке «Регистрация» администратор вводит ФИО нового пользователя, выбирает статус пользователя, после выбора появляется либо поле «Группа», либо поле «Предмет» в зависимости от выбранного статуса. В блоке «Отчёты» администратор указывает группу и предмет, затем появляется отчёт с успеваемостью студентов.

2.3 Вывод по главе

Таким образом, разработка технического задания выполнена. Техническое задание позволит разработать эффективный модуль по успеваемости.

Глава 3. ОБЗОР ПРОГРАММНЫХ, АППАРАТНЫХ И/ИЛИ ИНЖЕНЕРНО-ТЕХНИЧЕСКИХ СРЕДСТВ ДЛЯ РЕАЛИЗАЦИИ ПРАКТИЧЕСКОЙ ЧАСТИ ВКР. РЕАЛИЗАЦИЯ ТЕХНИЧЕСКОГО ЗАДАНИЯ

3.1 Обзор средств для реализации практической части выпускной квалификационной работы

Реализация практической части выпускной квалификационной работы проводилась на таких программных средствах как:

- Open Server;
- PhpMyAdmin;
- VS Code.

Реализация проводилась на ПК со следующими характеристиками:

- OS Windows 10
- RAM 12Gb
- Intel Core i5 4200H @ 2.80GHz.

3.2 Реализация технического задания

3.2.1. Создание базы данных

Для создания базы данных потребовалось веб-приложение phpMyAdmin. Доступ к веб-приложению обеспечит программа Open Server.

Для входа в phpMyAdmin необходимо авторизоваться (см.рис.7)

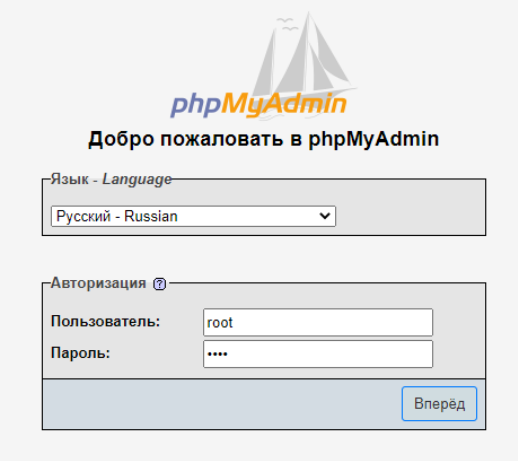


Рисунок 7 - Авторизация в phpMyAdmin

По умолчанию на локальном веб-сервере для phpMyAdmin имеется пользователь root с паролем root.

После авторизации появляется главное меню веб-приложения (см. рис.8).

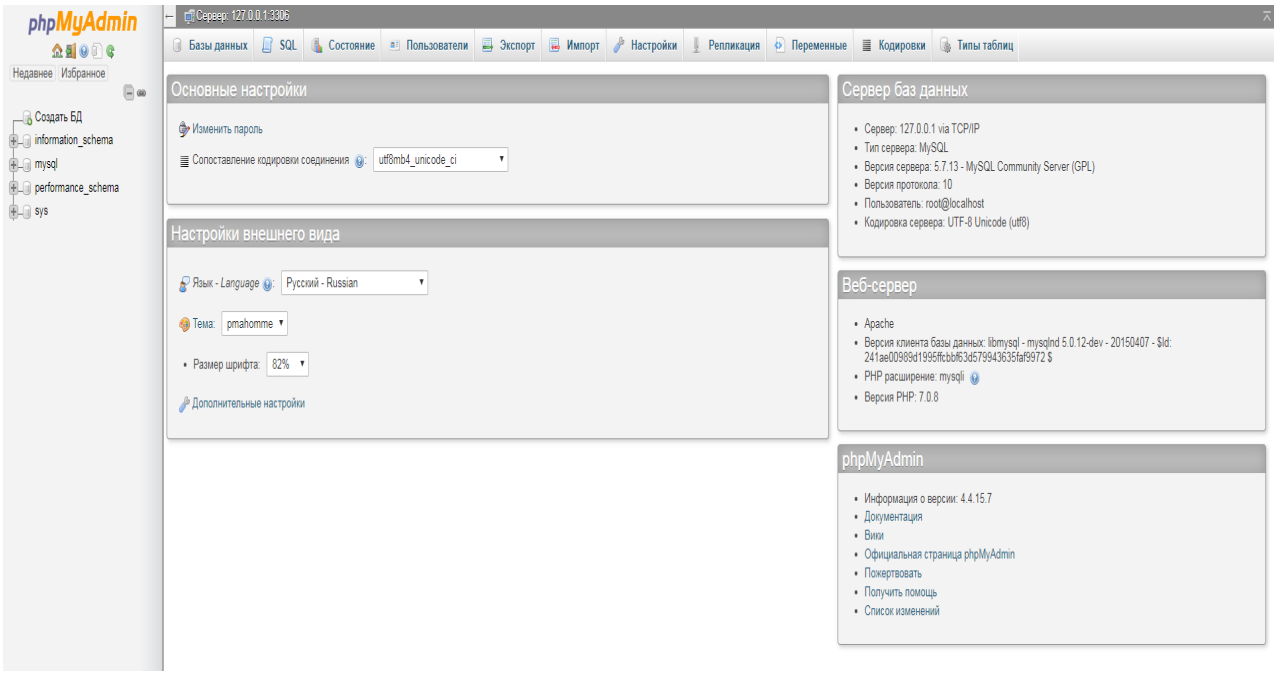


Рисунок 8 - Главное меню

В левой колонке главного меню расположены уже созданные базы данных.

В верхнем меню расположен основной функционал (также список баз данных, SQL режим, состояния веб-приложения, список пользователей, экспорт, импорт и т.д.).

Для создания базы данных необходимо в левой колонке нажать на кнопку «Создать БД». После нажатия на данную кнопку появилась форма создания базы данных (см. рис. 9).

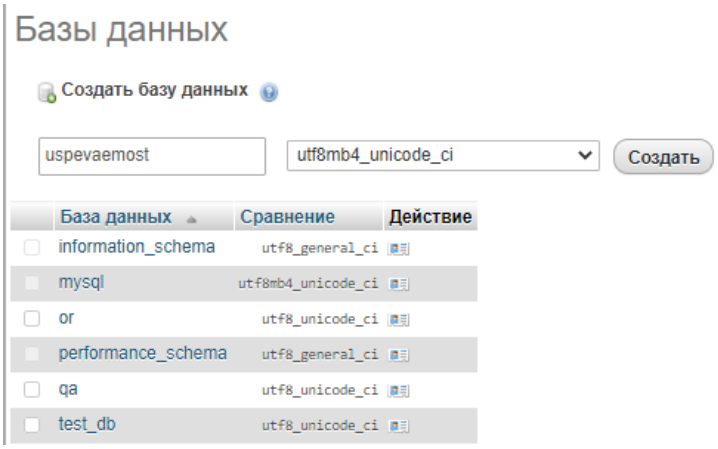


Рисунок 9 - Создание БД

После создания базы данных откроется форма с созданием таблиц (см.

рис.10)

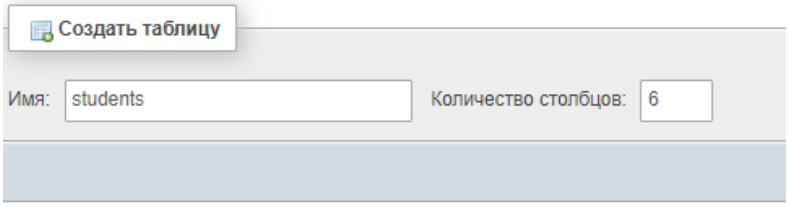


Рисунок 10 - Форма создания таблицы

Таблица «students» имеет шесть столбцов. Она хранит данные о студентов. После заполнения полей необходимо нажать кнопку «Вперёд» откроется структура таблицы «students» (см. рис.11).

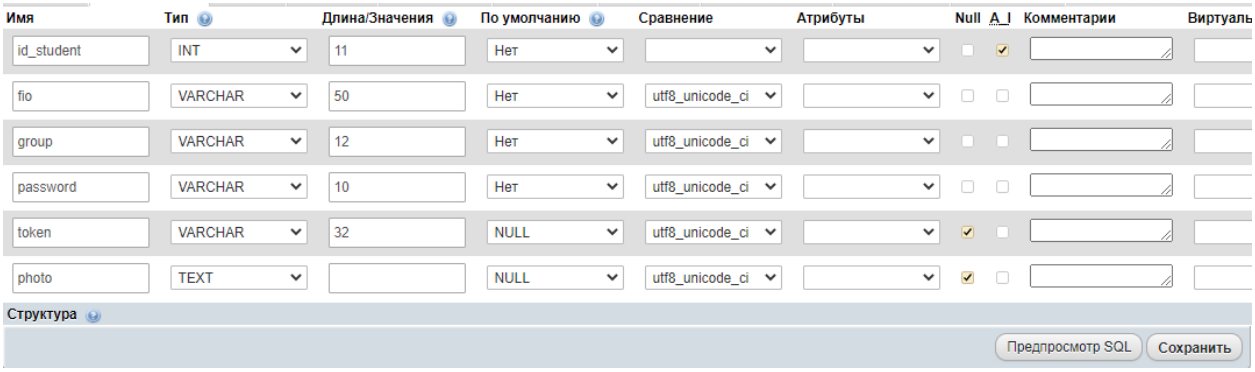


Рисунок 11 - Структура таблицы «students»

Поле «id_student» с типом INT и максимальной длиной в 11 символов служит идентифицирующим номером студента. Для того, чтобы номер студента всегда был индивидуальным, необходимо поставить галочку на пункте A_I.

Поле «fio» с типом VARCHAR и максимальной длиной в 50 символов хранит ФИО студентов.

Поле «group» с типом VARCHAR и максимальной длиной в 12 символов хранит группу студента.

Поле «password» с типом VARCHAR и максимальной длиной в 10 символов хранит пароль студента для входа в систему.

Поле «token» с типом VARCHAR и максимальной длиной в 32 символов и значением по умолчанию равное нулю хранит уникальный ключ для входа в личный кабинет. Для того, чтобы токен был уникальным, необходимо поставить галочку на пункте A_I.

Поле «photo» с типом TEXT значением по умолчанию равное нулю хранит путь к фото авторизованного пользователя. В системе будет отображаться

фото.

Следующая таблица названа «teachers», имеет шесть столбцов. Таблица хранит информацию о преподавателе (см.рис.12).

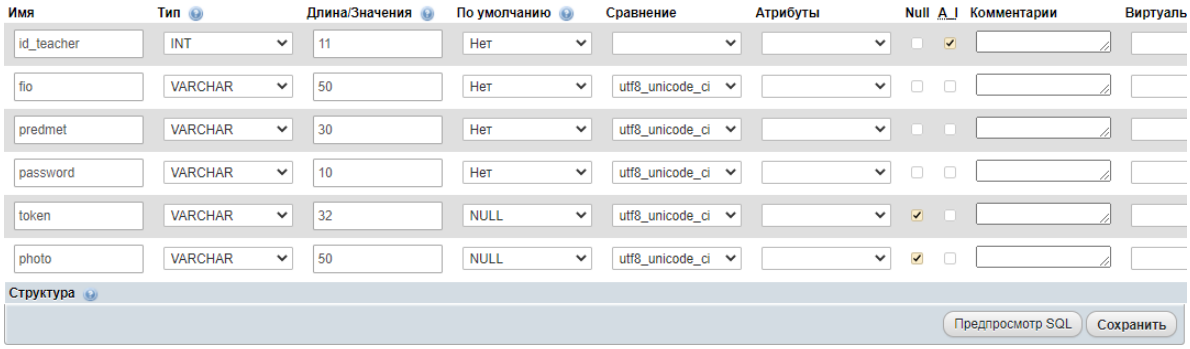


Рисунок 12 - Структура таблицы «teachers»

Поле «id_teachers» с типом INT и максимальной длиной в 11 символов служит идентифицирующим номером преподавателя. Для того, чтобы номер преподавателя всегда был индивидуальным, необходимо поставить галочку на пункте A_I.

Поле «fio» с типом VARCHAR и максимальной длиной в 50 символов хранит ФИО преподавателя.

Поле «predmet» с типом VARCHAR и максимальной длиной в 30 символов хранит предмет, который преподаёт пользователь.

Поле «password» с типом VARCHAR и максимальной длиной в 10 символов хранит пароль преподавателя для входа в систему.

Поле «token» с типом VARCHAR и максимальной длиной в 32 символов и значением по умолчанию равное нулю хранит уникальный ключ для входа в личный кабинет. Для того, чтобы токен был уникальным, необходимо поставить галочку на пункте A_I.

Поле «photo» с типом TEXT значением по умолчанию равное нулю хранит путь к фото авторизованного пользователя. В системе будет отображаться фото.

Следующая таблица названа «groups» имеет один столбец. Таблица хранит список групп (см.рис.13).

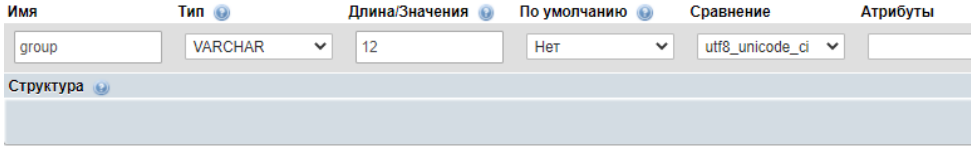


Рисунок 13 - Структура таблицы «groups»

Поле «group» с типом VARCHAR и максимальной длиной в 12 символов хранит название группы.

Следующая таблица названа «lession» имеет четыре столбца. Хранит информацию о проведённых парах (см.рис.14).

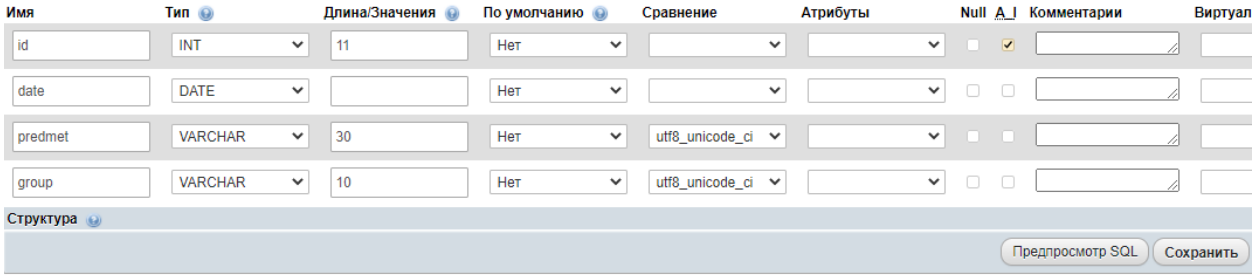


Рисунок 14 - Структура таблицы «lession»

Поле «id» с типом INT и максимальной длиной в 11 символов служит идентифицирующим номером пары. Для того, чтобы номер пары всегда был индивидуальным, необходимо поставить галочку на пункте A_I.

Поле «Date» с типом DATE хранит дату проведённой пары.

Поле «predmet» с типом VARCHAR и максимальной длиной в 30 символов хранит предмет, который преподаёт пользователь.

Поле «group» с типом VARCHAR и максимальной длиной в 12 символов хранит название группы.

Следующая таблица названа «performance» хранит информацию об успеваемости студентов (см. рис.15).

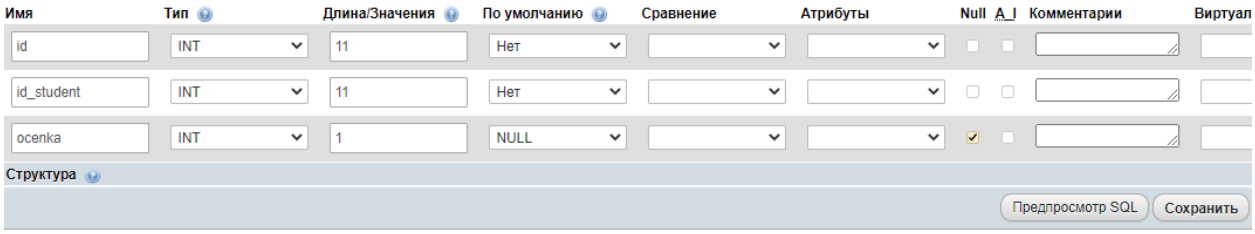


Рисунок 15 - Структура таблицы «performance»

Поле «id» с типом INT и максимальной длиной в 11 символов служит идентифицирующим номером пары.

Поле «id_student» с типом INT и максимальной длиной в 11 символов служит идентифицирующим номером студента.

Поле «ocenka» с типом INT и максимальной длиной в 1 символ содержит оценку студента за пару.

3.2.2. Написание программного кода

Практическая часть ВКР разрабатывалась в редакторе VS Code.

На рисунке 16 изображено подключение системы к базе данным.

```php
<?php
    $connection = mysqli_connect("localhost", "root", "root", "uspevaemost");
    if ($connection == false){
        echo 'Не получилось подключиться к базе данных<br>';
        echo mysqli_connect_error();
        exit();
    }
?>
```

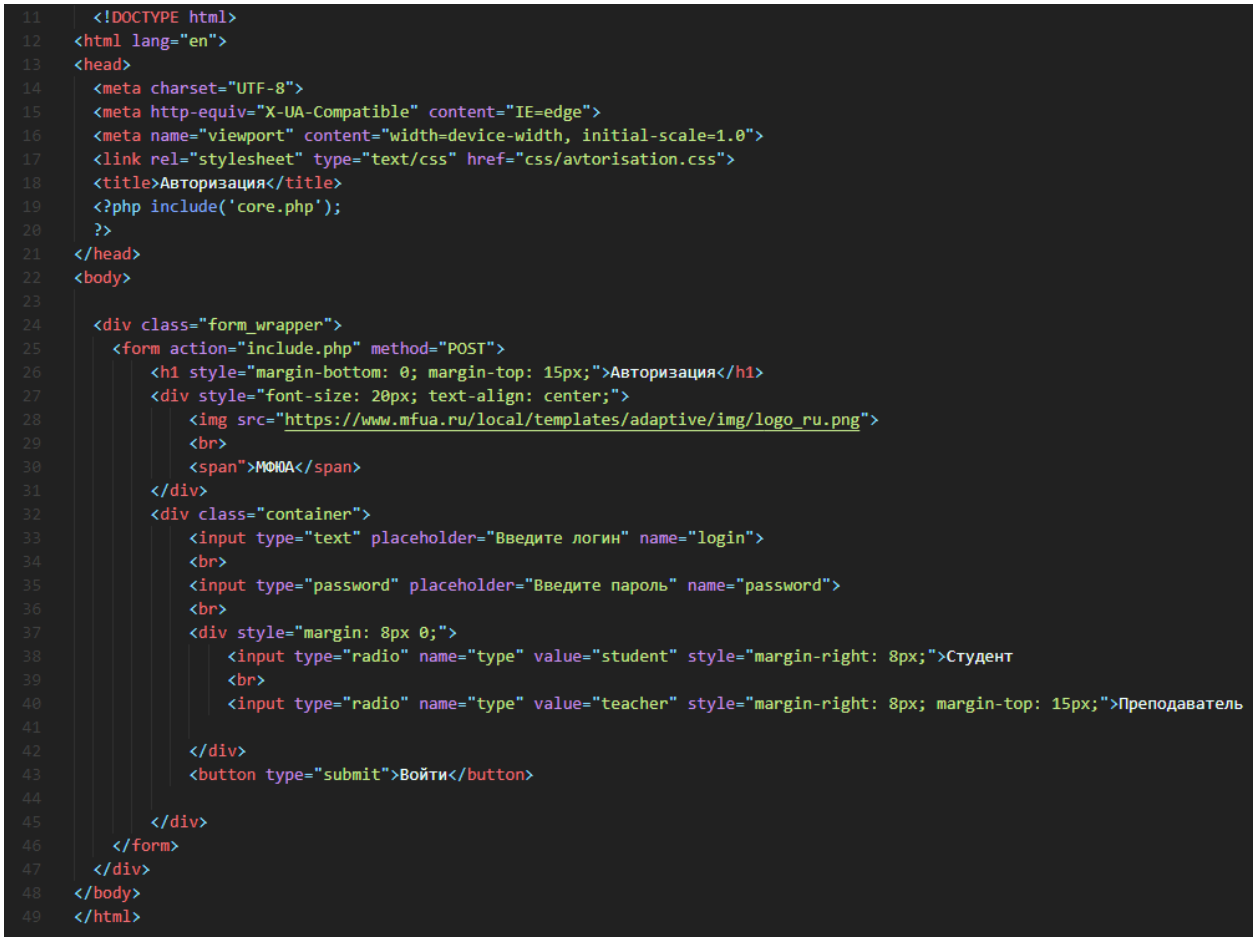
Рисунок 16 - Подключение к базе данных

На рисунке 17 показана проверка токена. В зависимости от статуса авторизированного пользователя ему показывается соответствующая страница. Для пользователя с токеном «token_student» откроется страница студента. Для пользователя с токеном «token_teacher» откроется страница преподавателя. Для пользователя с токеном «token_admin» откроется страница администратора. Код написан в файле core.php.

```php
<?
session_start();
if (isset($_SESSION['token_student'])){
    echo "<script>window.location = 'student.php';</script>";
} else if (isset($_SESSION['token_teacher'])) {
  echo "<script>window.location = 'teacher.php';</script>";
} else if (isset($_SESSION['token_admin'])) {
  echo "<script>window.location = 'admin.php';</script>";
}
?>
```

Рисунок 17 – Проверка токена.

На рисунке 18 показан код страницы авторизации. Этот код написан в файле index.php.



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" type="text/css" href="css/avtorisation.css">
  <title>Авторизация</title>
  <?php include('core.php');
  ?>
</head>
<body>

  <div class="form_wrapper">
    <form action="include.php" method="POST">
        <h1 style="margin-bottom: 0; margin-top: 15px;">Авторизация</h1>
        <div style="font-size: 20px; text-align: center;">
            <img src="https://www.mfua.ru/local/templates/adaptive/img/logo_ru.png">
            <br>
            <span">МФЮА</span>
        </div>
        <div class="container">
            <input type="text" placeholder="Введите логин" name="login">
            <br>
            <input type="password" placeholder="Введите пароль" name="password">
            <br>
            <div style="margin: 8px 0;">
                <input type="radio" name="type" value="student" style="margin-right: 8px;">Студент
                <br>
                <input type="radio" name="type" value="teacher" style="margin-right: 8px; margin-top: 15px;">Преподаватель

            </div>
            <button type="submit">Войти</button>

        </div>
    </form>
  </div>
</body>
</html>
```

Рисунок 18 – Код страницы авторизации

К файлу index.php подключается файл avtorisation.css и файл core.php. На странице расположены два поля: «login» и «password», два переключателя: «Студент» и «Преподаватель» и кнопка «Войти». При нажатии на кнопку «Войти» php-код подключается к базе данных «uspevaemost». Затем сравнивается логин и пароль с введёнными данными.

На рисунке 19 изображён код для проверки логина и пароля. Код находится в файле include.php.

```php
<?php
    include ('core.php');
    session_start();
    $login = $_POST['login'];
    $password = $_POST['password'];
    $status = $_POST['type'];
    if ($status == 'student'){
        $count = mysqli_query($connection, "SELECT * FROM `students`
        WHERE `id_student` = '$login' AND `password` = '$password'");

        if (mysqli_num_rows($count) == 0){
            echo '<script>alert("Неправильно введён логин или пароль");
            window.location = "index.php";</script>';
        } else {
            echo "<script>window.location = 'student.php';</script>";
            $_SESSION['token_student'] = md5(round(microtime(true) * 1000));
            mysqli_query($connection, "UPDATE students
            SET token = '".$_SESSION['token_student']."' WHERE id_student = '$login'");
        }
    } else if ($status == 'teacher'){
        $count = mysqli_query($connection, "SELECT * FROM `teachers`
        WHERE `id_teacher` = '$login' AND `password` = '$password'");

        if (mysqli_num_rows($count) == 0){
            echo '<script>alert("Неправильно введён логин или пароль");
            window.location = "index.php";</script>';
        } else {
            echo "<script>window.location = 'teacher.php';</script>";
            $_SESSION['token_teacher'] = md5(round(microtime(true) * 1000));
            mysqli_query($connection, "UPDATE teachers
            SET token = '".$_SESSION['token_teacher']."' WHERE id_teacher = '$login'");
        }
    } else if ($login == 'admin' and $password == 'admin132'){
        echo  '<script> window.location = "admin.php"; </script>';
        $_SESSION['token_admin'] = md5(round(microtime(true) * 1000));
    } else if(!isset($status) || !isset($login) || !isset($password)){
        echo  '<script> window.location = "index.php"; alert("Вы что-то забыли ввести)")</script>';
    }
?>
```

Рисунок 19 – Код проверки логина и пароля

Файл include.php подключается к файлу core.php для сравнения введённых данных с данными из базы. Если в странице авторизации был выбран пункт «Студент», то проверяется таблица «students». Если был выбран пункт «Преподаватель», то проверяется таблица «teachers». Если был введён логин и пароль администратора, то введённые данные сравниваются с данными из условия. Если логин и пароль были введены верно, то откроется соответствующая страница. В противном случае, выскочит сообщение об ошибке.

На рисунке 20 изображён фрагмент кода страницы студента, который выводит в title ФИО авторизованного пользователя, а также подключает файл main.css. ФИО берётся с базы данных

```
<title>
  <?php
    $row = mysqli_query($connection, "SELECT fio FROM `students`
    WHERE `token` = '".$_SESSION['token_student']."'");
    $fio = $row->fetch_assoc();
    echo 'Студент - ' .  $fio['fio'];
  ?>
</title>
<link href="css/main.css" rel="stylesheet" type="text/css">
```

Рисунок 20 – Фрагмент кода страницы студента

На рисунке 21 изображён фрагмент кода, отвечающий за блок профиля пользователя.

```
<div class="content">
  <div class="profile">
    <div class="header">Профиль</div>
      <div class="profile_content">
        <div class="profile_photo">
        <?php
            $row = mysqli_query($connection, "SELECT photo FROM `students`
            WHERE `token` = '".$_SESSION['token_student']."' AND photo IS NOT NULL");
            if (mysqli_num_rows($row) > 0) {
              $fio = mysqli_fetch_assoc($row);
              echo '<img src="/image/'.$fio['photo'].'">';
            } else {
              echo '<img src="/image/nophoto.png">';
            }
          ?>
        </div>
        <div class="profile_name">
          <?php
            $row = mysqli_query($connection, "SELECT fio FROM `students`
            WHERE `token` = '".$_SESSION['token_student']."'");
            $fio = $row->fetch_assoc();
            echo 'Привет, ' .  $fio['fio'] . '!';
          ?>
        </div>
        <form action="student.php" method="post">
          <button type="submit" name="exit" style="width: 100px;
          background-color: red; padding: 10px; margin-top: 15px;">Выход</button>
        </form>
      </div>
  </div>
```

Рисунок 21 – Код блока профиля

Блок профиля состоит из фотографии, приветствия и кнопки «Выход». В приветствии имеется ФИО авторизованного пользователя. Для получение ФИО с базы данных был применён SQL – запрос.

На рисунке 22 изображён код первого фрагмента блока «Рабочая область» в странице студента.

```
<div class="work_area">
  <div class="header">Рабочая область</div>
    <div class="work_area_content">
      <div style="overflow: auto">
        <table class="table">
          <thead>
            <tr>
              <th class="table-th-sticky">Предмет\Число</th>
                <?php
                  $sql = mysqli_query($connection, "SELECT performance.`id_student`,
                  lession.`predmet`, lession.`date`, performance.`ocenka`
                  FROM students INNER JOIN (lession INNER JOIN performance
                  ON lession.id=performance.id) ON (students.id_student=performance.id_student)
                  WHERE students.`token` = '".$_SESSION['token_student']."' GROUP BY `date`");
                  $i = 0;
                  $dates = array();
                  while ($result = mysqli_fetch_array($sql)) {
                    $date = date('d.m.Y', strtotime($result['date']));
                    echo "<th>{$date}</th> ";
                    $dates[$i] = $result['date'];
                    $i++;
                  }
                ?><th class="table-th-sticky" style="right:0;">Средний балл</th>
            </tr>

          </thead>
```

Рисунок 22 – Код первого фрагмента блока «Рабочая область»

Рабочая область на странице студента состоит из одной таблицы, в которой отображается его успеваемость. Этот фрагмент содержит столбцы таблицы. В столбцах содержатся даты проведённых пар, в которых студент получил оценку. Даты выводятся из базы данных в формате дд-мм-гг. Также имеется столбец «Средний балл», для удобства он сделан фиксированным. Для вывода дат был использован SQL-запрос

На рисунке 23 изображён код второго фрагмента блока «Рабочая область» в странице студента.

В этом фрагменте содержатся предметы и оценки к ним, а средний балл по каждому предмету. Для вывода предметов и оценок был использован SQL-запрос. А для подсчёта среднего балла был использован массив $marks с переменными ‘average’ и ‘count’.

```
<tbody>
  <?php
    $sql = mysqli_query($connection, "SELECT lession.`predmet`, lession.`date`, performance.`ocenka`
    FROM students INNER JOIN (lession INNER JOIN performance
    ON lession.id=performance.id) ON (students.id_student=performance.id_student)
    WHERE students.`token` = '".$_SESSION['token_student']."'");
    $pr = array();
    while($result = $sql ->fetch_assoc()) {
      $pr[$result['predmet']][] = $result;
    }
  ?>
  <?php foreach($pr as $predmet => $values): ?>
  <tr>
    <td class="table_title table-th-sticky"><?php echo $predmet; ?></td>
      <?php
        $marks = array('average' => 0, 'count' => 0);
        for ($i = 0; $i < count($dates); $i++) {
          echo "<td>";
          foreach($values as $ocenka) {
            if ($dates[$i] == $ocenka['date']) {
              $marks['average'] += $ocenka['ocenka'];
              if ($ocenka['ocenka'] !== null){
                $marks['count']++;
              }

              echo $ocenka['ocenka'];
            }
          }
          echo "</td>";

        }
        echo '<td class="table-th-sticky table_title" style="right:0;">';
        echo $marks['average'] / $marks['count'];

        echo '</td>';
      ?>
  </tr>
  <?php endforeach; ?>
</tbody>
</table>
```

Рисунок 23 – Код второго фрагмента блока «Рабочая область»

Блок профиля во всех страницах, кроме авторизации, одинаковый. Блок «Рабочая область» на странице преподавателя состоит из селекта с группами

На рисунке 24 изображён селект с дизайном, который расположен на странице преподавателя.

```
<div class="select" style="margin-left: 20px">
  <div class="select-input">
    <span>Выберите группу...</span>
      <i class="fa fa-caret-down" style="font-size: 26px; color: #379fc1;"></i>
  </div>
  <ul class="select-list">
    <?php
      $sql = mysqli_query($connection, "SELECT `group` FROM `groups`");
      while ($groups = mysqli_fetch_array($sql)) {
        echo "<li class=\"select-value\">{$groups['group']}</li>";
      }
    ?>
  </ul>
</div>
```

Рисунок 24 – Селект с дизайном

На рисунке 25 изображён фрагмент кода, отвечающий за появление кнопок

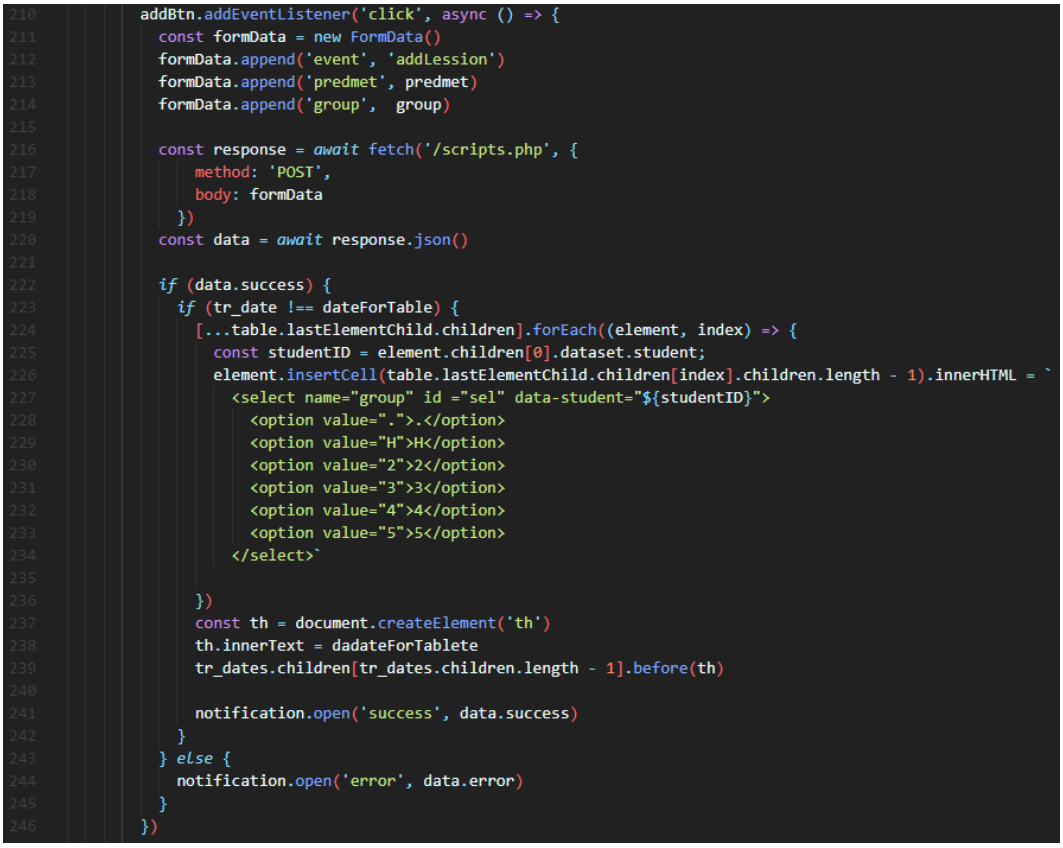
«Добавить пару» и «Сохранить» после выбора группы в селекте.

```
<input id="group_name" type="text" value="09ЦЕ8423" name="group" hidden>
<button type="submit" value="Выбрать" style="display: inline-flex; j
ustify-content: center; align-items: center; text-align: center;
margin: 15px 0 20px 20px; width: 90px; height: 30px;
background-color: rgb(100, 160, 216)">Выбрать</button>
<?php
  if (isset($_POST['group'])) {
?>
<div class="buttons">
  <input name="addLession" id="add" type="button" value="Добавить пару"
  style="height: 30px; width: 120px; background-color: rgb(100, 160, 216);
  margin: 20px 0 0 10px; padding: 0px"></input>
  <input name="save" id="save" type="button" value ="Сохранить"
  style="height: 30px; width: 120px; background-color: #76e27d;
  margin: 20px 0 0 10px; padding: 0px"></input>
</div>
```

Рисунок 25 – Появление кнопок

Условие isset($_POST[‘group’]) проверяет выбрана ли группа или нет. Если да, то появляются кнопки и таблица.

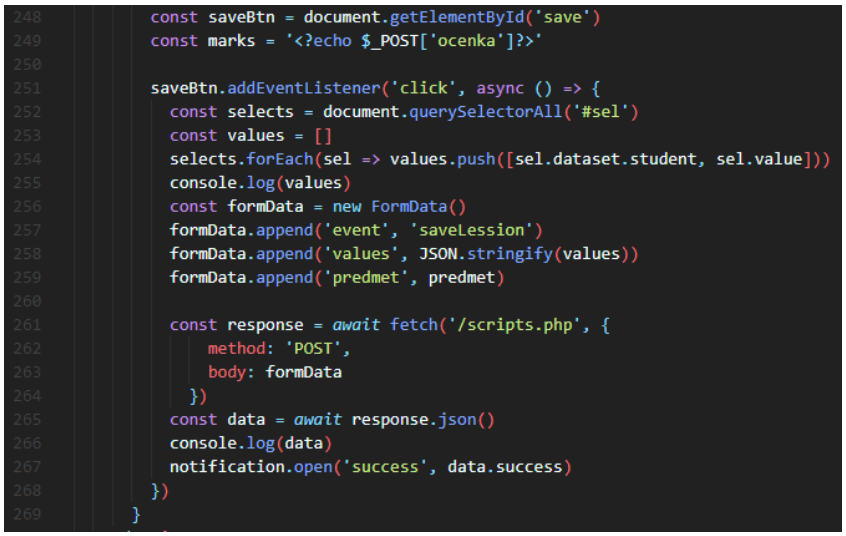
На рисунке 26 изображён JS код, который создаёт прослушивание события нажатия на кнопку «Добавить пару». Этот код отправляет запрос с данными на файл scripts.php, на котором реализовано проверка на существование пары за сегодняшний день. Если пара существует, то на клиент отправляется ответ об ошибке. Если пара не существует, то на клиент отправляется ответ об успешном создании пары, после чего JS выводит сообщение о создании пары и динамически создаёт колонку, позволяющую редактировать успеваемость студентов за сегодняшний день.



```
addBtn.addEventListener('click', async () => {
  const formData = new FormData()
  formData.append('event', 'addLession')
  formData.append('predmet', predmet)
  formData.append('group',  group)

  const response = await fetch('/scripts.php', {
      method: 'POST',
      body: formData
    })
  const data = await response.json()

  if (data.success) {
    if (tr_date !== dateForTable) {
      [...table.lastElementChild.children].forEach((element, index) => {
        const studentID = element.children[0].dataset.student;
        element.insertCell(table.lastElementChild.children[index].children.length - 1).innerHTML = `
          <select name="group" id ="sel" data-student="${studentID}">
            <option value=".">.</option>
            <option value="H">H</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
            <option value="5">5</option>
          </select>`

      })
      const th = document.createElement('th')
      th.innerText = dadateForTablete
      tr_dates.children[tr_dates.children.length - 1].before(th)

      notification.open('success', data.success)
    }
  } else {
    notification.open('error', data.error)
  }
})
```

Рисунок 26 – Прослушивание события нажатия на кнопку «Добавить пару»

На рисунке 27 изображён JS код, который создаёт прослушивание события нажатия на кнопку «Сохранить». Этот код отправляет запрос на файл scripts.php, на котором реализовано внесение введённых пользователем данных таблицы в базу данных. При успешном сохранении JS отправляет сообщение о сохранении оценок.



```
const saveBtn = document.getElementById('save')
const marks = '<?echo $_POST['ocenka']?>'

saveBtn.addEventListener('click', async () => {
  const selects = document.querySelectorAll('#sel')
  const values = []
  selects.forEach(sel => values.push([sel.dataset.student, sel.value]))
  console.log(values)
  const formData = new FormData()
  formData.append('event', 'saveLession')
  formData.append('values', JSON.stringify(values))
  formData.append('predmet', predmet)

  const response = await fetch('/scripts.php', {
      method: 'POST',
      body: formData
    })
  const data = await response.json()
  console.log(data)
  notification.open('success', data.success)
})
}
```

Рисунок 27 – Прослушивание события нажатия на кнопку «Сохранить»

На рисунке 28 указан фрагмент файла scripts.php, отвечающий за

обработку запроса таких кнопок, как «Добавить пару» и «Сохранить». Код проверяет, какое событие произошло, а затем выполняет SQL – запрос. Если произошло событие «addLession», то в базу данных добавляется строка с парой, а затем добавляет строки успеваемости без оценок. Если произошло событие «saveLession», то введённые в таблице оценки добавляются в недавно созданную строку.

```
if ($event == 'addLession') {
  $control = mysqli_query($connection, "SELECT * FROM lession WHERE `date` = CURDATE()
  AND lession.predmet = (SELECT predmet FROM teachers WHERE token = '".$data["predmet"]."') AND lession.group = '".$data["group"]."'");
  if (mysqli_num_rows($control) == 0) {
    $update = mysqli_query($connection, "INSERT INTO lession
    VALUES (null, CURDATE(),  (SELECT predmet FROM teachers WHERE token ='" . $data["predmet"] . "'), '" . $data["group"] . "')");
    $update2 = mysqli_query($connection, "INSERT INTO performance(id, id_student)
    SELECT DISTINCT lession.id, students.id_student FROM lession INNER JOIN students ON lession.group = students.group
    WHERE lession.predmet = (SELECT predmet FROM teachers WHERE token = '".$data["predmet"]."')
    AND students.group = '".$data["group"]."' AND lession.date = CURDATE()");
    $data = array('success' => 'Пара добавлена');
    echo json_encode($data);
  } else {
    $data = array('error' => 'За сегодня пара уже была добавлена');
    echo json_encode($data);
  }
if ($event == 'saveLession') {
  $values = json_decode($data['values']);
  for ($i = 0; $i < count($values); $i++) {
    $save = mysqli_query($connection, "UPDATE performance LEFT JOIN students ON performance.id_student = students.id_student
    LEFT JOIN lession ON performance.id=lession.id SET ocenka = '".$values[$i][1]."'
    WHERE lession.predmet = (SELECT predmet FROM teachers WHERE token = '".$data["predmet"]."')
    AND students.id_student = '".$values[$i][0]."' AND lession.date = CURDATE()");
  }
  $data = array('success' => "Оценки сохранены");
  echo json_encode($data);
}
```

Рисунок 28 – Обработка запроса кнопок «Добавить пару» и «Сохранить»

Страница администратора состоит из трёх блоков: «Блок профиля», «Регистрация» и «Отчёты».

На рисунке 29 изображён код, отвечающий за блок «Регистрация». Первоначально этот блок состоит из статуса пользователя, ФИО, пароля и кнопки «Зарегистрировать», после выбора статуса появляется ещё поле «Предмет» или выпадающий список «Группа» в зависимости от выбранного статуса. Значения выпадающего списка взяты из базы данных.

```
<div style="display: flex; align-items: center;">
  <label for="status" id="status">Статус: </label>
    <input type="radio" name="status" value="student"
    <? if($_POST['status'] == 'student') {echo "checked";} ?>>Студент
    <input type="radio" name="status" value="teacher"
    <? if($_POST['status'] == 'teacher') {echo "checked";} ?>>Преподаватель
</div>
<button type="submit" id="type" value="Выбрать">Выбрать</button><br>
<label id="fio" for="fio">ФИО: </label>
  <input type="text" name="fio" placeholder="Введите ФИО" id="fio1" value=""><br>
<label id="pas" for="password">Пароль: </label>
  <input type="password" name="password" placeholder="Введите пароль" id="password"><br>
<?
    if($_POST['status'] == 'student') {
?>
<br><label id="group" for="group">Группа: </label>
  <select id="group1" style="margin-left: 20px; width: 173px; height: 25px">
  <option selected value="">Выберите группу</option>
  <?php
      $sql = mysqli_query($connection, "SELECT `group` FROM `groups`");
      while ($groups = mysqli_fetch_array($sql)) {
        echo "<option value=\"{$groups['group']}\">{$groups['group']}</option>";
      }
    ?>
  </select>
<?
    } else if ($_POST['status'] == 'teacher') {
      echo '<label for="predmet" style="margin-left: -2px;">Предмет: </label>
      <input type="text" name="predmet" placeholder="Введите предмет" id="predmet"><br>';
    }
?>
<button type="button" name="registration" id="registration" value="Зарегистрировать">Зарегистрировать</button>
```

Рисунок 29 – Блок регистрации

На рисунке 30 изображён JS код для прослушивания события нажатия на кнопку «Зарегистрировать». Этот код отправляет запрос на файл scripts.php, в котором реализовано регистрация пользователя в базу данных. Если регистрация получается успешно, то на клиент отправляется сообщение об успешной регистрации. Если регистрация не получилась, то на клиент отправляется сообщение об ошибке.

```
const addUser = document.getElementById('registration')
addUser.addEventListener('click', async () => {
  const st = document.querySelectorAll('[name="status"]')
  const fio = document.getElementById('fio1').value
  const password = document.getElementById('password').value
  let status
  st.forEach(element => {
    if (element.checked) {
      status = element.value
    }
  })
  const formData = new FormData()
  formData.append('event', 'addUser')
  formData.append('fio', fio)
  formData.append('password', password)
  formData.append('status', status)
  if (status === "student"){

    const n = document.getElementById('group1').options.selectedIndex
    const group = document.getElementById('group1').options[n].value
  formData.append('group', group)
  } else if (status === "teacher") {

    const predmet = document.getElementById('predmet').value
    formData.append('predmet', predmet)
  }

  const response = await fetch('/scripts.php', {
    method: 'POST',
    body: formData
  })
  const data = await response.json()
  console.log(data)
  if (data.success) {
    notification.open('success', data.success)
  } else if (data.error) {
    notification.open('error', data.error)
  }
})
```

Рисунок 30 – Прослушивание на событие нажатия на кнопку «Зарегистрировать»

На рисунке 31 изображён фрагмент файла scripts.php, который отвечает на обработку запроса кнопки «Зарегистрировать». Код использует условие, где проверяется статус нового пользователя (студент или преподаватель), потом используется SQL – запрос. Если значение статуса равен «student» то пользователь добавляется в таблицу «students». Если значение статуса равен «teacher» то пользователь добавляется в таблицу «teachers».

```
    if ($event == 'addUser') {
      if ($data['status'] == 'student') {
        $addUser = mysqli_query($connection, "INSERT INTO students VALUES
        (null, '".$data["fio"]."', '" . $data["group"] . "', '" . $data["password"] . "', null, null)");
      } else if ($data['status'] == 'teacher') {
        $addUser = mysqli_query($connection, "INSERT INTO teachers VALUES
        (null, '".$data["fio"]."', '" . $data["predmet"] . "', '" . $data["password"] . "', null, null)");
      }
      if ($data['status'] != 'undefined') {
        $data = array('success' => "Пользователь добавлен");
      } else {
        $data = array('error' => "Укажите статус");
      }
      echo json_encode($data);
    }
  }
```

Рисунок 31 – Обработка запроса кнопки «Зарегистрировать»

Блок «Отчёты» состоит из двух выпадающих списков: «Выберите группу» и «Выберите предмет», а также из кнопки «Сформировать отчёт». После нажатия на кнопку «Сформировать отчёт» появляется таблица с успеваемостью указанной группы по указанному предмету.

На рисунке 32 изображён фрагмент кода, отвечающий за два выпадающих списка. Для получения значений был использован SQL - запрос.

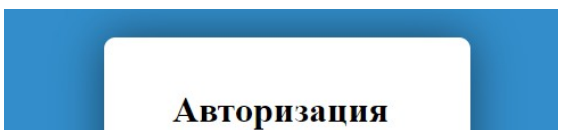
```
<select style="margin-left: 20px;" name="selgroup">
  <option>Выберите группу</option>
  <?php
    $sql = mysqli_query($connection, "SELECT `group` FROM `groups`");
    while ($groups = mysqli_fetch_array($sql)) {
      echo "<option value=\"{$groups['group']}\">{$groups['group']}</option>";
    }
  ?>
</select>
<select style="margin-left: 20px;" name="selpredmet">
  <option>Выберите предмет</option>
  <?php
    $sql = mysqli_query($connection, "SELECT `predmet` FROM `lession` GROUP BY predmet");
    while ($predmets = mysqli_fetch_array($sql)) {
      echo "<option value=\"{$predmets['predmet']}\">{$predmets['predmet']}</option>";
    }
  ?>
</select>
<button type="submit" value="Сформировать отчёт" id="repots">Сформировать отчёт</button>
<div style="overflow: auto">
```

Рисунок 32 – Выпадающие списки

Структура кода отображения таблицы с успеваемостью на странице администратора похожа на структуру кода на странице преподавателя.

3.2.3 Выполнение отладки тестирования

Для входа в систему пользователю необходимо ввести логин и пароль на странице авторизации (см. рис. 33)



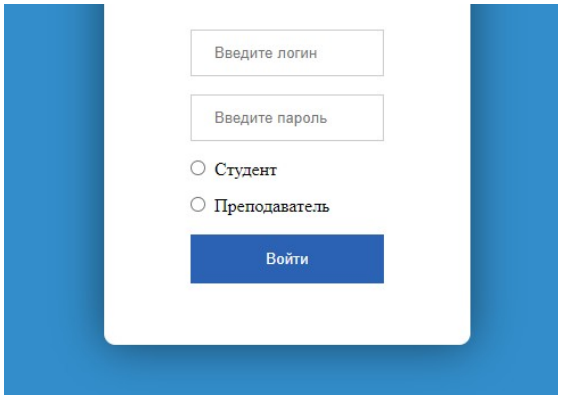


Рисунок 33 - Авторизация

При вводе неправильных данных система выведет ошибку (см. рис. 34)

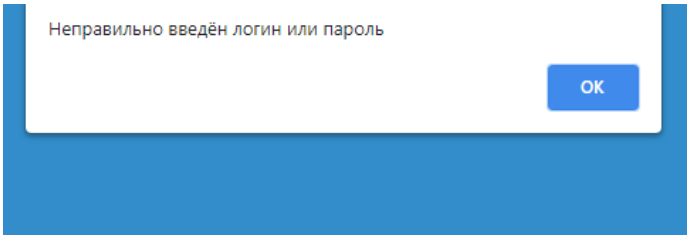


Рисунок 34 - Сообщение об ошибке

При успешной авторизации пользователь заходит в систему.

Студент в своей странице просматривает свои оценки по предметам. После просмотра он выходит из системы, нажав на кнопку выход (см. рис. 35).

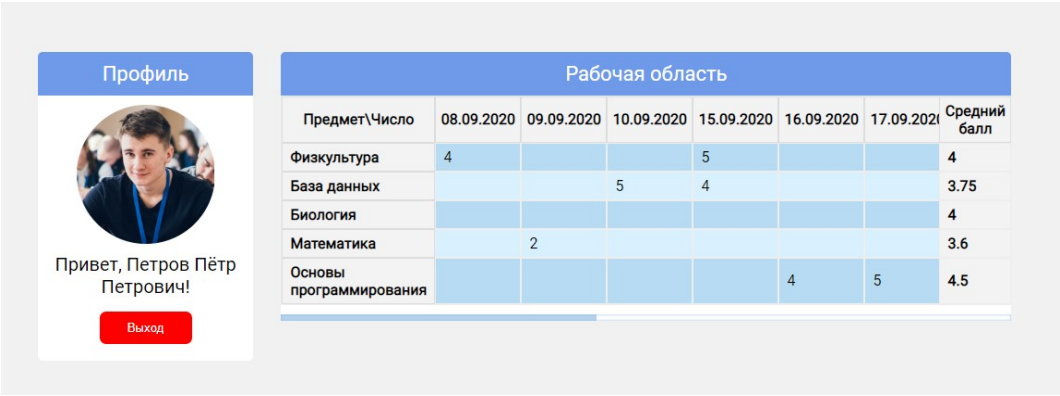


Рисунок 35 - Страница студента

Преподаватель после выбора группы проставляет оценки студентам и затем сохраняет изменения. Эти изменения появляются в базе данных (см. рис. 36).

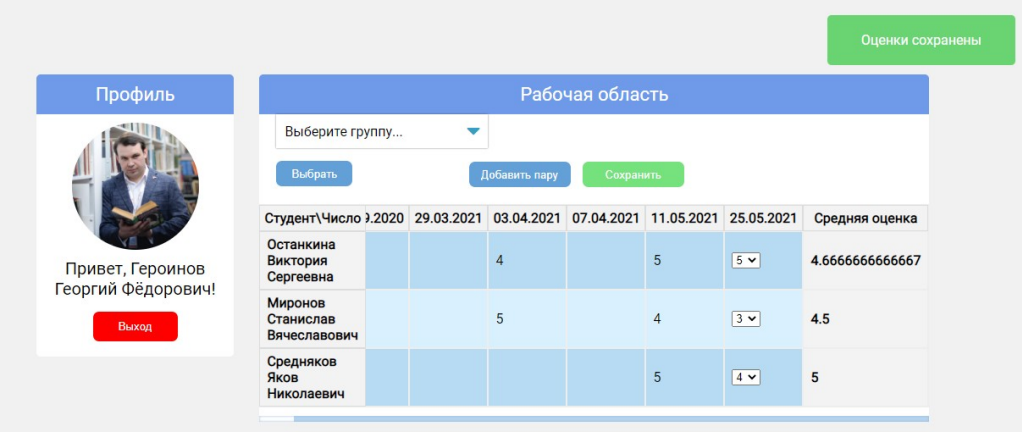


Рисунок 36 - Страница преподавателя

Администратор добавляет нового пользователя в базу данных. После регистрации выводится сообщение об успешном добавлении пользователя (см. рис 37).

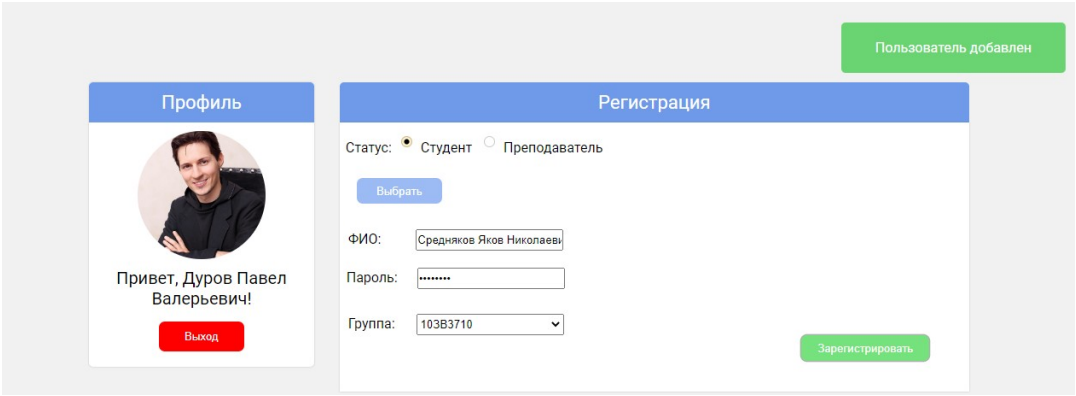


Рисунок 37 - Страница администратора

Таким образом, проведение тестирования завершено.

3.2.4 Оформление документации на программного обеспечения

Описание программного обеспечения

Функциональная часть системы разрабатывалась на основании функциональных требований, изложенных в техническом задании.

Система разрабатывалась как модуль портала образовательного учреждения с единой базой. Система базируется на технологиях Visual Studio Code, Open Server и PhpMyAdmin.

Программа и методика испытания ПО

Объектом испытаний является система успеваемости студентов.

Целью проведения испытаний системы является:

- проверка работоспособности системы;

- проверка соответствия требованиям, приведённых в техническом задании.

Испытание проводилось на личном оборудовании с использованием Visual Studio Code.

Испытание проводилось менее 1 часа, и система прошла испытание на работоспособность и соответствие с техническим заданием.

Руководство пользователю

Для входа в систему пользователю необходимо ввести логин, пароль и указать тип пользователя (см. рис.38).

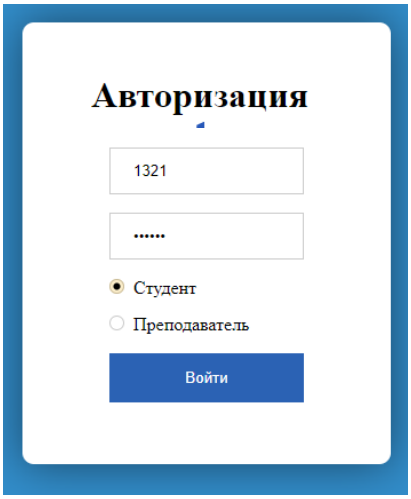


Рисунок 38 - Авторизация

На странице студента можно просматривать успеваемость по предметам авторизованного пользователя. Данные выводятся из базы данных (см. рис 39).

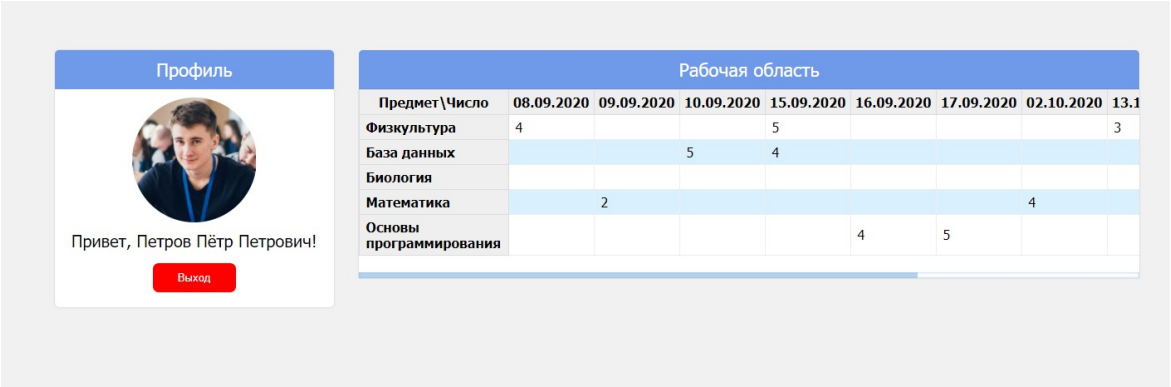


Рисунок 39 - Страница студента

Преподаватель может добавлять пары и проставлять оценки студентам только за текущий день. Все изменения вносятся в базу данных (см. рис. 40).

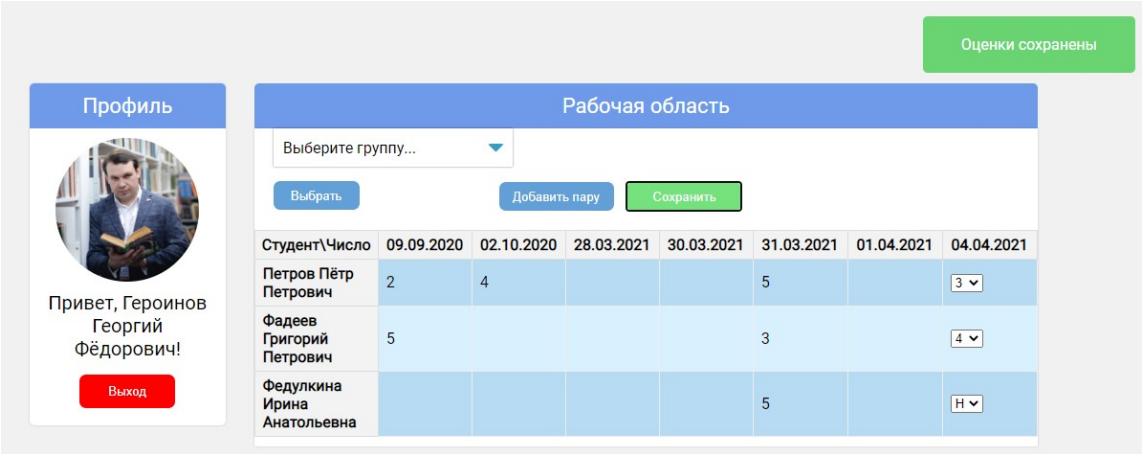


Рисунок 40 - Проставление оценок

Администратор при необходимости регистрирует нового пользователя. Новый пользователь добавляется в базу данных. После этого высветится сообщение о добавлении пользователя (см. рис 41).

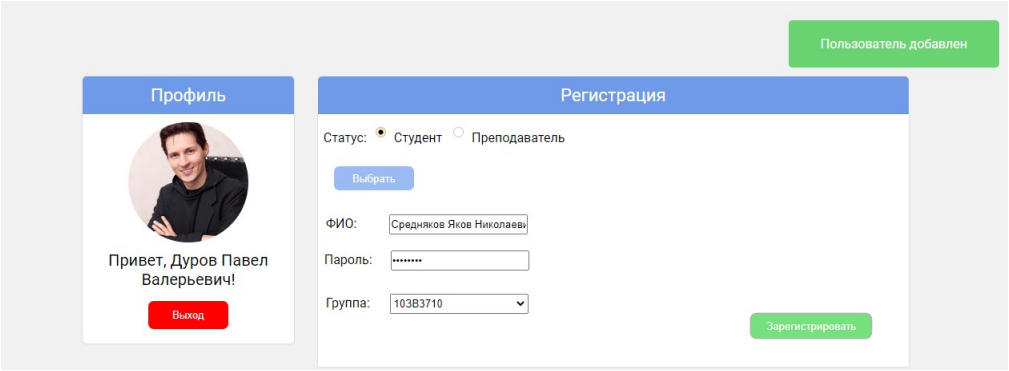


Рисунок 41 - Добавление пользователя

Таким образом, оформление документации на программное обеспечение завершилось.

3.3 Экономическая часть

Для оценки затрат необходимо вычислить капитальные затраты.

Капитальные затраты (К) на создание проекта складываются из четырёх составляющих:

$$К = Зр + Зв + Зэ + Зт$$

где:

Зр - затраты на разработку сайта;

Зв - затраты на внедрение сайта;

Зэ - затраты на эксплуатацию сайта;

Зт - затраты на создание вычислительных центров, приобретение необходимого оборудования и вычислительной техники.

Затраты на разработку сайта (Зр) вычисляется по следующей формуле:

$$Зр = Звр * Зз.п.прог$$

где:

Звр = временные затраты;

Зз.п.прог = стоимость одного часа работы программиста.

К временным затратам относятся:

- анализ предметной области (7 часов);
- разработка алгоритма решения задачи (10 часов);
- создание дизайна (8 часов);
- программирование (40 часов);
- тестирование (3 часа).

Таким образом, Звр = 68 часов.

Зз.п.прог = 90 000 / 120 = 750 руб./час.

Следовательно, Зр= 68 * 750 = 51 000 руб.

Затраты на внедрение сайта (Зв) состоит из регистрации доменного имени.

Значит, Зв = 200 руб.

Затраты на эксплуатацию сайта (Зэ) состоит из:

- стоимости поддержания модуля портала в рабочем состоянии (14 400 руб./год);
- продление доменного имени на 1 год (200 руб.);
- аренда хостинга на 1 год (1400 руб.).

Таким образом, Зэ = 16 000 руб./год.

Зт = 0, т.к. необходимое оборудование уже имеется в учреждении.

Исходя из вышеперечисленного,

К = 51 000 +200 + 16 000 + 0 =67 200 руб.

Модуль позволит контролировать учебную деятельность. А также в случае потери журнала, можно его восстановить. Затраты на бумагу составляют примерно 230 000 рублей. Модуль сокращает затраты на бумагу для отчётов на 30%. Таким образом высвобожденные средства составляют 69 000 рублей.

Для подсчёта коэффициента экономической эффективности за первый год используется следующая формула:

$$Ер1 = Эг / К$$

где:

Ер1 – коэффициент экономической эффективности за первый год;

Эг – экономическая эффективность равна сумме высвобожденных средств;

К – капитальные расходы.

69 000/67 200 = 1,03

Для подсчёта коэффициента экономической эффективности за последующие годы используется следующая формула:

$$Ер2 = Эг / К$$

где:

Ер2 – коэффициент экономической эффективности за последующие годы;

Эг – экономическая эффективность равна сумме высвобожденных средств;

К – капитальные расходы без затрат на разработку модуля.

69 00/16 200 = 4,26

3.4 Вывод по главе

Таким образом, разработка практической части выпускной квалификационной работы закончена.

Расчёт экономической эффективности показал, что внедрение системы успеваемости студентов образовательного учреждения повысит контроль за учебной деятельностью и снизит затраты.

ЗАКЛЮЧЕНИЕ

Портал – это сайт, предлагающий пользователю множество различных сервисов, которые работают в рамках этого сайта. Модуль портала – это самостоятельная часть портала, выполняющую определённую функцию.

Результатом выпускной квалификационной работы является разработка проекта автоматизации для клиентской части системы успеваемости студентов образовательного учреждения. Система позволит просматривать и проставлять оценки студентам, а также выводить отчёты по успеваемости студентов.

В ходе разработки были реализованы следующие задачи:

- провести анализ предметной области;
- спроектировать и разработать систему успеваемости студентов образовательного учреждения.

Для реализации выпускной квалификационной работы использовались программные средства – Visual Studio Code, Open Server, PhpMyAdmin.

Модуль портала позволит контролировать учебный процесс учреждения.

Экономическая часть показывает, что модуль портала является экономически выгодным.

Созданный модуль портала удовлетворяет техническому заданию. Таким образом, можно утверждать, что цель выпускной квалификационной работы достигнута, задачи выполнены в полном объёме.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

Учебники, монографии, брошюры:

1. Базы данных. Учебник для прикладного бакалавриата. Б.Я. Советов, В.В. Цехановский, В. Д. Чертовской. 3-е издание. «Юрайт», 2018. – 201 с.
2. HTML и CSS. Разработка и дизайн веб-сайтов. Джон Даккет, 2019. – 174 с.
3. Илюшечкин, В. М. Основы использования и проектирования баз данных. Учебник / В.М. Илюшечкин. - М.: Юрайт, 2019. - 214 с.
4. Стружкин, Н. П. Базы данных. Проектирование. Учебник / Н.П. Стружкин, В.В. Годин. - М.: Юрайт, 2018. - 478 с
5. HTML5 + CSS3. Основы современного WEB-дизайна. Кириченко А.В., Хрусталев А.А, 2018. – 343 с.
6. Изучаем программирование на JavaScript. Фримен Эрик, Робсон Элизабет. 4-е издание. «Питер», 2019. – 190 с.
7. PHP7 для начинающих с пошаговыми инструкциями. Михаил Райтман, Майк Мак Гра, 2018. – 175 с.

Электронные ресурсы:

1. https://studbooks.net/2259584/informatika/obzor_suschestvuyuschih_sistem_ucheta_kontrolya_uspevaemosti_poseschaemosti_studentov_vuzah - Обзор существующих систем учёта и контроля успеваемости и посещаемости студентов.
2. https://siblec.ru/informatika-i-vychislitelnaya-tekhnika/bazy-dannykh - Базы данных. Учебное пособие.
3. https://hostiq.ua/wiki/database/ - База данных (БД). Что это такое?
4. http://htmlbook.ru/html - Справочник по HTML
5. https://learn.javascript.ru/ - Современный учебник JavaScript
6. https://www.php.net/ - PHP: Hypertext Preprocessor
7. https://html5css.ru/html/html_css.php - Стили - CSS HTML уроки для начинающих академия

8. https://htmlacademy.ru/tutorial/php/mysql - Работа с MySQL в PHP – Учебник по PHP – HTML Academy

9. https://sql-language.ru/ - Язык запросов SQL

10. https://scienceforum.ru/2017/article/2017038108 - Проектирование АС Учёта успеваемости студентов

11. https://php-myadmin.ru/doc/ - Руководство и FAQ по phpMyAdmin. Материалы для веб-разработчиков.