



МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЮУрГГПУ»)

ФАКУЛЬТЕТ МАТЕМАТИКИ, ФИЗИКИ, ИНФОРМАТИКИ
КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ

Разработка автоматизированной системы для создания отчета по учебной и
производственной практике

Выпускная квалификационная работа
по направлению 09.03.02 Информационные системы и технологии

Направленность программы бакалавриата
«Информационные технологии в образовании»

Форма обучения очная

Проверка на объем заимствований:
_____ % авторского текста

Работа _____ к защите
рекомендована/не рекомендована

« ___ » _____ 20__ г.
зав. кафедрой И, ИТ и МОИ

_____ Рузаков А.А.

Выполнил:
Студент группы ОФ-413/095-4-1
Журахова Анастасия Александровна

Научный руководитель:
доцент кафедры, кандидат педагогических
наук
Дмитриева Ольга Александровна

Челябинск
2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И НАЗНАЧЕНИЕ ПРОЕКТИРУЕМОЙ СИСТЕМЫ	5
1.1 Характеристика объекта автоматизации.....	5
1.2 Описание проектируемой системы	8
1.3 Сравнительный анализ аналогов разрабатываемой системы	10
1.4 Выводы по первой главе	13
ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ.....	14
2.1 Выбор архитектуры информационной среды и инструментальных средств реализации	14
2.2 Описание модели данных.....	17
2.3 Описание реализации основных функций системы	25
2.4 Выводы по второй главе.....	34
ГЛАВА 3. ДОКУМЕНТИРОВАНИЕ ГОТОВОГО ПРОГРАММНОГО ПРОДУКТА	35
3.1 Испытания автоматизированной системы	35
3.2 Руководство пользователей информационной системы.....	36
3.3 Техничко-экономическое обоснование разработки информационной системы	43
3.4 Выводы по третьей главе	44
ЗАКЛЮЧЕНИЕ	46
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	47

ВВЕДЕНИЕ

Студент любой специализации на протяжении своего обучения проходит учебную, производственную, преддипломную практику, в результате которой формируются отчеты как со стороны студента, так и со стороны преподавателя. Содержание отчета свидетельствует о закрепленных учащимися знаниях, умениях, приобретении практического опыта, а также о формировании общих и профессиональных компетенций.

Тема выпускной квалификационной работы (ВКР) «Разработка автоматизированной системы для создания отчёта по учебной и производственной практике» была сформирована после сбора информации и анализа выявленных проблем в педагогическом отделении ГБПОУ «Саткинский горно-керамический колледж им. А.К. Савина».

В условиях развития информационного общества данная тема является актуальной, так как автоматизация процесса формирования отчета позволит сэкономить время, снизит объем монотонной работы, а также разгрузит руководителя и преподавателей.

Цель работы: разработка автоматизированной системы для создания отчёта по учебной и производственной практике.

В соответствии с этой целью были поставлены следующие задачи:

1. Изучить объект автоматизации.
2. Описать структуру организации.
3. Описать бизнес-процессы организации.
4. Определить проблему в структуре деятельности предприятия заказчика.
5. Описать и утвердить техническое задание на разработку информационной системы.
6. Проанализировать аналоги проектируемой системы.
7. Разработать автоматизированную систему.
8. Написать руководство пользователя.

9. Выполнить технико-экономическое обоснование разработанной системы.

10. Внедрить информационную систему в организацию.

Практическая значимость ВКР заключается в том, что разработанная автоматизированная система снизит объем работы руководителя структурного подразделения.

ГЛАВА 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И НАЗНАЧЕНИЕ ПРОЕКТИРУЕМОЙ СИСТЕМЫ

1.1 Характеристика ГБПОУ «Саткинский горно-керамический колледж им. А.К. Савина»

Полное наименование: Государственное бюджетное профессиональное образовательное учреждение «Саткинский горно-керамический колледж имени А.К. Савина».

Сокращенное наименование: ГБПОУ «СГКК им. А.К. Савина»

Дата создания образовательной организации: 1942 год.

Официальный сайт: <https://sgkk.ru> [1].

Цель колледжа: подготовить выпускника не только к профессиональной деятельности, но и к жизни в изменяющемся мире.

Основными задачами колледжа являются:

- создание условий для подготовки квалифицированных рабочих и специалистов с профессионально-техническим и средним специальным образованиями;
- удовлетворение потребностей личности в интеллектуальном, культурном и нравственном развитии;
- создание условий для физического совершенствования обучающихся, овладения ценностями и навыками здорового образа жизни;
- обеспечение социальной защиты обучающихся и установленных законодательством социальных гарантий;
- формирование гражданской ответственности, патриотизма и национального самосознания на основе государственной идеологии и т.д.

Целевой результат данного образовательного учреждения – это специалист, который будет обладать компетенциями современного мира для достижения успеха в профессии и жизни благодаря поддерживающей образовательной среде, практико-ориентированному обучению с помощью наиболее эффективных методов и форм обучения и воспитания.

На рисунке 1 представлена организационная структура ГБПОУ «Саткинский горно-керамический колледж им. А.К. Савина»

Структура управления ГБПОУ «Саткинский Горно-керамический колледж имени А.К Савина»

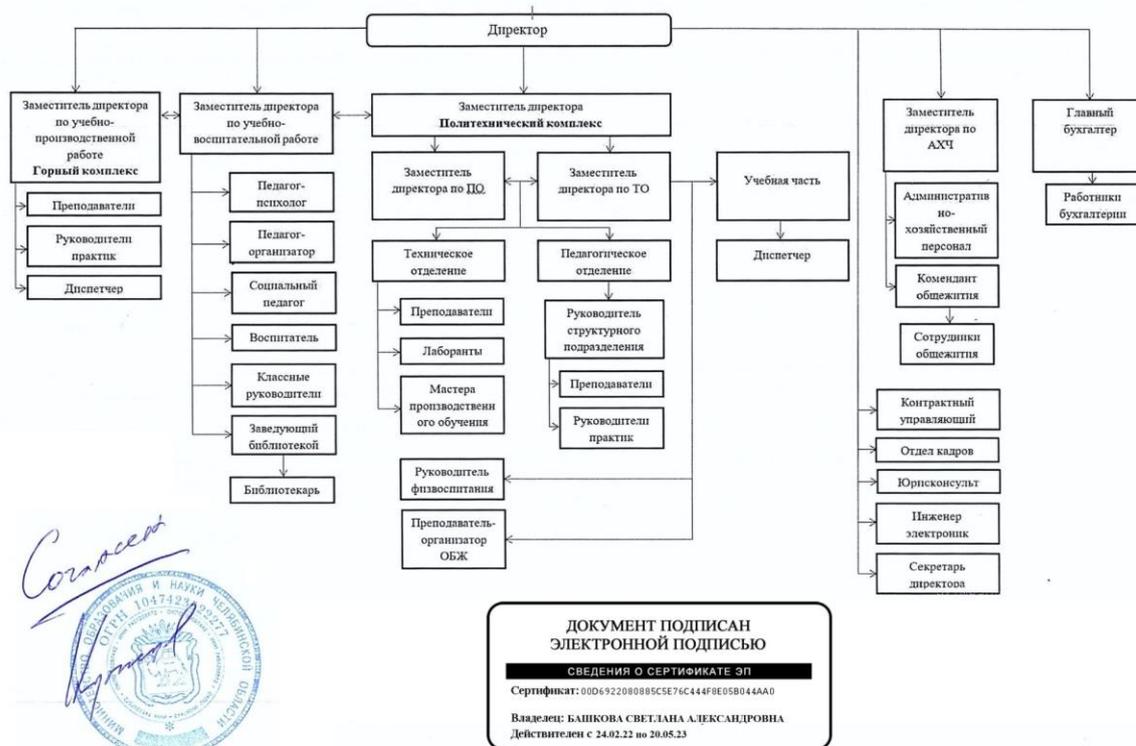


Рис 1 – Организационная структура ГБПОУ «СГКК им. А.К. Савина»

В организационной структуре ГБПОУ СГКК выделим педагогическое отделение политехнического комплекса. В него входит:

1. Руководитель структурного подразделения.
2. Преподаватели.
3. Руководители практики.

Для лучшего понимания особенностей функционирования педагогического отделения опишем бизнес-процессы организации [6].

Бизнес-процесс организации – это логическая последовательность человеческих действий (или группы лиц) в рабочем коллективе. Целью его описания является анализ и регламентация определенных событий в штате работников.

В данной работе для описания бизнес-процессов организации использовалась методология IDEF0 [2].

Описание системы с помощью IDEF0 называется функциональной моделью. В функциональной модели используются как естественный, так и графический языки. Для передачи информации о конкретной системе источником графического языка является сама методология IDEF0.

Одним из основных бизнес-процессов педагогического отделения колледжа является формирование отчета по учебной и производственной практике. Вся информация хранится отдельно у каждого преподавателя на бумажных носителях, MS Word и MS Excel, что впоследствии увеличивает время по формированию общего отчета.

Функциональная модель, приведённая на рисунках 2 и 3, показывает каким образом можно оптимизировать существующую работу по формированию отчета по учебной и производственной практике.

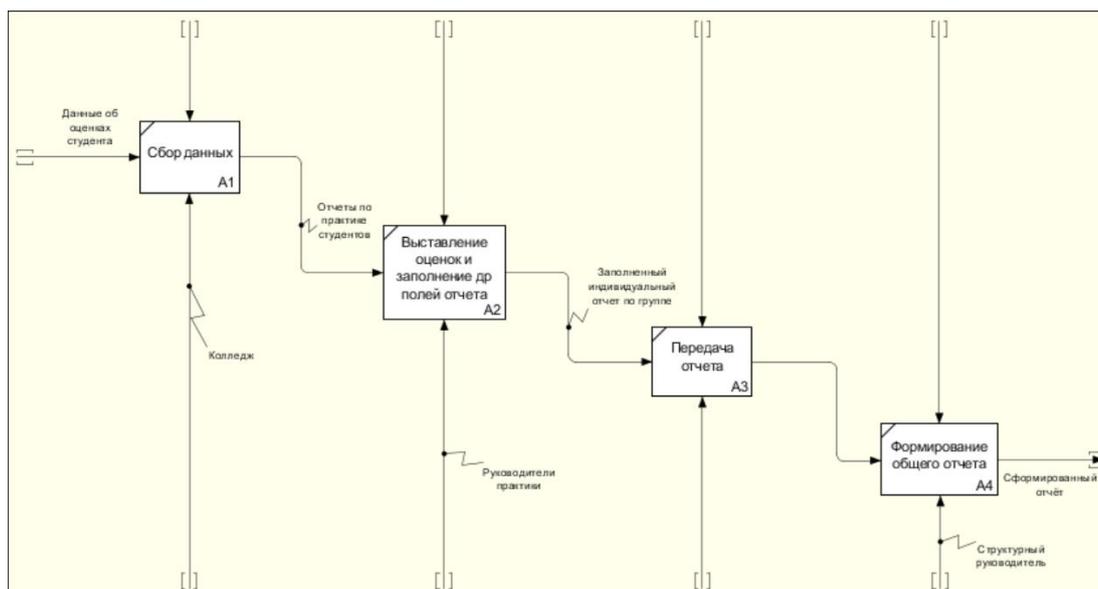


Рис 2 – Функциональная модель «как есть»

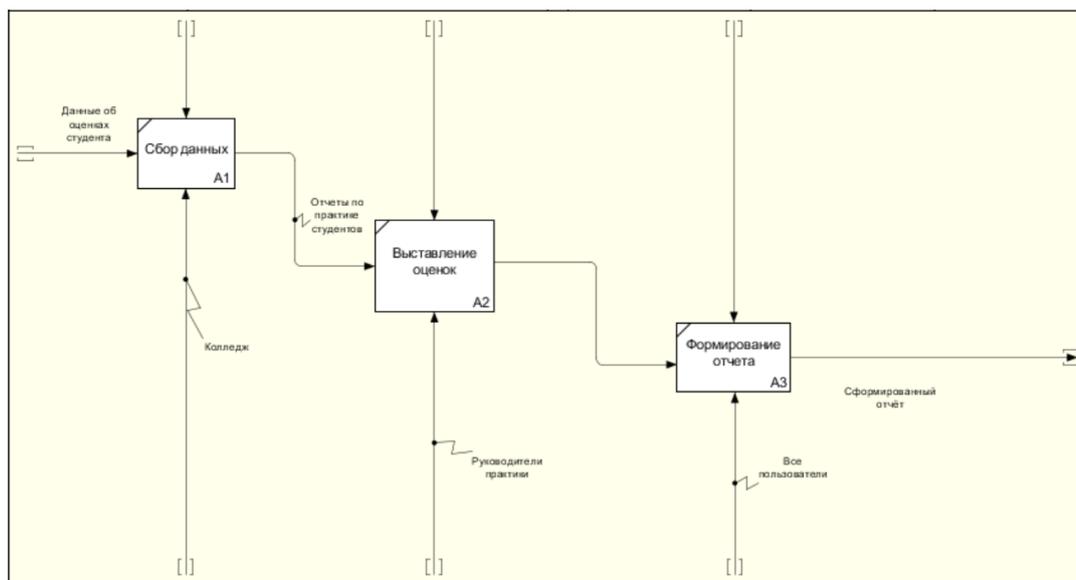


Рис 3 – Функциональная модель «как будет»

Внедряемая автоматизированная система должна:

1. Формировать таблицы по заполненным модулям и дисциплинам.
2. Выводить средние баллы по каждому ученику и по каждому модулю.
3. Формировать и скачивать готовый отчет в формате PDF.

1.2 Описание проектируемой системы

Основным назначением разрабатываемой системы является формирование отчета по учебной и производственной практике.

Система предназначена для структурного руководителя и руководителей практики.

Целью настоящей работы является создание клиент-серверного приложения, для автоматизации процесса формирования отчета, сокращения монотонной работы структурного руководителя педагогического отделения.

Структурный руководитель является администратором системы. Ему доступны функции:

1. Регистрация пользователей.
2. Добавление модулей и дисциплин.
3. Добавление студентов.

4. Просмотр отчетов.
5. Скачивание в формате PDF.

Руководителям практики доступны функции:

1. Добавление студентов.
2. Выставление оценок по модулям.
3. Просмотр отчетов.
4. Скачивание в формате PDF.

Неавторизованный пользователь не может просматривать информацию.

Автоматизированная система создается как веб-приложение, с возможностью преобразования и сохранения отчетов в формате PDF. Реализуется на персональном компьютере, который подключен к общей локальной сети и имеет доступ к сети Интернет.

Структурная схема веб-приложения представлена на рисунке 4.

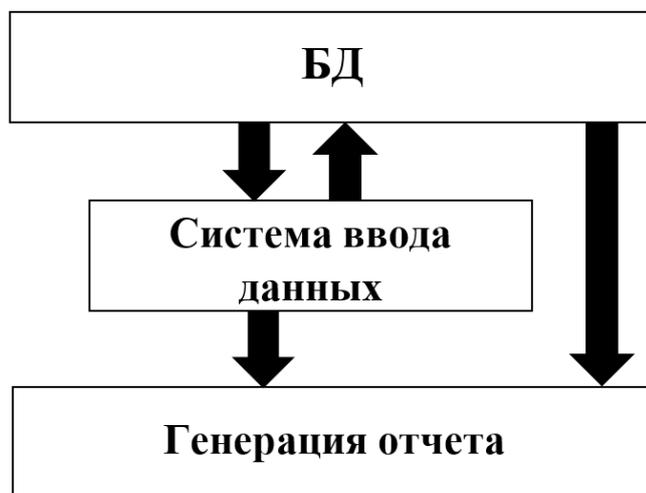


Рис 4 – Структурная схема автоматизированной системы для формирования отчета

В состав разрабатываемой системы должны входить следующие функциональные подсистемы:

- подсистема ввода данных;
- подсистема хранения информации;
- подсистема генерации отчёта.

Подсистема ввода данных предназначена для реализации процессов ввода данных необходимых для наполнения подсистемы хранения. Она должна обеспечивать сбор информации о студентах, модулях и дисциплинах, по которым выставляются оценки за практику.

Подсистема хранения информации должна хранить необходимую информацию в базе данных для корректной работы системы.

Подсистема генерации отчёта должна генерировать отчёт в формате PDF.

1.3 Сравнительный анализ аналогов разрабатываемой системы

Рассмотрим аналоги разрабатываемой системы. Самыми простыми и доступными являются MS Word и MS Excel [3].

Microsoft Word – мощный текстовый процессор, предназначенный для выполнения всех процессов обработки текста (рисунок 5).

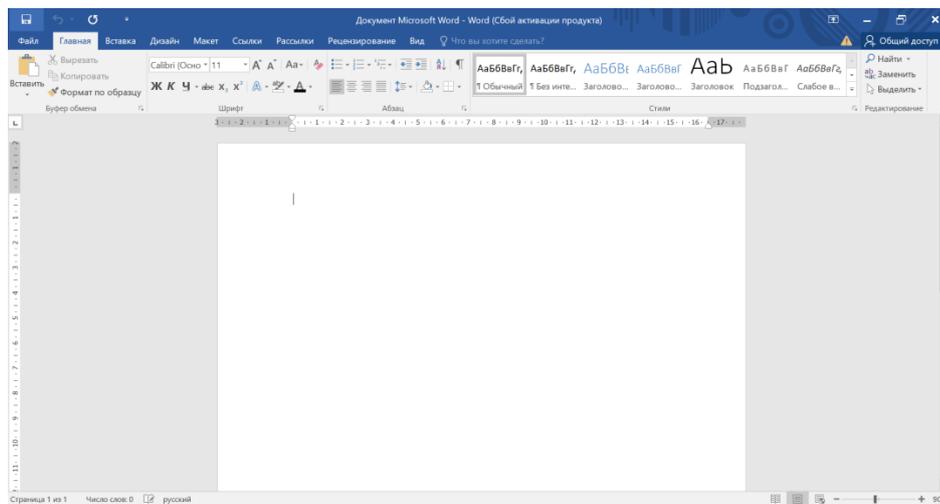


Рис 5 – Интерфейс Microsoft Word

Разработчик: компания Microsoft.

Среди преимуществ можно выделить:

1. Универсальность
2. Надежность.
3. Занимает мало места в памяти устройства.
4. Удобный интерфейс.
5. Скорость.

6. Экономия времени.
7. Возможность обработки документов различного формата.
8. Широкий спектр распространения.
9. Возможность иметь доступ к файлам вне своего устройства.
10. Удобное управление и доступные к персонализации настройки.
11. Высокая работоспособность и качественное выполнение поставленных задач.

Недостатки Microsoft Word включают:

1. Высокая цена.
2. Несовместимость форматов различных версий.
3. Трудности и проблемы различий в интерфейсе.

Microsoft Excel – программа, позволяющая работать с электронными таблицами. Можно собирать, преобразовывать и анализировать данные, проводить визуализацию информации, автоматизировать вычисления и выполнять еще ряд полезных и необходимых в работе задач (рисунок 6).

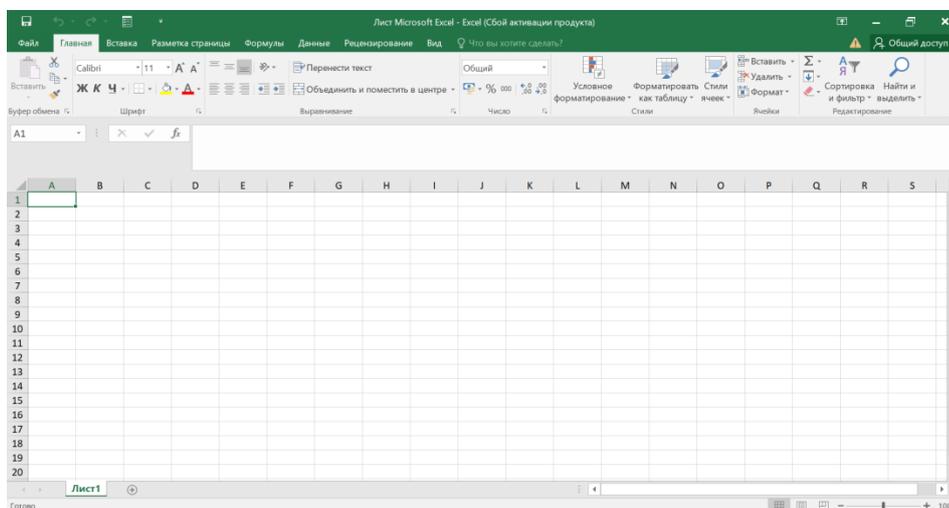


Рис 6 – Интерфейс Microsoft Excel

Разработчик: компания Microsoft.

Преимущества электронных таблиц:

1. Облегчает создание мгновенных расчетов.
2. Их легко изменить.
3. Обеспечивает эффективную организацию данных.
4. Большинство электронных таблиц предлагают стороннюю поддержку.

5. Есть варианты интеграции.
6. Возможно создать несколько электронных таблиц в одном файле.

Недостатки электронных таблиц:

1. Чтобы выучить формулы, нужно время.
2. Приложение или программное обеспечение может не распознавать ошибки вычислений.
3. Для ввода данных в каждую ячейку требуется время.
4. Некоторые файлы электронных таблиц могут быть несовместимы.
5. Программы для работы с электронными таблицами иногда могут быть дорогими.

Так же как аналог разрабатываемой системы можно рассмотреть 1С [4].

Программа «1С» – это продукт для автоматизации предприятий. Интерфейс программы представлен на рисунке 7.

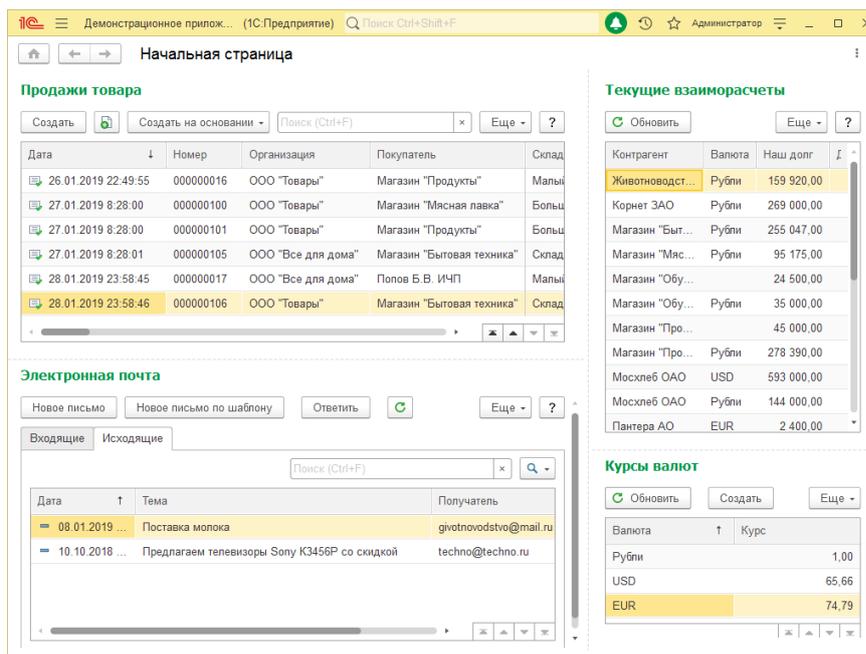


Рис 7 – Интерфейс 1С

Из плюсов данного продукта можно выделить:

- высокий уровень функциональности, позволяющий решать широкий спектр задач в автоматическом режиме, что даёт возможность экономить время и другие ресурсы;
- наличие единой технической платформы и т.д.

Из минусов можно отметить необходимость привлечения к запуску квалифицированных программистов и платные обновления.

Выводы по первой главе

В первой главе квалификационной работы была определена цель и задачи для разработки автоматизированной системы.

Рассмотрена структура объекта автоматизации ГБПОУ «Саткинский горно-керамический колледж им. А.К. Савина».

Также было проведено описание разрабатываемой системы и необходимых требований.

В заключении рассмотрены аналогичные программные продукты для формирования отчета по учебной и производственной практике.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 Выбор архитектуры информационной среды и инструментальных средств реализации

Автоматизированная система для формирования отчёта по учебной и производственной практике разработана в виде клиент-серверного приложения, которое имеет ряд преимуществ:

1. Доступность. Доступ возможен с различных браузеров, а также может осуществляться со всех устройств.
2. Масштаб. Одновременное использование большой аудиторией.
3. Простота в использовании для пользователя. Пользователю не нужно загружать себя лишней информацией по настройке, обновлению и т.п.

Для разработки веб-приложения выбраны следующие средства:

1. Microsoft Visual Studio 2022 Community [5].

Microsoft Visual Studio 2022 Community – бесплатная версия среды разработки Visual Studio для разработки приложений на платформа .Net Framework. Microsoft Visual Studio 2022 Community включает в себя:

- a. Создание приложений для любой платформы.
- b. Конструкторы, редакторы, отладчики и профилировщики в одном инструменте.
- c. Доступ к тысячам расширений.
- d. Программирование на C#, Visual Basic, F#, C++, HTML, JavaScript, TypeScript, Python и других языках.
- e. Упрощенная модульная установка.
- f. Подключение к облаку.
- g. Доступ к бесплатным инструментам и ресурсам, а именно к широкому набору бесплатных инструментов разработчика, ряду предоставляемых по запросу учебных курсов Xamarin University, Pluralsight, день-

гам на счете в Azure и многому другому в составе Visual Studio Dev Essentials.

h. Любой индивидуальный разработчик может создавать бесплатные или платные приложения с помощью Visual Studio Community.

i. Visual Studio Community может использовать неограниченное число пользователей в организации в следующих случаях: в учебных аудиториях, для научных исследований или участия в проектах с открытым кодом.

2. SQL Server 2016 Express.

SQL Server 2016 Express – мощная и надежная бесплатная система управления данными с широкими функциональными возможностями, обеспечивающая функциональное и надежное хранилище данных для веб-сайтов и настольных приложений. Она идеально подходит для обучения, разработки и обеспечения работы классических, небольших серверных приложений и веб-приложений, а также для распространения независимыми поставщиками программного обеспечения.

3. Microsoft SQL Server Management Studio.

Microsoft SQL Server Management Studio – основной, стандартный и полнофункциональный инструмент для работы с Microsoft SQL Server, разработанный компанией Microsoft, который предназначен как для разработчиков, так и для администраторов SQL Server. С помощью SSMS можно разрабатывать базы данных, выполнять инструкции T-SQL, а также администрировать Microsoft SQL Server.

4. Entity Framework.

Entity Framework – объектно-ориентированная технология доступа к данным, является object-relational mapping (ORM) решением для .NET Framework от Microsoft. Entity Framework позволяет значительно сократить код работы с базами данных. При этом он предоставляет большие возможности. Например, можно использовать:

1. Foreign keys.

2. Связи one-to-one, one-to-many и many-to-many.
3. Параметризацию запросов.
4. Хранимые процедуры.

Непосредственно разработка системы была осуществлена с помощью нескольких языков:

1. Языка программирования C#.

C# («Си Шарп») – объектно-ориентированный язык программирования. Является модификацией языка C от компании Microsoft. Синтаксис приближен к Java и C++. Среди особенностей языка можно выделить: строгую типизацию, сохранение концепций объектно-ориентированного программирования, функциональность, достаточно мощный инструментарий, стабильную работу через Visual Studio, компактный и легко читаемый код, понятный даже новичкам синтаксис.

2. Языка гипертекстовой разметки страницы HTML.

HTML (HyperText Markup Language) – язык гипертекстовой разметки страницы. Он применяется для того, чтобы дать браузеру понять, как нужно отображать загруженный сайт. Язык состоит из тегов – это своеобразные команды, которые преобразовываются в визуальные объекты в браузере пользователя.

3. Для визуального отображения сайта был использован язык программирования CSS.

CSS – язык, который отвечает за описание внешнего вида HTML-документа. Подавляющее большинство современных веб-сайтов работают на основе связки HTML+CSS. Если HTML структурирует контент на странице, то CSS позволяет отформатировать его, сделать более привлекательным для читателя.

4. Декларативного языка программирования SQL (Structured Query Language).

SQL – это структурированный язык запросов, созданный для того, чтобы получать из базы данных необходимую информацию. Если описать

схему работы SQL простыми словами, то специалист формирует запрос и направляет его в базу. Та в свою очередь обрабатывает эту информацию, «понимает», что именно нужно специалисту, и отправляет ответ.

5. Языка скриптов JavaScript.

JavaScript – это язык программирования, который используют для написания frontend- и backend- частей сайтов, а также мобильных приложений. Часто в текстах и обучающих материалах название языка сокращают до JS. Это язык программирования высокого уровня, то есть код на нем понятный и хорошо читается.

6. Языка программирования PHP.

PHP – интерпретируемый язык программирования, позволяющий создавать программы в процедурном и объектно-ориентированном стиле. Основная задача PHP – это «оживление» HTML страниц. Обычные HTML-страницы статичны. Статичность (или неизменность) означает, что после того, как страницу создали и загрузили на сайт, при каждом обращении к этой странице браузер покажет её любому пользователю в неизменном виде. PHP позволяет изменять веб–страницу на сервере непосредственно перед тем, как она будет отправлена браузеру.

Формирование отчетов происходит с использованием открытого генератора отчетов FastReport OpenSource.

2.2 Описание модели данных

В автоматизированной системе для формирования отчета по учебным и производственным практикам для хранения данных о пользователях системы была использована база данных (БД) в SQL Server 2016 Express.

База данных состоит из двух частей:

1. Examination – успеваемость обучающихся. Состоит из 7 таблиц.
2. Users – пользователи системы. Состоит из 4 таблиц.

Определим список сущностей (таблиц) и их свойств, необходимых для работы системы:

1. Таблица Users – пользователи системы (хранит данные о пользователях):
 - a. Id – идентификатор. Тип данных: int (11).
 - b. Login – логин. Тип данных: varchar (32).
 - c. Password – пароль. Тип данных: varchar (32).
 - d. LastName – фамилия. Тип данных: varchar (100).
 - e. FirstName – имя. Тип данных: varchar (100).
 - f. MiddleName – отчество. Тип данных: varchar (100).
 - g. RoleId – роль в системе: администратор, преподаватель. Тип данных: varchar (32).
2. Таблица Roles – роли пользователей (хранит данные о ролях пользователей):
 - a. Id – идентификатор. Тип данных: int (11).
 - b. RoleName – наименование: администратор, преподаватель. Тип данных: varchar (32).
3. Таблица Institutions – учебные заведения (хранит данные об ГБПОУ «СГКК им. Савина»):
 - a. Id – идентификатор. Тип данных: int (11).
 - b. InstitutionName – название учебного заведения: ГБПОУ «СГКК им. Савина». Тип данных: varchar (100).
4. Таблица InstitutionAssignment – закрепление учебных заведений за пользователями (хранит данные о закреплении пользователя за определенным учебным заведением):
 - a. Id – идентификатор. Тип данных: int (11).
 - b. UserId – идентификатор пользователя (Users). Тип данных: int (11).
 - c. InstitutionId – идентификатор учебного заведения (Institutions). Тип данных: int (11).

5. Таблица Specialities – направления подготовки/специальности учебных заведений (хранит данные о направлениях подготовки/специальности учебного заведения):
 - a. Id – идентификатор. Тип данных: int (11).
 - b. SpecialityCode – код направления/специальности. Тип данных: int (11).
 - c. SpecialityName – название направления/специальности. Тип данных: varchar (100).
 - d. InstitutionId – идентификатор учебного заведения (Institutions). Тип данных: int (11).
6. DisciplineModules – модули дисциплин направления подготовки/специальности:
 - a. Id – идентификатор. Тип данных: int (11).
 - b. ModuleCode – код модуля. Тип данных: int (11).
 - c. ModuleName – название модуля. Тип данных: varchar (100).
 - d. SpecialityId – идентификатор направления/специальности (Specialities). Тип данных: int (11).
7. Таблица Disciplines – дисциплины (хранит данные о дисциплинах):
 - a. Id – идентификатор. Тип данных: varchar (100).
 - b. DisciplineName – название дисциплины. Тип данных: varchar (100).
 - c. Semester – семестр, в котором подается дисциплина (Semesters).
8. Таблица Semesters – семестры (хранит данные о семестрах):
 - a. Id – идентификатор. Тип данных: int (11).
 - b. Caption – наименование. Тип данных: varchar (100).
9. Таблица Students – обучающиеся (хранит данные об учащихся, обучающихся в данном учебном заведении):
 - a. Id – идентификатор. Тип данных: int (11).

- b. LastName – фамилия. Тип данных: varchar (100).
- c. FirstName – имя. Тип данных: varchar (100).
- d. MiddleName – отчество. Тип данных: varchar (100).
- e. SpecialityId – идентификатор направления/специальности (Specialities). Тип данных: int (11).
- f. Kurs – курс. Тип данных: int (11).

10. Таблица Examination – успеваемость обучающихся (хранит данные об успеваемости учащихся):

- a. Id – идентификатор. Тип данных: int (11).
- b. DisciplineId – идентификатор дисциплины (Disciplines). Тип данных: int (11).
- c. StudentId – идентификатор обучающегося (Students). Тип данных: int (11).
- d. Mark – показатель успеваемости (Marks). Тип данных: int (11).

11. Таблица Marks – показатели успеваемости (хранит данные о показателях успеваемости учащихся):

- a. Mark – показатель успеваемости. Тип данных: int (11).

Схема модели данных автоматизированной системы создана с помощью программы DbSchema и представлена на рисунке 8.

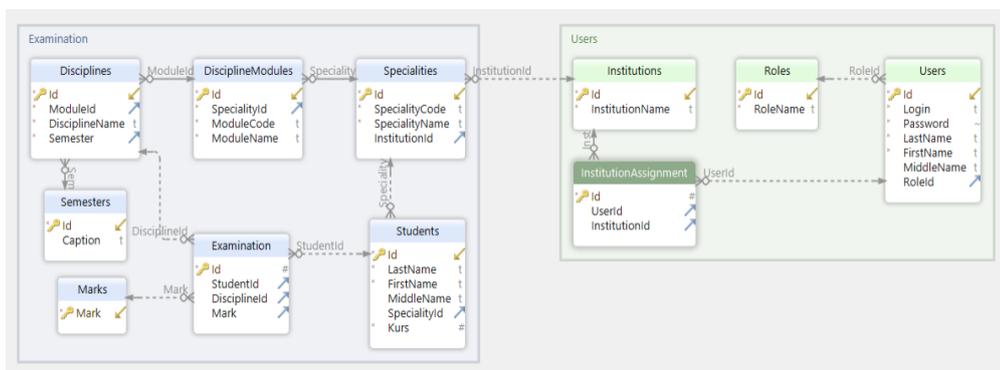


Рис 8 – Схема модели данных

Рассмотрим связи «Один-к-одному» между таблицами.

На рисунке 9 представлена таблица Users и таблица Roles, которые связаны по полю RoleId. У одного пользователя, может быть только одна роль (Администратор или преподаватель).

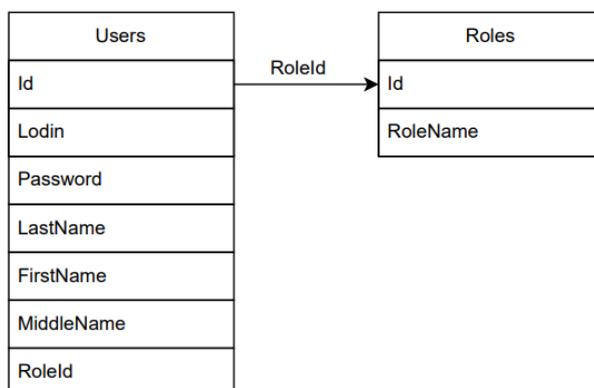


Рис 9 – Связь между таблицей Users и таблицей Roles

На рисунке 10 представлена таблица Users и таблица InstitutionAssignment, которые связаны по полю UserId. Один пользователь прикреплен к одному учебному заведению.

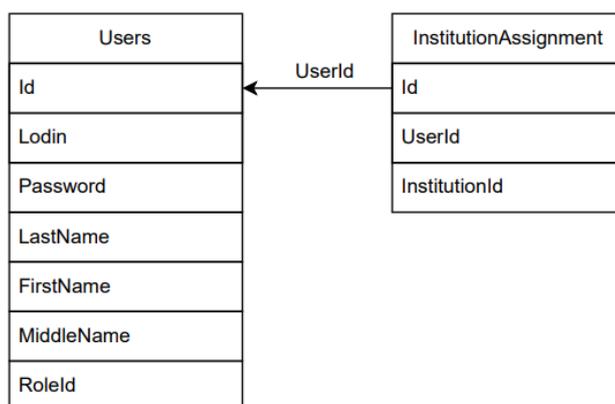


Рис 10 – Связь между таблицей Users и таблицей InstitutionAssignment

На рисунке 11 представлена таблица InstitutionAssignment и таблица Institutions, которые связаны по полю Id. У одного учебного заведения есть одно название.

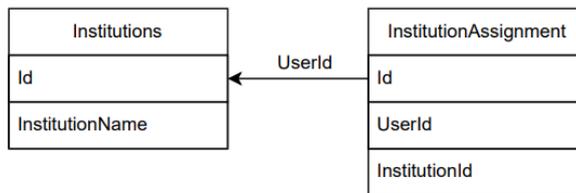


Рис 11 – Связь между таблицей Institutions и таблицей InstitutionAssignment

На рисунке 12 представлена Таблица Specialities и таблица Student, которые связаны по полю SpecialityId. Один студент учиться на одной специальности.

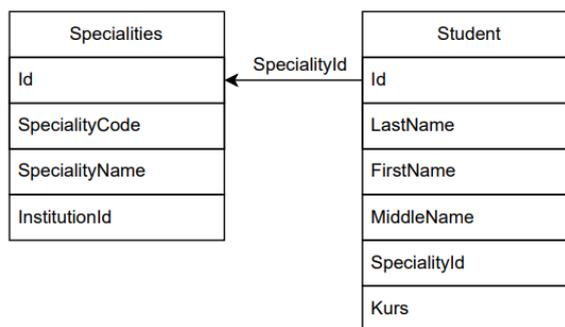


Рис 12 – Связь между таблицей Specialities и таблицей Student

На рисунке 13 представлена таблица Student и таблица Examination, которые связаны по полю StudentId. Один студент имеет одну успеваемость.

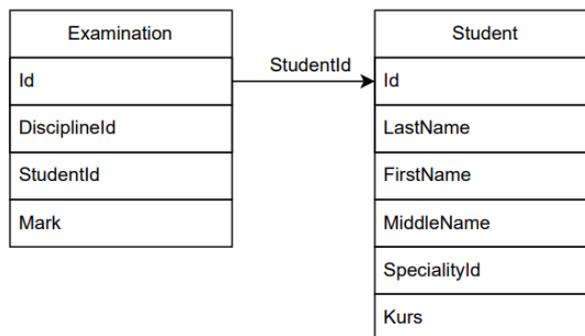


Рис 13 – Связь между таблицей Examination и таблицей Student

На рисунке 14 представлена таблица DisciplineModules и таблица Disciplines, которые связаны по полю ModuleId. У одной дисциплины есть одно название.

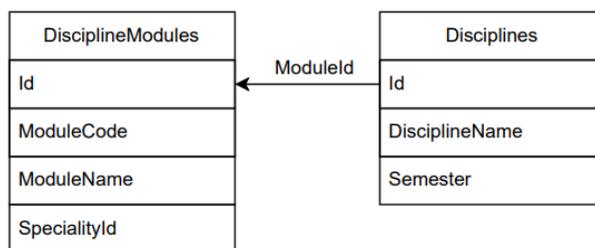


Рис 14 – Связь между таблицей DisciplineModules и таблицей Disciplines
 На рисунке 15 представлена таблица Examination и таблица Disciplines
 связаны по полю DisciplineId.

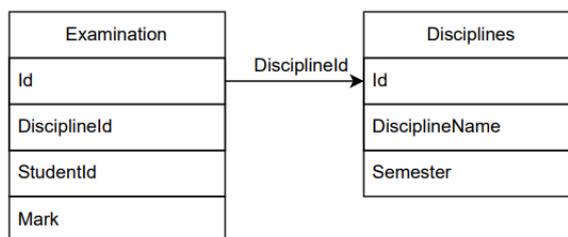


Рис 15 – Связь между таблицей Examination и таблицей Disciplines
 Рассмотрим связи «Один-ко-многим» между таблицами.

На рисунке 16 представлена таблица Institutions и таблица Specialities, которые связаны по полю InstitutionId. У одного учебного заведения есть несколько направлений подготовки/специальностей.

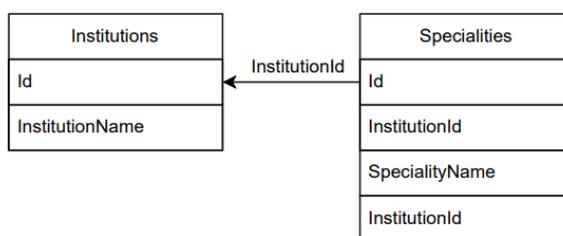


Рис 16 – Связь между таблицей Institutions и таблицей Specialities

На рисунке 17 представлена таблица Specialities и таблица DisciplineModules, которые связаны по полю SpecialityId. На одном направлении подготовки есть несколько модулей.

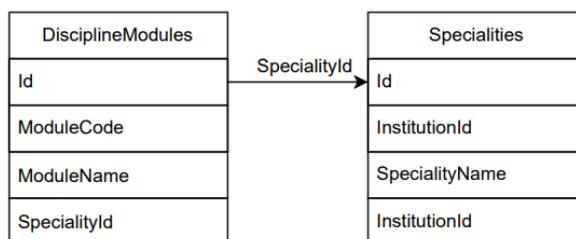


Рис 17 – Связь между таблицей Specialities и таблицей DisciplineModules

На рисунке 18 представлена таблица Examination и таблица Marks, которые связаны по полю Mark. У одного студента есть несколько оценок.

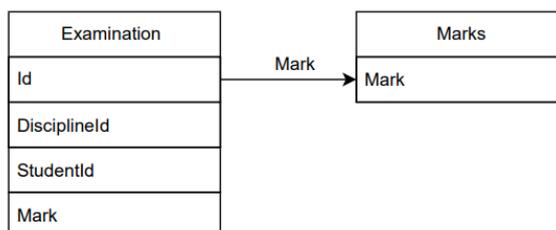


Рис 18 – Связь между таблицей Examination и таблицей Marks

На рисунке 19 представлена таблица Examination и таблица Disciplines, которые связаны по полю DisciplineId.

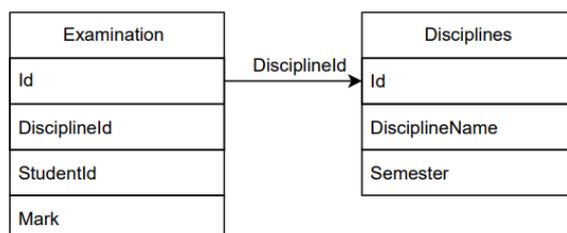


Рис 19 – Связь между таблицей Examination и таблицей Disciplines

На рисунке 20 представлена таблица Disciplines и таблица Semesters, которые связаны по полю SemestersId. В семестр есть несколько дисциплин.

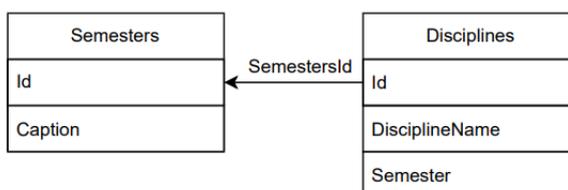


Рис 20 – Связь между таблицей Semesters и таблицей Disciplines

2.3 Описание реализации основных функций системы

Схема приложения представлена на рисунке 21.

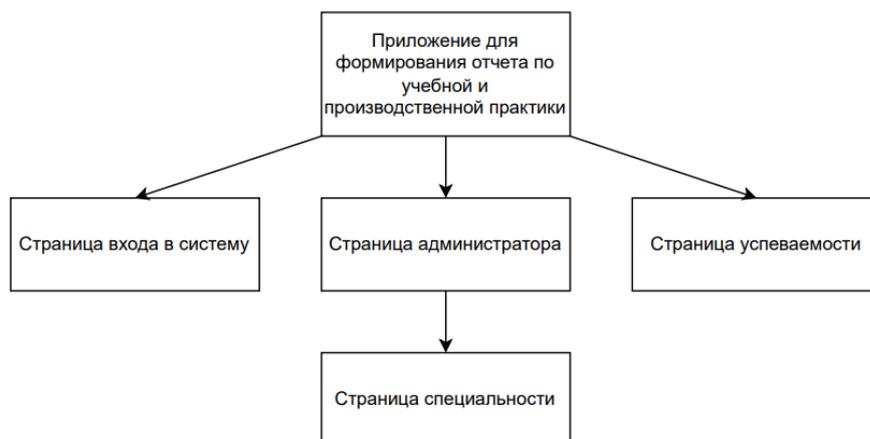


Рис 21 – Схема приложения

1. Структура приложения.

Функционал приложения разделен на следующие основные страницы:

- a. Страница входа в систему.
- b. Страница администратора.
- c. Страница успеваемости.

2. Страница входа в систему (рисунок 22).

Для корректной работы системы контроллер Home и действие Index вызываются по умолчанию при открытии пользователем веб-приложения, поэтому он будет использоваться в качестве страницы авторизации пользователей.

Для работы с БД добавим в контроллер поле «DbContext _context» (DbContext – сгенерированный Entity Framework класс с коллекциями всех сущностей БД, и реализующий функции чтения и записи данных), добавим конструктор с таким же параметром и запишем параметр в поле. Данное поле будет добавлено во все контроллеры приложения.

В представлении Index добавим форму авторизации с полями для ввода логина и пароля и укажем элементу form атрибут method="Post". Для

обработки формы добавим в контроллер действие Index с параметрами, передаваемыми из формы, и отметим его как «HttpPost».

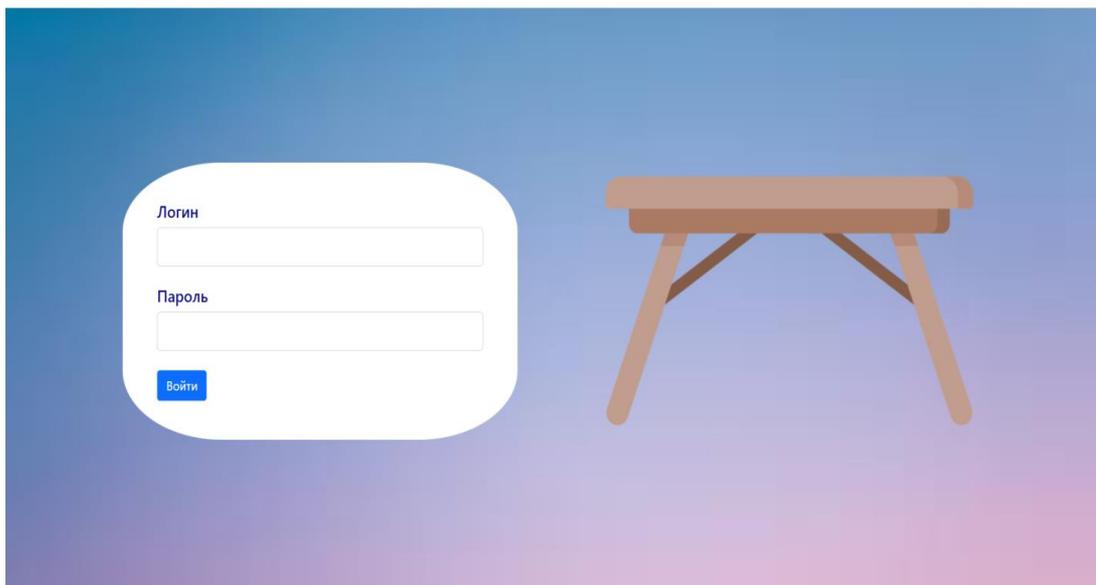


Рисунок 22 – Страница входа в систему

3. Авторизация пользователей.

Обработка формы авторизации проводит поиск пользователя по введенным логину и паролю среди всех пользователей с помощью метода FirstOrDefault. Поскольку пароль хранится в БД в зашифрованном виде, добавим в папку Models проекта статический класс Helper для реализации методов, необходимых в разных частях приложения.

В класс Helper добавим статический метод CreateMD5 для шифрования, введенного на форме авторизации, пароля по алгоритму MD5.

Для хранения данных авторизованного пользователя на время работы приложения запишем его данные, найденные в коллекции пользователей, в сессию.

Для работы с сессией добавим в файл Program.cs строку `builder.Services.AddSession(options => { options.IdleTimeout = TimeSpan.FromHours(1); });`. Свойство IdleTimeout определяет время хранения сессии в памяти если пользователь не совершает никаких действий.

4. Ограничение доступа неавторизованных пользователей.

Доступ к функционалу веб-приложения может осуществляться не только через интерфейс html-страниц, но и через адресную строку браузе-

ра. Для того чтобы пользователь не мог обращаться к функциям приложения без авторизации через адресную строку создадим метод ShowPage в классы Helper и передадим ему текущую сессию приложения и имя запрашиваемой страницы. Метод будет возвращать значения true/false в зависимости от доступности функционала пользователю и перенаправлять на страницу авторизации или иную доступную страницу. Данный метод будет вызываться во всех действиях контроллеров.

5. Страница администратора (рисунок 23).

После успешной авторизации пользователь с ролью «Администратор» автоматически перенаправляется на страницу администратора. На странице будут отображаться данные пользователей и информация об учебных заведениях. В действии Index запишем список пользователей и учебных заведений из БД в соответствующие переменные «users» и «institutions» с помощью метода ToListAsync у коллекций Users и Institutions, доступных в поле _context, для включения связанных сущностей будет использован метод Include с указанием поля, представляющего связанную сущность. Поскольку представления могут иметь только один тип модели, создадим переменную анонимного класса и запишем в свойства с соответствующими названиями переменные «users» и «institutions» и передадим как модель представления в методе View.

Для действия Index создадим представление Index.cshtml. При создании модели укажем тип «пустое представление», поскольку передаваемая модель не является строго типизированным классом, определенным в проекте.

Для отображения данных модели в представлении создадим в папке Shared частичное представление для каждого из свойств модели и выведем их с помощью метода «Html.RenderPartial» указав имя представления, модель и дополнительные параметры. В частичных представлениях выведем данные в виде таблице с кнопками действий добавления, удаления и редактирования.

Для каждой кнопки действия в представлении Index добавим соответствующие формы, которые будут отображаться при нажатии кнопки с помощью стилей CSS и заполняться при необходимости с помощью javascript функций, вызываемых при срабатывании события onclick.

Все используемые javascript функции будут определены в файле site.js проекта. Для каждой формы добавим в контроллер Administration действия с параметрами, идентичными указанным на формах и атрибутом HttpPost.

Формы добавления и редактирования для каждой сущности объединим в одну форму, для определения необходимого действия добавления или редактирования в поле идентификатора на форме с помощью функции javascript будет добавлен идентификатор из модели представления или значение «-1».

Для полей форм с выбором из списка в действии Index добавим в коллекцию ViewBag необходимые списки типа SelectList и привяжем с помощью html-атрибута asp-items в представлении.

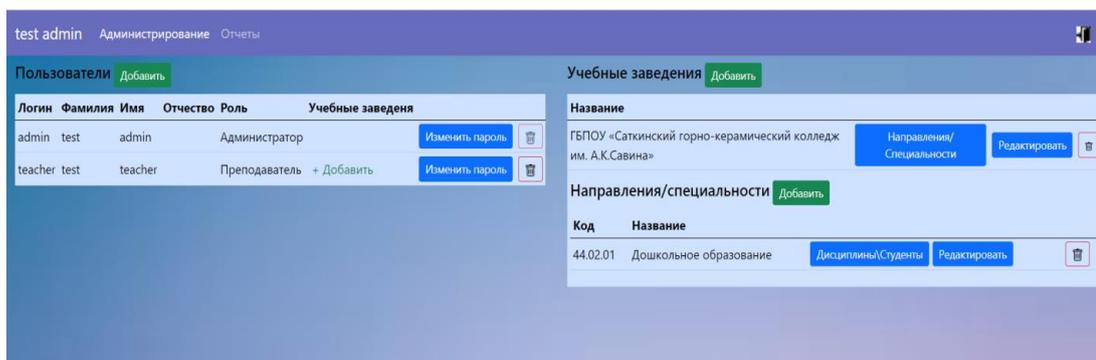


Рисунок 23 – Страница администратора

6. Страница специальности (рисунок 24).

На странице специальности будет отображаться следующая информация:

1. Список модулей специальности.
2. Список практик.
3. Список студентов.

Поскольку страница является подразделом страницы администратора, для нее не будет создан отдельный контроллер, а добавлено действие Speciality в контроллер Administration с параметром id, переданным при нажатии кнопки просмотра специальности.

В действии Speciality с помощью метода Where, указав переданный параметр id, выберем и запишем в соответствующие переменные список модулей специальностей, практик и студентов, запишем переменные в свойства анонимного класса и передадим в метод View.

Аналогично основной странице администратора для действия Speciality добавим представление с пустой моделью, а для каждого списка добавим частичные представления с таблицами и кнопками действий.

Список возможных действий для каждой сущности на странице специальности:

1. Для сущности «Модуль дисциплины»:
 - a. Добавление модуля дисциплины.
 - b. Редактирование кода и названия.
 - c. Удаление модуля дисциплины.
2. Для сущности «Практика»:
 - a. Добавление названия практики.
 - b. Редактирование названия практики.
 - c. Удаление названия практики.
3. Для сущности студент:
 - a. Добавление ФИО студента.
 - b. Редактирование ФИО студента.
 - c. Удаление студента.

Для каждого действия в представлении Speciality добавим формы и действия в контроллере, и в действии Speciality добавим необходимые списки для выбора в коллекцию ViewBag.

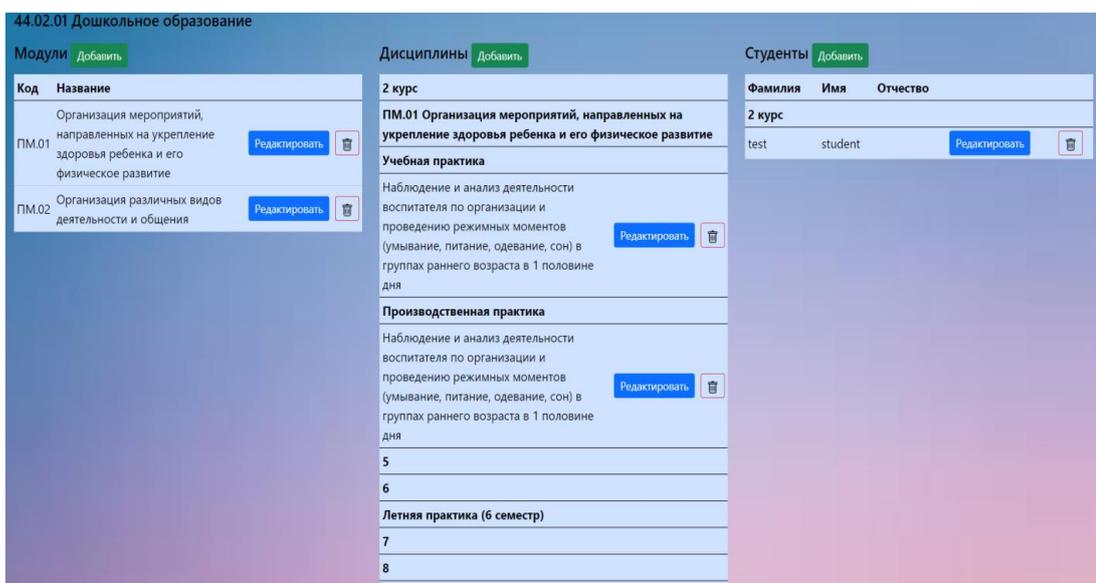


Рисунок 24 – Страница специальности

7. Страница успеваемости (рисунок 26).

После авторизации пользователя с ролью «Преподаватель», автоматически будет открыта страница успеваемости. Формирование содержимого страницы зависит от выбора параметров на форме в верхней части страницы. На рисунке 25 представлена форма выбора параметров для открытия страницы успеваемости.

Рисунок 25 – Форма выбора параметров

Список учебных заведений, доступных пользователю в зависимости от идентификатора и роли, хранящихся в сессии приложения, будет передан в коллекции ViewBag из действия Index контроллера Examination.

При выборе значения из списка учебных заведений будет вызвана javascript функция из файла site.js для заполнения списка специальностей с помощью вызова действия в контроллере, переданного в параметрах функции. Аналогично заполняется список семестров при выборе специальности.

После выбора параметров и нажатия кнопки «Показать» будет повторно вызвано действие `Index`, но уже с параметрами, указанными на форме.

В модели с помощью методов `Include` и `Where` будет передан список оценок по практикам за выбранный семестр по выбранной специальности для студентов, обучающихся на соответствующем семестру курсе или список практик, если студенты не были ранее добавлены на странице специальности.

В представлении `Index` контроллера `Examination` с помощью цикла `foreach` и метода `GroupBy` у модели сгруппируем оценки по модулям дисциплин и видам практик. Для получившегося сгруппированного набора создадим частичное представление `Examination` и выведем с помощью метода `Html.RenderPartialAsync`.

Для вывода оценки в представлении `Examination` создадим частичное представление `Mark` с полем отображения оценки и скрытым полем редактирования и добавим его в представление `Examination` с помощью `html` тэга `partial`, указав имя представления и модель.

Режим редактирования оценки вызывается с помощью `javascript` функции `beginEdit` при двойном нажатии левой кнопкой мыши по полю отображения.

Сохранение изменения оценки происходит с помощью `javascript` функции `endEdit`, вызываемой при снятии фокуса с поля ввода.

8. Добавление студентов.

При отсутствии предварительно добавленных студентов их можно добавить прямо в таблицу успеваемости, нажав кнопку «+» в любой из таблиц успеваемости.

ГБПОУ «Саткинский горно-керамический колледж им. А.К.Савина»			
Отчет по учебной и производственной практике по профессиональным модулям:			
ПМ.01. Организация мероприятий, направленных на укрепление здоровья ребенка и его физическое развитие			
Курс: 2			
Специальность: 44.02.01 Дошкольное образование			
1. Виды и объем работ, оценка за выполненные обучающимся работы во время учебной практики по ПМ.01 Организация мероприятий, направленных на укрепление здоровья ребенка и его физическое развитие			
		test student	+
Наблюдение и анализ деятельности воспитателя по организации и проведению режимных моментов (умывание, питание, одевание, сон) в группах раннего возраста в 1 половине дня	5		5
	5		
2. Виды и объем работ, оценка за выполненные обучающимся работы во время производственной практики по ПМ.01 Организация мероприятий, направленных на укрепление здоровья ребенка и его физическое развитие			
		test student	+
Наблюдение и анализ деятельности воспитателя по организации и проведению режимных моментов (умывание, питание, одевание, сон) в группах раннего возраста в 1 половине дня			

Скачать

Рисунок 26 – Страница успеваемости

9. Скачивание отчета.

Для скачивания отчета в формате PDF на странице успеваемости необходимо нажать кнопку «Скачать» внизу страницы.

Скачиваемый отчет формируется с помощью открытого генератора отчетов FastReport OpenSource, добавленного в проект с помощью менеджера пакетов NuGet. Макет отчета создан с помощью дизайнера отчетов FastReport Community и хранится в папке проекта. Макет отчета представлен на рисунке 27.

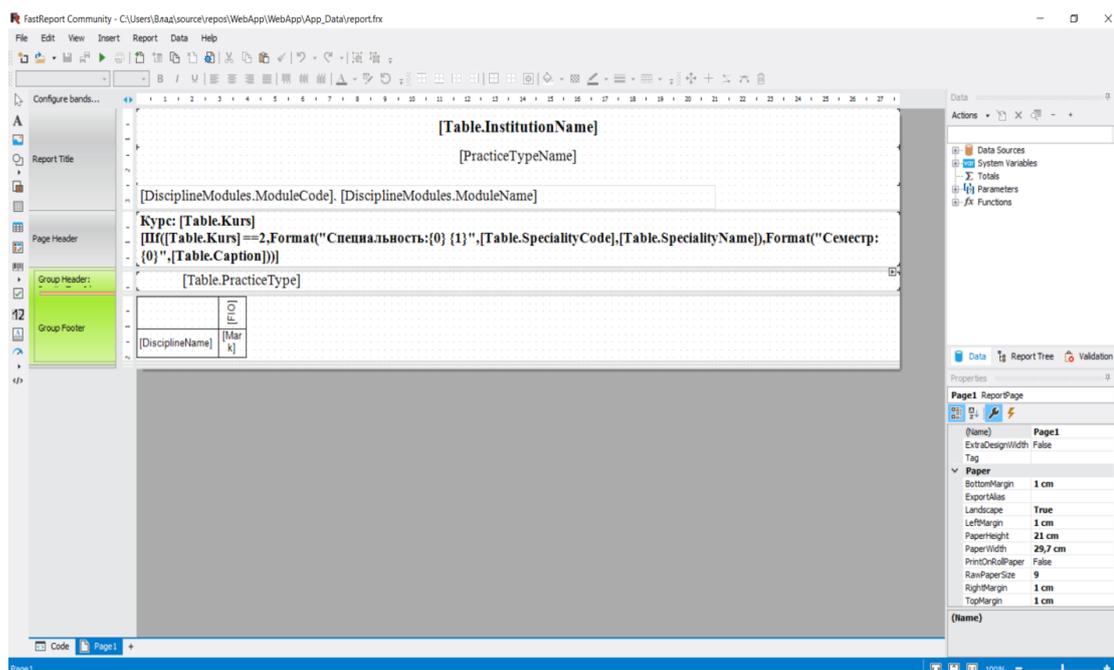


Рисунок 27– Макет отчета успеваемости

Нажатие кнопки «Скачать» вызывает действие Export контроллера Examination с параметрами, выбранными на форме страницы успеваемости.

Действие Export передает в макет отчета параметры и строку подключения к базе данных, после чего отчет будет подготовлен и сохранен в формат PDF для скачивания. PDF-файл отчета успеваемости представлен на рисунке 28.

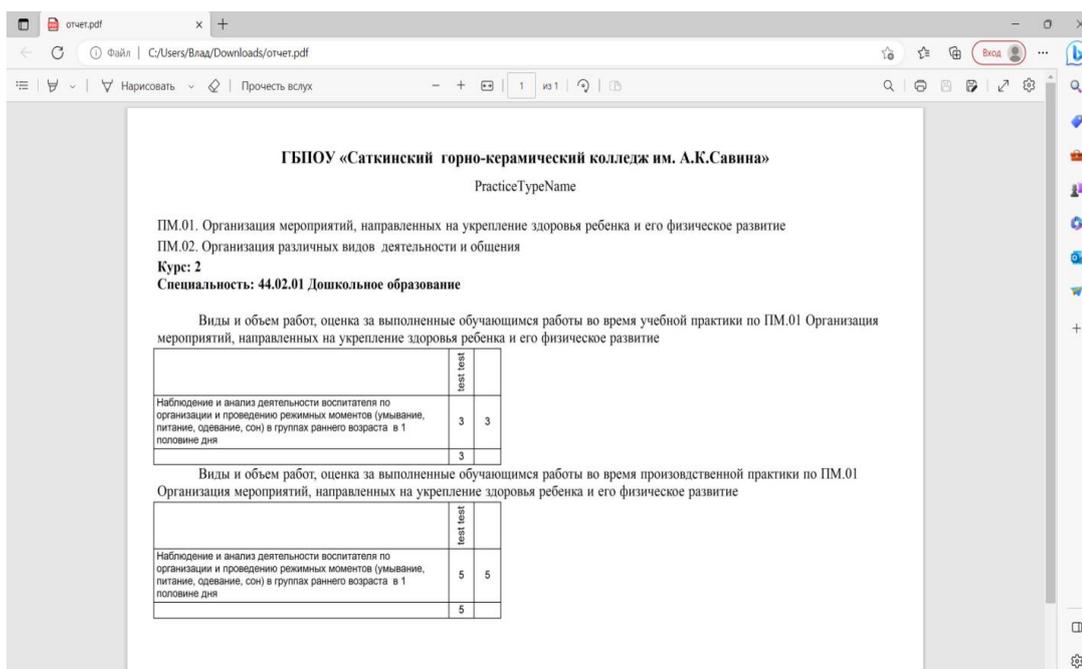


Рисунок 28 – PDF-файл отчета успеваемости

10. Публикация и хостинг приложения

Публикация приложения для развертывания на сервере происходит с помощью функции «Опубликовать» Visual Studio. Профиль публикации зависит от целевой платформы развертывания приложения. Для демонстрации проекта проект будет опубликован как папка с файлами для загрузки на сервер хостинга. Для развертывания приложения необходимы следующие условия:

1. Веб-сервер с поддержкой приложений на платформе .net 6.0 (пакет поддержки для ОС Windows и Linux доступен на сайте Microsoft).
2. Microsoft SQL Server 2016 и выше.

3. Настройка подключения к серверу в файле конфигурации приложения.
4. Регистрация домена при необходимости работы по сети интернет.

Выводы по второй главе

Во второй главе квалификационной работы было рассмотрено проектирование автоматизированной системы для создания отчета по учебной и производственной практике. Выделены средства разработки, такие как Microsoft Visual Studio 2022 Community, SQL Server 2016 Express, Microsoft SQL Server Management Studio, Entity Framework, а также: язык программирования C#, язык гипертекстовой разметки страницы HTML, декларативный язык программирования SQL, язык скриптов JavaScript, язык программирования PHP, для визуального отображения сайта был использован язык программирования CSS.

Также было выполнено описание модели данных. Представлена схема, в которой описаны все поля таблиц, рассмотрены связи между таблицами. После этого было выполнено описание основных функций разрабатываемой системы.

ГЛАВА 3. ДОКУМЕНТИРОВАНИЕ ГОТОВОГО ПРОГРАММНОГО ПРОДУКТА

3.1 Испытания автоматизированной системы

Испытание автоматизированной системы по формированию отчета по учебной и производственной практике можно разбить на несколько этапов (модулей).

Первый этап заключается в проверке системы на различных браузерах. Все должно открываться и отображаться без проблем.

На втором этапе необходимо испытать систему авторизации. Для этого необходимо войти в систему введя свой «Логин» и «Пароль». Всех пользователей в системе регистрирует администратор. Если пользователь ввел все данные правильно, то перед ним открылась главная страница, иначе система попросит ввести корректные данные.

На третьем этапе необходимо испытать системы на всевозможные ошибки.

При роли «Администратор» должны выполняться следующие действия:

1. Регистрация новых пользователей.
2. Внесение списков, обучающихся.
3. Добавление/редактирование учебного заведения.
4. Добавление/редактирование специальностей, дисциплин.
5. Выбор любого отчета для просмотра.
6. Просмотр отчета.
7. Скачивание в формате PDF.

При роли «Преподаватель» должны выполняться следующие действия:

1. Открытие любого предложенного отчета.
2. Добавление, непосредственно уже в саму таблицу отчета новых студентов.
3. Выставление оценок по модулям.

4. Скачивание в формате PDF.

Таким образом, после тестирования системы ошибок не выявлено. Разработанная автоматизированная система по формированию отчета по учебной и производственной практике работает исправно.

3.2 Руководство пользователей информационной системы

Для работы в системе всем пользователям необходимо сначала пройти авторизацию (рисунок 29). Для этого необходимо в открывшемся окне корректно ввести «Логин» и «Пароль». Если пользователь еще не зарегистрирован в системе, то ему нужно обратиться к администратору.

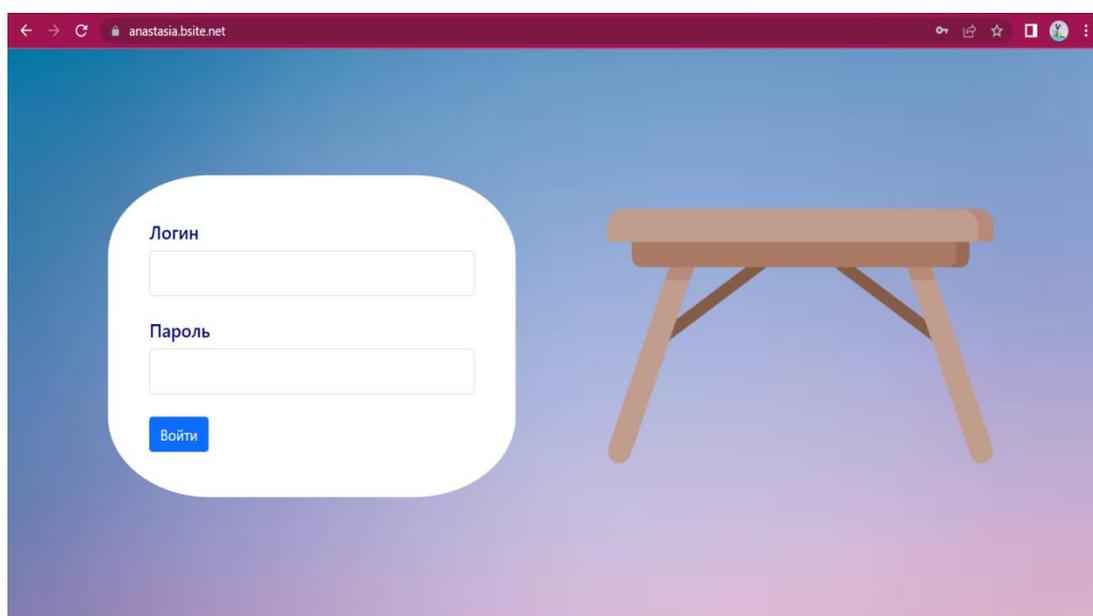


Рисунок 29 – Страница авторизации

3.2.1 Руководство для администратора системы

После входа в систему откроется окно «Администрирование» (рисунок 30.)

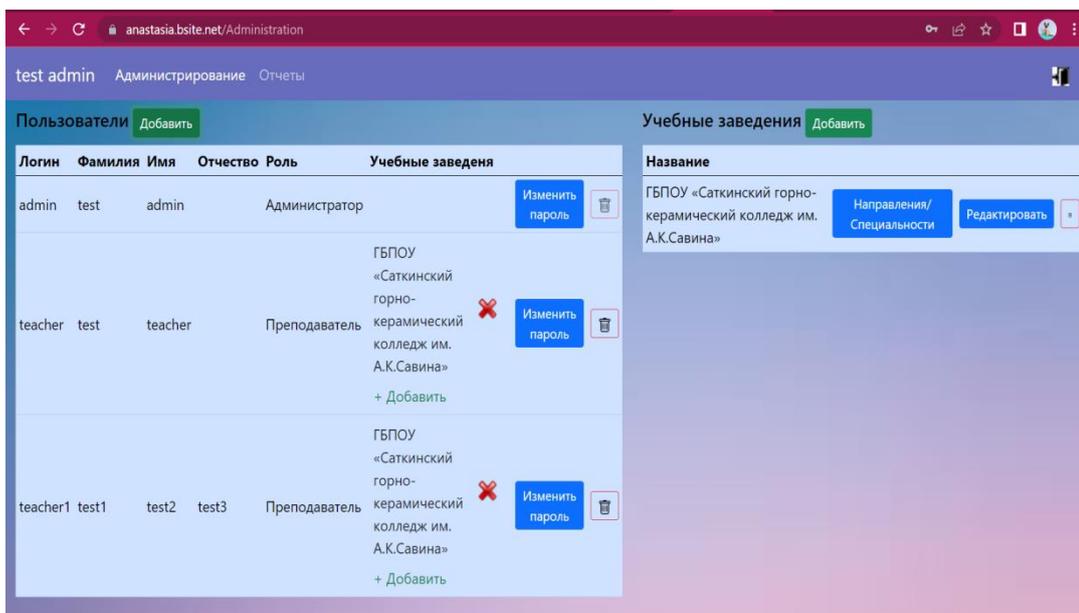


Рисунок 30 – Окно «Администрирование»

Далее администратор может выполнять следующие действия:

1. Добавление пользователей (рисунок 31).

Для этого необходимо нажать на кнопку «Добавить» и заполнить поля: логин; пароль; фамилия; имя; отчество. А также выбрать роль: администратор или учитель.

Новый пользователь ×

Логин

Пароль

Фамилия

Имя

Отчество

Роль

▼

Рисунок 31 – Добавление пользователей

2. Добавление учебного заведения (рисунок 32).

Для этого необходимо нажать на кнопку «Добавить» и заполнить поле «Название».

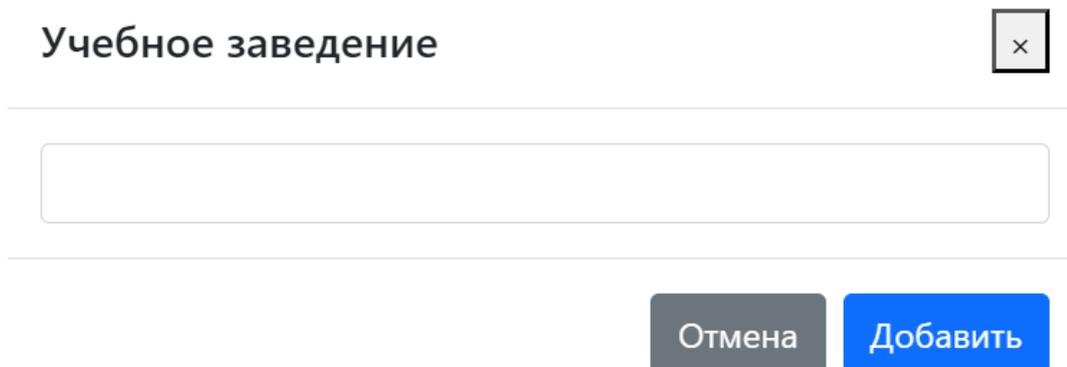


Рисунок 32 – Добавление учебного заведения

3. Редактирование уже имеющегося названия учебного заведения (рисунок 33).

Если в окне «Учебные заведения» уже есть название учебного заведения, то его можно отредактировать. Для этого необходимо нажать кнопку «Редактировать» и исправить название.

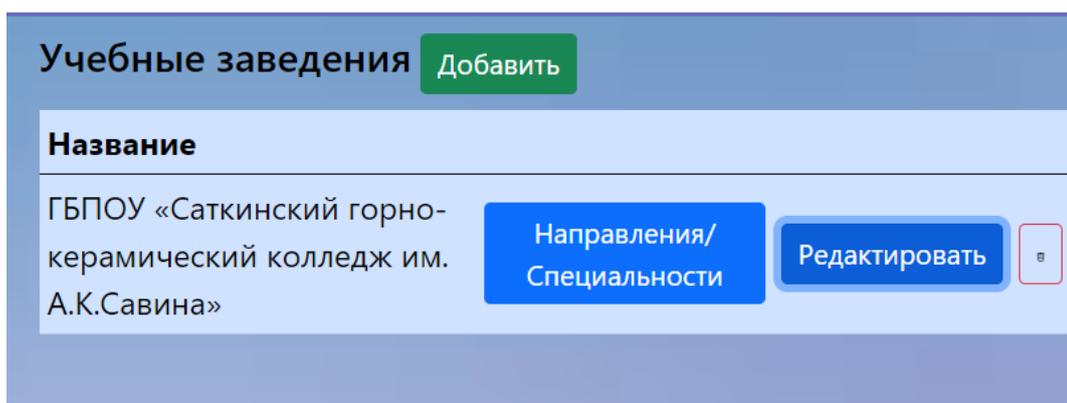


Рисунок 33 – Редактирование уже имеющегося названия учебного заведения

Так же учебное заведение можно полностью удалить.

4. Добавление направления/специальности (рисунок 35).

Для этого необходимо нажать на кнопку «Направления/Специальности». После этого список раскроется (рисунок 34).

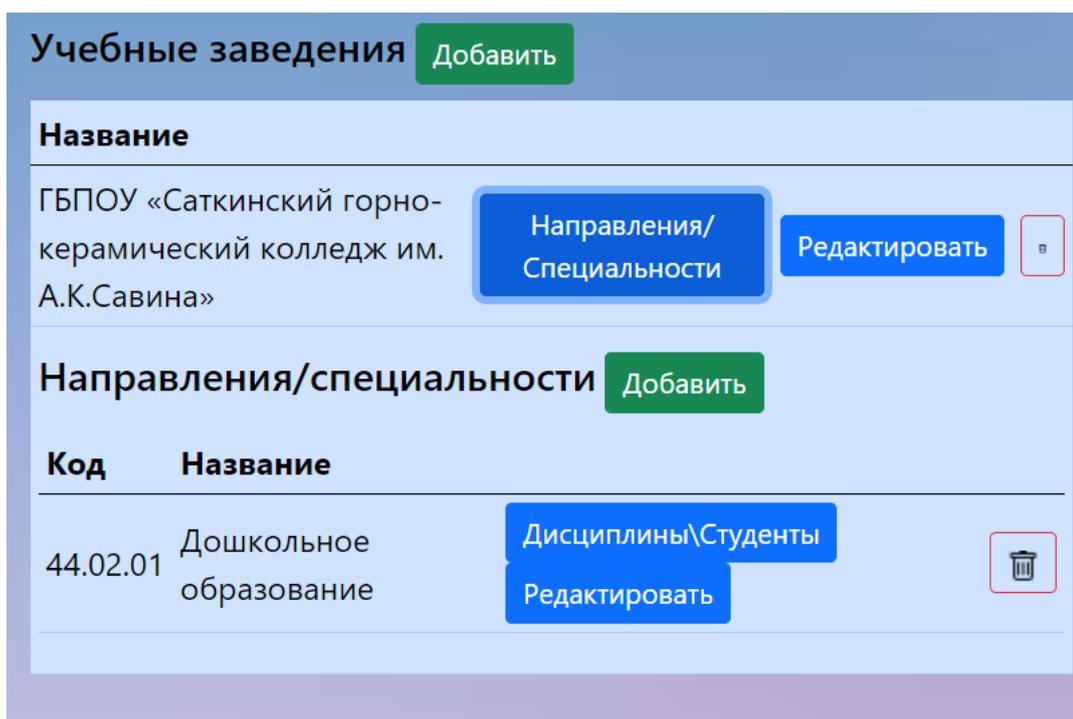


Рисунок 34 – Направление/специальности

Далее администратор может добавить новое направление/ специальность. Для этого необходимо нажать на кнопку «Добавить» и заполнить поля: код; название.

Направление/специальность ✕

Код

Название

Рисунок 35 – Добавление направления/специальности

Также есть возможность отредактировать уже имеющееся направление/специальность.

5. Добавление дисциплин и студентов.

Для перехода к добавлению необходимо нажать на кнопку «дисциплины/студенты». Откроется окно для добавления (рисунок 36).

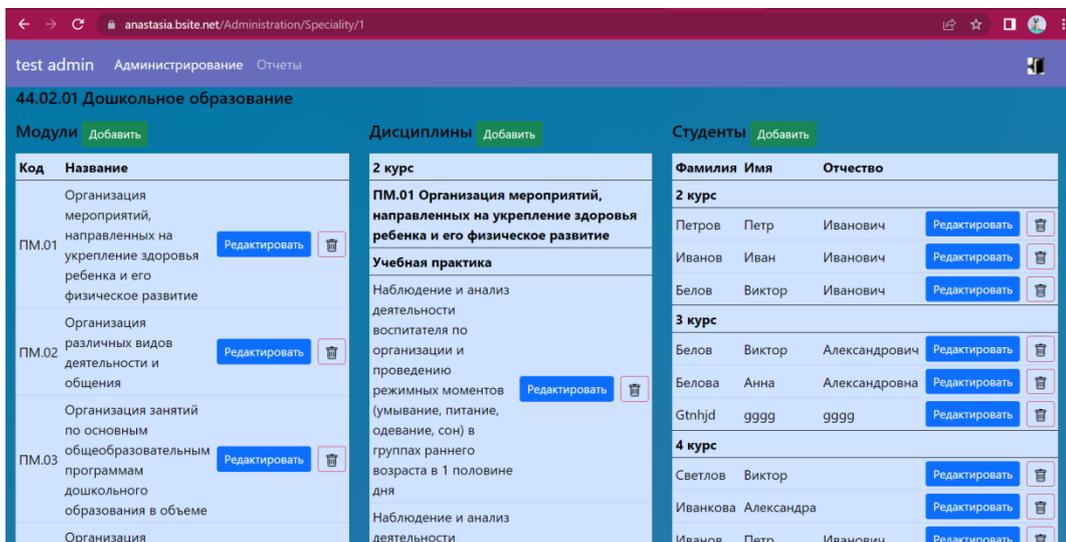


Рисунок 36 – Окно добавления дисциплин и студентов

Чтобы добавить модуль, необходимо нажать кнопку «Добавить» и заполнить поля: код, название.

Чтобы добавить дисциплину, необходимо нажать кнопку «Добавить» и заполнить поле «Название». И выбрать модуль, семестр, вид практики.

Чтобы добавить студента, необходимо нажать кнопку «Добавить» и заполнить поля: фамилия, имя, отчество. Выбрать курс.

Названия модулей, дисциплин, а также данные о студентах можно редактировать.

6. Просмотр отчетов.

Для этого необходимо перейти на вкладку отчет и выбрать: учебное заведение, направление/специальность, семестр (рисунок 37).

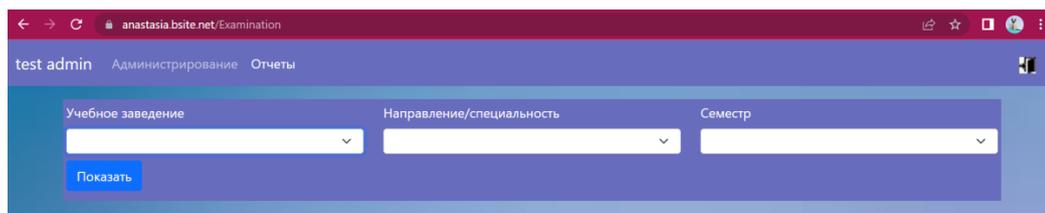


Рисунок 37 – Вкладка отчеты

Далее открывается непосредственно сам отчет (рисунок 38).

ГБПОУ «Саткинский горно-керамический колледж им. А.К.Савина»

Отчет по учебной и производственной практике по профессиональным модулям:

ПМ.01. Организация мероприятий, направленных на укрепление здоровья ребенка и его физическое развитие
 ПМ.02. Организация различных видов деятельности и общения
 ПМ.03. Организация занятий по основным общеобразовательным программам дошкольного образования в объеме

Курс: 2
 Специальность: 44.02.01 Дошкольное образование

1. Виды и объем работ, оценка за выполненные обучающимся работы во время учебной практики по ПМ.01
 Организация мероприятий, направленных на укрепление здоровья ребенка и его физическое развитие

	Петров Петр Иванович	Иванов Иван Иванович	Белов Виктор Иванович	
Наблюдение и анализ деятельности воспитателя по организации и проведению режимных моментов (умывание, питание, одевание, сон) в группах раннего возраста в 1 половине дня	5	3	3	4
Наблюдение и анализ деятельности воспитателя по организации и проведению режимных моментов (умывание, питание, одевание, сон) в группах раннего возраста во 2 половине дня	5	5	3	4

Рисунок 38 – Отчет

Редактировать оценки администратор не может.

7. Скачивание отчета.

Для этого необходимо снизу страницы нажать на кнопку «Скачать» (рисунок 39).

6. Виды и объем работ, оценка за выполненные обучающимся работы во время производственной практики по ПМ.03 Организация занятий по основным общеобразовательным программам дошкольного образования в объеме

	Петров Петр Иванович	Иванов Иван Иванович	Белов Виктор Иванович	
Проведение занятия по развитию речи детей раннего возраста				

Скачать

Рисунок 39 – Кнопка «Скачать»

3.2.2 Руководство для преподавателей

После входа в систему преподавателю доступно окно «Успеваемость» (рисунок 40) для заполнения отчетов по практике.

test teacher Успеваемость

Учебное заведение Направление/специальность Семестр

Показать

Рисунок 40 – Окно «Успеваемость»

Далее необходимо выбрать учебное заведение, направление/специальность и семестр для перехода к работе с отчетом (рисунок 41).

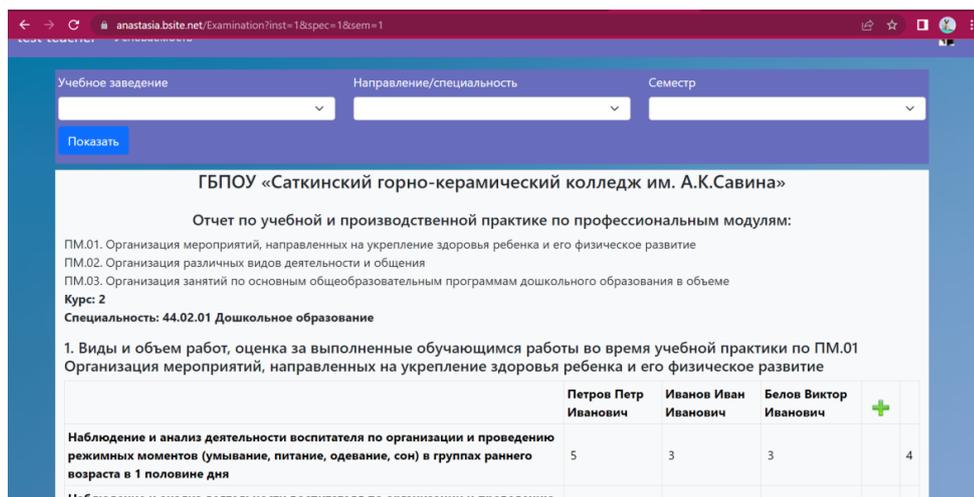


Рисунок 41 – Отчет

Для того, чтобы поле стало доступным необходимо нажать на ячейку двойным щелчком левой кнопкой мыши. Далее оценки можно ввести вручную, а также выбрать из списка (рисунок 42).

	Петров Петр Иванович	Иванов Иван Иванович	Белов Виктор Иванович	+
Наблюдение и анализ деятельности воспитателя по организации и проведению режимных моментов (умывание, питание, одевание, сон) в группах раннего возраста в 1 половине дня	5	3	3	4
Наблюдение и анализ деятельности воспитателя по организации и проведению режимных моментов (умывание, питание, одевание, сон) в группах раннего возраста во 2 половине дня	5	5	3	4
Наблюдение и анализ деятельности воспитателя по организации и проведению утренней гимнастики в группах раннего возраста	5	4	3	4
Составление конспекта утренней гимнастики для детей раннего возраста	4	3	3	3
Наблюдение и анализ деятельности воспитателя по организации и проведению подвижных игр в группах раннего возраста.	4	4	3	4

Рисунок 42 – Выставление оценок

У пользователя «Преподаватель» есть возможность добавить студентов сразу в таблицу. Для этого необходимо нажать на «+» и заполнить поля: фамилия, имя, отчество.

Для скачивания отчета необходимо внизу страницы нажать кнопку «Скачать» (рисунок 43).

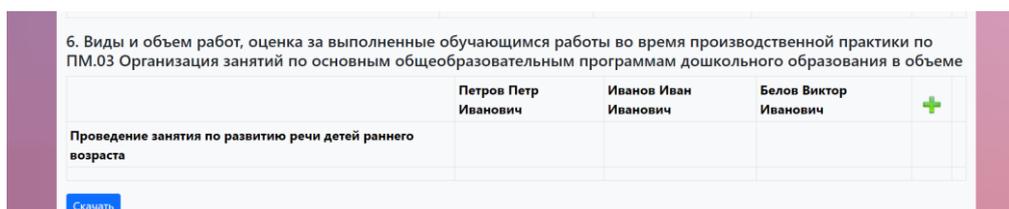


Рисунок 43 – Скачивание отчета

3.3 Техничко-экономическое обоснование разработки информационной системы

Техничко-экономическое обоснование (ТЭО) – это анализ, расчет, оценка экономической целесообразности осуществления предлагаемого проекта, в данном случае – проекта по разработке автоматизированной системы по формированию отчета по учебной и производственной практике. ТЭО основано на сопоставительной оценке затрат и результатов, установлении эффективности использования, срока окупаемости вложений.

1. Расчет затрат на оплату труда представлен в таблице 1.

Таблица 1 – Расчет затрат на оплату труда

№ п/п	Показатель	Единица измерения	Величина затрат
1	Затраты времени исполнителя на разработку	Дни	50
2	Ставка дневной заработной платы исполнителя	Руб.	420
3	Величина заработной платы исполнителя за выполнение разработки	Руб.	25000
4	Ставка страхового взноса в Пенсионный фонд[12]	%	22
5	Ставка страховых взносов в Фонд социального страхования	%	2,9
6	Ставка страховых взносов в Федеральный фонд обязательного медицинского страхования	%	5,1
7	Совокупный процент ставки страховых взносов	%	30
8	Величина страховых взносов	Руб.	7500
9	Страховые тарифы на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний	%	0,2
10	Отчисления на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний	Руб.	50
11	Общая сумма страховых взносов	Руб.	7550
12	Суммарные затраты на оплату труда с учетом страховых взносов	Руб.	32550

2. Расчет затрат на материалы, приобретаемые для выполнения разработки представлен в таблице 2.

Таблица 2 – Расчет затрат на материалы

№ п/п	Наименование	Единица измерения	Количество	Цена за единицу, руб.	Сумма затрат, руб.
1	Бумага формата А4	Лист	150	5,5	550
2	Ручка шариковая (синяя)	Шт.	3	21	63
	ИТОГО				613

3. Расчет прочих прямых затрат представлен в таблице 3.

Таблица 3 – Расчет прочих затрат

№ п/п	Наименование	Единица измерения	Количество	Цена за единицу, руб.	Сумма затрат, руб.
1	Интернет	Месяц	4	250	1000
2	Бензин	Литры	80	42	3360
	ИТОГО				4360

4. Расчет полной себестоимости на разработку программного продукта представлена в таблице 4.

Таблица 4 – Расчет полной себестоимости системы

№ п/п	Показатель	Единица измерения	Величина затрат
1	2	3	4
1	Заработная плата исполнителей с учетом страховых взносов	Руб.	32550
2	Затраты на материалы	Руб.	613
3	Прочие прямые затраты	Руб.	4360
4	Полная себестоимость проекта	Руб.	37523
5	Средний уровень прибыльности (рентабельности) проектов	%	22,5
6	Планируемый размер прибыли	Руб.	8442
7	Планируемая договорная цена разработки	Руб.	50000
8	Фактическая цена разработки программного продукта	Руб.	45 965

Выводы по третьей главе

В третьей главе квалификационной работы были проведены испытания модулей системы: авторизации, администратора, преподавателя. После тестирования системы ошибок не выявлено. Разработанная автоматизированная система работает исправно.

Также были разработаны подробные руководства для пользования, которые предоставят пользователям помощь в использовании данной системы. Руководства сопровождаются иллюстрациями.

В завершении было проведено технико-экономическое обоснование разработки автоматизированной системы по формированию отчета по учебной и производственной практике. Исходя из всех расчетов фактическая стоимость разработки данной системы составила 45 965 рублей.

ЗАКЛЮЧЕНИЕ

В ходе выполнения квалификационной работы была проанализирована организационная структура ГБПОУ «Саткинский горно-керамический колледж им. А.К. Савина». Сформулированы цели, назначение и требования к разрабатываемой автоматизированной системе.

Рассмотрены средства для разработки веб-приложения, а также описана модель данных и функции системы.

Автоматизированная система прошла испытания и была успешно внедрена в ГБПОУ «Саткинский горно-керамический колледж им. А.К. Савина».

В завершении было проведено технико-экономическое обоснование, демонстрирующее экономическую целесообразность разработки автоматизированной системы по формированию отчета по учебной и производственной практике.

В ходе выполнения работы цель достигнута, задачи выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГБПОУ «Саткинский горно-керамический колледж имени А.К. Савина»: официальный сайт. – Сатка. – URL: <https://sgkk.ru/> (дата обращения: 01.02.2023г).
2. Методология функционального моделирования IDEF0 / руководящий документ. – Москва: Госстандарт России, 2000. – 75 с. – URL: <https://nsu.ru/smk/files/idef.PDF> (дата обращения: 22.02.2023). – Текст: электронный.
3. Продукты Майкрософт»: официальный сайт. – Microsoft. URL: <https://visualstudio.microsoft.com/ru/vs/> (дата обращения: 01.02.2023г).
4. Фирма «1С»: официальный сайт. – URL: <https://1c.ru/> (Дата обращения: 10.04.2023г).
5. Продукты Майкрософт: // Visual Studio 2022: [сайт]. – 2022. – URL: <https://www.microsoft.com/ru-ru> (дата обращения: 10.04.2023г).
6. Бизнес-процессы: языки моделирования, методы, инструменты / Ф. Шёнталер [и др.]. — Москва : Альпина Паблишер, 2019. — 264 с. — ISBN 978-5-9614-2022-7. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/124474.html> (дата обращения: 20.05.2023). — Режим доступа: для авторизир. пользователей
7. Биллиг В.А. Основы объектного программирования на С# (С# 3.0, Visual Studio 2008) : учебник / Биллиг В.А.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 409 с. — ISBN 978-5-4497-0880-9. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102029.html> (дата обращения: 01.03.2023). — Режим доступа: для авторизир. пользователей
8. Дунаев, Вадим Сценарии для Web-сайта. PHP и JavaScript / Вадим Дунаев. - М.: "БХВ-Петербург", 2012. - 576 с. Кариев Ч.А. Разработка Windows-приложений на основе Visual С# : учебное пособие / Кариев Ч.А.. — Москва : Интернет-Университет Информационных Технологий (ИН-

ТУИТ), Ай Пи Ар Медиа, 2021. — 978 с. — ISBN 978-5-4497-0909-7. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102057.html> (дата обращения: 10.03.2023). — Режим доступа: для авторизир. пользователей

9. Кириченко А. Справочник HTML. Кратко, быстро, под рукой / А. Кириченко, Е. Дубовик. – Санкт-Петербург: Наука и Техника, 2021. – 288 с. – ISBN 978-5-94387-275-4.

10. Мартишин, С.А. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для применения проектирования информационных систем. Учебное пособие. Гриф МО РФ / С.А. Мартишин. - М.: Инфра-М, Форум, 2017. - 82 с

11. Островский, К. А. Типы требований к Web-приложению для обработки экспериментальных данных / К. А. Островский. — Текст : непосредственный // Молодой ученый. — 2010. — № 5 (16). — Т. 1. — С. 101-103. — URL: <https://moluch.ru/archive/16/1568/> (дата обращения: 24.01.2023).

12. Пенсионный фонд Российской Федерации: официальный сайт. Обновляется в течение суток. — URL: http://www.pfrf.ru/sved_o_per_pens_nak/uplata_str_vz/ (дата обращения: 10.05.2023). — Текст: электронный.

13. Поляков, Е. В. PHP на примерах: учебное пособие / Е. В. Поляков. — Санкт-Петербург: Наука и Техника, 2017. — 256 с. — ISBN 978-5-94387-733-9. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/101553> (Дата обращения: 10.04.2022г).

14. Страховые взносы, тарифы, ставки, размеры: официальный сайт. — Москва, 2011 — . — URL: <https://www.glavbukh.ru/rubrika/157> (дата обращения: 16.04.2023). — Текст: электронный

15. Флойд К.С. Введение в программирование на PHP5 : учебное пособие / Флойд К.С.. — Москва : Интернет-Университет Информационных

Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 280 с. — ISBN 978-5-4497-0886-1. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/101998.html> (дата обращения: 20.02.2023). — Режим доступа: для авторизир. пользователей

16. Фонд социального страхования Российской Федерации: официальный сайт. — Москва, 2005 — . — URL: http://fss.ru/ru/fund/activity/accident_insurance/index.shtml (дата обращения: 10.05.2022). — Текст: электронный.