

Уважаемый пользователь!

Обращаем ваше внимание, что система Антиплагиус отвечает на вопрос, является тот или иной фрагмент текста заимствованным или нет. Ответ на вопрос, является ли заимствованный фрагмент именно плагиатом, а не законной цитатой, система оставляет на ваше усмотрение.

## Отчет о проверке № 9132935

Дата загрузки: 2025-02-10 00:07:16  
Пользователь: [ivan.suhonenkov@gmail.com](mailto:ivan.suhonenkov@gmail.com), ID: 9132935

Отчет предоставлен сервисом «Антиплагиат»  
на сайте [antiplagius.ru/](https://antiplagius.ru/)

### Информация о документе

№ документа: 9132935  
Имя исходного файла: Пояснительная записка.docx  
Размер файла: 2.58 МБ  
Размер текста: 44593  
Слов в тексте: 6173  
Число предложений: 1001

### Информация об отчете

Дата: 2025-02-10 00:07:16 - Последний готовый отчет  
Оценка оригинальности: 96%  
Заимствования: 4%



### Источники:

Доля в тексте	Ссылка
16.70%	<a href="https://appmaster.io/ru/blog/vidy-testirovaniia-programmnogo-obe...">https://appmaster.io/ru/blog/vidy-testirovaniia-programmnogo-obe...</a>
14.70%	<a href="https://www.zaptest.com/ru/%D0%B4%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%...">https://www.zaptest.com/ru/%D0%B4%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%...</a>
12.50%	<a href="https://domajisto.ru/blog/principy-raboty-orm-osnovy-obektno-rel...">https://domajisto.ru/blog/principy-raboty-orm-osnovy-obektno-rel...</a>
10.80%	<a href="https://ru.hexlet.io/blog/posts/vidy-testirovaniya">https://ru.hexlet.io/blog/posts/vidy-testirovaniya</a>
10.30%	<a href="https://tquality.ru/blog/staticheskoe-i-dinamicheskoe-testirovan...">https://tquality.ru/blog/staticheskoe-i-dinamicheskoe-testirovan...</a>
9.40%	<a href="https://mate.academy/dev-blog/ru/qa-ru/static-and-dynamic-testin...">https://mate.academy/dev-blog/ru/qa-ru/static-and-dynamic-testin...</a>
9.20%	<a href="https://peredelka38.ru/d/kak-rabotaet-orm">https://peredelka38.ru/d/kak-rabotaet-orm</a>
8.10%	<a href="https://adoneast.ru/chto-takoe-orm-java.html">https://adoneast.ru/chto-takoe-orm-java.html</a>
8.00%	<a href="https://appttractor.ru/info/articles/chto-takoe-arhitektura-prilo...">https://appttractor.ru/info/articles/chto-takoe-arhitektura-prilo...</a>
7.30%	<a href="https://multiurok.ru/index.php/files/tema-uroka-zashchita-inform...">https://multiurok.ru/index.php/files/tema-uroka-zashchita-inform...</a>

### Информация о документе:

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ федеральное государственное бюджетное образовательное учреждение высшего образования "Российский экономический университет им. Г.В. Плеханова" Московский приборостроительный техникум Курсовой проект ПМ 11 Разработка, администрирование и защита баз данных МДК 11.01 "Технология разработки и защиты баз данных" Специальность 09.02.07 "Информационные системы и программирование" Квалификация: Программист Тема: "Разработка информационной системы по учету посещаемости и успеваемости" Пояснительная записка Листов: Руководитель \_\_\_\_\_ / Е.Ю. Бойцова "\_\_\_\_\_" \_\_\_\_\_ 2024 год Исполнитель \_\_\_\_\_ / И.А. Сухоненков "\_\_\_\_\_" \_\_\_\_\_ 2025 год 2025 СОДЕРЖАНИЕ ВВЕДЕНИЕ 4 1. ОБЩАЯ ЧАСТЬ 5 1.1. Цель разработки 5 1.2. Средства разработки 5 2. СПЕЦИАЛЬНАЯ ЧАСТЬ 8 2.1. Постановка задачи 8 2.1.1. Входные данные предметной

области 9 2.1.2. Выходные данные предметной области 9 2.1.3. Требования к проекту 10 2.2. Внешняя спецификация 12 2.2.1. Описание задачи 19 2.2.2. Входные и выходные данные 22 2.2.3. Методы 27 2.2.4. Тесты 30 2.2.5. Контроль целостности данных 31 2.3. Проектирование 35 2.3.1. Схема архитектуры приложения 35 2.3.2. Логическая схема данных 37 2.3.3. Физическая схема данных 38 2.3.4. Структурная схема 44 2.3.5. Функциональная схема 48 2.3.6. Диаграмма классов 49 2.3.7. Схема тестирования 52 2.3.8. Схема пользовательского интерфейса 52 2.4. Результат работы программы 53 3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 60 3.1. Инструментальные средства 60 3.2. Отладка программы 61 3.3. Защитное программирование 62 3.4. Характеристики программы 64 ЗАКЛЮЧЕНИЕ 65 СПИСОК ИСПОЛЪЗУЕМЫХ МАТЕРИАЛОВ 67 ПРИЛОЖЕНИЕ А. Текст программы ПРИЛОЖЕНИЕ Б. Сценарий и результаты тестовых испытаний ПРИЛОЖЕНИЕ В. Руководство пользователя ПРИЛОЖЕНИЕ Г. Скрипт базы данных ВВЕДЕНИЕ В настоящее время цифровые технологии активно трансформируют все сферы образования, и одним из важных направлений становится автоматизация процессов учёта посещаемости и успеваемости. Традиционные методы, при которых учителя вручную ведут журналы, занимают много времени и нередко приводят к задержкам с предоставлением информации. Чтобы решить эти проблемы, была разработана информационная система, выступающая электронным аналогом бумажного журнала, где реализовано удобное взаимодействие между учителем и учеником. В основе проекта лежит фреймворк Flask, который обеспечивает гибкость и простоту разработки, а также легко расширяемую архитектуру. Система позволяет учителям выставлять оценки, фиксировать посещаемость и просматривать общую статистику по каждому ученику. Ученик, в свою очередь, может в реальное время получать доступ к своим баллам и другим важным данным об успеваемости, что повышает прозрачность оценочного процесса и стимулирует его вовлечённость в учебный процесс. Таким образом, данная разработка упрощает ведение учётных записей, снижает вероятность ошибок, характерных для бумажных журналов, и улучшает коммуникацию между всеми участниками образовательного процесса. В перспективе система может быть интегрирована с другими образовательными ресурсами и дополнена новыми функциями, способствуя дальнейшей цифровизации обучения.

1. ОБЩАЯ ЧАСТЬ 1.1. Цель разработки Цель разработки заключается в повышении эффективности процессов учёта посещаемости и успеваемости, а также в улучшении взаимодействия между учителем и учеником. Информационная система предназначена для автоматизации выставления оценок, контроля пропусков и оперативного анализа успеваемости, что помогает сократить временные затраты и повысить прозрачность образовательного процесса. Такая оптимизация позволяет учителям более эффективно управлять учебным процессом, а ученикам - своевременно отслеживать собственные результаты и принимать меры для их улучшения.

1.2. Средства разработки Для проектирования, разработки и тестирования веб приложения были использованы программные средства, представленные в Таблице 1.

№	Средства разработки	№	Тип средства	Название средства
1	Операционная система	Windows 11	Организация взаимодействия программ и пользователя	3. Инструментальные средства разработки программного решения
2	JetBrains	PyCharm	Разработка веб-приложения	5. Графическая оболочка
3	Admin4	Разработка базы данных	6. СУБД	PostgreSQL
4	Разработка базы данных	7. Браузер	Mozilla Firefox	Просмотр веб приложения

В таблице 2 представлены минимальные и рекомендованные технические средства, на базе которых возможно комфортное использование реализуемого программного обеспечения.

Таблица 2	Минимальные и рекомендованные технические средства	№	Тип оборудования	Наименование оборудования	
1	Минимальные технические требования	Веб-приложение	1	Размер экрана: 14,0"	
2	Разрешение экрана: 1366x768	3	Линейка процессора: Intel Core i3	4	Количество ядер процессора: 25
5	Оперативная память: 4	6	ГБ	Тип видеокарты: Встроенная	
7	Видеокарта: Intel HD Graphics / AMD аналог	8	Конфигурация накопителей: HDD	9	Общий объем всех накопителей: 128
10	ГБ	Рекомендуемые технические требования	Веб приложение	1	Размер экрана: 15,6"
2	Разрешение экрана: 1920x1080	3	Линейка процессора: Intel Core i5	4	Количество ядер процессора: 45
5	Оперативная память: 8	6	ГБ	Тип видеокарты: Встроенная или дискретная	
7	Видеокарта: NVIDIA GeForce GTX 1660Ti	8	Конфигурация накопителей: SSD	9	Общий объем всех накопителей: 512
10	ГБ	2. СПЕЦИАЛЬНАЯ ЧАСТЬ 2.1. Постановка задачи	Необходимо разработать веб-приложение, позволяющее автоматизировать процессы учёта посещаемости и успеваемости учащихся на основе клиент-серверной архитектуры с использованием фреймворка Flask и СУБД PostgreSQL. Система должна обеспечивать удобное взаимодействие для четырёх ролей пользователей: администратор, методист, учитель и ученик. Каждая из этих ролей обладает собственным набором прав, определяющим доступ к конкретным функциям и данным. Основная цель приложения заключается в упрощении ведения электронного журнала благодаря возможностям создания, чтения, обновления и удаления данных (CRUD-операции) о посещаемости и оценках. Помимо этого, система должна предоставлять инструменты для поиска, сортировки и фильтрации информации с целью быстрой выборки требуемых сведений и формирования отчётов. Предполагается наличие механизма для авторизации и регистрации с учётом распределения прав доступа, чтобы обеспечить безопасность и удобство работы различных категорий пользователей. Важной частью функционала является поддержка операций импорта и экспорта данных в форматах SQL и CSV, что позволит интегрировать приложение с другими системами и легко переносить необходимую информацию. Кроме того, требуется реализовать резервное копирование и восстановление базы данных для повышения надёжности, а также вести логирование действий, связанных с изменением или удалением записей, чтобы контролировать безопасность и отслеживать возможные ошибки. Ещё одним значимым компонентом является визуализация аналитических показателей, в частности наглядное отображение среднего балла по каждому предмету, что позволит быстро оценивать общую динамику успеваемости и выявлять проблемные области. Таким образом, разработанное веб-приложение станет эффективным инструментом для оптимизации		

учебного процесса, предоставляя структурированный и гибкий набор функций для всех пользователей, связанных с процессом учёта и анализа успеваемости и посещаемости.

2.1.1. Входные данные предметной области

Входные данные предметной области представляют собой совокупность сведений, необходимых для функционирования веб-приложения по учёту успеваемости и посещаемости. Все эти данные хранятся и обрабатываются в базе данных PostgreSQL. Основные категории входных данных включают:

- Аутентификационные данные пользователя: логин и пароль, используемые для авторизации в системе и разграничения прав доступа.
- Данные об учетных записях (User): фамилия и имя, дата регистрации, аватар (при необходимости), статус, а также ссылка на соответствующую роль и класс (для учеников).
- Данные о ролях (Role): наименование каждой роли и её уникальный идентификатор, позволяющие разграничивать функционал между администратором, методистом, учителем и учеником.
- Данные о классах (Class): уникальный идентификатор, наименование класса и связь с учениками, зарегистрированными в данном классе.
- Данные о предметах (Subject): уникальный идентификатор, название и количество академических часов (hours), предусмотренных учебным планом.
- Данные о занятиях (Lesson): идентификаторы связанного класса, учителя и предмета, а также дата проведения урока. Эта сущность необходима для формирования расписания и учёта проведённых занятий.
- Данные об оценках (Grade): идентификатор ученика, идентификатор занятия (урока), оценка или признак отсутствия на уроке. На основе этих данных в дальнейшем строится аналитика успеваемости.
- Данные для импорта и экспорта: файлы в форматах CSV или SQL, позволяющие переносить данные между разными системами, загружать новые списки пользователей, классов или предметов и создавать резервные копии базы данных.

Все перечисленные данные поступают в систему через формы веб-интерфейса или загружаются в виде отдельных файлов, а их валидация и сохранение осуществляются на стороне сервера.

2.1.2. Выходные данные предметной области

Выходные данные предметной области представляют результаты работы веб-приложения, связанные с учётом успеваемости и посещаемости учащихся. Эти данные формируются на основании записей в базе данных, пользовательских запросов и функций приложения. К основным категориям выходных данных относятся:

- Список пользователей с учетом их ролей и классов. Веб-приложение предоставляет структурированную информацию о зарегистрированных пользователях (администратор, методист, учитель, ученик), включая их ФИО, статус, принадлежность к определённому классу и другую справочную информацию.
- Информация об успеваемости. Система позволяет просматривать выставленные оценки, в том числе отдельные оценки по каждому занятию (уроку), общее количество оценок за определённый период и усреднённые показатели успеваемости по предметам или группам.
- Данные о посещаемости.
- Отчёты и аналитика. Приложение формирует график успеваемости.
- Результаты поиска, фильтрации и сортировки. По запросам пользователей система выводит выборки данных, соответствующие заданным критериям.
- Файлы экспорта. Приложение генерирует выходные файлы в форматах CSV или SQL, которые содержат данные об оценках ученика по конкретному предмету.
- Логи и статистика использования. При включённом механизме логирования система выдаёт записи об ошибках на сервере.

2.1.3. Требования к проекту

Для системы учёта посещаемости и успеваемости требуется обеспечить полноценный функционал с учётом интересов всех ролей, участвующих в процессе обучения: администраторов, методистов, учителей и учеников. Основное внимание уделяется сбору, хранению и обработке данных об учебном процессе, а также предоставлению статистических данных для анализа успеваемости и посещаемости. База данных должна содержать не менее восьми таблиц, отражающих основные сущности: пользователи, роли, классы, предметы, занятия (уроки), оценки, а также дополнительные таблицы для логирования и резервного копирования. БД необходимо нормализовать до третьей нормальной формы, чтобы минимизировать избыточность и обеспечить целостность данных. Все сущности должны поддерживать операции добавления, редактирования и удаления записей (CRUD). Пользовательский интерфейс приложения обязан предоставлять возможности сортировки, фильтрации и **поиска** данных, чтобы **упростить доступ** к информации об учениках, предметах, расписании, оценках и посещаемости. Для удобного обращения к повторяющимся выборкам данных или агрегированным результатам рекомендуется разработать представления (виртуальные таблицы), предоставляющие **быстрый** доступ к наиболее часто **используемой** информации. Для обеспечения безопасности данных необходимо реализовать шифрование паролей и использовать механизмы защиты от ошибок как на стороне сервера, так и на стороне клиента. Также система должна включать средства резервного копирования и восстановления базы данных, а результаты действий пользователей (создание, изменение или удаление записей, авторизация) должны логироваться для обеспечения контроля и аудита. Неотъемлемой частью функционала является возможность экспорта и импорта данных в форматах SQL и CSV, что существенно облегчает интеграцию приложения с другими системами и повышает удобство ведения электронного журнала. Система должна формировать статистические визуализации (графики) о ходе обучения ученика. Панель администратора **должна** включать:

- \* Регистрацию новых **пользователей** с **выбором ролей**;
- \* Просмотр списка пользователей
- \* Изменение данных пользователей.
- \* **Просмотр** логов **системы**;
- \* Создание резервного копирования базы данных.
- \* Управление собственной учетной записью.

Панель методиста должна включать:

- \* Создание и редактирование данных об уроках
- \* Создание и редактирование данных о классах
- \* Создание и редактирование данных о предметах
- \* Управление собственной учетной записью

Панель учителя должна включать:

- \* Возможность просмотра своих классов.
- \* Выставление и изменение оценок.
- \* Экспорт и импорт данных в форматах sql и csv.

Панель ученика должна включать:

- \* Просмотр своих оценок и пропусков;
- \* Возможность фильтрации, поиска и сортировки оценок.
- \* Возможность просмотра графика с статистикой по среднему баллу каждого предмета.

2.2. Внешняя спецификация

На рисунке 1 представлен контекстный уровень диаграммы IDEF0. Данная диаграмма отображает детальный анализ процессов управления посещаемостью и успеваемостью учеников. На ней указаны документы, регламентирующие данные процессы, исполнители и выходные документы. На первом уровне диаграммы представлена генеральная цель системы - обеспечение учета посещаемости и успеваемости учащихся. В рамках данного курсового проекта был проведён анализ существующих методов учёта посещаемости и успеваемости в образовательных организациях. До внедрения данной информационной системы все процессы велись вручную или с помощью разрозненных инструментов, что усложняло обработку данных, замедляло работу преподавателей и снижало прозрачность контроля

результатов. После интеграции новой системы удалось достичь генеральной цели - оптимизации процессов учёта посещаемости и успеваемости. Это привело к повышению эффективности, сокращению времени на внесение и обработку данных, а также к более оперативному предоставлению информации всем участникам образовательного процесса. Рисунок 1 - IDEF0 контекстный уровень На рисунке 2 представлен первый уровень декомпозиции диаграммы IDEF0, показывающий детализированный анализ процессов учёта посещаемости и успеваемости. На данном уровне отражены отдельные подцели, каждая из которых вносит вклад в достижение генеральной цели системы. Реализация перечисленных подцелей обеспечивает выполнение основной задачи - оптимизации управления учебным процессом. Помимо распределения процессов между исполнителями, первый уровень декомпозиции наглядно демонстрирует, какие именно структурные подразделения и участники образовательного процесса (администраторы, методисты, учителя) отвечают за конкретные операции. Также указываются регламентирующие документы и входные данные, необходимые для корректного исполнения соответствующих функций. Рисунок 2 - IDEF0 первый уровень декомпозиции Диаграмма потоков данных (Data Flow Diagram, DFD) даёт возможность подробно отобразить входные и выходные данные информационной системы, а также показать, как эти данные перемещаются между внешними субъектами и хранилищами. Подобно нотации IDEF0, в DFD есть контекстный уровень, который предоставляет общее представление о взаимодействии потоков данных, и уровни декомпозиции, позволяющие более детально рассмотреть процессы и маршруты передачи информации. На рисунке 3 контекстная диаграмма с функцией "Администрирование системы" администратор ИС. Рисунок 3 - Dfd диаграмма администратора На рисунке 4 изображен общий план декомпозиции информационных потоков, проходящих в процессе администрирования системы. Рисунок 4 - Декомпозиция dfd диаграммы администратора На рисунке 5 контекстная диаграмма с функцией "Управление учебными данными в электронном журнале" учитель и методист. Рисунок 5 - Dfd диаграмма учителя и методиста На рисунке 6 декомпозиция функции "Управление учебными данными в электронном журнале" Рисунок 6 - Декомпозиция dfd диаграммы учителя и методиста На рисунке 7 представлена контекстная диаграмма функции "Формирование отчета успеваемости ученика" ученик. Рисунок 7 - Dfd диаграмма ученика На рисунке 8 представлена декомпозиция процесса "Формирование отчета успеваемости ученика" Рисунок 8 - Декомпозиция dfd диаграммы ученика 2.2.1. Описание задачи Требуется разработать веб-систему для учёта посещаемости и успеваемости учащихся, обеспечивающую удобный интерфейс для внесения, просмотра и анализа данных об учебном процессе. Особое внимание уделяется проектированию структуры базы данных, которая должна быть нормализована до третьей нормальной формы, а также содержать триггеры, функции и хранимые процедуры. Необходимо предусмотреть инструменты для импорта и экспорта данных в форматах SQL и CSV, чтобы упростить перенос и интеграцию информации. Важным элементом системы является поддержка отчётности и статистики, а также возможность визуализации данных. Безопасность является ключевым аспектом решения. Функционал должен быть распределён по ролям (администратор, методист, учитель, ученик), каждая из которых имеет доступ только к необходимым функциям и данным. Администратор обладает полным набором прав, включая управление пользователями, настройками системы и оперативный контроль за процессом обучения. Методист и учитель работают с данными о посещаемости и оценках, а ученики просматривают результаты успеваемости. Все пользовательские данные должны защищаться с использованием современных методов криптографической защиты, включая хэширование паролей. Неотъемлемым компонентом системы выступает подсистема резервного копирования. При запуске процедуры резервирования система должна автоматически создавать архив базы данных и сохранять его на компьютер. Администратор может выбрать нужную копию для восстановления и загрузить её. Все операции по созданию и восстановлению резервных копий должны быть доступны через административную панель. На рисунке 11 представлена блок-схема процесса скачивания резервной копии. Рисунок 11 - блок-схема процесса скачивания резервной копии На рисунке 12 представлена диаграмма прецедентов. Диаграмма прецедентов показывает, какой функционал доступен для различных ролей пользователей. Система разделяет доступ к данным и операциям в зависимости от роли пользователя. Администратор имеет полный доступ к управлению системой, включая управление пользователями. Методист может контролировать ведение учебного процесса: управление уроками, классами и предметами. Учитель отвечает за выставление оценок и посещаемости. Ученик, в свою очередь, имеет доступ к просмотру своих оценок и статистике по общей оценке. Рисунок 12 - Диаграмма прецедентов 2.2.2. Входные и выходные данные В таблице 4 представлены входные данные Таблица 4 - Входные данные

Входные данные
Имя
Тип
Ограничение
Формат ввода
Описание
12345
Формы авторизации
loginСтрока[A-Za-z0-9@] {5, 100}
Поле вводаПароль пользователяpasswordСтрока[A-Za-z0-9]{8, 32}
Поле вводаПароль пользователяФорма регистрацииNameСтрока[A-Za-z0-9]{8, 32}
Поле вводаИмя пользователяavatarфото.jpg, png <2мб
Поле выбора файлаАватар пользователяloginСтрока[A-Za-z0-9@] {5, 100}
Поле вводаЛогин пользователяpasswordСтрока[A-Za-z0-9]{8, 32}
Поле вводаПароль пользователяpassword(повторение)Строка[A-Za-z0-9]{8, 32}
Поле вводаПодтверждение пароляФорма редактирования пользователя (Администратор)loginСтрока[A-Za-z0-9@] {5, 100}
Поле вводаЛогин пользователяpasswordСтрока[A-Za-z0-9]{8, 32}
Поле вводаПароль пользователяroleДата1-4
Выпадающий списокСписок всех доступных классовФорма создания урокаclassСтрока[A-Яа-я0-9]{2, 128}
Выпадающий списокСписок доступных классовTeacherСтрока[A-Яа-я0-9]{2, 128}
Выпадающий списокСписок доступных учителейSubjectСтрока [A-Яа-я0-9]{2, 128}
Выпадающий списокСписок предметовDateДата YYYY-MM-DD
Поле вводаДата проведения урокаФорма создания предметаSubjectСтрока [A-Яа-я0-9]{2, 128}
Поле вводаНаименование предметаФорма резервного копированияbackup_fileФайл .sql, .tar.gz, .csv
Поле выбора файлаФайл резервной копии
Форма импорта оценок
classajournal_class_subjectФайл Csv, sql
Поле выбора файлаФайл для импорта оценокФорма создания классаclassСтрока [A-Яа-я0-9]{2, 128}
Поле вводаНаименование класса

В таблице 5 представлены выходные данные Таблица 5- Выходные данные

Выходные данные
Имя
Тип
Ограничение
Формат ввода
Описание
12345
Формы авторизации
loginСтрока[A-Za-z0-9@] {5, 100}
Поле выводаЛогин пользователяpasswordСтрока[A-Za-z0-9] {5, 32}
Поле выводаПароль пользователяФорма регистрацииNameСтрока[A-Za-z0-9]{8, 32}
Поле выводаИмя пользователяavatarфото.jpg, png <2мб
Поле выбора файлаАватар



пользователяloginСтрока[A-Za-z0-9@] {5, 100}Поле выводаЛогин пользователяpasswordСтрока[A-Za-z0-9]{8, 32}Поле вывода  
Пароль пользователяpassword(повторение)Строка[A-Za-z0-9]{8, 32}Поле выводаПодтверждение пароляФорма создания  
урокаclassСтрока[A-Яа-я0-9]{2, 128} Выпадающий списокСписок доступных классовTeacherСтрока[A-Яа-я0-9]{2, 128}  
Выпадающий списокСписок доступных учителейSubjectСтрока [A-Яа-я0-9]{2, 128} Выпадающий списокСписок  
предметовDateДата YYYY-MM-DDПоле выводаДата проведения урокаФорма создания предметаSubjectСтрока [A-Яа-я0-9]{2, 128}  
Поле выводаНаименование предметаФорма создания классаclassСтрока [A-Яа-я0-9]{2, 128} Поле выводаНаименование  
классаОценки ученикаsubjectСтрока[A-Яа-я0-9]{2, 128}Поле выводаНаименование предметаDateСтрока[A-Яа-я0-9]{2, 128}Поле  
выводаЧисло, за которое стоит оценкаGradeЧисло{2, 128}Поле выводаОценкаПанель учителяClassСтрока[A-Яа-я0-9]{2,  
128}Поле выводаПреподаваемый классSubjectСтрока[A-Яа-я0-9]{2, 128}Поле выводаПреподаваемый предметФорма резервного  
копированияbackup\_fileФайл .sql, .tar.gz, .csvПоле выбора файлаФайл резервной копии Форма экспорта оценок  
классаjournal\_class\_subjectФайл Csv, sqlКнопка для загрузки файлаЭкспортированный файл с оценками 2.2.3. Методы  
Необходимо реализовать готовые методы для валидации и обработки значительного объема списочных данных, располагая  
базу данных на удаленном сервере, отдельно от клиентской части. Для организации взаимодействия между клиентскими  
компонентами и сервером базы данных следует разработать соответствующий программный интерфейс. Клиентские модули  
при этом должны соответствовать принципам объектно-ориентированного программирования и включать в себя элементы  
реактивного программирования. Важным аспектом является упрощение процесса администрирования системы, включая анализ  
и настройку логов, а также контроль стабильности работы. Кроме того, следует использовать инкапсуляцию, защищающую  
информацию от несанкционированного доступа и изменений, и обеспечивающую удобную поддержку, и повторное  
использование кода. Применение реактивного подхода, при котором все изменения входных данных автоматически отражаются  
в выходных данных, позволит создавать более отзывчивые и масштабируемые приложения, способные оперативно реагировать  
на любые изменения. В процессе разработки приложения необходимо регулярно проводить рефакторинг, **который** помогает  
улучшить **структуру и качество** кода, а также устраняет технический долг. Помимо рефакторинга особое внимание следует  
уделять оптимизации на всех уровнях системы: от корректного выбора алгоритмов и структуры данных до **повышения** скорости  
отклика и сокращения расхода ресурсов. Это позволит добиться **высокой производительности и стабильности приложения** при  
работе с большими объемами данных. Рисунок 12 - код без рефакторинга Рисунок 13 - Рефакторинг Рисунок 14 - Код без  
оптимизации Оптимизация может включать изменение кода приложения, чтобы он работал быстрее, или использование  
оптимизированных библиотек и инструментов. Примеры оптимизации включают улучшение алгоритмов, оптимизацию  
использования памяти и ускорение процесса загрузки приложения. Рисунок 15 - Код после **оптимизации 2.2.4. Тесты 1.** По  
формальности **тестирования**. Тестирование по тестам предназначено для проверки функциональности веб-приложения в  
соответствии с написанными **тест-кейсами**. Это **позволяет с помощью определенных шагов убедиться в корректной работе**  
**программы**. 2. По исполнению кода. **Динамическое тестирование** фокусируется на проверке программного кода в процессе его  
выполнения. Оно используется **в созданном веб-приложении для выявления ошибок**, которые могут быть упущены при  
статическом тестировании. 3. По уровню тестирования. Системное тестирование направлено на проверку функционирования  
всей системы **в соответствии с заявленными требованиями** к программному **продукту**. Это позволяет обнаружить ошибки,  
**связанные с взаимодействием** различных компонентов приложения. 4. По целям. **Функциональное тестирование сосредоточено**  
**на проверке** реализации функций **программного** продукта и их точности, что помогает убедиться в правильной **работе**  
**разработанного приложения** в соответствии с ожиданиями пользователей. 5. По **степени автоматизации**. **Ручное тестирование**  
**осуществляется без использования дополнительных программных средств** и позволяет оценить работу **разработанного** веб-  
приложения с точки **зрения пользователя**. 6. По **позитивности сценария**. Позитивное тестирование проверяет соответствие  
**функций** веб-приложения ожидаемому **поведению**, в то время как негативное тестирование исследует работу программы в  
ситуациях, когда поведение пользователя отличается от ожидаемого, что помогает выявить ошибки, которые могут возникнуть  
в реальном использовании приложения. 7. По знанию системы. Тестирование "белого ящика" направлено на тестирование  
мобильного приложения с целью глубокого понимания его работы и выявления ошибок, связанных с кодом. Подробные  
сценарии и результаты тестовых испытаний веб-приложения представлены в Приложении В Сценарий и результаты тестовых  
испытаний. 8. По разработке тестовых испытаний. Тестирование на основе требований сосредоточено на проверке  
разработанного веб-приложения для подтверждения его соответствия установленным требованиям. 2.2.5. Контроль  
целостности данных В процессе разработки были реализованы методы валидации входных данных в информационной системе,  
которые обеспечивают целостность этих входных данных. Контроль целостности, описывающих ситуации и реакции  
приложения на выполнения функций представлен в таблице 6. Таблица 6 - Контроль целостности данных  
№СитуацияАномалияРеакцияПримечание123451Регистрация пользователяПользователь оставляет поле "Имя" пустым Поле  
"Имя" является обязательным для заполнения. Пользователь получит сообщение "Поле имя обязательное для заполнения" Для  
успешной регистрации необходимо заполнить все обязательные поляВ поле аватар пользователь загружает файл с  
разрешением exe1.Аватар не поддерживает файлы с разрешением exe1. Пользователь получит сообщение "Допустимые форматы  
jpg, png"Для загрузки аватара необходимо учитывать разрешение файла для загрузкиПользователь вводит пароль длиной  
менее 8 символов.Пароль должен содержать более 8 символов. Пользователь получит сообщение "Пароль должен содержать не  
менее 8 символов!"Пароль должен быть длинной не меньше 8 символовПользователь вводит пароль "Qq12345678" в поле  
"повторите пароль" пользователь вводит другой пароль, например, "Qq11111111" Пароли не совпадают, пользователь получит  
сообщение "Введенные пароли не совпадают"Пароли должны совпадатьПользователь вводит логин student\_ivan, но такой логин  
уже существует в системе.Пользователь ввел логин, который уже существует в системе. Сообщение: "Данное имя пользователя  
уже занято"Логин пользователя должен быть уникальным2Создание урока. МетодистПользователь оставил пустым поле  
"Класс"При создании урока пользователь не выбрал класс. Сообщение "все поля обязательны для заполнения"Обязательные

Страница 6 из 9

manage\_lessons.html Разметка страницы для управления уроками  
7. edit\_lessons.html Разметка страницы для редактирования существующего урока  
8. manage\_subject.html Разметка страницы для управления предметами  
9. edit\_subject.html Разметка страницы для редактирования существующего предмета  
10. manage\_classes.html Разметка страницы для управления классами  
11. edit\_classes.html Разметка страницы для редактирования существующего класса  
12. teacher/dashboard.html Разметка страницы для отображения журналов учителя  
13. class\_journal.html Разметка страницы для отображения журнала класса с оценками  
14. grades.html Разметка страницы студента для отображения собственных оценок  
15. student\_statistics.html Разметка страницы для отрисовки графика со средним баллом  
16. admin/dashboard.html Разметка страницы для управления пользователями  
17. admin/createbackup.html Разметка страницы для создания и загрузки резервной копии бд  
18. logs.html Разметка страницы для просмотра логов приложения  
19. create\_user.html Разметка страницы для создания нового пользователя  
20. edit\_user.html Разметка страницы для редактирования данных пользователя

2.3.5. Функциональная схема На рисунке 20 представлена функциональная схема веб-приложения, на котором графически изображены возможности пользователя в процессе эксплуатации. Рисунок 20 - Функциональная схема 2.3.6. Схема тестирования На рисунке 22 представлена схема тестирования, по которой проводилось тестирование веб-приложения. Данная схема подробно рассматривается в документе "ПРИЛОЖЕНИЕ В. Результат тестовых испытаний" Рисунок 22 - Схема тестирования 2.3.7. Схема пользовательского интерфейса На рисунке 23 представлена схема пользовательского интерфейса. Данная схема отображает возможные переходы пользователя по интерфейсу. Рисунок 23 - Схема пользовательского интерфейса 2.4. Результат работы программы В результате выполнения поставленной задачи был разработан требуемый программный продукт в виде информационной системы по учету успеваемости и посещаемости. На рисунках 24-33 представлена часть полученного результата работы программы. Полную версию результата программы можно посмотреть в приложении В (Руководство пользователя) Пользователя встречает приветствующая страница. Рисунок 24 - Страница заставка Окно авторизации/регистрации. Пользователь имеет возможность ввести свои данные (логин, пароль) и авторизоваться в системе. Рисунок 25 - окно авторизации Если пользователь отсутствует в системе он может перейти на страницу регистрации и зарегистрироваться. По умолчанию зарегистрированный пользователь будет с ролью student. Для регистрации необходимо ввести: имя, логин, прикрепить аватар, пароль и повторить введенный пароль. После успешной регистрации пользователя переправит на страницу авторизации. Рисунок 26 - окно регистрации После успешной авторизации в зависимости от роли пользователя перед ним открывает следующий функционал. Админ. Роль администратора предполагает управление системой. Поэтому на админ-панели располагается таблица для управления пользователями системы. Администратор может удалить пользователя, изменить его данные и создать нового пользователя. Рисунок 27 - админ панель Также у администратора есть возможность создать или загрузить резервную копию базы данных. Рисунок 28 - backup системы У каждого пользователя включая администратора есть вкладка "учетная запись" перейдя на которую пользователь сможет изменить свои данные в системе. Рисунок 29 - учетная запись Студент. У студента есть возможность просматривать свои оценки. Для удобства просмотра предусмотрены функции фильтра, поиска и сортировки данных. Также студент может посмотреть статистику в виде графика, где будет показан средний балл по всем предметам. Рисунок 30 - оценки ученика Рисунок 31 - статистика успеваемости Методист. Основная цель методиста - это ведение учебного процесса, поэтому у методиста есть специальная панель. Используя эту панель, методист может управлять предметами, занятиями и классами, а именно создавать, редактировать и удалять данные. Рисунок 32 - панель методиста Учитель. Учитель может выставлять оценки и пропуски ученикам. Также у учителя есть возможность импортировать и экспортировать данные в форматах sql и csv. Рисунок 33 - журнал успеваемости и посещаемости

3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 3.1. Инструментальные средства Для разработки веб-приложения "Система по учету успеваемости и посещаемости" была использована интегрированная среда разработки JetBrains PyCharm. Эта среда обладает мощными инструментами для создания и отладки приложений, написанных на языке программирования Python. PyCharm предоставляет удобный интерфейс для работы с фреймворками, библиотеками и модулями, автоматизирует процессы настройки виртуальных окружений и управления зависимостями, а также включает встроенные инструменты для работы с системами контроля версий, такими как Git. В качестве основного языка программирования для разработки приложения был выбран Python. Это высокоуровневый язык программирования с простым и читаемым синтаксисом, который широко применяется для создания веб-приложений, автоматизации процессов, анализа данных и других задач. Python имеет большое количество библиотек и фреймворков, которые значительно ускоряют процесс разработки. Для создания веб-интерфейса и API был использован фреймворк Flask. Flask - это легковесный веб-фреймворк с минимальным количеством зависимостей, который предоставляет гибкость для построения различных архитектур, таких как MVC и RESTful API. Flask позволяет эффективно реализовывать маршруты, обработчики запросов, шаблоны и подключения к базам данных. В качестве основного движка шаблонов был выбран Jinja2, который позволяет создавать динамические HTML-страницы с использованием простого и интуитивно понятного синтаксиса. Для разработки и управления базой данных была использована система управления базами данных PostgreSQL. Это мощная реляционная СУБД, которая предоставляет высокую производительность, поддержку сложных запросов и надёжность хранения данных. Для управления PostgreSQL использовался инструмент pgAdmin, который обладает удобным графическим интерфейсом для администрирования баз данных, создания и редактирования таблиц, написания SQL-запросов, а также для построения диаграмм ERD. Кроме того, pgAdmin предоставляет средства для мониторинга активности сервера, настройки пользователей и выполнения резервного копирования базы данных. Для реализации взаимодействия с базой данных использовалась библиотека SQLAlchemy, которая является мощным инструментом для работы с реляционными базами данных и предоставляет ORM (Object-Relational Mapping). Это позволило абстрагироваться от низкоуровневого SQL и сосредоточиться на логике приложения. Для обеспечения контроля версий и совместной работы над проектом была выбрана система Git, а в качестве платформы для хостинга репозитория использовался GitHub. GitHub предоставляет широкий функционал для управления проектами, включая ведение репозитория, создание и отслеживание задач. Это позволило организовать процесс разработки максимально

эффективно и структурированно. Таким образом, использование указанных инструментов и технологий обеспечило удобство, гибкость и надёжность при создании веб-приложения "Система по учету успеваемости и посещаемости учащихся". 3.2. Отладка программы В процессе разработки веб-приложения "Система управления задачами и проектами" возникали различные ошибки, связанные с логикой взаимодействия между фронтендом и бэкендом, а также с операциями над базой данных. Большинство проблем касались неправильной обработки данных на этапе сериализации и десериализации, некорректных запросов к базе данных и ошибок валидации пользовательского ввода. Для обнаружения и исправления ошибок активно использовались встроенные инструменты отладки среды разработки JetBrains PyCharm, включая дебаггер и консоль для анализа логов. Это позволило пошагово анализировать выполнение программы, выявлять проблемные участки кода и проверять корректность выполнения SQL-запросов. Одним из ключевых инструментов отладки стали логи, настроенные на сбор информации об ошибках и действиях пользователей. В проекте был реализован механизм записи логов, который фиксирует критические ошибки, предупреждения и успешные операции, что значительно упростило процесс анализа причин возникновения ошибок. В случаях, когда возникали сложные задачи или неизвестные ошибки, требующие дополнительных знаний, использовались внешние ресурсы, такие как официальная документация Flask, SQLAlchemy и PostgreSQL, а также профессиональные форумы и сообщества, например Stack Overflow и тематические группы разработчиков. Это помогало оперативно находить решение и углублять знания в области разработки. Благодаря систематическому подходу к отладке, использованию мощных инструментов и обращению к дополнительным источникам информации удалось своевременно исправлять ошибки, улучшать функциональность приложения и обеспечить его стабильную работу. Полный перечень источников, которые использовались при разработке, представлен в разделе "Список используемых материалов". 3.3. Защитное программирование В программном коде предусмотрены защиты от ошибок и недоступности сервиса в различных частях кода. Для этого применяются проверки входных данных и условий с использованием блоков if...else, глобальные обработчики ошибок, а также механизмы ORM SQLAlchemy, предотвращающие SQL-инъекции. На рисунке 34 представлена защита от некорректных данных и ошибок с базой. Если переданный user\_id не является числом, возникает ошибка ValueError, которая может привести к сбою работы системы. В этом случае обработчик предотвращает ошибку и возвращает None, не допуская некорректного выполнения кода. Также предусмотрена защита от возможных ошибок SQLAlchemy, которые могут возникнуть при взаимодействии с базой данных. Если пользователь с таким user\_id отсутствует, код корректно возвращает None, что исключает потенциальные уязвимости, связанные с доступом к несуществующим данным. Рисунок 34 - защита от некорректных данных На рисунке 35 представлена защита хеширования паролей. При регистрации и обновлении профиля пароли не хранятся в открытом виде, а шифруются с помощью bcrypt. Рисунок 35 - хеширование паролей. На рисунке 36 реализована проверка прав доступа, которая предотвращает просмотр и изменение профиля другого пользователя. Рисунок 36 - проверка прав доступа 3.4. Характеристики программы Разработанное веб-приложение может быть доступно на любом браузере. Характеристики программы представлены в таблице 1 "Модули" Приложения А "Текст программы". ЗАКЛЮЧЕНИЕ В результате выполнения курсового проекта была разработана "информационная система для учёта посещаемости и успеваемости" учащихся, отвечающая современным требованиям удобства, функциональности и безопасности. В ходе работы были реализованы все ключевые задачи, включая проектирование архитектуры приложения, разработку базы данных, внедрение механизмов защиты данных и создание инструментов для анализа учебного процесса. На этапе проектирования системы был проведён детальный анализ предметной области, который включал построение диаграммы IDEF0 для описания функциональных процессов и логической модели данных для структурирования информации о пользователях, классах, предметах, оценках и посещаемости. Это позволило глубже понять автоматизируемые процессы и разработать оптимальную структуру хранения данных. Приложение было создано на основе Flask с использованием PostgreSQL в качестве системы управления базами данных. Архитектурный подход основан на модели MVC (Model-View-Controller) с интеграцией RESTful API, что обеспечило чёткое разделение логики, удобство масштабирования и гибкость системы. Взаимодействие с базой данных реализовано с помощью ORM SQLAlchemy, что позволило безопасно и эффективно работать с данными. Для повышения отказоустойчивости системы были реализованы механизмы резервного копирования с возможностью восстановления данных, а также логирование событий, что позволяет отслеживать изменения в базе данных и анализировать работу пользователей. Безопасность обеспечивается системой аутентификации и разграничения прав доступа, которая разделяет функциональные возможности между администратором, методистом, учителем и учеником. В процессе разработки проводилось тестирование, включающее функциональное тестирование, тестирование методом чёрного ящика и модульное тестирование. Это позволило выявить и устранить возможные ошибки, обеспечив стабильную и корректную работу приложения. Таким образом, разработанное приложение полностью соответствует поставленным требованиям и готово к дальнейшему использованию и возможному расширению. Оно обеспечивает эффективный контроль успеваемости и посещаемости учащихся, а также повышает прозрачность учебного процесса, что способствует улучшению организации образовательной деятельности. СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ 1. Flask Documentation. URL: <https://flask.palletsprojects.com/en/2.3.x/> (Дата обращения: 05.09.2024) 2. Python 3.11 Documentation. URL: <https://docs.python.org/3.11/> (Дата обращения: 07.09.2024) 3. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs/> (Дата обращения: 09.09.2024) 4. JetBrains PyCharm Documentation. URL: <https://www.jetbrains.com/pycharm/documentation/> (Дата обращения: 11.09.2024) 5. GitHub Documentation. URL: <https://docs.github.com/> (Дата обращения: 13.09.2024) 6. SQLAlchemy Documentation. URL: <https://docs.sqlalchemy.org/en/20/> (Дата обращения: 15.09.2024) 7. Flask-SQLAlchemy Documentation. URL: <https://flask-sqlalchemy.palletsprojects.com/en/3.x/> (Дата обращения: 18.09.2024) 8. Jinja2 Documentation. URL: <https://jinja.palletsprojects.com/en/3.1.x/> (Дата обращения: 20.09.2024) 9. Flask-Login Documentation. URL: <https://flask-login.readthedocs.io/en/latest/> (Дата обращения: 23.09.2024) 10. Flask-WTF Documentation. URL: <https://flask-wtf.readthedocs.io/en/stable/> (Дата обращения: 26.09.2024) 11. Git Documentation. URL: <https://git-scm.com/doc> (Дата обращения: 30.09.2024) 12. RESTful API Design Best Practices. URL: <https://restfulapi.net/> (Дата обращения: 04.10.2024) 13. SQL



Optimization Techniques. URL: <https://use-the-index-luke.com/> (Дата обращения: 08.10.2024) 14. Flask Security Best Practices. URL: <https://flask.palletsprojects.com/en/2.3.x/security/> (Дата обращения: 12.10.2024) 15. PostgreSQL Performance Tuning. URL: <https://www.postgresql.org/docs/current/performance-tips.html> (Дата обращения: 17.10.2024) 16. Object-Relational Mapping Concepts. URL: <https://martinfowler.com/eaCatalog/orm.html> (Дата обращения: 23.10.2024) 17. Flask-Migrate Documentation. URL: <https://flask-migrate.readthedocs.io/en/latest/> (Дата обращения: 01.11.2024) 18. GitHub Actions Documentation. URL: <https://docs.github.com/en/actions> (Дата обращения: 10.11.2024) 19. Web Application Security Guidelines. URL: <https://owasp.org/> (Дата обращения: 22.11.2024) 20. Best Practices for Flask Deployment. URL: <https://flask.palletsprojects.com/en/2.3.x/deploying/> (Дата обращения: 20.12.2024) 2