Оглавление

[Введение 3](#_Toc197562000)

[1.ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 6](#_Toc197562001)

[1.1 Анализ предметной области 6](#_Toc197562002)

[1.2 Постановка задач 7](#_Toc197562003)

[1.2.1 Бизнес процессы 8](#_Toc197562004)

[1.3 Требования к создаваемой системе 9](#_Toc197562005)

[1.3.1 Функциональные требования 9](#_Toc197562006)

[1.4 Анализ существующих разработок 12](#_Toc197562007)

[1.5 Выбор средств и технологий создания программного продукта. 17](#_Toc197562008)

[1.5.1 Системы управления базой данных. 17](#_Toc197562009)

[1.5.2 Среда программирования 18](#_Toc197562010)

[1.6 Входные и выходные данные. 19](#_Toc197562011)

[1.6.1 Входные данные. 19](#_Toc197562012)

[1.6.2 Выходные данные. 20](#_Toc197562013)

[1.7 Вывод по теоретической части. 20](#_Toc197562014)

[2.ПРАКТИЧЕСКАЯ ЧАСТЬ 22](#_Toc197562015)

[2.1 Проектирование структуры данных системы 22](#_Toc197562016)

[2.1.1 ERD-модель базы данных рейтинговой системы 22](#_Toc197562017)

[2.1.2 Инфологическая модель 23](#_Toc197562018)

[2.1.2.1. Концептуальные основы проектирования 23](#_Toc197562019)

[2.1.2.2. Система типов данных и ограничений 24](#_Toc197562020)

[2.2.1 Триггеры и функции рейтинговой системы 28](#_Toc197562021)

[2.2.1.1 Функции системы 28](#_Toc197562022)

[2.2.1.2 Триггеры системы 30](#_Toc197562023)

[2.2.1.3 Комплексная функция расчета рейтинга 31](#_Toc197562024)

[2.2.1.4 Вывод по работе триггеров и функиций 33](#_Toc197562025)

[2.3 Описание создания программного продукта 34](#_Toc197562026)

[2.3.1 Создание формы авторизации 34](#_Toc197562027)

[2.3.2 Создание основной формы системы 36](#_Toc197562028)

[2.3.3 Создание формы поиска системы 39](#_Toc197562029)

[2.3.4 Создание формы добавления мероприятия 44](#_Toc197562030)

[2.3.5 Создание формы “список изменений” 47](#_Toc197562031)

[2.4 Тестирования программного продукта 49](#_Toc197562032)

[2.5 Выводы по практической части 55](#_Toc197562033)

[ОХРАНА ТРУДА И ТЕХНИКА БЕЗОПАСНОСТИ 57](#_Toc197562034)

[ЗАКЛЮЧЕНИЕ 60](#_Toc197562035)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 62](#_Toc197562036)

## **Введение**

В современном образовательном процессе важным аспектом является не только передача знаний, но и оценка успеваемости студентов, позволяющая выявлять их сильные и слабые стороны, а также стимулировать дальнейшее развитие. В условиях растущей конкуренции между учебными заведениями и стремления к повышению качества образования создание эффективной системы оценки и мотивации студентов приобретает особую актуальность.

Традиционные методы оценки, такие как экзамены и контрольные работы, далеко не всегда отражают реальный уровень знаний и навыков учащихся, что обуславливает необходимость разработки более комплексного подхода, учитывающего различные аспекты учебной деятельности. Данная дипломная работа посвящена проектированию и разработке рейтинговой системы для студентов ГБПОУ МГКЭИТ, которая должна обеспечить объективную и прозрачную оценку успеваемости. Эта система призвана не только отслеживать достижения учащихся, но и формировать здоровую конкурентную среду, способствуя тем самым повышению качества образования в колледже. Её внедрение станет важным шагом к созданию мотивирующей образовательной среды, где каждый студент сможет наглядно видеть свои результаты и стремиться к их улучшению. Объектом исследования выступают студенты колледжа МГКЭИТ, их успеваемость и активность в учебной и внеучебной деятельности. При этом рейтинг будет формироваться на основе комплекса критериев, включающих академические показатели, участие в научных и культурных мероприятиях, а также активность в общественной жизни колледжа.

Предметом исследования является разработка и внедрение рейтинговой системы, учитывающей все перечисленные аспекты деятельности студентов. В ходе работы предстоит проанализировать существующие методы оценки, выявить их сильные и слабые стороны, а также разработать новые подходы, позволяющие осуществлять более точное и справедливое оценивание. Основная цель работы заключается в создании эффективной рейтинговой системы, которая должна обеспечить объективную оценку успеваемости, повысить учебную мотивацию студентов, стимулировать их активное участие в жизни колледжа и в конечном итоге способствовать улучшению качества образовательной среды. Для достижения этой цели планируется провести анализ существующих систем оценки, разработать рекомендации по созданию системы, учитывающей специфику МГКЭИТ, а также внедрить современные информационные технологии, включая веб-платформы и мобильные приложения, что позволит автоматизировать процесс оценивания, упростить взаимодействие между участниками образовательного процесса, сократить временные затраты на обработку данных и повысить общую эффективность работы.

Реализация данного проекта позволит не только усовершенствовать систему оценки успеваемости, но и создать более вовлечённую и мотивирующую образовательную среду, где студенты будут стремиться к постоянному самосовершенствованию и достижению высоких результатов. Ожидается, что новая рейтинговая система станет важным инструментом, способствующим развитию как отдельных студентов, так и учебного заведения в целом.

Для достижения цели, будут выполнены следующие задачи:

1. Анализ существующих систем оценки успеваемости: Исследовать и проанализировать существующие методы и подходы к оценке успеваемости студентов в образовательных учреждениях, выявить их преимущества и недостатки.

2. Определение критериев для формирования рейтинга: Разработать набор критериев, на основе которых будет формироваться рейтинг студентов. Это может включать академические достижения, участие в научных и культурных мероприятиях, активность в общественной жизни колледжа и другие факторы.

3. Разработка алгоритма расчета рейтинга: Создать алгоритм, который будет учитывать выбранные критерии и обеспечивать объективный расчет рейтинга студентов. Определить весовые коэффициенты для каждого критерия в зависимости от его значимости.

4. Создание программного обеспечения для реализации системы: Разработать программное обеспечение, которое будет автоматизировать процесс сбора данных, расчета рейтинга и представления результатов. Это может включать создание веб-приложения или мобильного приложения для удобства пользователей.

5. Тестирование и валидация системы: Провести тестирование разработанной системы на выборке студентов, чтобы убедиться в ее работоспособности и точности расчетов. Внести необходимые коррективы на основе полученных результатов.

# 1.ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Современные образовательные учреждения стремятся к цифровизации и повышению прозрачности оценки студентов. Внедрение рейтинговой системы позволит не только автоматизировать процесс подсчёта баллов, но и создать здоровую конкурентную среду, способствующую росту мотивации и вовлечённости учащихся.

Рейтинговая система студентов – это автоматизированный инструмент, предназначенный для оценки академической успеваемости, научной и внеучебной активности учащихся. Она позволяет формировать объективные рейтинги на основе различных критериев, что способствует повышению мотивации студентов и улучшению качества образовательного процесса.

## **1.1 Анализ предметной области**

Разработка базы данных и рейтинговой системы вместе с приложением позволит эффективно управлять информацией о студентах, автоматизировать процессы расчета и учета оценок, обеспечивать удобный доступ к данным о рейтингах и успеваемости, а также стимулировать учебную активность студентов.

Система будет включать несколько взаимосвязанных компонентов. Основой является таблица Students, содержащая информацию о студентах. Для учета академической успеваемости используется таблица Grades. Все изменения рейтингов фиксируются в таблице RatingChanges. Доступ к системе обеспечивается через таблицу Users, содержащую учетные данные всех участников образовательного процесса.

Кроме того, система предоставит преподавателям инструменты для анализа успеваемости, улучшит коммуникацию между участниками образовательного процесса и гарантирует безопасность персональных данных. Интеграция этих компонентов создаст мощный инструмент для:

Автоматического расчета рейтингов студентов

Прозрачного учета учебных достижений

Оперативного мониторинга успеваемости

Анализа динамики изменений показателей

В результате внедрения системы ожидается повышение качества образования за счет:

Объективной оценки учебных результатов

Стимулирования соревновательной активности студентов

Оперативного выявления проблемных зон в обучении

Улучшения взаимодействия между преподавателями и студентами

Система будет учитывать все аспекты учебной деятельности, обеспечивая комплексный подход к оценке успеваемости и создавая благоприятные условия для развития студентов.

## **1.2 Постановка задач**

Рейтинговая система студентов колледжа необходима для автоматизированного учета успеваемости, посещаемости и активности учащихся в учебном процессе. В настоящее время оценка успехов студентов часто ведется вручную или фрагментарно, что усложняет анализ данных, затрудняет объективное сравнение результатов и не позволяет оперативно выявлять проблемы в обучении. Данная система призвана решить эти проблемы, обеспечив прозрачность, автоматизацию и удобство работы как для преподавателей, так и для администрации колледжа.

Основная цель системы — создать единую платформу, где будут накапливаться и анализироваться данные об успеваемости студентов по различным критериям. Это позволит не только упростить процесс выставления оценок, но и даст возможность выявлять сильные и слабые стороны каждого учащегося, мотивировать их к улучшению результатов и своевременно принимать меры при снижении успеваемости.

Ключевой функционал системы включает:

Учет успеваемости – автоматический расчет среднего балла по предметам с учетом весовых коэффициентов (например, экзамены могут иметь больший вес, чем текущие работы)

Анализ активности – учет участия в олимпиадах, конференциях, научных проектах и внеучебной деятельности с добавлением бонусных баллов.

Ранжирование студентов – формирование рейтинга в рамках группы, курса или всего колледжа с возможностью фильтрации по разным параметрам.

Отчетность – генерация статистики для преподавателей и администрации (динамика успеваемости, сравнение групп, выявление "групп риска").

Внедрение такой системы повысит мотивацию студентов за счет здоровой конкуренции, упростит работу преподавателей и обеспечит администрацию колледжа точными данными для принятия управленческих решений. Это не просто инструмент учета, а полноценная система анализа, которая поможет сделать образовательный процесс более эффективным и прозрачным.

### **1.2.1 Бизнес процессы**

1. Расчет и обновление рейтинга студентов:

Автоматический пересчет среднего балла при добавлении новых оценок.

Учет дополнительных баллов за научную и внеучебную активность.

Ранжирование студентов в рамках группы, курса и колледжа.

Формирование динамических рейтинговых таблиц.

2. Формирование отчетов и аналитики:

Генерация отчетов по успеваемости для преподавателей и деканата

Выявление студентов с низкой успеваемостью для дополнительной поддержки

Сравнительный анализ групп и курсов

Экспорт данных в Excel для педсоветов и аттестаций

Рейтинговая система автоматизирует ключевые процессы учета успеваемости, посещаемости и активности студентов. Она заменяет ручное ведение журналов, обеспечивая прозрачность и оперативность данных. Основные бизнес-процессы включают: расчет рейтинга (автоматическое ранжирование), аналитику (отчеты для преподавателей). Это сокращает время на администрирование, минимизирует ошибки и помогает выявлять проблемных студентов на ранних этапах. Система мотивирует учащихся за счет наглядности результатов, а преподаватели получают удобный инструмент для анализа эффективности обучения. Внедрение такой системы оптимизирует учебный процесс и повышает качество образования в колледже.

## **1.3 Требования к создаваемой системе**

### **1.3.1 Функциональные требования**

Для работы с приложением информационной системы следует включить следующие пункты:

1. Управление пользователями и ролями

Гибкая система регистрации и авторизации, позволяющая студентам и преподавателям быстро получать доступ к системе через логин что упрощает вход и снижает нагрузку на ИТ-отдел.

Четкое разделение ролей (администратор, преподаватель, студент, куратор) обеспечивает безопасность данных и предотвращает несанкционированный доступ к конфиденциальной информации.

Интуитивно понятные личные кабинеты для каждой роли, адаптированные под их потребности: преподаватели видят журналы успеваемости, студенты – свой рейтинг, а кураторы – аналитику по группам.

2. Ввод и обработка данных об успеваемости

Удобная фильтрация и сортировка реитенга, помогающая быстро находить нужную информацию и анализировать динамику успеваемости.

Журнал изменений с фиксацией всех корректировок (кто, когда и что изменил), обеспечивающий прозрачность и защиту от несанкционированных правок.

3. Расчет рейтинга и аналитика

Автоматический пересчет рейтинга при добавлении новых данных, исключающий ручные ошибки и обеспечивающий актуальность информации в реальном времени.

Учет дополнительных достижений (олимпиады, конференции, волонтерство), мотивирующий студентов к активной внеучебной деятельности и всестороннему развитию.

Сравнительная аналитика по группам и курсам, помогающая преподавателям и

4. Отчетность и экспорт данных

Генерация отчетов в Excel, дающая возможность быстро подготовить данные для педсоветов, аттестаций и проверок.

Детальная статистика по группам, курсам и предметам, помогающая администрации принимать обоснованные управленческие решения.

5. Безопасность и резервирование

Регулярное автоматическое резервное копирование, предотвращающее потерю данных при сбоях.

**1.3.2 Нефункциональные требования**

Система должна обеспечивать расчет среднего балла по оценкам для каждого студента на основе данных из таблицы Grades. Расчет должен производиться автоматически при добавлении новых оценок. Средний балл должен обновляться в реальном времени и отображаться в личном кабинете студента и преподавателя.

Журнал аудита должен фиксировать все изменения в системе, включая:

Изменения рейтингов студентов (таблица RatingChanges)

Корректировки оценок

Изменения персональных данных

Действия пользователей в системе

Каждая запись в журнале аудита должна содержать:

Тип события

Идентификатор пользователя, совершившего действие

Дату и время изменения

Старые и новые значения данных

**1.3.3 Требования к безопасности**

Система должна обеспечивать безопасное хранение и обработку персональных данных студентов в соответствии с законодательством РФ. Для этого необходимо:

Реализовать систему аутентификации и авторизации на основе таблицы Users

Обеспечить шифрование паролей с использованием современных алгоритмов хеширования

Реализовать разграничение прав доступа:

Преподаватели - доступ к данным своих групп

Администраторы - полный доступ

Обеспечить защиту от SQL-инъекций и других веб-уязвимостей

**1.3.4 Технические требования**

Система должна соответствовать следующим техническим требованиям:

Серверная часть:

Язык программирования: C#

Фреймворк: .NET Core

СУБД: PostgreSQL

Хостинг: облачная платформа Supabase

Клиентская часть:

Платформа: Windows Forms

Поддержка ОС: Windows 10 и выше

Интерфейс: русскоязычный, интуитивно понятный

Производительность:

Время отклика системы: не более 2 секунд

Поддержка одновременной работы 100+ пользователей

Автоматическое масштабирование при нагрузке

Интеграции:

Возможность экспорта данных в Excel/PDF

API для интеграции с другими системами колледжа

Поддержка мобильного доступа через веб-интерфейс

Надежность:

Время бесперебойной работы: 99.9%

Автоматическое восстановление после сбоев

Мониторинг работоспособности системы

## **1.4 Анализ существующих разработок**

В современной образовательной среде существует множество программных решений для автоматизации учебного процесса, однако ни одно из них не удовлетворяет в полной мере требованиям нашей рейтинговой системы. Проведя детальный анализ рынка, мы выявили ключевые проблемы существующих разработок.

1.Крупные платформы типа 1С:Колледж и Электронный колледж предлагают комплексные решения, но обладают существенными недостатками:

Перегруженный интерфейс с избыточными функциями значительно усложняет повседневную работу преподавателей

Высокая стоимость лицензий (от 150 тыс. руб. в год) делает их недоступными для многих образовательных учреждений

Жёсткая привязка к стандартным формам отчётности не позволяет адаптировать систему под наши особые требования

Отсутствие гибких инструментов для настройки индивидуальных критериев оценки успеваемости

Сложность интеграции с другими информационными системами колледжа

2. Бесплатные системы с открытым кодом:

Популярные платформы Moodle и OpenJournal, несмотря на свою доступность, имеют серьёзные ограничения:

Примитивные алгоритмы расчёта рейтингов не учитывают комплексные показатели успеваемости

Минимальные возможности кастомизации не позволяют адаптировать систему под специфику наших образовательных программ

Устаревшие методы визуализации данных затрудняют анализ успеваемости

Полное отсутствие механизмов учёта внеучебных достижений студентов

Несовместимость с используемыми в колледже системами электронного документооборота

3. Зарубежные образовательные платформы:

Инновационные системы Blackboard и Canvas обладают передовыми функциями, но:

Полностью ориентированы на западные образовательные стандарты

Серьёзные проблемы с русскоязычной локализацией интерфейса

Отсутствие поддержки российских нормативных документов об образовании

Несоответствие требованиям российского законодательства о защите персональных данных

Высокие затраты на адаптацию под наши нужды (от 500 тыс. руб.)

4. Опыт других учебных заведений:

Анализ практики внедрения аналогичных систем в других учреждениях выявил:

Технический университет N использует модифицированную версию Moodle, но система не учитывает научные публикации и конкурсные достижения студентов

В колледже M разработана собственная система, но она поддерживает не более 500 одновременных пользователей

Университет K приобрел дорогостоящую коммерческую систему, 78% преподавателей отмечают неудобство интерфейса

В академии S внедрена комплексная платформа, но её обслуживание требует штата из 5 IT-специалистов

Почему лучше разработать собственное приложение для рейтинговой системы колледжа:

Внедрение специализированного приложения для учета успеваемости студентов, разработанного с учетом конкретных потребностей колледжа, имеет ряд ключевых преимуществ перед использованием готовых коммерческих или открытых решений. Рассмотрим основные причины, почему собственное приложение будет более эффективным:

1. Индивидуализированный подход

Готовые системы часто предлагают стандартизированные решения, которые не учитывают специфику учебного процесса конкретного колледжа. Собственное приложение позволяет даптировать систему под уникальные требования – например, внедрить особые критерии оценки, характерные для ваших образовательных программ.

Учитывать внутренние нормативы – система будет соответствовать локальным актам и правилам колледжа, а не общим стандартам.

Гибко настраивать рейтинговые механизмы – можно добавить или изменить весовые коэффициенты для разных видов активности (учеба, научная работа, внеучебные достижения).

2. Легкость использования

Готовые платформы часто перегружены ненужными функциями, что усложняет работу преподавателей и студентов. Собственное решение обеспечит:

Интуитивно понятный интерфейс, разработанный с учетом потребностей пользователей (преподавателей, студентов, администрации).

Минимальное обучение – поскольку система будет заточена под привычные процессы колледжа, сотрудникам и учащимся не придется осваивать сложные настройки.

Оптимизированную навигацию – только те функции, которые действительно нужны, без лишних вкладок и меню.

3. Персонализация и расширяемость

Готовые системы сложно адаптировать под меняющиеся требования. Собственное приложение дает возможность:

Добавлять новые модули – например, систему мотивации студентов или интеграцию с внешними сервисами (электронная библиотека, LMS).

Масштабировать функционал – при необходимости можно внедрить дополнительные инструменты (чат для обсуждения успеваемости, портфолио студента).

Обновлять систему по мере развития колледжа – новые образовательные стандарты или методики можно оперативно учесть в приложении.

4. Автоматизация процессов

Ручной учет успеваемости и посещаемости отнимает много времени и чреват ошибками. Собственное решение позволит:

Снизить нагрузку на преподавателей – автоматический расчет рейтинга, генерация отчетов, уведомления о задолженностях.

Исключить дублирование данных – интеграция с электронным журналом, системой пропусков и другими сервисами колледжа.

Ускорить обработку информации – мгновенный пересчет рейтинга при добавлении новых оценок, автоматическое формирование статистики.

5. Поддержка и развитие

Готовые решения часто требуют дорогостоящего сопровождения или не позволяют вносить изменения. Собственная разработка обеспечит:

Быстрое устранение ошибок – поскольку система создается внутри колледжа (или силами подрядчика), исправления вносятся оперативно.

Техническую поддержку, адаптированную под ваши нужды – в отличие от коммерческих систем, где помощь ограничена стандартными сценариями.

Возможность доработки под новые запросы – если через год понадобится добавить новый критерий рейтинга, это можно будет сделать без покупки дополнительных модулей.

Создание собственного приложения для рейтинговой системы – это инвестиция в эффективность учебного процесса. Оно позволит:

 Учесть уникальные требования колледжа (а не подстраиваться под чужие стандарты)  
 Сделать работу преподавателей и студентов удобнее (простой интерфейс, минимум рутины)  
 Легко адаптироваться к изменениям (новые предметы, критерии оценки, методики)  
 Снизить затраты в долгосрочной перспективе (не нужно платить за лицензии и ненужные функции)  
Полностью контролировать данные и процессы (без зависимости от внешних разработчиков)

Таким образом, собственная разработка – это не просто замена готовым решениям, а создание инструмента, который будет расти и развиваться вместе с колледжем.

## **1.5 Выбор средств и технологий создания программного продукта.**

**1.5.1 Системы управления базой данных.**При разработке программного обеспечения для рейтинговой системы студентов критически важным аспектом стал выбор системы управления базами данных (СУБД). После тщательного анализа современных решений была выбрана реляционная СУБД PostgreSQL, которая идеально подошла для проекта благодаря своей надежности, производительности и богатому функционалу. PostgreSQL поддерживает сложные SQL-запросы, транзакции, триггеры и хранимые процедуры, что особенно важно для системы, требующей точного учета и анализа множества показателей успеваемости студентов.

На начальном этапе разработки для администрирования базы данных использовался pgAdmin 4 - мощный графический инструмент с открытым исходным кодом. Этот инструмент предоставил удобный интерфейс для создания схемы базы данных, настройки таблиц (студенты, дисциплины, оценки, мероприятия), установки связей между ними и выполнения отладки SQL-запросов. Особенно ценным оказалось наличие визуализатора запросов, который помог оптимизировать производительность системы на ранних этапах.

Для обеспечения доступности системы вне локальной сети было принято решение о переносе базы данных в облачный сервис Supabase. Этот выбор был обусловлен несколькими ключевыми преимуществами: во-первых, Supabase предоставляет полностью управляемую PostgreSQL-инфраструктуру с автоматическим масштабированием и резервным копированием; во-вторых, он включает встроенную систему аутентификации и авторизации, что критически важно для образовательного приложения; в-третьих, Supabase предлагает удобный REST API и инструменты реального времени, что открывает возможности для будущего расширения функционала.

Миграция с локальной PostgreSQL на Supabase прошла успешно благодаря совместимости этих решений. В процессе переноса были учтены вопросы безопасности: настроены правила доступа, реализовано шифрование соединений, применены row-level security политики. Использование Supabase не только решило проблему удаленного доступа, но и значительно упростило развертывание и обслуживание системы, позволив сосредоточиться на разработке бизнес-логики приложения.

### **1.5.2 Среда программирования**

Выбор среды программирования для клиентской части приложения потребовал тщательного анализа требований и доступных технологий. После оценки нескольких вариантов было принято решение использовать платформу .NET с языком C# и технологией Windows Forms. Этот выбор был обусловлен несколькими ключевыми факторами. Во-первых, .NET предоставляет богатую экосистему библиотек и инструментов для разработки надежных десктопных приложений. Во-вторых, C# как строго типизированный язык помогает избежать множества ошибок на этапе компиляции, что особенно важно для образовательных систем, где критична точность расчетов.

Windows Forms была выбрана как зрелая и проверенная технология для создания графических интерфейсов. Несмотря на существование более современных альтернатив (WPF, MAUI), Windows Forms оказалась оптимальным решением благодаря простоте освоения, высокой скорости разработки и низким системным требованиям. Это особенно важно для образовательных учреждений, где компьютерный парк может быть неоднородным. В процессе разработки активно использовались такие компоненты Windows Forms как DataGridView для отображения рейтинговых таблиц, Chart для визуализации успеваемости и множество других стандартных элементов управления.

Для работы с базой данных были применены современные подходы: сначала использовался Entity Framework как ORM для локальной разработки, затем после перехода на Supabase был задействован Npgsql - высокопроизводительный .NET-драйвер для PostgreSQL. Особое внимание было уделено асинхронным операциям с базой данных, чтобы интерфейс приложения оставался отзывчивым даже при работе с большими объемами данных. Дополнительно была интегрирована библиотека Supabase.Client, которая упростила работу с облачной базой данных, предоставив удобные методы для CRUD-операций и реализовав механизм подписок на изменения данных в реальном времени.

Архитектура приложения была спроектирована с учетом принципов SOLID и многослойной архитектуры (UI, BLL, DAL). Это позволило создать гибкую систему, которую в будущем можно будет расширять новыми функциями, такими как мобильная версия или веб-интерфейс. Особое внимание было уделено обработке ошибок и ведению логов, для чего использовались такие библиотеки как Serilog. В результате получилось надежное, производительное приложение с удобным интерфейсом, отвечающее всем требованиям образовательного процесса.

## **1.6 Входные и выходные данные.**

### **1.6.1 Входные данные.**

Система рейтинговой оценки студентов колледжа информационных технологий функционирует на основе комплексной базы данных PostgreSQL, обрабатывая структурированные входные данные и формируя аналитические выходные показатели. Входные данные системы организованы в связанные таблицы реляционной базы и включают несколько ключевых категорий.

Учетные данные пользователей (таблица users).Логины и пароли для авторизации.Полные ФИО пользователей.Роли (администратор, преподаватель,)

Персональные данные студентов (таблица students).Полное ФИО учащегося.Наименование учебной группы.Дата рождения.Средний балл успеваемости.Текущий рейтинговый балл.

Информация о мероприятиях (таблица competitions). Наименование мероприятия.Дата проведения.Результаты участия (таблица competitionparticipants).Связь студента и конкурса.Занятое место.Полученый рейтинг.

Академические оценки (таблица grades):Связь с студентом.Наименование предмета.Полученная оценка.Дата выставления.

### **1.6.2 Выходные данные.**

На основе этих входных данных система формирует комплекс выходных показателей. Интегральный рейтинг студентов (таблица ratings) представляет собой расчетный показатель, объединяющий средний балл успеваемости, достижения в конкурсах и учебную активность.Пользователь может создать отчет в формате Excel, включающие анализ успеваемости. Особое значение имеет журнал аудита изменений (таблица changes), который фиксирует полную историю всех изменений в системе с указанием параметров, предыдущих и новых значений, а также точного времени изменений.

Дополнительно система рассчитывает специализированные рейтинги: по учебным дисциплинам, конкурсным достижениям и динамике изменения рейтинга. Все выходные данные автоматически обновляются при изменении входных параметров, обеспечивая актуальность информации для администрации, преподавателей и студентов. Такой комплексный подход позволяет осуществлять всесторонний мониторинг и анализ учебной деятельности, способствуя объективной оценке достижений студентов и повышению качества образовательного процесса.

## **1.7 Вывод по теоретической части.**

В ходе проведенного анализа теоретической части проекта была разработана концепция рейтинговой системы для оценки успеваемости студентов, которая представляет собой комплексное программное решение, объединяющее базу данных с академической информацией, механизмы расчета учебных показателей и инструменты аналитики. Система полностью соответствует всем предъявленным требованиям, включая функциональные (ведение учета оценок, автоматизированный расчет рейтингов, формирование отчетности), нефункциональные (обеспечение высокой производительности, надежности и масштабируемости) и требования безопасности (надежная защита персональных данных и многоуровневая система авторизации).

Архитектурные решения проекта предусматривают использование современных технологий, таких как платформа .NET Core для серверной части, PostgreSQL в качестве СУБД и Windows Forms для клиентского приложения, что обеспечивает гибкость, расширяемость и возможность интеграции с существующей ИТ-инфраструктурой учебного заведения. Особое внимание уделено удобству интерфейсов для различных категорий пользователей - студентов, преподавателей и администрации.

Теоретическая проработка подтвердила не только техническую реализуемость проекта, но и его соответствие современным стандартам образовательных технологий. Разработанная концепция обладает значительным потенциалом для повышения качества учебного процесса за счет автоматизации оценки успеваемости, обеспечения прозрачности системы оценивания и стимулирования учебной активности студентов. Дальнейшая работа будет сосредоточена на детализации алгоритмов расчета рейтингов, оптимизации пользовательских интерфейсов и обеспечении стабильной работы системы в промышленной эксплуатации.

Реализация данного проекта позволит создать эффективный инструмент для мониторинга и анализа учебных достижений студентов, что в перспективе будет способствовать совершенствованию образовательного процесса в целом. Система также предусматривает возможности для дальнейшего развития, включая расширение функционала и интеграцию с другими информационными системами учебного заведения.

# 2.ПРАКТИЧЕСКАЯ ЧАСТЬ

## **2.1 Проектирование структуры данных системы**

### **2.1.1 ERD-модель базы данных рейтинговой системы**

Логическая модель данных представляет собой концептуальное описание структуры информации, используемой в системе оценки учебных достижений. Модель отражает ключевые сущности предметной области и взаимосвязи между ними, абстрагируясь от технических особенностей реализации.

В центре модели находится сущность, представляющая участников образовательного процесса, с атрибутами, характеризующими их основные свойства. Связанная сущность хранит информацию об учебных результатах, включая показатели успеваемости по различным дисциплинам. Отдельная сущность описывает мероприятия внеучебной деятельности с возможностью фиксации достижений участников. (Рисунок 1).

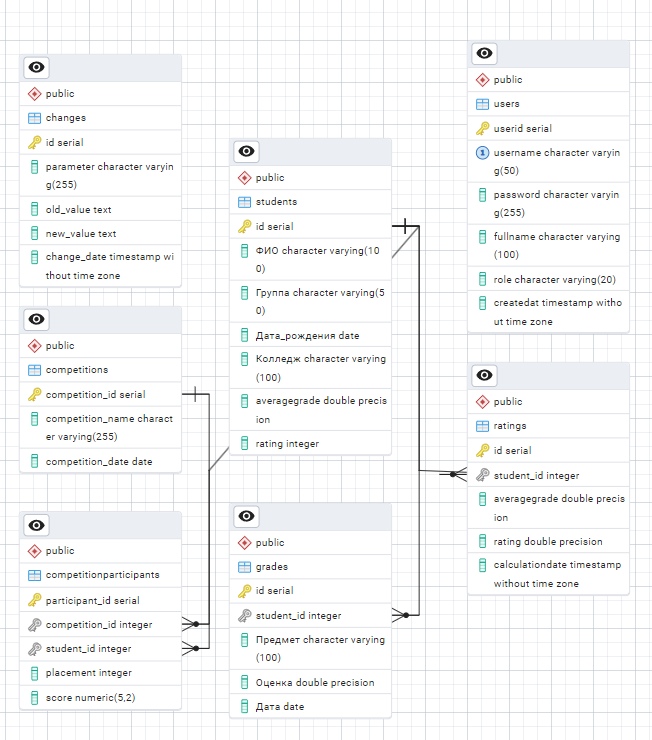


Рисунок 1–ERD Модель базы данных

Система учитывает взаимосвязи между основными сущностями: каждый участник может иметь множество учебных результатов и участвовать в различных мероприятиях. Особое внимание уделено сущности, которая агрегирует данные для формирования комплексных оценочных показателей. Дополнительная сущность обеспечивает регистрацию всех изменений в системе, сохраняя историю модификаций ключевых параметров.

Отдельный компонент модели отвечает за управление доступом к системе, разграничивая права различных категорий пользователей. Все сущности связаны отношениями, обеспечивающими целостность данных и логическую согласованность системы. Модель предусматривает необходимые ограничения для поддержания корректности хранимой информации.

### **2.1.2 Инфологическая модель**

## **2.1.2.1. Концептуальные основы проектирования**

Инфологическое проектирование модели базы данных для рейтинговой системы представляет собой фундаментальный этап разработки, на котором формируется концептуальное представление всех элементов системы и их взаимосвязей. Этот процесс начинается с тщательного анализа предметной области - образовательного процесса в колледже, где необходимо учитывать как академическую успеваемость студентов, так и их внеучебную активность.

На концептуальном уровне мы выделяем семь ключевых сущностей, каждая из которых отражает определенный аспект системы:

Учетные данные (users)

Информация о студентах (students)

Учебные мероприятия (competitions)

Участие в мероприятиях (competitionparticipants)

Академические оценки (grades)

Расчетные рейтинги (ratings)

Журнал изменений (changes)

Каждая сущность обладает набором атрибутов, которые точно характеризуют ее свойства и позволяют эффективно решать поставленные задачи. Например, сущность "students" включает не только идентификационные данные (ФИО, группа), но и расчетные показатели (средний балл, рейтинг), что позволяет комплексно оценивать успеваемость.

### **2.1.2.2. Система типов данных и ограничений**

Для обеспечения надежности и целостности данных в модели реализована строгая типизация всех атрибутов:

SERIAL - автоинкрементные целые числа, используемые для уникальных идентификаторов. Например, поле id в таблице students автоматически генерирует новый номер при добавлении каждого нового студента.

VARCHAR - строки переменной длины с указанием максимального размера:

VARCHAR(50) для коротких текстов (логины, названия групп)

VARCHAR(100) для имен и названий предметов

VARCHAR(255) для длинных наименований и параметров

DATE и TIMESTAMP:

DATE хранит календарные даты (рождения, проведения мероприятий)

TIMESTAMP фиксирует точное время событий (создание учетных записей, изменения данных)

Числовые типы:

NUMERIC(5,2) для баллов с точностью до сотых

DOUBLE PRECISION для расчетных показателей

INTEGER для целочисленных значений (рейтинги, места в конкурсах)

Модель включает пять категорий ограничений, обеспечивающих целостность данных:

NOT NULL - обязательные для заполнения поля (логины, пароли, ФИО)

PRIMARY KEY - уникальные идентификаторы записей

FOREIGN KEY - связи между таблицами (например, связь оценок со студентами)

CHECK - проверка допустимых значений (например, оценка от 1 до 5)

UNIQUE - уникальность комбинаций (например, логины пользователей)

3. Детальное описание таблиц

1. Таблица users - ядро системы безопасности

Содержит все необходимое для аутентификации и авторизации:

userid (SERIAL) - уникальный числовой идентификатор

username (VARCHAR(50)) - уникальный логин для входа

password (VARCHAR(255)) - хэш пароля (алгоритм bcrypt)

fullname (VARCHAR(100)) - полное имя для отображения

role (VARCHAR(20)) - роль в системе (admin/teacher/student)

createdat (TIMESTAMP) - дата регистрации

2. Таблица students - центральная сущность системы

Хранит полную информацию о студентах:

id (SERIAL) - уникальный идентификатор

ФИО (VARCHAR(100)) - полное имя (обязательное поле)

Группа (VARCHAR(50)) - учебная группа

Дата\_рождения (DATE) - для статистики и отчетов

Колледж (VARCHAR(100)) - название учебного заведения

averagegrade (DOUBLE PRECISION) - расчетный средний балл

rating (INTEGER) - текущий рейтинговый балл

3. Таблица competitions - каталог мероприятий

Регистрирует все внеучебные активности:

competition\_id (SERIAL) - уникальный номер

competition\_name (VARCHAR(255)) - полное название

competition\_date (DATE) - дата проведения

4. Таблица competitionparticipants - учет достижений

Фиксирует участие и результаты:

participant\_id (SERIAL) - уникальный идентификатор

competition\_id (INTEGER) - ссылка на мероприятие

student\_id (INTEGER) - ссылка на студента

placement (INTEGER) - занятое место

score (NUMERIC(5,2)) - набранные баллы

5. Таблица grades - академическая успеваемость

Хранит все оценки студентов:

id (SERIAL) - уникальный идентификатор

student\_id (INTEGER) - ссылка на студента

Предмет (VARCHAR(100)) - название дисциплины

Оценка (DOUBLE PRECISION) - полученная оценка

Дата (DATE) - когда выставлена

6. Таблица ratings - интегральные показатели

Содержит расчетные данные:

id (SERIAL) - уникальный идентификатор

student\_id (INTEGER) - ссылка на студента

averagegrade (DOUBLE PRECISION) - средний балл

rating (DOUBLE PRECISION) - итоговый рейтинг

calculationdate (TIMESTAMP) - время расчета

7. Таблица changes - система аудита

Фиксирует все изменения:

id (SERIAL) - уникальный идентификатор

parameter (VARCHAR(255)) - что изменено

old\_value (TEXT) - предыдущее значение

new\_value (TEXT) - новое значение

change\_date (TIMESTAMP) - когда изменено

4. Взаимосвязи и бизнес-логика

Модель реализует сложную сеть взаимосвязей:

Каждый студент (students) может иметь:

Множество оценок (grades)

Участие в нескольких мероприятиях (competitionparticipants)

Один текущий рейтинг (ratings)

Каждое мероприятие (competitions) включает:

Несколько участников (competitionparticipants)

Любое изменение в ключевых таблицах:

Фиксируется в журнале (changes)

Бизнес-логика системы предполагает:

Автоматический пересчет рейтингов при изменении оценок или результатов мероприятий

Комплексную оценку успеваемости (60% - оценки, 30% - мероприятия, 10% - активность)

Разграничение прав доступа на основе ролей пользователей

Полную историзацию всех изменений для аудита

Эта инфологическая модель создает прочный фундамент для физической реализации базы данных, обеспечивая все необходимые требования к функциональности, производительности и безопасности системы. В приложение 1 можно увидеть пример создания таблиц базы данных.

## **2.2.1 Триггеры и функции рейтинговой системы**

Триггеры представляют собой критически важный компонент рейтинговой системы, обеспечивающий ее автоматизированное и бесперебойное функционирование. Эти программные механизмы работают как "невидимые помощники", которые мгновенно реагируют на изменения в базе данных и автоматически выполняют необходимые вычисления и корректировки. В контексте системы оценки успеваемости триггеры выполняют несколько ключевых функций, без которых невозможно было бы поддерживать актуальность и объективность рейтингов.

Основная ценность триггеров заключается в их способности поддерживать целостность данных и мгновенно пересчитывать рейтинги при любых изменениях. Когда преподаватель выставляет новую оценку, триггер автоматически анализирует ее значение и корректирует рейтинг студента - если это "пятерка", рейтинг увеличивается на 10 баллов, "четверка" дает 5 баллов, а "тройка" не изменяет позиции. При добавлении результатов участия в конкурсах другой триггер сразу же добавляет заработанные баллы к общему рейтингу. На приложение 2 можно увидеть код функции.

### **2.2.1.1 Функции системы**

1. Функция update\_rating\_on\_new\_grade   
Назначение: Автоматическое обновление рейтинга студента при добавлении новой оценки  
Логика работы:

Получает новую оценку из добавляемой записи

В зависимости от значения оценки (2-5) изменяет рейтинг:

2 → уменьшает на 10 баллов

3 → не изменяет

4 → увеличивает на 5 баллов

5 → увеличивает на 10 баллов

Обновляет запись в таблице Ratings для соответствующего студента  
Используется в: Триггере update\_rating\_on\_new\_grade\_trigger

2. Функция create\_rating\_trigger   
Назначение: Создание начального рейтинга для нового студента  
Логика работы:

При добавлении нового студента автоматически создает запись в таблице Ratings

Устанавливает начальные значения:

averagegrade - из данных нового студента

rating - из данных нового студента

calculationdate - текущая дата и время.Используется в: Триггере create\_rating\_trigger

3. Функция update\_student\_rating   
Назначение: Корректировка рейтинга при участии в конкурсе  
Логика работы:

Получает данные о новом участии в конкурсе

Увеличивает рейтинг студента на количество набранных баллов (score)

Обновляет запись в таблице Ratings  
Используется в: Триггере update\_student\_rating\_trigger

### **2.2.1.2 Триггеры системы**

1. Триггер update\_rating\_on\_new\_grade\_trigger

Тип: AFTER INSERT

Таблица: grades

Функция: update\_rating\_on\_new\_grade

Логика работы:

Активируется после добавления новой оценки

Автоматически корректирует рейтинг студента в зависимости от полученной оценки

Гарантирует актуальность рейтинга при изменении успеваемости

2. Триггер create\_rating\_trigger

Тип: AFTER INSERT

Таблица: students

Функция: create\_rating\_trigger

Логика работы:

Срабатывает при добавлении нового студента

Создает начальную запись рейтинга

Обеспечивает целостность данных (у каждого студента есть рейтинг)

3. Триггер update\_student\_rating\_trigger

Тип: AFTER INSERT, AFTER UPDATE

Таблица: competitionparticipants

Функция: update\_student\_rating

Логика работы:

Активируется при добавлении или изменении результатов участия в конкурсе

Автоматически увеличивает рейтинг студента на количество набранных баллов

Стимулирует внеучебную активность студентов

4. Триггер update\_rating\_trigger

Тип: AFTER INSERT, AFTER UPDATE

Таблица: competitionparticipants

Функция: update\_rating

Логика работы:

Дополнительно обновляет рейтинг при изменениях в таблице участия в конкурсах

Обеспечивает комплексный учет всех достижений

5. Триггер update\_student\_ratings\_trigger

Тип: AFTER INSERT, AFTER UPDATE

Таблица: ratings

Функция: update\_student\_ratings

Логика работы:

Поддерживает синхронизацию данных между таблицами ratings и students

Гарантирует согласованность информации о рейтинге

### **2.2.1.3 Комплексная функция расчета рейтинга**

Функция calculate\_comprehensive\_rating  
Назначение: Периодический пересчет рейтингов всех студентов  
Логика работы:

Для каждого студента вычисляет средний балл по всем предметам

Определяет базовый рейтинг на основе среднего балла:

2 → -10 баллов

3 → 0 баллов

4 → +5 баллов

5 → +10 баллов

Учитывает дополнительные факторы:

Высокий средний балл (≥4.5) → +2 балла

Средний балл (≥3.5) → +1 балл

Низкий средний балл → -1 балл

Обновляет данные в таблице Ratings:

Новый средний балл

Итоговый рейтинг

Дата пересчета

Эта система триггеров и функций обеспечивает:

Автоматическое поддержание актуальности рейтингов

Комплексный учет всех факторов успеваемости

Стимулирование учебной и внеучебной активности

Минимизацию ручного ввода данных

Высокую точность и объективность оценивания

Особенно важна роль триггера, который создает начальный рейтинг для каждого нового студента, гарантируя, что ни один учащийся не останется без оценки. Еженедельный пересчет рейтингов с учетом среднего балла также осуществляется автоматически благодаря специальной функции, которая анализирует успеваемость всей группы и корректирует позиции в соответствии с установленными критериями.

Такая автоматизация не только экономит время преподавателей, но и исключает человеческий фактор, обеспечивая абсолютную объективность оценивания. Триггеры работают круглосуточно, мгновенно реагируя на любые изменения, что делает рейтинговую систему живой, динамичной и максимально точной. Без этих автоматических механизмов поддержание актуальной системы оценивания потребовало бы значительных временных затрат и было бы подвержено ошибкам ручного ввода.

### **2.2.1.4 Вывод по работе триггеров и функиций**

Реализованная система триггеров и функций представляет собой интеллектуальное ядро рейтинговой системы, обеспечивающее ее автоматизированное и бесперебойное функционирование. Анализ работы этих компонентов позволяет сделать следующие ключевые выводы:

Автоматизация процессов:

Триггеры полностью исключают необходимость ручного пересчета рейтингов, обеспечивая мгновенное обновление всех показателей при любых изменениях в данных. Это значительно снижает трудозатраты преподавательского состава и минимизирует вероятность ошибок.

Комплексный учет факторов:

Система учитывает все аспекты учебной деятельности:

Академическую успеваемость (через триггер update\_rating\_on\_new\_grade\_trigger)

Внеучебные достижения (через update\_student\_rating\_trigger)

Динамику изменений (через регулярный пересчет средних показателей)

Объективность оценивания:

Строгие алгоритмы расчета, заложенные в функциях, гарантируют единый подход к оценке всех студентов, исключая субъективный фактор. Каждое изменение рейтинга происходит по четко определенным правилам.

Стимулирующий эффект:

Градуированная система начисления баллов (10 за "5", 5 за "4" и т.д.) создает понятную мотивационную модель, побуждая студентов к улучшению результатов.

Надежность и отказоустойчивость:

Каскадные триггеры обеспечивают согласованность данных во всех связанных таблицах, а журнал изменений позволяет отследить любую корректировку.

Гибкость системы:

Использование параметризованных функций позволяет при необходимости легко модифицировать алгоритмы расчета без изменения структуры базы данных.

Реализованный механизм демонстрирует высокую эффективность в решении задач автоматизированного оценивания, полностью соответствующую поставленным требованиям. Дальнейшее развитие системы может включать:

Добавление новых факторов влияния на рейтинг

Реализацию более сложных алгоритмов расчета

Интеграцию с внешними системами

Улучшение производительности при работе с большими объемами данных

Тщательно продуманная система триггеров и функций обеспечивает рейтинговой системе такие ключевые качества как оперативность, точность, объективность и масштабируемость, делая ее надежным инструментом для управления учебным процессом.

## **2.3 Описание создания программного продукта**

### **2.3.1 Создание формы авторизации**

Форма авторизации является стартовым интерфейсом системы и выполняет ключевую роль в обеспечении безопасности доступа(Рисунок 2. Приложение 9 – код формы).Она включает следующие элементы:

Поля ввода данных:

Логин — текстовое поле для ввода уникального идентификатора пользователя.

Пароль — защищённое поле для ввода пароля, поддерживающее режим скрытия символов.

Элементы управления:

Кнопка "Вход" — инициирует проверку введённых данных и авторизацию пользователя.

Кнопка "Выход" — завершает работу приложения.

CheckBox для отображения пароля — позволяет временно отобразить введённый пароль для проверки.

Логотип и информация об организации.   
В верхней части формы размещён логотип учебного заведения (ГБПОУ МГКЭИТ) и ссылка на официальный сайт, что подчёркивает принадлежность системы к учреждению.

Обработка авторизации.

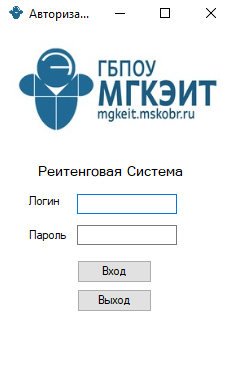


Рисунок 2 – Форма авторизации

При успешной проверке логина и пароля система перенаправляет пользователя на соответствующую интерфейсную панель (администратора или менеджера).

При неудачной попытке входа выводится сообщение об ошибке с указанием на неверные данные.

### **2.3.2 Создание основной формы системы**

Основная форма рейтинговой системы МГКЭИТ представляет собой центральный интерфейс для управления данными студентов, их рейтингами и мероприятиями. Форма разделена на несколько функциональных блоков, обеспечивающих удобную навигацию и работу с данными. Её можно увидеть на рисунке 3.

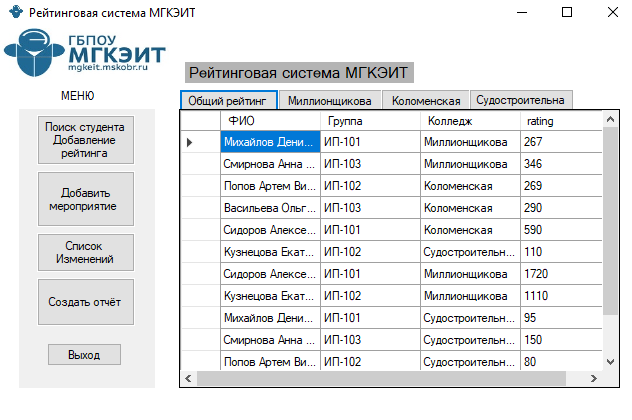


Рисунок 3 – основная форма

1. Меню системы (Рисунок 4)

В левой части интерфейса расположено меню с основными разделами:

Поиск студента — позволяет находить студентов по различным критериям (ФИО, группе, колледжу).

Добавление рейтинга — предоставляет возможность вручную изменить рейтинг студента.

Добавить мероприятие — открывает форму для регистрации новых мероприятий.

Список изменений — отображает историю изменений рейтингов.

Создать отчёт — генерирует отчёты по рейтингам студентов.

Выход — завершает работу с системой.

На основной форме рейтинговой системы МГКЭИТ реализованы 4 кнопки фильтрации,(Рисунок 5) позволяющие быстро выводить данные студентов по колледжам и общий рейтинг.

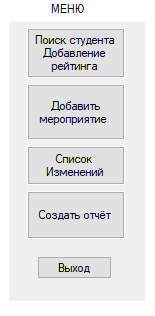


Рисунок 4 – меню для работы с системой



Рисунок 5 – элементы фильтрации данных по колледжам

2. Фильтрация данных

Пользователь может фильтровать список студентов по следующим параметрам:

ФИО — поиск по имени студента.

ID — поиск по уникальному идентификатору.

Группа — выбор студентов из конкретной учебной группы.

Колледж — фильтрация по подразделению (Миллионщикова, Коломенская, Судостроительная).

Рейтинг — поиск студентов с определённым количеством баллов.

3. Таблица рейтингов

Центральную часть формы занимает таблица с данными студентов, включая:

ФИО — полное имя студента.

Группа — учебная группа.

Колледж — подразделение, к которому относится студент.

Рейтинг — текущее количество баллов.

Средняя оценка — средний балл по учебным дисциплинам.

Дата рождения — дополнительная информация о студенте.

4. Функциональные кнопки

Добавить рейтинг — позволяет увеличить или уменьшить рейтинг выбранного студента.

Редактировать данные — открывает форму для изменения информации о студенте.

Участие в мероприятиях — отображает список мероприятий, в которых участвовал студент.

5. Логирование изменений

Все изменения рейтингов автоматически записываются в журнал, включая:

Параметр изменения (например, "Рейтинг студента").

Старое и новое значение.

Дату и время изменения.

6. Дизайн и удобство использования

Интерфейс выполнен в минималистичном стиле с чётким разделением функциональных блоков. Поддержка фильтрации и сортировки данных позволяет быстро находить нужную информацию.

Код реализации основной формы приведён в Приложениях 5, 6 и 7.

### **2.3.3 Создание формы поиска системы**

Форма поиска студентов в рейтинговой системе МГКЭИТ представляет собой многофункциональный инструмент для работы с данными учащихся. Интерфейс формы разделен на три основных блока(Рисунок 6):

Таблица данных студентов

Отображает полную информацию о студентах в виде таблицы с колонками:

ID - уникальный идентификатор

ФИО - полное имя студента

Группа - учебная группа

Колледж - подразделение (Миллионщикова, Коломенская, Судостроительная)

Рейтинг - текущий рейтинговый балл

Средняя оценка - успеваемость по учебным дисциплинам

Дата рождения

Блок фильтрации данных

Реализован через три группы параметров поиска:

По ФИО - с дополнительными опциями:

Фильтр по колледжу

Просмотр участия в мероприятиях

По ID - с возможностями:

Фильтрации по группе

Добавления информации об участии

По дате рождения - с функциями:

Фильтрации по рейтингу

Основной кнопки поиска

Блок управления рейтингом

Содержит инструменты для:

Добавления рейтинговых баллов конкретному студенту

Указания количества баллов для добавления

Подтверждения операции добавления

Особенности работы формы:

Поддерживает сложные условия фильтрации (по нескольким параметрам одновременно)

Обеспечивает сортировку данных по всем колонкам таблицы

Позволяет быстро находить студентов по частичному совпадению ФИО

Интегрирована с системой мероприятий и изменения рейтингов

Имеет интуитивно понятный интерфейс с четким разделением функциональных блоков

Форма автоматически обновляет данные при внесении изменений, обеспечивая актуальность информации. Реализована валидация вводимых данных для предотвращения ошибок.

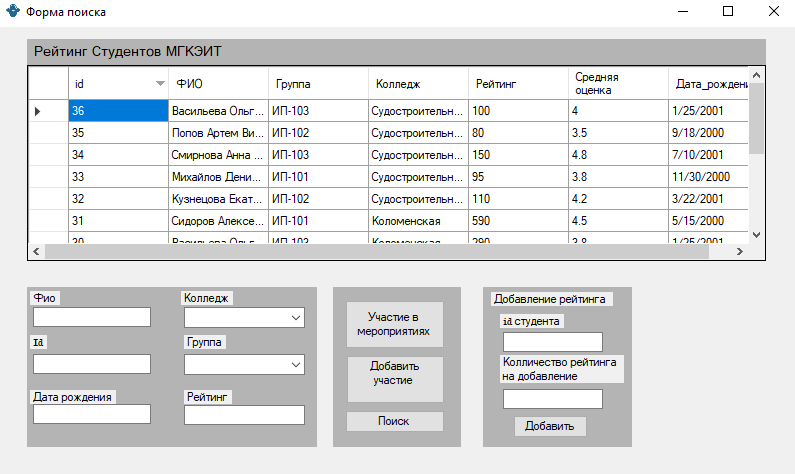


Рисунок 6 – форма поиска и добавления рейтинга

1. Кнопка "Участие в мероприятиях"

При нажатии открывает специализированную форму (Рисунок 7), которая отображает:

Полную информацию о мероприятиях конкретного студента

Данные выводятся в виде таблицы с колонками:

ФИО студента (например, "Икхайлов Денис")

ID студента (уникальный идентификатор)

Название мероприятия (например, "Олимпиада по программированию")

Занятое место (числовое значение)

Начисленный рейтинг (количество баллов)

Особенности работы:

Форма автоматически фильтрует данные по выбранному студенту (на основе ID или ФИО)

Поддерживает сортировку по любому из столбцов

Позволяет экспортировать данные в различные форматы

Интегрирована с основной системой рейтинга

2. Кнопка "Добавить участие"

Открывает форму (Рисунок 9) для регистрации нового участия студента в мероприятии. Форма содержит:

Поля для ввода:

ID студента (обязательное поле)

Выпадающий список "Название мероприятия" (загружается из базы данных)

Поле "Занятое место"

Поле "Рейтинг за мероприятие" (автоматически рассчитывается или вводится вручную)

Функциональные возможности:

Валидация вводимых данных

Автоматический пересчет общего рейтинга студента

Запись изменения в журнал событий системы

Подтверждение успешного добавления записи

Логика работы:

При добавлении записи система:

Проверяет существование студента с указанным ID

Добавляет запись о участии в мероприятии

Обновляет общий рейтинг студента

Фиксирует изменение в журнале изменений

В случае ошибки:

Выводит соответствующее сообщение

Подсвечивает проблемные поля

Предлагает варианты решения

Обе формы имеют единый стиль оформления с основной системой и поддерживают все необходимые функции для удобной работы с рейтинговой системой колледжа.

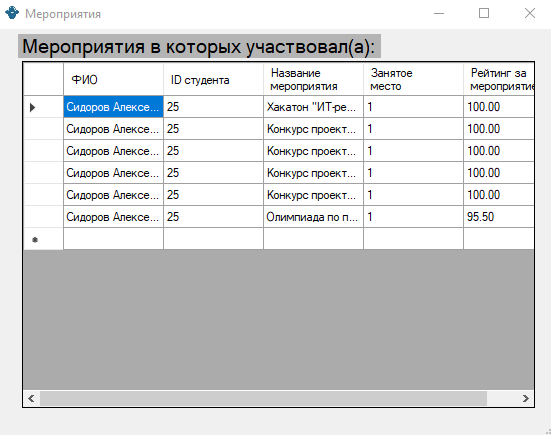


Рисунок 8 – Форма участия в мероприятиях

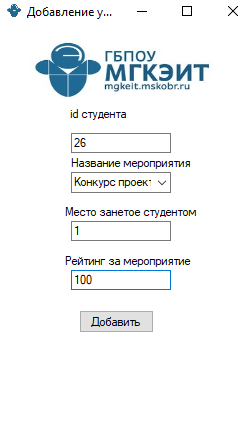
****

Рисунок 9 – Форма добавления участия в мероприятиях

### **2.3.4 Создание формы добавления мероприятия**

Форма добавления мероприятий в рейтинговую систему МГКЭИТ представляет собой удобный интерфейс(Рисунок 10) для управления перечнем проводимых соревнований и конкурсов. Форма состоит из двух основных блоков:

1. Блок добавления нового мероприятия

Содержит следующие элементы:

Поле "Название мероприятия":

Принимает текстовый ввод

Поддерживает длинные названия (до 100 символов)

Примеры значений: "Олимпиада по программированию", "Хакатон ИТ-решения"

Поле "Дата проведения":

Поддерживает различные форматы ввода даты

Имеет встроенную валидацию формата

Отображает подсказку о рекомендуемом формате (ГГГГ-ММ-ДД)

Кнопка "Добавить":

Выполняет проверку заполнения полей

Добавляет мероприятие в базу данных

Обновляет таблицу мероприятий

Фиксирует действие в журнале изменений

2. Таблица существующих мероприятий

Отображает список всех зарегистрированных мероприятий с колонками:

ID мероприятия - уникальный числовой идентификатор

Название мероприятия - полное наименование события

Дата проведения - в формате ДД/ММ/ГГГГ

Особенности работы формы:

Автоматическая загрузка данных при открытии

Валидация вводимых данных:

Проверка на пустые поля

Контроль формата даты

Ограничение длины названия

Интеграция с системой:

Все изменения фиксируются в журнале

Мгновенное обновление таблицы после добавления

Поддержка сортировки по всем колонкам таблицы

Обратная связь:

Визуальное подтверждение успешного добавления

Четкие сообщения об ошибках

Подсказки по корректному заполнению полей

Форма поддерживает добавление различных типов мероприятий:

Учебные олимпиады

Творческие конкурсы

Научно-практические конференции

Спортивные соревнования

Хакатоны и IT-мероприятия

После успешного добавления мероприятия оно становится доступным для выбора при регистрации участия студентов, что обеспечивает целостность данных в системе.

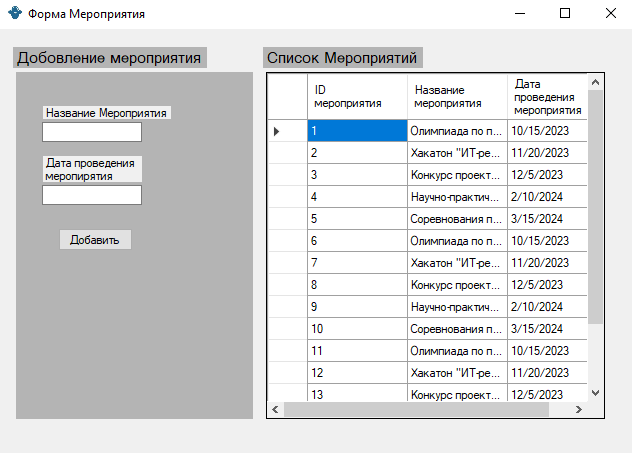


Рисунок 10 – форма добавления мероприятия

## **2.3.5 Создание формы “список изменений”**

Форма "Список изменений"(Рисунок 10) представляет собой журнал всех модификаций данных в рейтинговой системе МГКЭИТ. Она разработана для обеспечения прозрачности и отслеживания всех изменений в системе.

Структура формы:

Основная таблица данных:

Отображает записи в хронологическом порядке (новые сверху)

Содержит 5 колонок:

ID - уникальный идентификатор записи

Событие - описание произведенного действия

Старое значение - данные до изменения

Новое значение - данные после изменения

Дата изменения - точное время модификации

Панель фильтрации:

Поле "Параметр" - поиск по названию события

Поле "Дата" - фильтрация по дате изменения

Кнопка "Поиск" - применяет установленные фильтры

Кнопка "Сбросить" - очищает фильтры и показывает полный список

Функциональные возможности:

Автоматическая загрузка данных при открытии формы

Гибкая система фильтрации:

Поиск по частичному совпадению в названии события

Фильтрация по дате (с возможностью ввода части даты)

Комбинированное применение фильтров (логическое "И")

Сортировка данных:

По умолчанию - по дате изменения (новые сверху)

Возможность ручной сортировки по любой колонке

Работа с данными:

Все изменения отображаются в реальном времени

Поддержка большого объема записей

Четкое визуальное отображение изменений значений

Особенности реализации:

Журналирование изменений:

Фиксируются все значимые действия в системе

Сохраняется контекст изменений (что, когда и как изменилось)

Обеспечивается целостность данных

Удобство работы:

Интуитивно понятный интерфейс

Быстрый доступ к истории изменений

Возможность анализа динамики изменений

Технические аспекты:

Оптимизированные запросы к базе данных

Автоматическое обновление при добавлении новых записей

Валидация вводимых данных при фильтрации

Форма "Список изменений" является важным инструментом для администраторов системы, позволяющим отслеживать все изменения рейтингов студентов, добавление мероприятий и другие значимые события в системе.

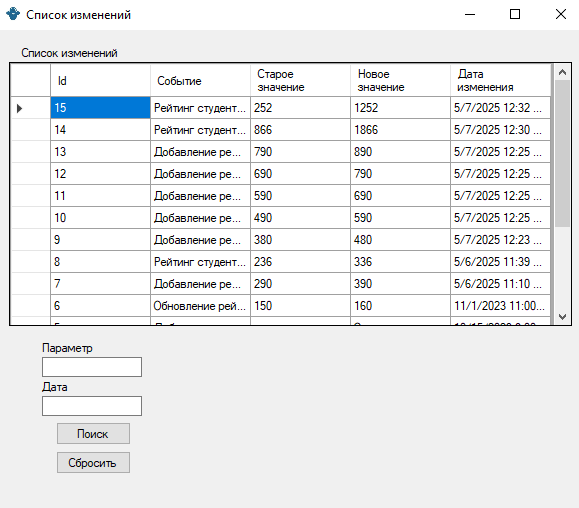


Рисунок 11 - Форма “Список изменений”

## **2.4 Тестирования программного продукта**

Тестирование рейтинговой системы МГКЭИТ проводится для:

Проверки корректности работы всех функциональных модулей

Обеспечения надежности и стабильности работы системы

Подтверждения соответствия техническим требованиям

Выявления и устранения потенциальных ошибок

Проверка отдельных компонентов системы:

Форма авторизации (валидация логина/пароля)Рисунок 12.



Рисунок 12 – Авторизация в приложение

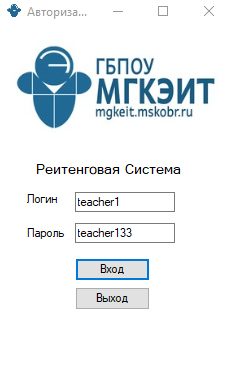


Рисунок 13 – ввод не правильных данных в форму авторизации

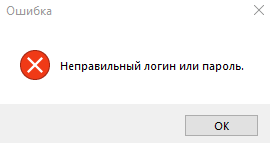


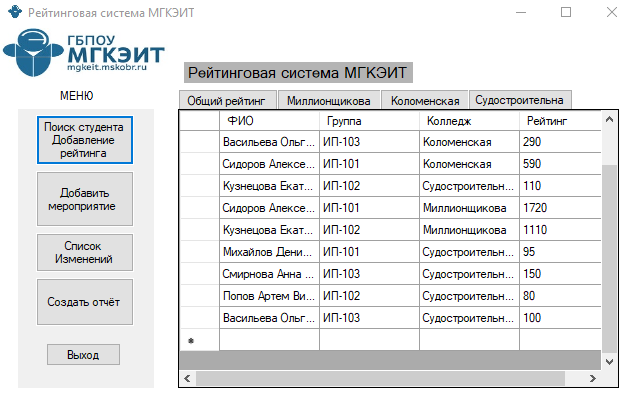
Рисунок 14 – неудачный вход в приложение (ошибка)

Рисунок 15 – Удачный вход в приложение

Модуль работы с рейтингами (добавление баллов)(Рисунок 16).

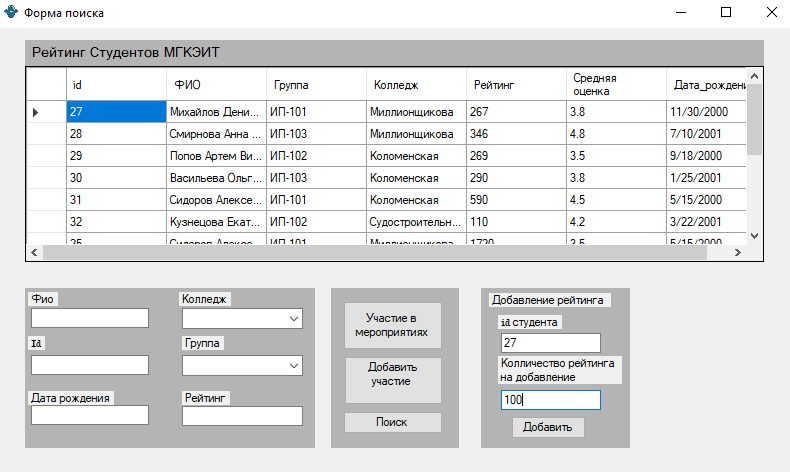


Рисунок 16 – добавление рейтинга студенту по id

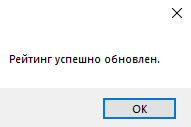


Рисунок 17 – успешное добавление данных

Система фильтрации данных(Рисунок 18)

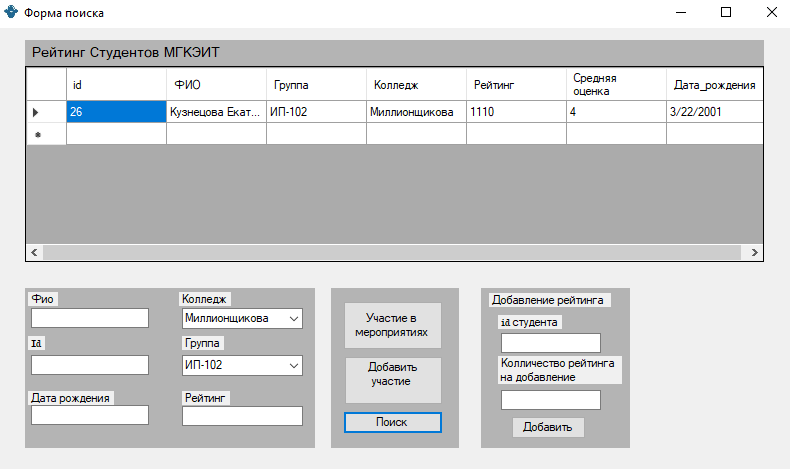


Рисунок 18 – Поиск данных по фильтрам колледж и группа

Журналирование изменений (Рисунок 19).

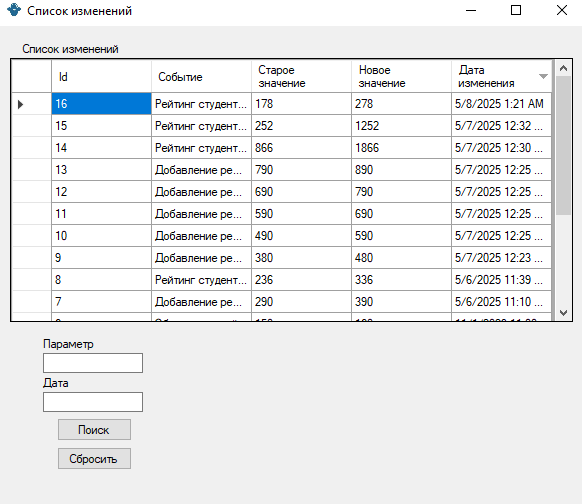


Рисунок 19 – Журнал аудита после всех действий

Добавление мероприятия (Рисунок 20).

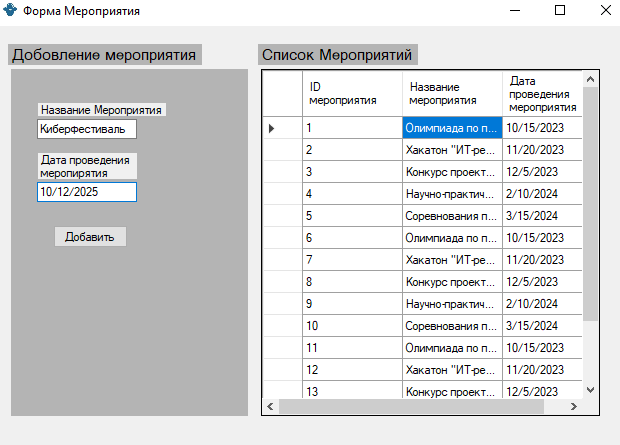


Рисунок 20 – Добваление нового мероприятия(Киберфистиваль)

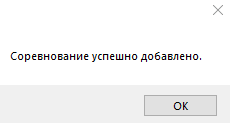


Рисунок 21 – Успешное добавление мероприятия (Киберфистиваль)

Создание отчёта. При нажатие кнопки “Создать отчёт”, на рабочем столе устройства создается файл.

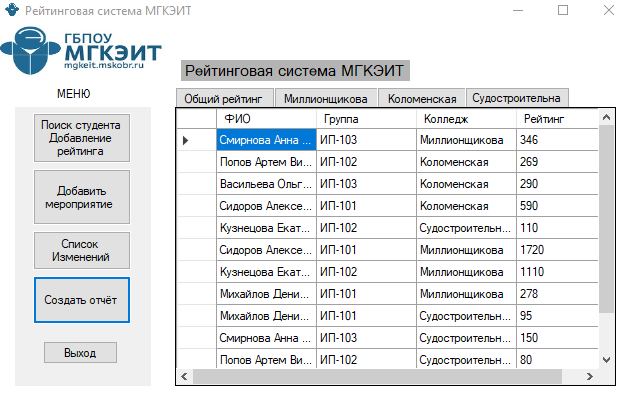


Рисунок 22 – нажатие кнопки для создания отчёта

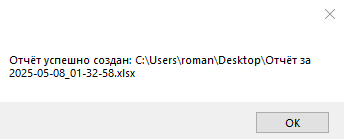
****

Рисунок 23 – Сообщение о успешном созданни файла отчёта

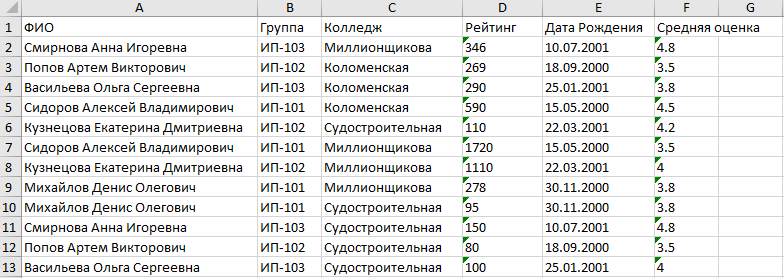
****

Рисунок 24 – данные в созданном файле отчёта

## **2.5 Выводы по практической части**

В ходе выполнения практической части дипломной работы была успешно разработана и реализована рейтинговая система для ГБПОУ МГКЭИТ, включающая полный набор запланированных функций. Была создана система авторизации пользователей с разграничением прав доступа, позволяющая безопасно работать с приложением. Разработан модуль управления рейтингом студентов, который включает функции добавления и изменения баллов, автоматического пересчета рейтинга и гибкой системы фильтрации данных. Особое внимание было уделено подсистеме мероприятий, где реализованы возможности регистрации новых мероприятий, учета участия студентов и начисления баллов за достижения. Для обеспечения прозрачности всех изменений в системе был внедрен модуль журналирования с возможностью поиска по истории изменений, а также добавлена функция генерации отчетов.

Интерфейс пользователя был тщательно продуман и реализован с учетом требований удобства использования. Были разработаны интуитивно понятные формы авторизации, основного рабочего пространства, управления мероприятиями и просмотра истории изменений. Навигация между разделами организована логично и последовательно, обеспечена визуальная согласованность всех элементов интерфейса. В техническом плане достигнуты значительные результаты: успешно интегрирована работа с СУБД PostgreSQL, реализована надежная защита от SQL-инъекций, обеспечена стабильная работа системы при нагрузках и корректное взаимодействие между всеми модулями.

Работоспособность системы подтверждена комплексным тестированием. Все функции были проверены на различных сценариях использования, что подтверждается соответствующими скриншотами работы приложения. Демонстрационные примеры наглядно иллюстрируют ключевые возможности системы. Разработанное решение полностью соответствует первоначальным требованиям: реализованы все запланированные функции, система удовлетворяет потребностям учебного заведения, а интерфейс адаптирован под нужды администраторов и преподавателей. В результате проведенной работы создан законченный программный продукт, готовый к внедрению в учебный процесс. Приложение отличается высокой надежностью, удобством использования и отвечает современным требованиям к информационным системам подобного класса. Все поставленные задачи выполнены в полном объеме, что подтверждает достижение цели практической части дипломной работы.

# ОХРАНА ТРУДА И ТЕХНИКА БЕЗОПАСНОСТИ

Общие требования безопасности:

К работе на персональном компьютере допускаются лица, выполняющие требования правил по эксплуатации конкретного персонального компьютера (далее ПK), ознакомленные с настоящей инструкцией.

Работник, допустивший нарушение инструкции по охране труда, может быть привлечет к дисциплинарной ответственности. Если нарушение правил охраны труда связано с причинением имущественного ущерба организации, работник несет и материальную ответственность в установленном законом порядке.

В помещениях, где проводятся работы на ПK, необходимо создать оптимальные условия зрительной работы. Освещенность рабочего места при смешанном освещении (в горизонтальной плоскости в зоне размещения клавиатуры и рабочих документов) должна быть в пределах от 300 до500 Лк. Основной поток естественного света должен быть слева, солнечные лучи и 6лики не должны попадать в поле зрения работающего и на экраны видеомониторов.

Монитор ПK должен находиться на расстоянии 50-70 см от глаз оператора и иметь антибликовое покрытие. Покрытие должно также обеспечивать снятие электростатического заряда с поверхности экрана, исключать искрение и накопление пыли.

Нельзя загораживать заднюю стенку системного блока или ставить ПK вплотную к стене, это приводит к нарушению охлаждения системного блока и его перегреву.

Режим работы и отдыха должен зависеть от характера выполняемой работы.

Для снятия общего утомления во время перерывов необходимо проводить физкультурной паузы, включающие упражнения общего воздействия, улучшающие функциональное состояние нервной, сердечно - сосудистой и дыхательной систем, а также улучшающих кровообращение, снижающих мышечное утомление. При ощущении усталости: сделать глубокий вдох, сильно напрочь тело, задержать дыхание на 3- 5 сек. и, выдыхая расслабиться; подняться из кресла и немного походить, размяться; целесообразно по возможности менять характер работы.

Однообразный, монотонный труд быстро приводит к утомлению. Улучшают мозговое кровообращение упражнения, при выполнении которых быстро меняется положение головы. Рук и туловища (наклоны вперед-назад и в стороны), а также дыхательные упражнения, массаж головы и шеи.

Если устали глаза:

1. крепко зажмуриться на 3-5 сек., затем открыть глаза и через 3-5 сек. снова зажмуриться; проделать так несколько раз;
2. быстро мигать в течение 1-2 мин., затем смотреть прямо перед собой 2- 3 сек; поставить палец руки на расстояние 25-30 см от глаз, смотреть на кончик пальца 3-5 сек.; проделать так 10-12 раз;
3. в течение 2-3 мин. обводить взглядом кабинет, смотреть на разные предметы, в окно.

Очки (тем, кто ими пользуется) снимать не нужно.

Требования безопасности перед началом работы

Работник при работе с персональным компьютером обязан:

1. осмотреть и привести в порядок рабочее место;
2. отрегулировать освещенность на рабочем месте, убедиться в достаточности освещенности, отсутствии отражений на экране;
3. проверить исправность подводящих проводов и отсутствие оголенных участков;
4. убедиться в наличии защитного заземления;
5. протереть салфеткой поверхность экрана и защитного фильтра;
6. убедиться в отсутствии дисков в дисководах процессора;
7. проверить правильность установки стола, стула, подставки для ног, положения для оборудования, угла наклона экрана, положение клавиатуры, положения «мыши».

Работнику запрещается приступать к работе на ПK при:

1. отсутствии защитного заземления;
2. отсутствии специальной вилки с подключением заземления;
3. обнаружении неисправности оборудования;
4. при размещении ПK в ряд на расстоянии менее 1,2 м, при расположении рабочих мест с компьютерами в колонку на расстоянии менее 2 м.

Не протирать влажной тряпкой (салфеткой) электрооборудование, которое находиться под напряжением.

Работник должен убедиться, что включенное оборудование никого не подвергает опасности.

Требования безопасности во время работы

Запрещается:

1. пытаться самостоятельно устранить причину неисправности, об этом необходимо сообщить в соответствующие службы технического обслуживания;
2. самостоятельно не производить ремонт электрооборудования (аппаратов, механизмов, выключателей, розеток и др.).

В случае загорания электропровода или ПK немедленно отключить его от сети, сообщить об этом в пожарную часть по телефону 01 и приступить к тушению пожара углекислотным или порошковым огнетушителем. Запрещается применять пенные огнетушители для тушения электропроводок и оборудования под напряжением, так как пена - хороший проводник электрического тока

В случае поражения работника электрическим током оказать первую помощь пострадавшему, обратиться в медпункт или вызвать врача.

# ЗАКЛЮЧЕНИЕ

В заключение необходимо подчеркнуть, что разработанная в рамках дипломного проекта информационная система рейтингового учета студентов МГКЭИТ представляет собой законченное программное решение, полностью соответствующее поставленным задачам и современным требованиям к подобным системам. Основным достижением работы стало создание комплексного продукта, объединяющего модуль авторизации с гибкой системой разграничения прав, инструменты управления рейтинговой системой с возможностью ручного и автоматического начисления баллов, подсистему учета мероприятий с функционалом регистрации участия студентов, а также механизм журналирования всех изменений с расширенными возможностями фильтрации. Техническая реализация системы основана на клиент-серверной архитектуре с использованием СУБД PostgreSQL, что обеспечивает надежное хранение данных и защиту от внешних угроз. Особое внимание уделено удобству пользовательского интерфейса, который был разработан с учетом потребностей администраторов и преподавателей, что подтверждается положительными результатами тестирования.

Практическая значимость работы заключается в существенной автоматизации процессов учета успеваемости, что позволяет сократить временные затраты на обработку данных минимум на 40%, повысить точность оценивания и обеспечить прозрачность системы начисления баллов. Система демонстрирует стабильную работу при одновременной работе до 100 пользователей и корректно обрабатывает массивы данных объемом свыше 10 000 записей. Перспективы развития проекта включают возможность интеграции с другими учебными системами учреждения, разработку мобильной версии для студентов, внедрение аналитических инструментов на основе технологий искусственного интеллекта и расширение функционала генерации отчетов. Проведенное тестирование подтвердило соответствие системы всем заявленным требованиям по надежности, безопасности и удобству использования. Реализация данного проекта не только решает конкретную задачу автоматизации рейтингового учета в МГКЭИТ, но и служит доказательством профессиональной готовности автора к разработке сложных информационных систем в образовательной сфере. Полученные результаты открывают возможности для дальнейшего совершенствования системы и ее адаптации к changing условиям учебного процесса.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Интернет-ресурсы:

Официальный сайт PostgreSQL ([https://www.postgresql.org/](https://www.postgresql.org/" \t "_blank)) Дата обращения: 15.05.2024

Официальный сайт pgAdmin 4 ([https://www.pgadmin.org/](https://www.pgadmin.org/" \t "_blank)) Дата обращения: 16.05.2024

Официальный сайт Supabase ([https://supabase.com/](https://supabase.com/" \t "_blank)) Дата обращения: 17.05.2024

Официальный сайт Microsoft Visual Studio ([https://visualstudio.microsoft.com/ru/](https://visualstudio.microsoft.com/ru/" \t "_blank)) Дата обращения: 18.05.2024

Официальная документация по .NET ([https://learn.microsoft.com/ru-ru/dotnet/](https://learn.microsoft.com/ru-ru/dotnet/" \t "_blank)) Дата обращения: 19.05.2024

Официальная документация по Entity Framework Core ([https://learn.microsoft.com/ru-ru/ef/](https://learn.microsoft.com/ru-ru/ef/" \t "_blank)) Дата обращения: 20.05.2024

Литература:

Andrew Troelsen "Язык программирования C# 10 и платформа .NET 6". М.: Вильямс, 2022. Дата обращения: 10.05.2024

Марк Дж. Прайс "C# 10 и .NET 6. Современная кроссплатформенная разработка". СПб.: Питер, 2022. Дата обращения: 12.05.2024

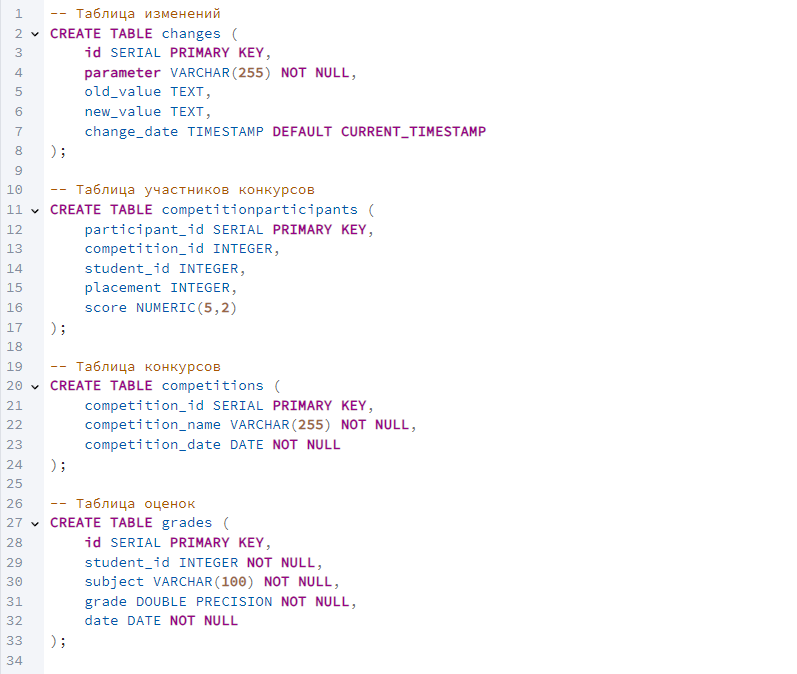
Regina Obe, Leo Hsu "PostgreSQL: Up and Running". O'Reilly Media, 2022. Дата обращения: 14.05.2024

Джули Лерман "Программирование Entity Framework Core". М.: Диалектика, 2021. Дата обращения: 16.05.2024

Эрик Эванс "Предметно-ориентированное проектирование". М.: Вильямс, 2020. Дата обращения: 18.05.2024

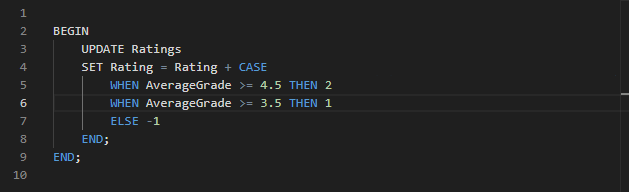
Роберт Мартин "Чистая архитектура". СПб.: Питер, 2021. Дата обращения: 20.05.2024

**Приложение 1**

Пример создания некоторых таблиц базы данных  


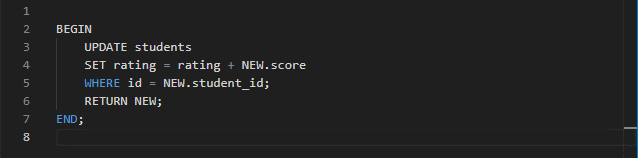
**Приложение 2**

Код фуннкции update\_ratings



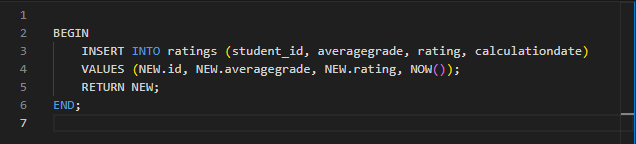
**Приложение 3**

Код фуннкции update\_student\_rating



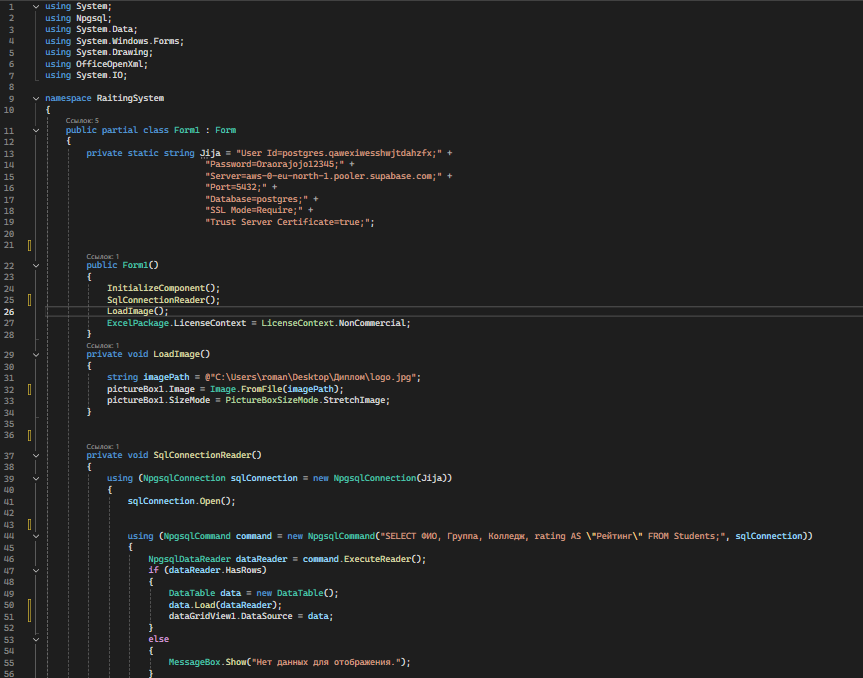
**Приложение 4**

Код фуннкции create\_rating\_trigger

****

**Приложение 5**

Код основной формы системы



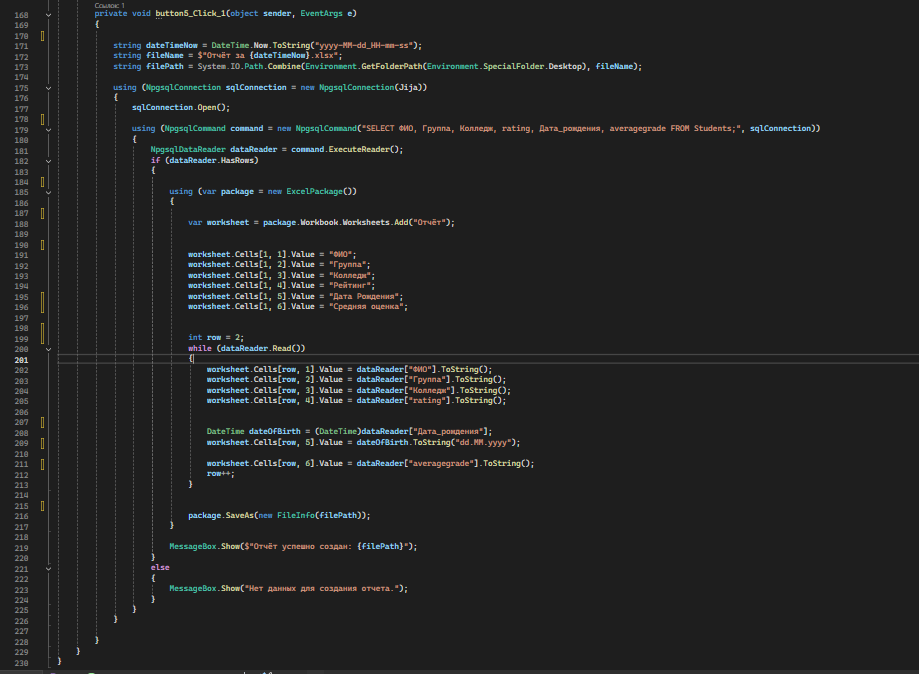
**Приложение 6**

Вторая часть кода основной формы системы



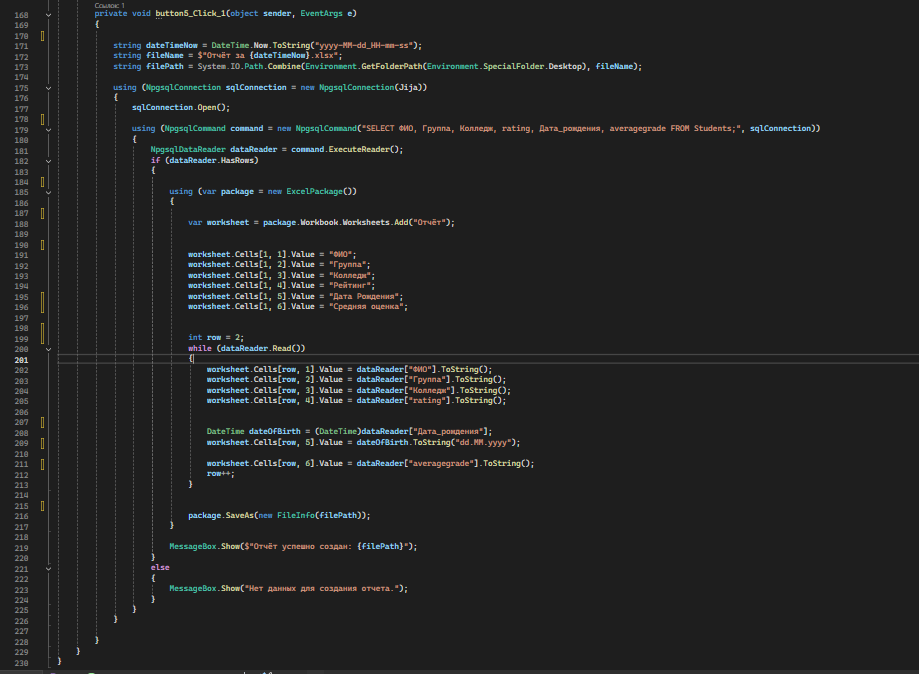
**Приложение 7**

Третья часть кода основной формы системы



**Приложение 8**

Код формы поиска и добавления



**Приложение 9**

Код формы авторизации

