**МИНИСТЕРСТВО ОБРАЗОВАНИЯ, НАУКИ И МОЛОДЕЖНОЙ ПОЛИТИКИ КРАСНОДАРСКОГО КРАЯ**

**ГАПОУ КК ЛАТ**

**КУРСОВАЯ РАБОТА**

по МДК 07.01 Управление и автоматизация баз данных

на тему:

*«Проектирование и разработка базы данных для учета и анализа активности участия обучающихся в мероприятиях ГАПОУ КК ЛАТ»*

|  |  |
| --- | --- |
|  | **Выполнил:**  Летунов Захар Денисович  студент 4 курса группы 541  специальности 09.02.07 «Информационные системы и программирование»  очной формы обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Руководитель работы:**  Пискунова Е.А.  должность: преподаватель |

Работа допущена к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя) (дата)

Работа выполнена и

защищена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члены комиссии\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(должность) (подпись) (И.О. Фамилия)

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

(должность) (подпись) (И.О. Фамилия)

Задание на курсовую

**Содержание**

[ВВЕДЕНИЕ 4](#_Toc1334793626)

[Глава 1. ТЕОРЕТИЧЕСКИЙ ОБЗОР ИССЛЕДУЕМОЙ ОБЛАСТИ 6](#_Toc536358339)

[1.1 Общая характеристика ГАПОУ КК ЛАТ 6](#_Toc1770029144)

[1.2 Обоснование необходимости разработки базы данных для учета и анализа активности 7](#_Toc1065380255)

[1.3 Проектирование БД 8](#_Toc640623721)

[Вывод по главе 10](#_Toc1991574609)

[Глава 2. РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ 12](#_Toc1858449391)

[2.1 Анализ и выбор программного обеспечения и инструментов для разработки 12](#_Toc1435181024)

[2.2 Реализация структуры базы данных с использованием СУБД PostgreSQL 14](#_Toc1643167163)

[2.3 Создание запросов и пользовательского интерфейса для взаимодействия с базой данных 17](#_Toc1406039400)

[Вывод по главе 23](#_Toc1588347455)

[ЗАКЛЮЧЕНИЕ 25](#_Toc793299234)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 27](#_Toc1994712006)

**ВВЕДЕНИЕ**

В современных условиях образовательные учреждения сталкиваются с необходимостью эффективного управления большими объемами данных, связанных с учебной и внеучебной деятельностью студентов. Одним из ключевых аспектов успешной организации образовательного процесса является учет и анализ активности участия обучающихся в различных мероприятиях. Это позволяет не только оценить вовлеченность студентов в жизнь учебного заведения, но и способствует формированию их профессиональных и личностных компетенций.

Государственное автономное профессиональное образовательное учреждение Краснодарского края «Лабинский аграрный техникум» (ГАПОУ КК ЛАТ) — это образовательное учреждение, которое готовит специалистов для аграрного сектора экономики. В рамках учебного процесса и внеучебной деятельности студенты техникума принимают участие в различных мероприятиях: научных конференциях, спортивных соревнованиях, культурных событиях, волонтерских акциях и других. Учет и анализ этой активности является важной задачей для администрации техникума, так как позволяет оценить эффективность воспитательной работы, выявить наиболее активных студентов и скорректировать планы мероприятий.

Для автоматизации процессов учета и анализа данных об участии студентов в мероприятиях необходимо разработать специализированную базу данных. Такая система позволит упростить сбор, хранение и обработку информации, а также предоставит возможность формировать отчеты и аналитические материалы для принятия управленческих решений.

Объектом исследования является процесс учета и анализа активности участия обучающихся ГАПОУ КК ЛАТ в мероприятиях.

Предметом исследования выступает проектирование и разработка базы данных для автоматизации учета и анализа активности студентов.

Практическая значимость работы заключается в создании инструмента, который позволит администрации техникума эффективно управлять данными об участии студентов в мероприятиях. Разработанная база данных может быть внедрена в деятельность ГАПОУ КК ЛАТ, что упростит процесс сбора и анализа информации, а также повысит качество управленческих решений. Кроме того, результаты работы могут быть адаптированы для использования в других образовательных учреждениях.

Целью данной курсовой работы является проектирование и разработка базы данных для учета и анализа активности участия обучающихся ГАПОУ КК ЛАТ в мероприятиях. В рамках работы будут решены следующие задачи:

1. Проведен анализ предметной области и определены основные требования к системе.

2. Разработана концептуальная и логическая модель базы данных.

3. Реализована физическая модель базы данных с использованием современных технологий.

4. Разработан пользовательский интерфейс для удобства работы с системой.

5. Проведено тестирование и оценка эффективности разработанной базы данных.

Актуальность данной работы обусловлена необходимостью автоматизации процессов учета и анализа данных в образовательных учреждениях, что способствует повышению качества управления и организации учебно-воспитательного процесса. Разработанная база данных может быть внедрена в деятельность ГАПОУ КК ЛАТ и других образовательных учреждений, что подтверждает ее практическую значимость.

**Глава 1. ТЕОРЕТИЧЕСКИЙ ОБЗОР ИССЛЕДУЕМОЙ ОБЛАСТИ**

1. **Общая характеристика ГАПОУ КК ЛАТ**

Государственное автономное профессиональное образовательное учреждение Краснодарского края «Лабинский аграрный техникум» (ГАПОУ КК ЛАТ) является некоммерческой организацией, созданной для предоставления образовательных услуг в сфере среднего профессионального образования. Учреждение находится в ведении Министерства образования и науки Краснодарского края, которое выполняет функции учредителя и осуществляет контроль за его деятельностью (п. 1.3, 1.4 Устава).

ГАПОУ КК ЛАТ осуществляет подготовку студентов по программам среднего профессионального образования, а также предоставляет услуги дополнительного профессионального образования, включая программы повышения квалификации и профессиональной переподготовки (п. 2.3.4 Устава). Основной целью деятельности техникума является подготовка квалифицированных специалистов для аграрного сектора экономики, а также содействие их трудоустройству и профессиональному развитию.

Техникум имеет право осуществлять различные виды деятельности, связанные с образовательным процессом, включая организацию учебно-производственных мастерских, проведение культурно-массовых и спортивных мероприятий, а также оказание платных образовательных услуг (п. 2.3 Устава). Учреждение также занимается реализацией собственной продукции, работ и услуг, таких как полевые работы, сбор урожая и строительные услуги (п. 2.3.6 Устава).

ГАПОУ КК ЛАТ является правопреемником государственного бюджетного образовательного учреждения начального профессионального образования профессионального училища № 13 Краснодарского края, которое было присоединено к техникуму в 2013 году (п. 1.1 Устава). Учреждение имеет статус юридического лица и осуществляет свою деятельность на основе самофинансирования, используя доходы от оказания платных услуг и реализации продукции для достижения уставных целей (п. 5.10 Устава).

Техникум активно взаимодействует с работодателями и социальными партнерами, что позволяет обеспечивать высокий уровень подготовки студентов и их востребованность на рынке труда. В своей деятельности ГАПОУ КК ЛАТ руководствуется законодательством Российской Федерации, нормативными актами Краснодарского края и внутренними локальными актами, регулирующими образовательный процесс и управление учреждением. [1]

1. **Обоснование необходимости разработки базы данных для учета и анализа активности**

В современных условиях образовательные учреждения, включая Государственное автономное профессиональное образовательное учреждение Краснодарского края «Лабинский аграрный техникум» (ГАПОУ КК ЛАТ), сталкиваются с необходимостью эффективного управления большими объемами данных, связанных с учебной и внеучебной деятельностью студентов. Учет и анализ активности студентов в мероприятиях является важной задачей, так как это позволяет оценить уровень их вовлеченности в жизнь учебного заведения, выявить наиболее активных и талантливых обучающихся, а также скорректировать планы мероприятий с учетом интересов и потребностей студентов.

В настоящее время процесс учета и анализа данных об участии студентов в мероприятиях в ГАПОУ КК ЛАТ осуществляется в основном вручную, что приводит к ряду проблем:

1. Трудоемкость процесса: Ручной сбор и обработка данных требуют значительных временных и человеческих ресурсов, что снижает оперативность работы.

2. Ошибки и неточности: При ручном вводе данных высока вероятность ошибок, что может привести к искажению информации и некорректным выводам.

3. Отсутствие систематизации: Данные об участии студентов в мероприятиях хранятся в различных форматах (бумажные журналы, электронные таблицы), что затрудняет их анализ и поиск.

Разработка базы данных для учета и анализа активности студентов позволит решить эти проблемы и обеспечит следующие преимущества:

1. Автоматизация процессов: База данных позволит автоматизировать сбор, хранение и обработку информации об участии студентов в мероприятиях, что значительно сократит временные затраты и уменьшит вероятность ошибок.

2. Централизованное хранение данных: Все данные будут храниться в единой системе, что упростит доступ к информации и ее анализ.

3. Повышение качества управления: На основе данных, полученных из базы, администрация техникума сможет принимать более обоснованные решения, связанные с организацией мероприятий и воспитательной работы.

5. Стимулирование активности студентов: Система учета активности может быть использована для поощрения наиболее активных студентов, что повысит их мотивацию к участию в мероприятиях.

Таким образом, разработка базы данных для учета и анализа активности студентов в ГАПОУ КК ЛАТ является актуальной задачей, которая позволит повысить эффективность управления внеучебной деятельностью, улучшить качество воспитательной работы и создать условия для всестороннего развития личности студентов.

1. **Проектирование БД**

Проектирование базы данных для учета и анализа активности студентов в мероприятиях ГАПОУ КК ЛАТ основывается на создании структуры, которая обеспечивает эффективное хранение, обработку и анализ данных. В качестве основы для проектирования используется ER-диаграмма, которая отражает взаимосвязи между сущностями базы данных. [10]

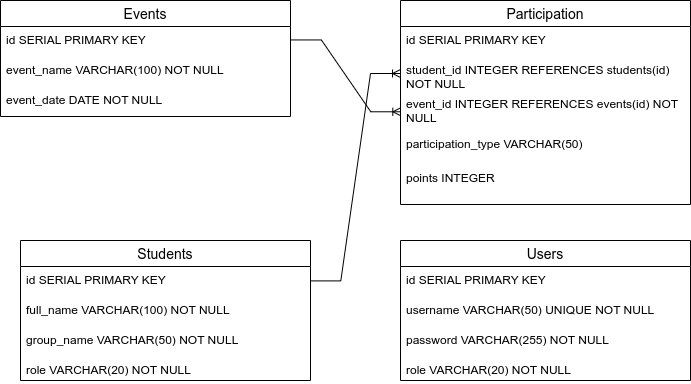
****

Рис. 2.1 - ER диаграмма

Основные сущности, представленные в схеме, включают:

* Пользователи (Users) – хранят информацию о пользователях системы, включая их учетные данные и роли.
* Студенты (Students) – содержат данные о студентах, такие как их полное имя, группа и роль.
* Мероприятия (Events) – хранят информацию о мероприятиях, включая название и дату проведения.
* Участие (Participation) – отражает связь между студентами и мероприятиями, а также тип участия и начисленные баллы.

Описание сущностей и их атрибутов

1. Пользователи (Users):

* id – уникальный идентификатор пользователя (первичный ключ).
* username – имя пользователя (логин), должно быть уникальным.
* password – пароль пользователя.
* role – роль пользователя в системе (например, администратор, преподаватель, студент).

1. Студенты (Students):

* id – уникальный идентификатор студента (первичный ключ).
* full\_name – полное имя студента.
* group\_name – название группы, в которой обучается студент.

1. Мероприятия (Events):

* id – уникальный идентификатор мероприятия (первичный ключ).
* event\_name – название мероприятия.
* event\_date – дата проведения мероприятия.

1. Участие (Participation):

* id – уникальный идентификатор записи об участии (первичный ключ).
* student\_id – внешний ключ, ссылающийся на таблицу "Students" (идентификатор студента).
* event\_id – внешний ключ, ссылающийся на таблицу "Events" (идентификатор мероприятия).
* participation\_type – тип участия студента в мероприятии (например, организатор, участник).
* points – количество баллов, начисленных студенту за участие в мероприятии.

Взаимосвязи между сущностями

Таблица "Participation" связывает студентов и мероприятия, что позволяет отслеживать участие каждого студента в конкретных событиях.

Внешние ключи student\_id и event\_id обеспечивают целостность данных, предотвращая возможность добавления записей о участии для несуществующих студентов или мероприятий. [14]

**Вывод по главе**

В первой главе проведен теоретический обзор исследуемой области, включающий общую характеристику Государственного автономного профессионального образовательного учреждения Краснодарского края «Лабинский аграрный техникум» (ГАПОУ КК ЛАТ), обоснование необходимости разработки базы данных для учета и анализа активности студентов, а также проектирование базы данных.

Таким образом, проведенный анализ показал, что разработка базы данных для учета и анализа активности студентов в ГАПОУ КК ЛАТ является актуальной и необходимой задачей. Она позволит не только упростить процессы учета и анализа данных, но и повысить эффективность управления внеучебной деятельностью, что в конечном итоге будет способствовать всестороннему развитию студентов и улучшению качества образовательного процесса в техникуме.

**Глава 2. РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ**

1. **Анализ и выбор программного обеспечения и инструментов для разработки**

Для успешной разработки программного обеспечения необходимо выбрать подходящие инструменты, которые обеспечат эффективность, удобство и соответствие требованиям проекта. В данном разделе проводится анализ и обоснование выбора программного обеспечения для разработки, включая систему управления базами данных (СУБД) и интегрированную среду разработки (IDE).

При выборе системы управления базами данных (СУБД) для разработки проекта важно учитывать такие критерии, как производительность, масштабируемость, поддержка стандартов SQL, расширяемость, безопасность и сообщество разработчиков. Рассмотрим две популярные СУБД: PostgreSQL и MySQL.

PostgreSQL — это мощная объектно-реляционная СУБД с открытым исходным кодом, которая поддерживает широкий спектр функций, включая сложные запросы, транзакции, триггеры, хранимые процедуры и расширения. Она строго соответствует стандартам SQL, что делает её универсальным решением для проектов любого масштаба. PostgreSQL также отличается высокой надежностью и поддержкой ACID (Atomicity, Consistency, Isolation, Durability), что критически важно для проектов с высокими требованиями к целостности данных. [2]

MySQL — это реляционная СУБД, которая также имеет открытый исходный код и широко используется в веб-разработке. MySQL известен своей простотой и высокой производительностью для операций чтения, что делает его популярным выбором для небольших и средних проектов. Однако MySQL имеет ограниченную поддержку сложных SQL-запросов и менее гибок в настройке по сравнению с PostgreSQL. [4]

Обоснование выбора PostgreSQL:

* Поддержка сложных запросов и операций. PostgreSQL лучше справляется с обработкой сложных запросов и больших объемов данных, что делает его предпочтительным для проектов с высокими требованиями к аналитике и обработке данных.
* Расширяемость. PostgreSQL поддерживает пользовательские типы данных, функции и расширения, что позволяет адаптировать СУБД под конкретные задачи.
* Надежность и соответствие стандартам. PostgreSQL строго соответствует стандартам SQL и обеспечивает высокий уровень надежности, что важно для проектов с критически важными данными.
* Сообщество и документация. PostgreSQL имеет активное сообщество разработчиков и подробную документацию, что упрощает поиск решений и поддержку. [7]

Таким образом, для проекта, требующего высокой надежности, гибкости и поддержки сложных операций, PostgreSQL является более подходящим выбором.

Для разработки программного обеспечения важно выбрать подходящую интегрированную среду разработки (IDE), которая обеспечивает удобство работы, поддержку необходимых языков программирования и инструментов для отладки и тестирования. Рассмотрим две популярные IDE: Visual Studio Community и PyCharm.

Visual Studio Community — это бесплатная версия мощной IDE от Microsoft, которая поддерживает множество языков программирования, включая C#, C++, Python, JavaScript и другие. Она предоставляет широкий набор инструментов для разработки, отладки, тестирования и интеграции с системами контроля версий. Visual Studio Community также поддерживает расширения, что позволяет адаптировать среду под конкретные нужды разработчика. [3]

PyCharm — это специализированная IDE для разработки на Python, созданная JetBrains. Она предлагает мощные инструменты для работы с Python, включая поддержку виртуальных окружений, отладку, рефакторинг и интеграцию с популярными фреймворками, такими как Django и Flask. PyCharm доступен в бесплатной версии (Community Edition) и платной (Professional Edition).

Обоснование выбора Visual Studio Community:

* Мультиязыковая поддержка. Visual Studio Community поддерживает множество языков программирования, что делает её универсальным инструментом для разработки, особенно если проект предполагает использование нескольких языков.
* Интеграция с экосистемой Microsoft. Visual Studio легко интегрируется с другими инструментами Microsoft, такими как Azure, SQL Server и GitHub, что упрощает разработку и развертывание приложений.
* Расширяемость. Поддержка расширений позволяет добавлять новые функции и инструменты, что делает Visual Studio гибким решением для различных задач.
* Бесплатная лицензия. Visual Studio Community предоставляет все необходимые функции для индивидуальных разработчиков и небольших команд без необходимости приобретения платной лицензии.

Таким образом, Visual Studio Community является более универсальным и гибким выбором для разработки, особенно если проект предполагает использование нескольких языков программирования и интеграцию с экосистемой Microsoft. [11]

1. **Реализация структуры базы данных с использованием СУБД PostgreSQL**

Для реализации структуры базы данных была выбрана СУБД PostgreSQL, которая обеспечивает высокую надежность, гибкость и соответствие стандартам SQL. Структура базы данных включает четыре таблицы: Users, Students, Events и Participation. Каждая таблица отвечает за хранение определенных данных и связана с другими таблицами через внешние ключи. [5]

Описание таблиц базы данных:

1. Таблица Users:

Эта таблица предназначена для хранения информации о пользователях системы. Она включает следующие поля:

* id — уникальный идентификатор пользователя (первичный ключ, автоматически генерируемый);
* username — уникальное имя пользователя, используемое для авторизации;
* password — пароль пользователя;
* role — роль пользователя в системе (например, администратор, преподаватель, студент).

SQL:

CREATE TABLE Users (

id SERIAL PRIMARY KEY,

username VARCHAR(50) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

role VARCHAR(20) NOT NULL

);

1. Таблица Students:

Таблица Students хранит информацию о студентах. Основные поля:

* id — уникальный идентификатор студента (первичный ключ, автоматически генерируемый);
* full\_name — полное имя студента;
* group\_name — название группы, к которой принадлежит студент.

SQL:

CREATE TABLE Students (

id SERIAL PRIMARY KEY,

full\_name VARCHAR(100) NOT NULL,

group\_name VARCHAR(50) NOT NULL

);

1. Таблица Events:

Таблица Events содержит данные о мероприятиях, проводимых в учебном заведении. Поля таблицы:

* id — уникальный идентификатор мероприятия (первичный ключ, автоматически генерируемый);
* event\_name — название мероприятия;
* event\_date — дата проведения мероприятия.

SQL:

CREATE TABLE Events (

id SERIAL PRIMARY KEY,

event\_name VARCHAR(100) NOT NULL,

event\_date DATE NOT NULL

);

1. Таблица Participation

Таблица Participation связывает студентов с мероприятиями и хранит информацию об их участии. Поля таблицы:

* id — уникальный идентификатор записи (первичный ключ, автоматически генерируемый);
* student\_id — внешний ключ, ссылающийся на таблицу Students;
* event\_id — внешний ключ, ссылающийся на таблицу Events;
* participation\_type — тип участия студента в мероприятии (например, докладчик, слушатель);
* points — количество баллов, начисленных студенту за участие.

SQL:

CREATE TABLE Participation (

id SERIAL PRIMARY KEY,

student\_id INTEGER REFERENCES Students(id) NOT NULL,

event\_id INTEGER REFERENCES Events(id) NOT NULL,

participation\_type VARCHAR(50),

points INTEGER

);

Особенности реализации:

* Целостность данных. Использование внешних ключей (student\_id и event\_id) в таблице Participation обеспечивает ссылочную целостность данных. Это гарантирует, что записи о участии студентов в мероприятиях будут корректно связаны с существующими записями в таблицах Students и Events.
* Уникальность. Поле username в таблице Users имеет ограничение UNIQUE, что предотвращает создание пользователей с одинаковыми именами.
* Гибкость. Таблица Participation позволяет хранить различную информацию о участии студентов в мероприятиях, включая тип участия и начисленные баллы.

1. **Создание запросов и пользовательского интерфейса для взаимодействия с базой данных**

Для обеспечения удобного взаимодействия пользователей с базой данных необходимо разработать набор SQL-запросов, которые будут выполнять основные операции (CRUD: Create, Read, Update, Delete), а также создать пользовательский интерфейс (UI), который позволит пользователям работать с данными без необходимости написания SQL-запросов вручную. Запросы реализованы в методах C# и выполняют основные операции с данными: аутентификацию пользователей, управление студентами, мероприятиями, участием студентов в мероприятиях и пользователями. [6]

Аутентификация пользователя:

Метод: login(string username, string pass)

Описание:

Этот метод выполняет аутентификацию пользователя, проверяя наличие записи в таблице Users с указанным именем пользователя (username) и паролем (password).

SQL-запрос:

SELECT \* FROM Users WHERE username = :username AND password = :pass;

Параметры:

username — имя пользователя.

pass — пароль пользователя.

Возвращаемые данные:

Метод возвращает DataTable, содержащий данные пользователя, если аутентификация прошла успешно.

Управление студентами:

Метод: createStudent(string fio, string group)

Описание:

Добавляет нового студента в таблицу Students.

SQL-запрос:

INSERT INTO Students (full\_name, group\_name) VALUES (:fio, :group);

Параметры:

fio — полное имя студента.

group — название группы.

Метод: getStudents()

Описание:

Возвращает список всех студентов из таблицы Students.

SQL-запрос:

SELECT \* FROM Students;

Возвращаемые данные:

Метод возвращает DataTable, содержащий данные всех студентов.

Управление мероприятиями:

Метод: createEvents(string event\_name, DateTime event\_data)

Описание:

Добавляет новое мероприятие в таблицу Events.

SQL-запрос:

INSERT INTO Events (event\_name, event\_date) VALUES (:event\_name, :event\_data); [9]

Параметры:

event\_name — название мероприятия.

event\_data — дата проведения мероприятия.

Управление участием студентов в мероприятиях:

Метод: createParticipation(int student\_id, int event\_id, string participation\_type, int point)

Описание:

Добавляет запись об участии студента в мероприятии в таблицу Participation.

SQL-запрос:

INSERT INTO Participation (student\_id, event\_id, participation\_type, points)

VALUES (:student\_id, :event\_id, :participation\_type, :point);

Параметры:

student\_id — идентификатор студента.

event\_id — идентификатор мероприятия.

participation\_type — тип участия (например, "докладчик", "слушатель").

point — количество баллов за участие.

Метод: getParticipation()

Описание:

Возвращает список всех записей об участии студентов в мероприятиях. Данные объединяются из таблиц Participation, Students и Events.

SQL-запрос:

SELECT

s.full\_name,

s.group\_name,

e.event\_name,

e.event\_date,

p.participation\_type,

p.points

FROM

Participation p

JOIN

Students s ON s.id = p.student\_id

JOIN

Events e ON e.id = p.event\_id;

Возвращаемые данные:

Метод возвращает DataTable, содержащий данные об участии студентов в мероприятиях.

Разработка пользовательского интерфейса:

Для взаимодействия с базой данных был разработан пользовательский интерфейс, который позволяет выполнять основные операции через удобные формы и кнопки. Интерфейс реализован с использованием языка программирования C# и Windows Forms , который обеспечивает простоту разработки и интеграцию с базой данных. [8]

Форма авторизации:

Пользователь вводит логин и пароль.

После успешной авторизации система определяет роль пользователя (администратор, преподаватель, студент) и предоставляет соответствующие права доступа. (Рис. 2.2) [12]

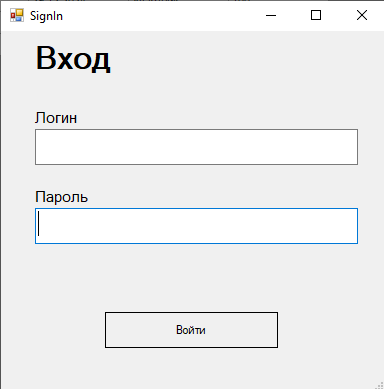


Рис. 2.2 Форма авторизации

Форма просмотра активностей:

Пользователь попадает туда при запуске системы и может просматривать активности студентов. (Рис. 2.3)

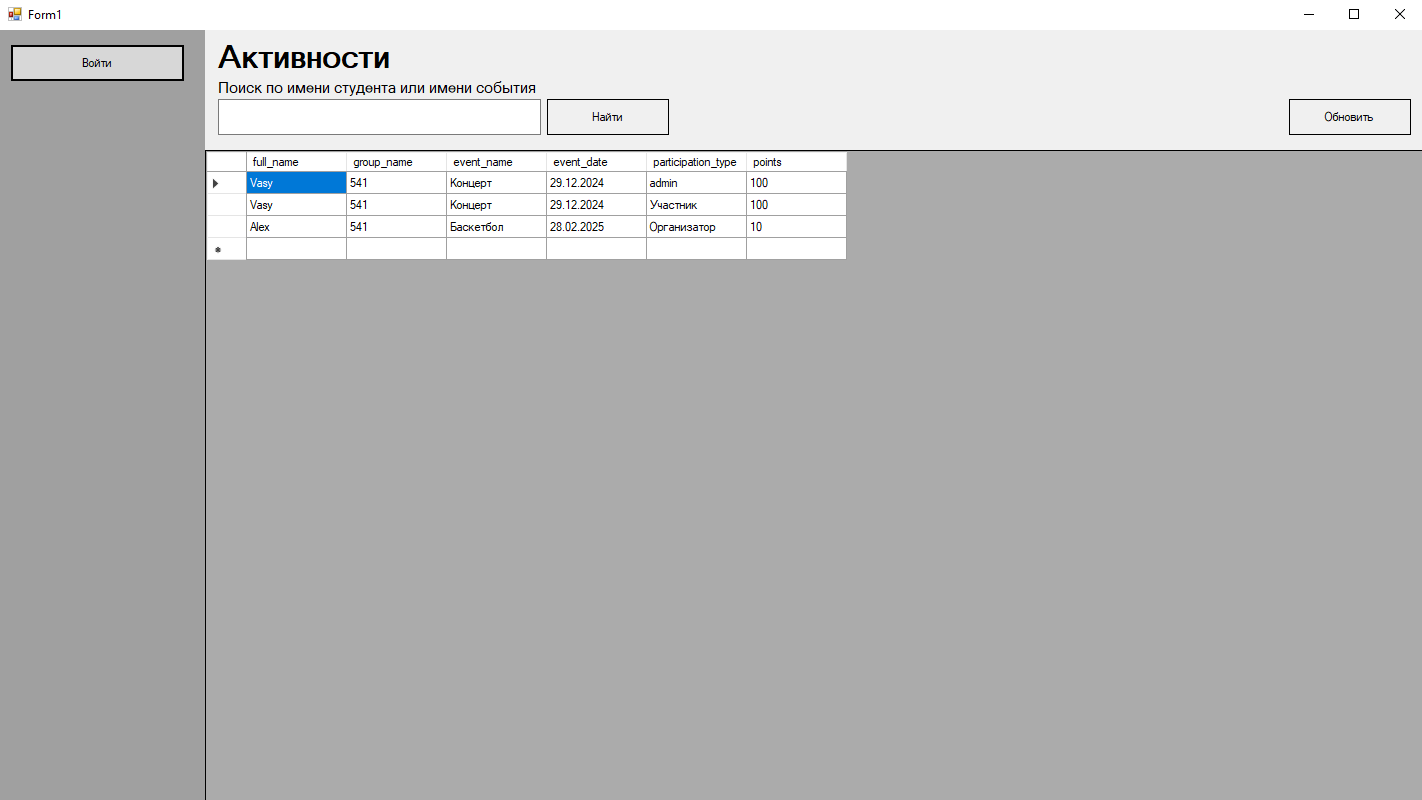


Рис. 2.3 Форма просмотра активностей

После входа в систему если пользователям является оператором системы, то у него появляется доступ к управлению студентами, активностями и событиями, а если пользователь является админом, то еще и к управлению пользователями (Рис. 2.5) [13]

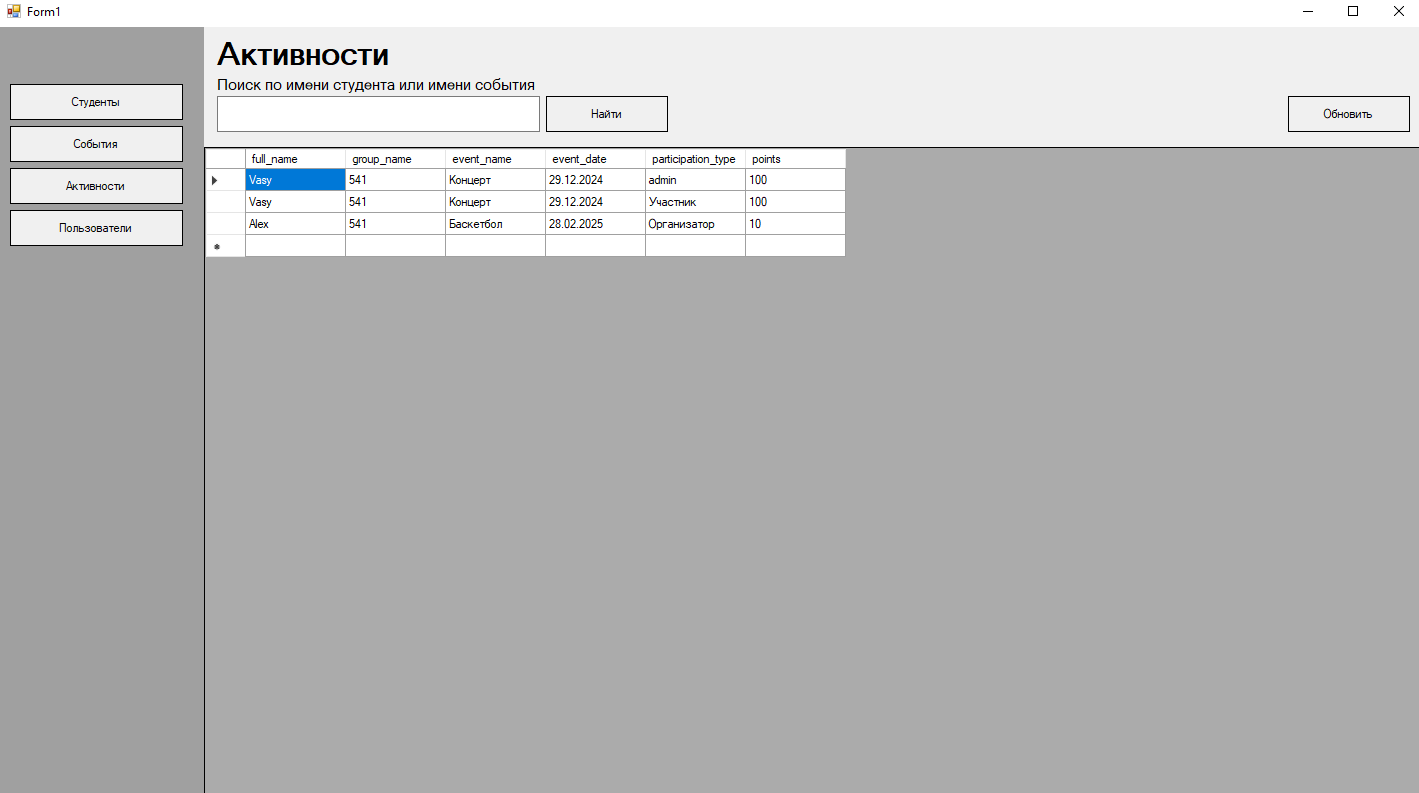


Рис. 2.4 Форма после авторизации

Форма студентов:

Пользователь после авторизации может перейти сюда и добавить студентов, а также просмотреть уже добавленных. (Рис. 2.5)

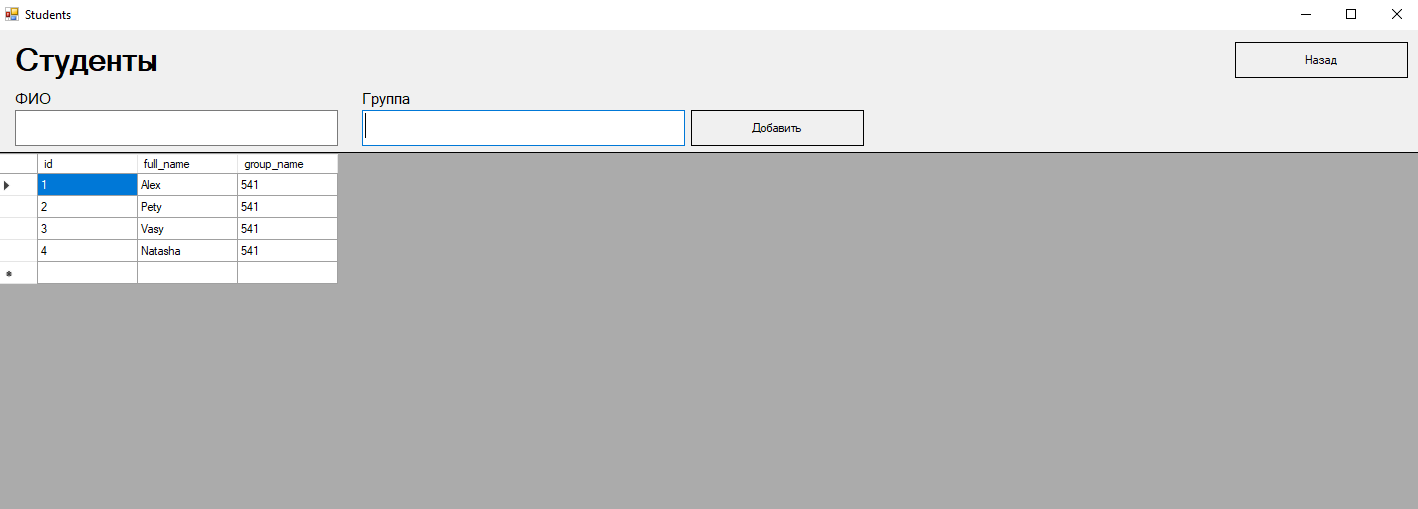


Рис. 2.5 Форма студентов

Форма управления активностями:

Предоставляет доступ к управлению активностями (Рис. 2.6)

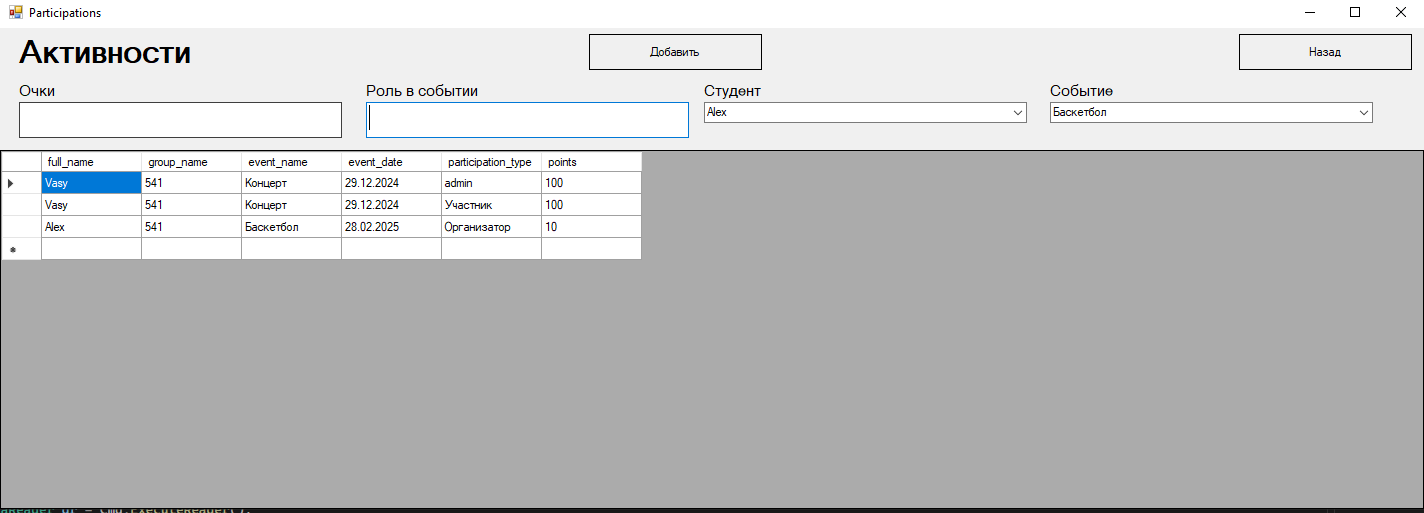


Рис. 2.6 Форма управления активностями

**Вывод по главе**

В рамках главы 2 была реализована база данных для системы управления студентами и их участием в мероприятиях. В качестве СУБД был выбран PostgreSQL, что обусловлено его высокой надежностью, поддержкой сложных запросов, расширяемостью и соответствием стандартам SQL. Для разработки программного обеспечения была выбрана интегрированная среда разработки Visual Studio Community, которая предоставляет широкие возможности для работы с различными языками программирования, включая C#, и поддерживает интеграцию с экосистемой Microsoft.

Структура базы данных включает четыре таблицы: Users, Students, Events и Participation, которые связаны между собой через внешние ключи. Это обеспечивает целостность данных и позволяет эффективно хранить информацию о пользователях, студентах, мероприятиях и участии студентов в мероприятиях. Для взаимодействия с базой данных были разработаны SQL-запросы, охватывающие основные операции (CRUD), а также создан пользовательский интерфейс на основе Windows Forms, который предоставляет удобный доступ к данным для пользователей с различными ролями (администратор, оператор, студент).

Реализованная система позволяет:

* Аутентифицировать пользователей и предоставлять им доступ в зависимости от их роли.
* Управлять данными о студентах, мероприятиях и участии студентов в мероприятиях.
* Просматривать и анализировать информацию об активностях студентов.
* Обеспечивать гибкость и масштабируемость за счет использования PostgreSQL и Visual Studio Community.

Таким образом, выбранные инструменты и реализованная структура базы данных полностью соответствуют требованиям проекта и обеспечивают удобство, надежность и эффективность работы системы.

**ЗАКЛЮЧЕНИЕ**

В рамках данной курсовой работы была разработана база данных для учета и анализа активности участия студентов Государственного автономного профессионального образовательного учреждения Краснодарского края «Лабинский аграрный техникум» (ГАПОУ КК ЛАТ) в различных мероприятиях. Проект направлен на автоматизацию процессов сбора, хранения и анализа данных, что позволит повысить эффективность управления внеучебной деятельностью и улучшить качество воспитательной работы в техникуме.

На этапе проектирования была разработана концептуальная и логическая модель базы данных, включающая четыре основные таблицы: Users, Students, Events и Participation. Эти таблицы связаны между собой через внешние ключи, что обеспечивает целостность данных и позволяет эффективно хранить информацию о пользователях, студентах, мероприятиях и участии студентов в мероприятиях. В качестве системы управления базами данных был выбран PostgreSQL, что обусловлено его высокой надежностью, поддержкой сложных запросов и соответствием стандартам SQL.

Для разработки программного обеспечения была выбрана интегрированная среда разработки Visual Studio Community, которая предоставляет широкие возможности для работы с языком программирования C# и интеграции с экосистемой Microsoft. Пользовательский интерфейс, реализованный на основе Windows Forms, обеспечивает удобный доступ к данным для пользователей с различными ролями (администратор, оператор, студент). Интерфейс включает формы для авторизации, управления студентами, мероприятиями и участием студентов в мероприятиях, что позволяет эффективно выполнять основные операции (CRUD) без необходимости написания SQL-запросов вручную.

Разработанная система позволяет:

* Автоматизировать процессы учета и анализа данных об участии студентов в мероприятиях.
* Обеспечить централизованное хранение данных, что упрощает доступ к информации и её анализ.
* Повысить качество управленческих решений за счет формирования отчетов и аналитических материалов.
* Стимулировать активность студентов через систему учета и поощрения их участия в мероприятиях.

Практическая значимость работы заключается в создании инструмента, который может быть внедрен в деятельность ГАПОУ КК ЛАТ для автоматизации процессов учета и анализа активности студентов. Разработанная база данных также может быть адаптирована для использования в других образовательных учреждениях, что подтверждает её универсальность и актуальность.

Таким образом, цели и задачи курсовой работы были успешно достигнуты. Разработанная система обеспечивает удобство, надежность и эффективность работы с данными, что способствует повышению качества образовательного процесса и всестороннему развитию студентов. В перспективе возможно дальнейшее развитие системы, включая добавление новых функций, таких как формирование отчетов в автоматическом режиме, интеграция с другими системами управления учебным процессом и расширение аналитических возможностей.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Устав Государственного автономного профессионального образовательного учреждения Краснодарского края «Лабинский аграрный техникум» URL: <https://lat-labinsk.ru/images/information/glavnoemenu/nomativniedocumenti/ystav/ustav.pdf>
2. PostgreSQL 15 Documentation. URL: <https://www.postgresql.org/docs/15/>
3. Microsoft. Visual Studio 2022 Documentation. URL: <https://docs.microsoft.com/ru-ru/visualstudio/>
4. MySQL 8.0 Reference Manual. URL: <https://dev.mysql.com/doc/refman/8.0/en/>
5. Современные подходы к проектированию баз данных. URL: <https://habr.com/ru/post/254773/>
6. Руководство по проектированию реляционных баз данных. URL: <https://proglib.io/p/relational-database-design/>
7. PostgreSQL vs MySQL: Сравнение СУБД. URL: <https://habr.com/ru/company/ruvds/blog/582880/>
8. Windows Forms: Создание приложений на C#. URL: <https://metanit.com/sharp/tutorial/>
9. SQL для начинающих: Полное руководство. URL: <https://proglib.io/p/sql-for-beginners/>
10. Кузнецов, С. Д. Основы баз данных. – М.: Интернет-Университет Информационных Технологий, 2020. – 480 с.
11. Гарнаев, А. Ю. Microsoft Visual Studio 2022. Первое знакомство. – СПб.: БХВ-Петербург, 2022. – 352 с.
12. Васильев, А. Н. Программирование на C# для начинающих. – М.: Эксмо, 2022. – 256 с.
13. Троелсен, Э. Язык программирования C# 9 и платформа .NET 5. – М.: Вильямс, 2021. – 1200 с.
14. Гарсиа-Молина, Г., Ульман, Д., Уидом, Дж. Системы баз данных: полный курс. – М.: Вильямс, 2020. – 1088 с.
15. Проектирование баз данных: лучшие практики. URL: <https://habr.com/ru/company/ruvds/blog/564382/>