**Техническое задание (ТЗ)**

**По теме:** Программа учета студентов колледжа

**Общее описание системы**

**Название системы:** "Учет студентов колледжа"

**Основное назначение:** Автоматизация процесса управления учебным процессом и учета студентов в колледже. Система предназначена для оптимизации работы преподавателей, административного персонала и руководства учебного заведения.

**Ожидаемые результаты внедрения:**

Упрощение и ускорение обработки информации о студентах (ввод, изменение, удаление данных).

* Повышение точности расчета успеваемости и отслеживания посещений.
* Автоматическое формирование отчетной документации.
* Оптимизация взаимодействия между различными подразделениями колледжа.

**Функциональные требования**

Система должна обеспечивать выполнение следующих функций:

1. **Учет студентов**
   * Ведение базы данных студентов (ФИО, номер группы, дата рождения, контактные данные и др.).
   * Регистрация новых студентов, редактирование и удаление записей.
2. **Контроль посещаемости**
   * Учет явки студентов на занятия.
   * Автоматическая фиксация пропусков с возможностью комментариев от преподавателя.
3. **Оценка успеваемости**
   * Введение оценок по каждому предмету.
   * Расчёт среднего балла студента по курсу/семестру.
   * Генерация сводных ведомостей по успеваемости.
4. **Административная панель**
   * Интерфейс для настройки системы администратором (добавление/редактирование учебных курсов, преподавателей, предметов и т.п.).

**Технические требования**

1. **Поддерживаемые платформы**
   * .EXE интерфейс (совместимость с основными: Windows, Mac).
2. **Операционная система сервера**
   * Windows
3. **База данных**Microsoft Access DataBase
4. **Язык программирования**   
   C#
5. **Интерфейсы и API**
   * RESTful API для взаимодействия с внешними сервисами.
   * Поддержка OAuth 2.0 для аутентификации и авторизации.
6. **Резервное копирование**
   * Автоматизированный процесс резервного копирования базы данных и важных файлов.
7. **Производительность и масштабируемость**
   * Система должна поддерживать одновременную работу нескольких сотен пользователей.
   * Легкость горизонтального масштабирования за счет контейнеризации (Docker, Kubernetes).
8. **Требования к безопасности**
   * Шифрование передачи данных (HTTPS).
   * Двухфакторная аутентификация для административных аккаунтов.
   * Логирование действий пользователей для аудита.