**Техническое задание (ТЗ)**

**Приложение для контроля и анализа успеваемости с визуализацией прогресса**

**Язык разработки:** C# (.NET, WinForms)   
**Целевая ОС:** Windows

**1. Введение**

**1.1. Назначение**

Приложение предназначено для контроля успеваемости учащихся (студентов, школьников) с возможностью анализа прогресса и визуализации данных в виде графиков и диаграмм.

**1.2. Цели**

* Упрощение учета оценок и посещаемости
* Наглядное отображение динамики успеваемости
* Анализ слабых и сильных сторон учащегося
* Формирование отчетов

**2. Требования к функционалу**

**2.1. Основные модули**

**2.1.1. Управление пользователями**

* Регистрация и авторизация (роли: **ученик, преподаватель, администратор**)
* Хранение данных пользователей (ФИО, группа/класс, контакты)

**2.1.2. Учет успеваемости**

* Ввод и редактирование оценок (по предметам, датам)
* Учет посещаемости (отметки о пропусках)
* Возможность добавления весовых коэффициентов для разных типов работ (контрольные, домашние задания и т. д.)

**2.1.3. Визуализация данных**

* Графики успеваемости (линейные, столбчатые диаграммы)
* Круговая диаграмма распределения оценок
* Сравнение с средним баллом группы/класса

**2.1.4. Аналитика и отчеты**

* Расчет среднего балла по предмету/периоду
* Прогнозирование итоговой оценки
* Экспорт отчетов в **PDF/Excel**

**2.1.5. Уведомления**

* Оповещения о низкой успеваемости
* Напоминания о контрольных работах

**3. Технические требования**

**3.1. Интерфейс**

* **Главное окно** с навигационным меню
* **Дашборд** с ключевыми метриками (средний балл, прогресс)
* **Таблицы** с оценками и посещаемостью
* **Графический редактор** для настройки визуализации

**3.2. Архитектура**

* **Локальная** (JSON)
* **MVC**

**3.3. Библиотеки и технологии**

* **Для графиков:** LiveCharts, OxyPlot, Microsoft Chart Controls
* **Для экспорта:** iTextSharp (PDF), EPPlus (Excel)
* **База данных:** PostgreSQL

**3.4. Безопасность**

* Хранение паролей в хешированном виде (bcrypt)
* Защита от SQL-инъекций
* Ограничение доступа к данным (ученик видит только свои оценки, преподаватель — свою группу)

**3.5. Производительность**

* Максимальное время загрузки данных: не более 2 секунд при 100+ записей.
* Оптимизация запросов к БД

**4. Этапы разработки**

**4.1. Проектирование (1 неделя)**

* Схема БД (таблицы: Users, Subjects, Grades, Attendance)
* Диаграмма классов
* Макеты интерфейса (Figma)

**4.2. Реализация ядра (2 недели)**

* Модуль аутентификации
* Логика расчёта среднего балла, прогнозирования
* API для работы с PostgreSQL (Entity Framework Core)

**4.3. Разработка UI (1 неделя)**

* Главная страница с дашбордом
* Формы ввода оценок/посещаемости
* Настройка графиков (выбор типа диаграммы, периодов)

**4.4. Тестирование (3-5 дней)**

* Юнит-тесты на ключевые модули (расчёты, экспорт)
* UI-тесты (автоматизированные через WinAppDriver)
* Ручное тестирование (проверка на разных разрешениях экрана)

**4.5. Документирование (2-3 дня)**

* Руководство пользователя (PDF)
* Комментарии в коде (XML-документация)

**5. Дополнительные требования**

* **Резервное копирование** данных
* **Гибкая система настроек** (темы, параметры аналитики)

**6. Ограничения**

* Сроки: 1 месяц