Приложение А

Техническое задание

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

«Проектирование, моделирование и разработка автоматизированной информационной системы Система учета успеваемости студентов учебных групп для ГБПОУ КСТ».

Разработчик:

Студент группы СИП 313/23

Барышников Владимир Сергеевич

На\_\_\_\_\_листах

Согласованно

Руководитель

курсовой(дипломной) работы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.В,Павлов

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2025 г.

Москва

2025

Таблица 1 – Обозначения и сокращения

|  |  |
| --- | --- |
| API (Application Programming Interface | это набор правил, протоколов и инструментов, который позволяет различным программам и системам обмениваться данными и функциями, |
| БД | База данных |
| ГБПОУ КСТ. | Государственное бюджетное профессиональное образовательное учреждение города Москвы "Колледж современных технологий имени Героя Советского Союза М.Ф. Панова" |
| Клиент-серверная | это модель организации вычислительных систем, где клиент запрашивает услуги или данные, а сервер обрабатывает эти запросы и предоставляет ответ. |
| Язык программирования Rust | это современный компилируемый язык программирования, созданный для обеспечения высокой скорости, безопасности и надежности, особенно в системном программировании и высоконагруженных приложениях. |
| СУБД | Система управления базой данных |
| SQLite | это лёгкая, встраиваемая (внутрипроцессная) реляционная база данных, которая не требует отдельного сервера и хранит всю базу данных в одном файле на диске. |
| Рендеринг | процесс создания двумерного изображения или анимации из компьютерной модели или сцены |
| Role-Based Access Control | это метод управления правами доступа к ресурсам, при котором привилегии назначаются не отдельным пользователям, а ролям, соответствующим их должностным обязанностям в организации. |
| **REST API** | **архитектурный стиль, или набор правил, для создания веб-сервисов, позволяющий приложениям обмениваться данными по сети, используя стандартные протоколы HTTP.** |
| JSON | **это лёгкий текстовый формат для хранения и обмена структурированными данными, который легко читается человеком и обрабатывается машинами.** |
| CSV | **это текстовый формат файла для представления табличных данных, где каждая строка представляет запись, а значения в столбцах разделены запятыми или другими разделителями** |
| UI/UX | **это комплексное направление в создании цифровых продуктов (сайтов, приложений, сервисов), которое фокусируется на том, чтобы сделать взаимодействие пользователя с продуктом удобным, понятным и привлекательным.** |
| Tauri | **это фреймворк для создания кросс-платформенных настольных приложений с использованием веб-технологий (HTML, CSS, JavaScript) для фронтенда и языка Rust для бэкенда** |
| ФИО | Фамилия имя отчество |

Оглавление

[**1.** **ОБЩИЕ СВЕДЕНИ** 3](#_Toc208782314)

[**1.1.** **НАИМЕНОВАНИЕ СИСТЕМЫ** 3](#_Toc208782315)

[**1.2.** **ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ** 4](#_Toc208782316)

[**1.3.** **УЧАСТНИКИ ПРОЕКТА** 4](#_Toc208782317)

[**1.4.** **СРОКИ ВЫПОЛНЕНИЯ** 4](#_Toc208782318)

[**1.5.** **ЭТАПЫ РАЗРАБОТКИ:** 4](#_Toc208782319)

[**1.6.** **ИСТОЧНИКИ ФИНАНСИРОВАНИЯ** 4](#_Toc208782320)

[**1.7.** **ПОРЯДОК СДАЧИ РЕЗУЛЬТАТОВ** 4](#_Toc208782321)

[**2.** **НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ** 4](#_Toc208782322)

[**2.1.** **НАЗНАЧЕНИЕ СИСТЕМЫ** 4](#_Toc208782323)

[**2.2 ЦЕЛИ СОЗДАНИЯ** 5](#_Toc208782324)

[**3.** **ТРЕБОВАНИЯ К СИСТЕМЕ** 5](#_Toc208782325)

[**3.1.** **ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ** 5](#_Toc208782326)

[**3.2.** **ТРЕБОВАНИЯ К ПОЛЬЗОВАТЕЛЬСКОМУ ИНТЕРФЕЙСУ** 5](#_Toc208782327)

[**3.3.** **ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ** 7](#_Toc208782328)

[**3.4.** **ТРЕБОВАНИЯ К ПРОИЗВОДИТЕЛЬНОСТИ** 8](#_Toc208782329)

[**3.5.** **ТРЕБОВАНИЯ К ИНТЕГРАЦИИ** 8](#_Toc208782330)

[**4.** **ЭТАПЫ РАЗРАБОТКИ** 8](#_Toc208782331)

[**4.1.** **АНАЛИЗ И ПРОЕКТИРОВАНИЕ:** 8](#_Toc208782332)

[**4.2.** **РАЗРАБОТКА:** 9](#_Toc208782333)

[**4.3.** **ТЕСТИРОВАНИЕ:** 9](#_Toc208782334)

[**4.4.** **ВНЕДРЕНИЕ:** 9](#_Toc208782335)

[**4.5.** **СОПРОВОЖДЕНИЕ (ГАРАНТИЙНЫЙ ПЕРИОД):** 9](#_Toc208782336)

[**5.** **ПОРЯДОК ПРИЕМКИ** 9](#_Toc208782337)

[**5.1.** **ПРЕДВАРИТЕЛЬНЫЕ ИСПЫТАНИЯ:** 9](#_Toc208782338)

[**5.2.** **ОПЫТНАЯ ЭКСПЛУАТАЦИЯ:** 9](#_Toc208782339)

[**5.3.** **ПРИЕМОЧНЫЕ ИСПЫТАНИЯ:** 9](#_Toc208782340)

[6. **ТРЕБОВАНИЯ К ДОКУМЕНТАЦИИ** 9](#_Toc208782341)

[**7.** **ИСТОЧНИКИ РАЗРАБОТКИ** 10](#_Toc208782342)

# **ОБЩИЕ СВЕДЕНИ**

* 1. **НАИМЕНОВАНИЕ СИСТЕМЫ** 
     + Полное наименование: Проектирование, моделирование и разработка автоматизированной информационной системы «Система учета успеваемости студентов учебных групп для ГБПОУ КСТ».
     + Краткое наименование: ИС «Учет успеваемости КСТ».
  2. **ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ**

Работа выполняется на основании:

учебного плана специальности 09.02.07 – Информационные системы и программирование – курсовая работа.

* 1. **УЧАСТНИКИ ПРОЕКТА** 
     + Заказчик:
       - Наименование: колледж современных технологий имени Героя Советского
       - Союза М. Ф. Панова, учебный корпус Ярославский.
       - Адрес: г. Москва, Хибинский проезд 10
       - Контакты: +7 (499) 182-67-62
     + Разработчик:
       - Разработчик: студент группы СИП-312/23 Барышников Владимир Сергеевич
       - Адрес фактический: Лубянская площадь, дом 2
       - Контакты: +7 (977) 782-11-27
  2. **СРОКИ ВЫПОЛНЕНИЯ** 
     + Начало работ: 1 сентября 2025 года.
     + Окончание работ: 21 октября 2025 года.
  3. **ЭТАПЫ РАЗРАБОТКИ:** 
     + Проектирование – 15.09.25 г.
     + Моделирование – 21.09.25 г.
     + Разработка – 01.10.25 г.
     + Тестирование – 07.10.25 г.
     + Предпроектная защита – 10.10.25 г.
  4. **ИСТОЧНИКИ ФИНАНСИРОВАНИЯ**

Задание выполняется в рамках курсовой работы. Финансирование осуществляется за счёт средств разработчика.

* 1. **ПОРЯДОК СДАЧИ РЕЗУЛЬТАТОВ**

Результаты работ предоставляются поэтапно. По завершении каждого этапа Заказчик получает:

* 1. Отчет о выполнении.
  2. Демонстрацию функционала.

1. **НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ** 
   1. **НАЗНАЧЕНИЕ СИСТЕМЫ**

Информационная система предназначена для:

* + - Автоматизации учет успеваемости студентов, также обеспечивается контроль над посещаемостью.
    - Обеспечения хранение и удобный доступ к информации.
    - Облегчения выполнения своих обязанностей преподавательского состава ГБПОУ КСТ.
    - Создание отчётов об успеваемости студентов как, в частности, так и в формате группы.
    - Обеспечения разграничения прав доступа к данным в зависимости от роли пользователя.

## **2.2 ЦЕЛИ СОЗДАНИЯ**

* Повышение эффективности работы сотрудников учебного учреждения.
* Снижение трудозатрат за счет автоматизации.
* Улучшение качества данных и скорости их обработки.
* Обеспечение безопасности информации внутри учебного заведения.
* Контроль качества обучения в удобном формате информационной системе
* Возможность создание и формирование электронных отчетов, которые внутри информационной системе могут быть доставлены руководителям учебного заведения.

1. **ТРЕБОВАНИЯ К СИСТЕМЕ** 
   1. **ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ** 
      * Основные функции:
        + Выставление отметок в информационную систему.
        + Просмотр выставленных ранее отметок
        + Управление посещаемостью студентов каждой группы.
      * Дополнительные функции**:**
        + Создание отчетов за месяц.
        + Просмотр статистики выставленных отметок для каждой группы отдельно.
   2. **ТРЕБОВАНИЯ К ПОЛЬЗОВАТЕЛЬСКОМУ ИНТЕРФЕЙСУ** 
      * Тип интерфейса: Десктоп-клиент, адаптируемый под стандартные размеры мониторов.
      * Дизайн: Строгий, лаконичный интерфейс, соответствующий корпоративному стилю образовательного учреждения.
      * Удобство: интуитивно понятный интерфейс с элементами управления, ориентированными на неподготовленного пользователя.
      * Общие сведения
        + Система: «Информационная система учёта успеваемости студентов группы ГБПОУ КСТ».
        + Развёртывание: локально на каждом отдельном ПК (нет клиент-серверной архитектуры, данные хранятся на локальной БД, например SQLite/MS Access).
        + Интерфейс: настольное приложение (Windows Forms / WPF / Qt / JavaFX — зависит от твоего выбора).
      * Структура интерфейса (основные окна)
        + Главное окно (меню навигации).
        + Окно учёта студентов.
        + Окно учёта посещаемости.
        + Окно учёта оценок.
        + Окно формирования отчётов.
        + Окно «О программе» / справка.
      * Главное окно
        + Кнопка "Студенты"
        + Назначение: открывает окно управления списком студентов.
        + Расположение: верхняя панель главного окна, слева.
        + Событие при нажатии: вызывает форму «Список студентов».
      * Кнопка "Посещаемость"
        + Назначение: переход к учёту посещаемости.
        + Действие: открывает форму «Посещаемость».
      * Кнопка "Успеваемость"
        + Назначение: переход к окну выставления и просмотра оценок.
        + Действие: открывает форму «Оценки».
      * Кнопка "Отчёты"
        + Назначение: генерация отчётных документов (Word/Excel/печать).
        + Действие: открывает форму «Формирование отчётов».
      * Кнопка "Выход"
        + Назначение: завершение работы программы.
        + Действие: закрывает приложение с подтверждением.
      * Окно «Студенты»
        + Кнопка "Добавить студента"
        + Назначение: добавить запись о новом студенте.
        + Действие: открывает модальное окно «Новый студент» с полями (ФИО, дата рождения, группа).
      * Кнопка "Удалить студента"
        + Назначение: удалить выбранного студента из списка.
        + Действие: при нажатии появляется окно подтверждения удаления.
      * Кнопка "Редактировать"
        + Назначение: изменить данные выбранного студента.
        + Действие: открывает форму редактирования с уже заполненными полями.
      * Кнопка "Поиск" (рядом с полем ввода)
        + Назначение: фильтрация списка студентов по ФИО/ID.
        + Действие: в таблице остаются только подходящие записи.
      * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
      * Окно «Посещаемость»
        + Календарь
        + Назначение: выбор даты занятия.
        + Действие: отображает список студентов и чекбоксы «Присутствовал/Отсутствовал».
      * Кнопка "Сохранить"
        + Назначение: записать данные посещаемости в локальную БД.
      * Кнопка "Экспорт"
        + Назначение: выгрузить список посещаемости за период в Excel.
      * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
      * Окно «Успеваемость»
        + Таблица с фамилиями студентов и колонками по дисциплинам.
        + Кнопка "Выставить оценку"
        + Назначение: открыть форму выставления оценки.
        + Поля: студент, предмет, дата, оценка, комментарий.
      * Кнопка "Сохранить изменения"
        + Назначение: записать изменения в локальную БД.
      * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
      * Окно «Отчёты»
        + Кнопка "Сформировать отчёт по успеваемости"
        + Назначение: построение сводной таблицы/графика по оценкам студентов.
        + Кнопка "Сформировать отчёт по посещаемости"
        + Назначение: построение отчёта за период.
        + Кнопка "Экспорт в PDF/Excel"
        + Назначение: выгрузка отчёта в файл.
   3. **ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ**
      * Архитектура: Клиент-серверная.
      * Серверная часть:
        + Языки программирования: Rust.
        + СУБД: SQLite.
      * Клиентская часть:
        + Необходимые характеристики: Windows 10, доступ к интернету.
        + Окружение: Рендеринг осуществляется в веб-вью, предоставляемом операционной системой (WebView2 на Windows, WKWebView на macOS, WebKitGTK на Linux). Это обеспечивает совместимость и высокую производительность.
      * Безопасность:
        + Аутентификация: По логину и паролю. Пароли хранятся в хэшированном виде с использованием соли (например, с помощью алгоритма bcrypt или Argon2).
        + **Авторизация:** на основе ролей (Role-Based Access Control), реализованная на уровне ядра (Rust).
        + **Защита от уязвимостей:** благодаря использованию Rust (память-безопасный язык) и подготовленных запросов к SQLite риски SQL-инъекций и ошибок, связанных с памятью, сведены к минимуму. Защита от XSS обеспечивается санитизацией данных на стороне ядра перед их возвратом в UI.
        + **Шифрование данных:** Шифрование соединения (HTTPS) не требуется, так как коммуникация между клиентом и ядром происходит локально через высокопроизводительный IPC-канал. Для шифрования файла базы данных по желанию пользователя могут быть использованы сторонние SQLite-расширения (например, SQLCipher).

## **ТРЕБОВАНИЯ К ПРОИЗВОДИТЕЛЬНОСТИ**

* + - Время отклика системы: не более 2 секунд.
    - Количество одновременных пользователей: до 40.
    - Объем обрабатываемых данных: до 20 ГБ.
  1. **ТРЕБОВАНИЯ К ИНТЕГРАЦИИ** 
     + **Внешние системы:** на первом этапе интеграция с внешними системами не предусмотрена. Система должна быть спроектирована с возможностью последующего расширения функционала. В будущем интеграция (например, выгрузка данных в 1C) может быть реализована через:
     + **REST API:** Развертывание отдельного минимального HTTP-сервера в фоновом режиме для обслуживания запросов.
     + **Локальные файлы:** Экспорт данных в стандартные форматы (CSV, JSON) для последующей обработки внешними системами.
     + **Форматы данных:** для импорта/экспорта данных система должна поддерживать форматы CSV и JSON. Генерация отчетов также может быть реализована в этих форматах.

1. **ЭТАПЫ РАЗРАБОТКИ** 
   1. **АНАЛИЗ И ПРОЕКТИРОВАНИЕ:**
      * Детальный сбор и формализация требований.
      * Проектирование логической и физической структуры базы данных.
      * Создание технического проекта и макетов пользовательского интерфейса (UI/UX).
      * Согласование ТЗ и макетов с Заказчиком.
   2. **РАЗРАБОТКА:**
      * **Реализация серверной части (Ядро / Backend)**на Rust с использованием фреймворка Tauri. Создание API для безопасного взаимодействия с фронтендом.
      * **Проектирование и миграция базы данных** SQLite.
      * **Разработка клиентской части (Frontend)** на JavaScript.
      * **Реализация механизмов аутентификации и авторизации** на стороне Rust.
      * **Написание модулей отчётности,** генерирующих данные для последующего отображения во фронтенде или экспорта в файлы.
   3. **ТЕСТИРОВАНИЕ:**
      * Функциональное тестирование всех модулей системы.
      * Нагрузочное тестирование (в рамках заявленных требований).
      * Тестирование безопасности.
      * Исправление выявленных ошибок.
   4. **ВНЕДРЕНИЕ:**
      * Развертывание системы на тестовом стенде Заказчика.
      * Презентация системы и обучение ключевых пользователей (администраторов).
      * Перенос данных из существующих учетных систем (при наличии и возможности).
      * Развертывание на рабочем сервере.
   5. **СОПРОВОЖДЕНИЕ (ГАРАНТИЙНЫЙ ПЕРИОД):**
      * Техническая поддержка пользователей.
      * Мониторинг работы системы.
      * Исправление критических ошибок, выявленных в процессе эксплуатации.
2. **ПОРЯДОК ПРИЕМКИ** 
   1. **ПРЕДВАРИТЕЛЬНЫЕ ИСПЫТАНИЯ:**

Проверка функционала на тестовом стенде Разработчика по заранее согласованному с Заказчиком тест-плану. Устранение всех критических и существенных замечаний.

* 1. **ОПЫТНАЯ ЭКСПЛУАТАЦИЯ:**

Тестирование системы ограниченным кругом пользователей Заказчика на тестовом стенде в течение [, например, 14] календарных дней. Фиксация всех замечаний.

* 1. **ПРИЕМОЧНЫЕ ИСПЫТАНИЯ:**

Окончательная проверка системы на рабочем сервере Заказчика по полному тест-плану. Подписание Акта сдачи-приемки работ после успешного прохождения испытаний и устранения всех замечаний.

1. **ТРЕБОВАНИЯ К ДОКУМЕНТАЦИИ**

* Техническая документация:
  + Руководство системного администратора (инструкция по установке, настройке и развертыванию).
  + Описание архитектуры системы и API (при наличии).
  + Описание базы данных (ER-диаграмма).
* Пользовательская документация:
  + Руководство пользователя для каждой роли (Администратор, Преподаватель, Лаборант) с пошаговыми инструкциями.
* Отчеты
  + Предоставление отчетов по итогам каждого этапа работ.

1. **ИСТОЧНИКИ РАЗРАБОТКИ**
   * + Федеральный закон № 152-ФЗ «О персональных данных».
     + Внутренние регламенты и документы по учету материально-технических ценностей ГБПОУ КСТ.
     + Настоящее техническое задание.

Приложения:

• Макеты интерфейсов (обязательно).

• Дополнительные технические спецификации.

Подписи сторон:

Заказчик: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / [ФИО]

Разработчик: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / [ФИО]

Дата: \_\_\_\_\_\_\_\_\_\_\_