**Командный кейс №5**

**«Управление спортивным инвентарем»**

**Участники команды:**

Тычкин Фёдор Юрьевич 10И

ГБОУ школа №2098

Гуркин Степан Александрович 10И

ГБОУ школа №2098

Шикин Илья Владимирович 10И

ГБОУ школа №2098

Климович Валентин Андреевич 10И

ГБОУ школа №2098

Машутиков Владислав Сергеевич 10И

ГБОУ школа №2098

Ссылка на репозиторий в GitHub: <https://github.com/Xzimya1/ProjectPredProf>

**Обоснование выбора языка программирования и используемых программных средств:**

**Оптимальный подбор инструментов для разработки базы данных и веб интерфейса.**

Для успешной разработки проекта, включающего базу данных и веб-интерфейс, критически важен выбор технологий. Мы выбрали SQLite3 и Python для работы с данными, а HTML и Flask - для создания веб-интерфейса, чтобы обеспечить эффективность и производительность системы.

**Простота и удобство: почему мы выбрали SQLite3 для проекта.**

SQLite3 — это простая и легковесная встроенная СУБД, которая хранит все данные в одном файле, что упрощает управление и переносимость. Она идеальна для небольших и средних приложений, не требует сложной серверной инфраструктуры и позволяет легко редактировать данные. Благодаря опыту работы с этой библиотекой, мы выбрали её для нашего проекта, так как она позволяет сосредоточиться на разработке функционала.

**Эффективная разработка приложений: Python и SQLite3 в одном пакете.**

Python в сочетании с SQLite3 упрощает разработку приложений благодаря простому синтаксису и удобным функциям библиотеки SQLite. SQL-запросы можно выполнять прямо из кода Python, что ускоряет работу с данными и облегчает разработку. Это позволяет быстро создавать и тестировать проекты без сложностей интеграции с другими системами.

**HTML: основной инструмент для создания веб-интерфейса.**

Для создания веб-интерфейса мы выбрали HTML благодаря его простоте и универсальности. HTML позволяет структурировать контент, добавлять формы, изображения и другие элементы, являясь основой веб-приложений. В сочетании с CSS и JavaScript он помогает создавать интерактивные и визуально привлекательные интерфейсы, улучшая взаимодействие с пользователями.

**Интерактивность и управляемость: преимущества использования Flask.**

HTML недостаточен для динамических веб-приложений, поэтому мы используем Flask - микрофреймворк для Python. Flask добавляет интерактивность и управляемость на стороне сервера, поддерживая маршрутизацию, обработку форм, управление сессиями и интеграцию с базами данных, включая SQLite3. Это упрощает реализацию необходимого функционала.

**Гибкость Flask: ключ к созданию разнообразных веб-приложений.**

Одной из ключевых особенностей Flask является его гибкость. Разработчик может самостоятельно решать, какие компоненты использовать, что делает его подходящим для различных типов приложений, от простых до сложных многофункциональных сервисов. Благодаря этому, мы можем включать только те модули и библиотеки, которые необходимы для нашего проекта, избегая ненужной нагрузки и увеличивая производительность.

**Простота документации Flask: ускорение процесса разработки.**

Кроме того, Flask имеет простую и понятную документацию, что значительно ускоряет процесс разработки, так как мы можем быстро находить нужную информацию о функциях и возможностях фреймворка. Это также позволяет легко интегрировать различные расширения, что дает возможность добавлять функционал без значительного увеличения сложности проекта.

**Гармоничное взаимодействие: объединение базы данных, веб-интерфейса и серверного функционала.**

Наш проект объединяет базу данных (SQLite3), веб-интерфейс и серверный функционал через Flask. Он обрабатывает запросы пользователя (например, из форм), взаимодействует с базой данных для сохранения, редактирования или получения данных, обеспечивая слаженную работу всех компонентов.

**Реальное время и адаптивный интерфейс: как Flask и SQLite3 повышают пользовательский опыт.**

Использование SQLite3 и Flask вместе с HTML создает мощный и гибкий стек технологий, который позволяет быстро разрабатывать и внедрять новые функции. Мы можем обновлять данные в реальном времени, создавать различные представления информации и адаптировать веб-интерфейс под потребности пользователей. Это особенно важно в условиях современного веб-разработки, где пользовательский опыт имеет первостепенное значение.

**SQLite 3**

SQLite 3 — это легковесная реляционная система управления базами данных, которая имеет множество характеристик, делающих её идеальным выбором для ряда приложений и ситуаций. Ниже приведены ключевые особенности SQLite и аргументы в пользу её использования.

**Особенности SQLite 3:**

1. Серверная архитектура:

SQLite является встраиваемой базой данных, что означает, что она не требует отдельного сервера. Вместо этого база данных хранится в одном файле на диске. Это упрощает управление и развертывание приложений.

2. Легковесность:

SQLite имеет небольшой размер (библиотека меньше 500 КБ) и минимальные системные требования, что делает её идеальной для мобильных и встраиваемых приложений.

3. Простота использования:

SQLite не требует настройки. Для начала работы достаточно добавить библиотеку в проект и создать базу данных, что делает её идеальным выбором для разработчиков с разным уровнем опыта.

4. Файловая система:

База данных хранится как один файл, что упрощает резервное копирование, перенос и управление данными. Это также облегчает параллельный доступ к файлу базы данных.

5. Поддержка SQL:

SQLite поддерживает большинство стандартного SQL, что делает его мощным инструментом для разработки приложений, использующих реляционные данные.

6. Кроссплатформенность:

SQLite работает на различных платформах: Windows, MacOS, Linux, iOS и Android. Это делает её подходящей для многоуровневых приложений и кроссплатформенной разработки.

7. Транзакционность:

SQLite поддерживает ACID-транзакции (атомарность, согласованность, изолированность и долговечность), что обеспечивает целостность данных.

8. Хранение в памяти:

SQLite поддерживает базы данных, которые могут храниться в оперативной памяти для максимальной производительности, что полезно для тестирования или приложений, где скорость доступа критична.

**Аргументы в пользу использования SQLite:**

1. Легкость развертывания: благодаря простоте и легковесности SQLite идеально подходит для небольших приложений, прототипирования и разработки. Не требуется сложной инфраструктуры для управления.

2. Снижение затрат: Отсутствие серверной части сокращает эксплуатационные расходы. Также не нужно лицензировать программное обеспечение, так как SQLite является свободным и открытым.

3. Простота бэкапа и миграции: Все данные хранятся в одном файле, что упрощает резервное копирование и перемещение данных между системами.

4. Идеально для мобильных приложений: SQLite часто используется в мобильной разработке, так как позволяет эффективно хранить данные на устройствах с ограниченными ресурсами.

5. Достаточная функциональность: хотя SQLite легковесен, он поддерживает множество функций, необходимых для управления базами данных, что делает его мощным инструментом в арсенале разработчика.

SQLite 3 является отличным выбором для разработчиков, которым нужна простая, эффективная и мощная база данных. Несмотря на свою легковесность, она предоставляет все необходимые инструменты для работы с данными. Это делает её идеальным выбором для мобильных приложений, небольших веб-приложений, прототипов и для разработчиков, ищущих быстроту и удобство без лишних затрат.

**Заключение: синергия технологий – SQLite3, Python, HTML и Flask для успешного проекта.**

Наш проект использует стек технологий (SQLite3, Python, HTML, Flask), которые обеспечивают создание удобного и эффективного приложения. Опыт работы с SQLite3 гарантирует надежность базы данных, а Flask позволяет гибко разрабатывать веб-интерфейс. Это позволяет сосредоточиться на функционале и предоставлении пользователям качественного продукта.

**Структурная схема**

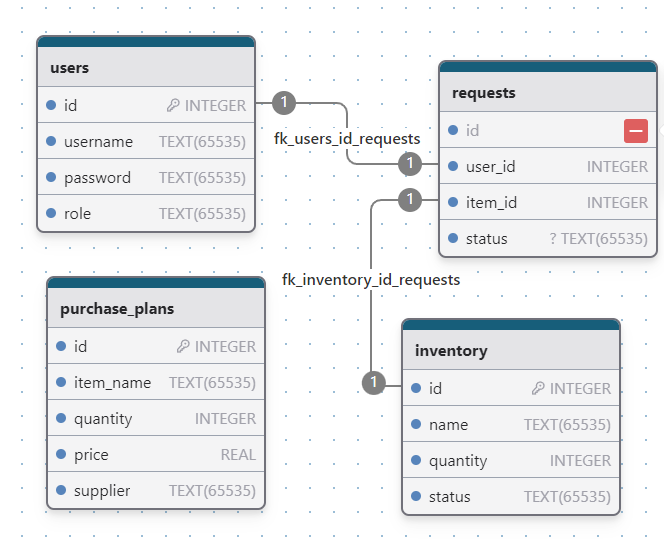
Пользователь > Интерфейс > Администратор>  База данных(Инвентарь, Заявки, Закупки)

Функции администратора

Авторизация, Добавление инвентаря, Редактирование инвентаря, Закрепление инвентаря, Планирование закупок, Создание отчетов

Функции пользователя

Регистрация, Авторизация, Просмотр инвентаря, Создание заявок, Отслеживание статуса



**Функциональная схема**

