**=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

Обоснование выбора языка программирования и используемых программных средств:

Оптимальный подбор инструментов для разработки базы данных и веб интерфейса.

При разработке проектов, связанных с хранением и обработкой данных, выбор подходящих технологий и инструментов играет критически важную роль. На этапе планирования нашего проекта, который включает в себя создание базы данных и веб-интерфейса, мы осознали важность выбора правильных инструментов, чтобы обеспечить эффективную работу и высокую производительность системы. В результате, для реализации базы данных был выбран SQLite3 в сочетании с языком программирования Python, а для веб-интерфейса мы решили использовать комбинацию HTML и Flask.

Простота и удобство: почему мы выбрали SQLite3 для проекта.

SQLite3 является легковесной, встроенной СУБД, которая идеально подходит для небольших и средних приложений. Основным преимуществом работы с SQLite3 является его простота и удобство в использовании. Важной частью этой базы данных является то, что она хранит всю информацию в одном файле, что упрощает управление данными и их переносимость. Поскольку у нас уже был достаточный опыт работы с этой библиотекой, мы уверенно решили использовать её в нашем проекте. Возможность легкого редактирования и взаимодействия с данными, а также работа без необходимости установки сложной серверной инфраструктуры выделяют SQLite3 среди других систем управления базами данных. Эти особенности позволяют сосредоточиться на разработке функционала, а не тратить время на настройку среды.

Эффективная разработка приложений: Python и SQLite3 в одном пакете.

Использование Python в сочетании с SQLite3 позволяет быстро и эффективно создавать приложения. Библиотека SQLite в Python обладает множеством полезных функций, которые делают взаимодействие с базой данных максимально простым. Например, возможность выполнять SQL-запросы прямо из кода Python значительно упрощает работу с данными и обеспечивает легкость в разработке. Благодаря простому синтаксису Python и мощным возможностям SQLite, мы можем быстро разрабатывать и тестировать различные аспекты нашего проекта, не отвлекаясь на сложности интеграции баз данных с другими системами.

HTML: основной инструмент для создания веб-интерфейса.

Что касается веб-интерфейса, мы выбрали HTML как основной инструмент для его создания. HTML — это стандартный язык разметки для создания веб-страниц. Его простота и универсальность делают его идеальным выбором для разработки веб-дизайна. HTML позволяет структурировать контент, добавлять формы, изображения и различные элементы, что делает его основой любого веб-приложения. Использование HTML совместно с CSS и JavaScript дает нам возможность создавать красивые и интерактивные интерфейсы, что значительно улучшает взаимодействие с пользователями.

Интерактивность и управляемость: преимущества использования Flask.

Однако HTML сам по себе не обеспечивает полноценного функционала, необходимого для динамических веб-приложений. Чтобы добавить интерактивность и управляемость на серверной стороне, мы используем Flask, микрофреймворк для Python, который прекрасно сочетается с нашими технологиями. Flask предлагает большое количество готовых решений для создания веб-приложений, что позволяет нам легко реализовать необходимый функционал. В том числе, он поддерживает маршрутизацию, обработку форм, управление сессиями и интеграцию с различными базами данных, в том числе SQLite3.

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.

В нашем проекте мы стремимся создать гармоничное взаимодействие между базой данных, веб-интерфейсом и серверным функционалом. Flask обеспечивает маршрутизацию запросов от пользователя, обрабатывает их и взаимодействует с базой данных SQLite3 для выполнения необходимых операций. Пользователь отправляет запрос через веб-интерфейс, например, заполняя формулу, а Flask обрабатывает этот запрос и, в случае необходимости, обращается к базе данных, чтобы сохранить, отредактировать или получить данные.

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.

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

В заключение, проект, который мы реализуем, опирается на тщательно подобранный стек технологий, включающий SQLite3, Python, HTML и Flask. Эти инструменты, обладая необходимыми свойствами и функциональностью, позволяют создавать эффективные и удобные приложения, которые отвечают всем современным требованиям. Наличие опыта работы с SQLite3 дает нам уверенность в построении надежной базы данных, в то время как Flask позволяет творчество в разработке веб-интерфейса. Таким образом, мы можем сосредоточиться на разработке функционала и предоставлении пользователям качественного продукта, который будет удобен, быстр и эффективен при работе с данными.

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.

Структурная схема

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

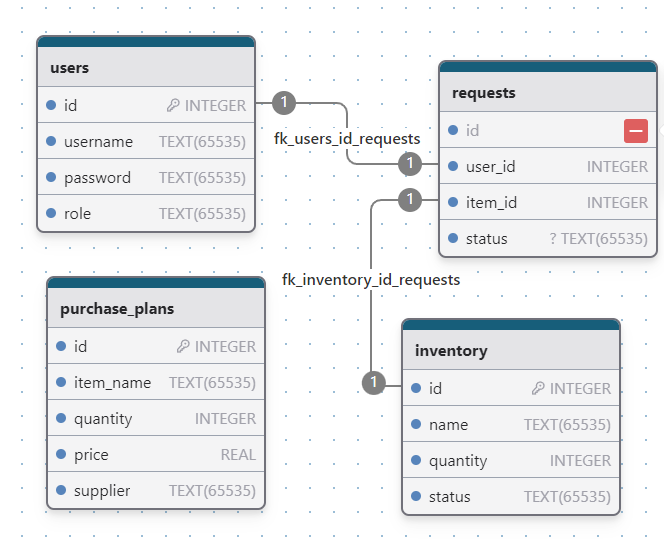
Функциональная схема:

Функции администратора

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

Функции пользователя

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



­­­

