**Техническое задание: Веб-сайт задачника**

**Введение**

Создание веб-сайта задачника, который будет отображать, добавлять и удалять задачи, хранить данные в формате CSV и представлять задачи в виде HTML-страницы. Сайт должен предоставлять функционал для пользователей по управлению своими задачами.

**Цели проекта**

1. Познакомиться с основами веб-разработки на Python.
2. Научиться работать с CSV-файлами для хранения данных.
3. Освоить основы HTML и CSS для создания интерфейса.
4. Реализовать CRUD-функционал (Create, Read, Update, Delete) для задач.

**Технические требования**

**1. Архитектура приложения**

* Язык программирования: Python
* Веб-фреймворк: Flask (или Django, по желанию)
* Хранение данных: CSV
* Фронтенд: HTML, CSS (возможно использование Bootstrap для стилизации)

**2. Функциональные требования**

**2.1. Главная страница**

* Отображение списка всех задач из CSV-файла.
* Каждая задача должна отображать:
  + Название
  + Описание
  + Статус (завершена/незавершена)
  + Дата создания

**2.2. Добавление задачи**

* Формуляр для добавления новой задачи.
* Поля для ввода:
  + Название
  + Описание
* Кнопка "Добавить" будет сохранять задачу в CSV-файл.

**2.3. Удаление задачи**

* У каждой задачи должна быть кнопка "Удалить".
* При нажатии на кнопку задача удаляется из CSV-файла.

**2.4. Редактирование задачи**

* При нажатии на задачу, пользователь должен иметь возможность редактировать её:
  + Изменение названия и описания.
  + Сохранение изменений в CSV-файл.

**2.5. Фильтрация и сортировка задач**

* Возможность фильтровать задачи по статусу (все, завершённые, незавершённые).
* Возможность сортировки задач по дате создания.

**3. Нефункциональные требования**

* Приложение должно иметь простой и интуитивно понятный интерфейс.
* Код должен быть написан с соблюдением стандартов PEP 8.
* Необходимо использовать Git для управления версиями кода.
* Требуется добавить комментарии в код.

**Этапы разработки**

1. **Исследование и планирование**
   * Проанализировать существующие решения.
   * Спроектировать структуру приложения.
2. **Настройка окружения**
   * Установить Flask и необходимые библиотеки.
   * Настроить структуру проекта.
3. **Создание функционала**
   * Реализовать CRUD-функционал.
   * Обработать данные из CSV.
4. **Создание интерфейса**
   * Разработать HTML-страницы с формами.
   * Стилизовать страницы с помощью CSS.
5. **Тестирование**
   * Провести тестирование функционала.
   * Исправить найденные ошибки.
6. **Документация**
   * Написать документацию для проекта.

**Завершение проекта**

* Опубликовать проект на GitHub.
* Подготовить краткое видео с демонстрацией функционала.