**Проект «Менеджер задач»** — это инструмент для управления задачами, который помогает пользователям организовывать свои дела и эффективно управлять временем. Основная цель — предоставить пользователю удобный интерфейс для создания, редактирования, удаления и отслеживания задач.

**Целевая аудитория –** это люди, которым нужно структурировать дела: от школьников, ведущих список домашних заданий, до руководителей, контролирующих проекты.

**Применение для получения прибыли:**

Программа «Менеджер задач» может использовать встроенную рекламу для получения прибыли. Рекламные объявления могут отображаться в интерфейсе программы.

Для пользователей, которые хотят получить доступ к расширенным функциям, предлагается подписочная модель. Подписка может включать дополнительные возможности, такие как синхронизация между устройствами, расширенные настройки задач или приоритетная поддержка, что делает её привлекательной для более требовательных пользователей.

Основные задачи включали:

* Проектирование и реализацию приложения с функционалом создания, редактирования, удаления и отслеживания задач.
* Обеспечение автосохранения данных.
* Разработку интуитивно понятного интерфейса.
* Тестирование и отладку программы.

Были рассмотрены следующие аналоги:

1. To-Do-List:

- Преимущества: простота установки и интерфейса.

- Недостатки: ограниченный функционал, отсутствие автосохранения.

2. ToDoList-Python:

- Преимущества: низкое потребление ресурсов.

- Недостатки: интерфейс командной строки, отсутствие категорий.

3. ToDo:

- Преимущества: удобный интерфейс, поддержка приоритетов.

- Недостатки: устаревший дизайн, невозможность редактирования задач.

Выявленные проблемы в аналогах:

- Отсутствие гибкости в управлении задачами.

- Недостаточная функциональность (например, отсутствие напоминаний или экспорта данных).

- Сложность интерфейса для неподготовленных пользователей.

Этапы проектирования

1. Настройка рабочего окружения:

- Выбор Python и библиотеки PySide6 для разработки GUI.

- Установка Git и GitHub для контроля версий.

- Настройка Visual Studio Code как среды разработки.

2. Анализ предметной области:

- Изучение требований к функционалу.

- Определение структуры данных (JSON для хранения задач).

3. Проектирование интерфейса:

- Создание макетов в Figma.

- Реализация главного окна, диалоговых окон для добавления и редактирования задач.

4. Реализация функционала:

- Разработка алгоритмов для добавления, редактирования, удаления задач.

- Реализация экспорта и импорта данных в формате iCalendar.

- Автоматическая проверка сроков выполнения задач.

5. Тестирование:

- Проверка корректности работы всех функций.

- Устранение выявленных ошибок.

Выводы

В ходе практики были успешно выполнены все поставленные задачи:

- Разработано приложение "Менеджер задач" с полным набором запланированных функций.

- Освоены современные инструменты разработки, включая Python, PySide6, Git и GitHub.

- Приобретены навыки проектирования, тестирования и документирования программного обеспечения.

Программа готова к эксплуатации и может быть доработана для коммерческого использования. Практика позволила углубить понимание процессов разработки ПО и применения полученных знаний на практике.