1. **ПРЕДМЕТНАЯ ОБЛАСТЬ**
   1. **История и описание**

**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.

Менеджеры задач, или приложения для управления задачами, возникли как инструменты для организации и отслеживания дел и проектов. Их история охватывает несколько ключевых этапов:

* Первые задатки управления задачами появились в древних цивилизациях с использованием глиняных табличек и папирусов для записи обязательств [1]
* С изобретением бумаги в средние века использовались записные книжки и ежедневники для учета задач
* В 1950-60-х годах появились формализованные подходы к управлению проектами, такие как графики Ганта
* С 1980-х годов начали использовать электронные таблицы и специализированные программы, что упростило организацию задач
* В 1990-х и 2000-х годах появились более сложные решения, такие как Microsoft Project и Trello, с интуитивно понятными интерфейсами
* Сегодня трекеры предлагают множество функций, включая интеграцию с другими сервисами, автоматизацию и возможности для командной работы
  1. **Целевая аудитория:**
* **Студенты** - Для организации учебного процесса, отслеживания дедлайнов по курсовым, проектам и экзаменам
* **Офисные сотрудники** - Для планирования рабочих задач, встреч и других обязанностей
* **Те, кто часто забывает о задачах**
* **Люди с высокой загруженностью** - Для тех, кто хочет структурировать свои дела и не забывать о важных задачах
  1. **Применение для получения прибыли:**
* Подписочная модель
* Корпоративное использование
* Встроенная реклама
* Мобильное приложение
  1. **Аналоги:**

1. To-Do-List, wynand1004, 2020 год

<https://github.com/wynand1004/Projects/tree/master/To-Do-List>

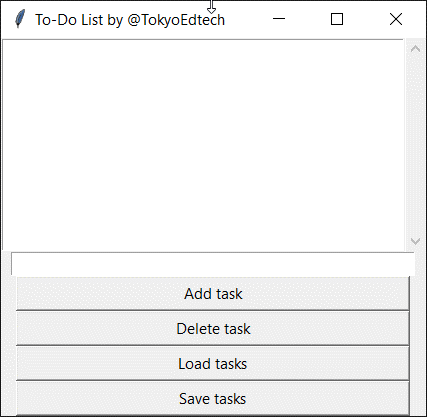


Рисунок 1 – Программа To-Do-List

**Плюсы:**

* Приложение имеет простую установку и не требует сложной настройки
* Простой и понятный интерфейс, пользователи могут быстро понять, как добавить, редактировать и удалять задачи

**Минусы:**

* Приложение ограничивается базовыми функциями, такими как добавление и удаление задач
* Отсутствие опции отметки задачи как выполненной
* Отсутствие функции автоматического сохранения может привести к потере данных в случае сбоев или закрытия приложения

1. ToDoList-Python, Varnithvarma, 2023 год

<https://github.com/Varnithvarma/ToDoList-Python>

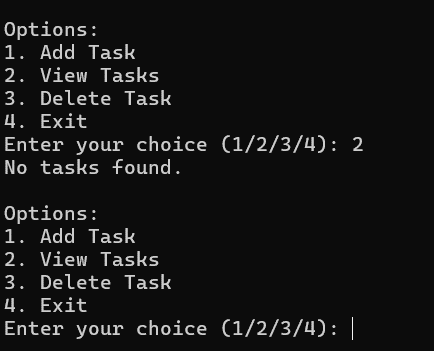


Рисунок 2 – Программа ToDoList-Python

**Плюсы:**

* Приложение не требует много ресурсов системы
* Имеет простой и быстрый текстовый интерфейс

**Минусы:**

* Работа в командной строке может отпугнуть пользователей, не имеющих соответствующего опыта
* Функциональность ограничена базовыми задачами
* Отсутствие функции категорий усложняет организацию задач

1. ToDo, ajaretemi, 2025 год

<https://github.com/ajaretemi/ToDo---Python>

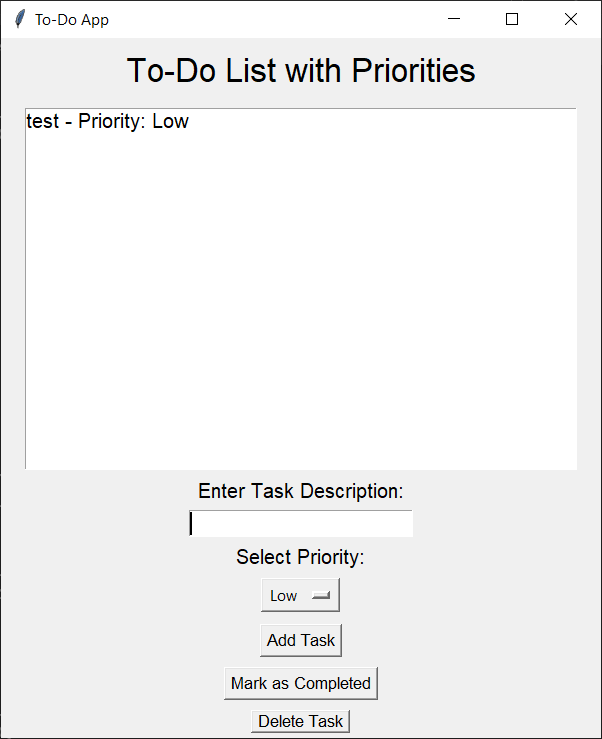


Рисунок 3 – Программа To-Do App

**Плюсы:**

* Интерфейс является удобным для пользователей
* Функции приоритета и возможности отметить задачу выполненной
* Приложение не нагружает систему и быстро запускается,

**Минусы:**

* Устаревший дизайн может показаться непривлекательным современным пользователям
* Отсутствие возможности группировать задачи по категориям снижает гибкость

**ВЫВОД**

При создании менеджера задач необходимо учесть несколько важных аспектов. Продукт должен быть простым в установке и настройке, чтобы пользователи с любым уровнем навыков могли быстро разобраться. Интерфейс приложения должен быть современным и удобным, с поддержкой категорий и отображением статусов задач. Важно предусмотреть создание, редактирование, удаление и отметку задач как выполненных, а также возможность группировки задач по категориям и автоматического сохранения данных для предотвращения их потери.

Для коммерческого успеха необходимо рассмотреть варианты монетизации, такие как подписочная модель, корпоративное использование с расширенными функциями для командной работы или встроенная реклама. Учитывая эти аспекты, можно создать конкурентоспособное приложение, которое привлечёт широкий круг пользователей и будет отвечать их требованиям, обеспечивая при этом стабильный доход для разработчиков.

1. **АЛГОРИТМЫ ИСПОЛЬЗОВАНИЯ ПРОГРАММЫ**

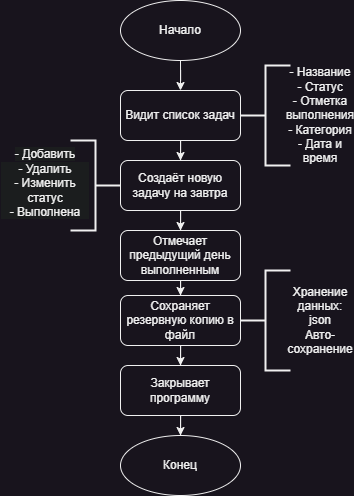
****

Рисунок 4 – Алгоритм действий пользователя

Видит список задач - Пользователь открывает приложение и видит текущий список задач. Список может включать задачи с разzличными состояниями (актуальные, завершенные, задачи на завтра). Это основной экран приложения.

Создаёт новую задачу на завтра - Пользователь может добавить новую задачу для выполнения на следующий день. При этом пользователь вводит название задачи, а также может добавить описание.

Отмечает предыдущий день выполненным - Пользователь может отметить задачи предыдущего дня как выполненные. Это действие фиксирует выполненные задачи и изменяет их статус.

Сохраняет резервную копию в файл - Программа автоматически создает резервную копию всех задач в виде файла

Закрывает программу - После завершения работы пользователь закрывает приложение

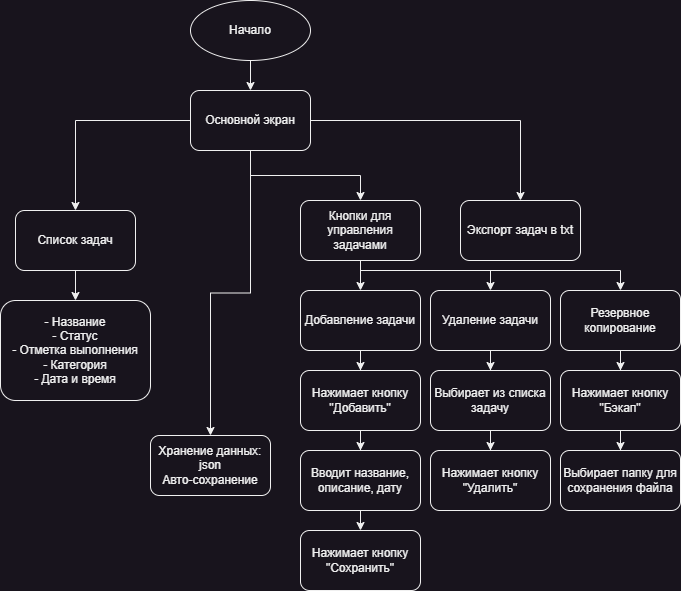


Рисунок 5 – Доступный функционал

**3. ВЫБОР ЯЗЫКА ПРОГРАММИРОВАНИЯ И НАСТРОЙКА РАБОЧЕГО ОКРУЖЕНИЯ**

В данном разделе представлена инструкция по выбору языка программирования и среды разработки, а также настройке рабочего окружения для создания нативных Desktop-приложений на Python с использованием библиотеки PySide6 и среды разработки Visual Studio Code (VSCode).

**3.1. Выбор языка программирования и среды разработки**

**Python** — высокоуровневый язык программирования, известный своей простотой и читаемостью. Он был создан Гвидо ван Россумом и впервые выпущен в 1991 году. Благодаря обширному выбору библиотек и поддержке множества сторонних модулей, Python подходит для различных областей разработки, включая создание нативных Desktop-приложений.

**Visual Studio Code** — редактор исходного кода от компании Microsoft. С 2018 года, благодаря расширению Python для Visual Studio Code с открытым исходным кодом, VSCode предоставляет разработчикам широкие возможности для редактирования, отладки и тестирования кода на языке Python.

**3.2. Причины выбора Python**

Python был выбран из-за его простоты, популярности и широкого набора библиотек, облегчающих разработку. Кроме того, Python имеет активное сообщество, что обеспечивает доступ к множеству ресурсов и поддержке. [2]

**3.3. Что такое Git и Git Bash**

**Git** — это распределенная система контроля версий, позволяющая отслеживать изменения в коде и работать совместно с другими разработчиками. Она обеспечивает эффективное управление версиями проекта и упрощает процесс разработки. [3]

**Git Bash** — это приложение, позволяющее использовать команды Git для работы с системой управления версиями.

**3.4. Описание библиотеки PySide6**

**PySide6** — это официальная Python-библиотека проекта Qt for Python, предоставляющая доступ к полному набору возможностей фреймворка Qt 6.0+. Она позволяет создавать кроссплатформенные графические интерфейсы пользователя (GUI) с использованием языка Python. [4]

**3.5. Этапы установки и настройки рабочего окружения**

**Шаг 1: Установка Python**

Перейдите на официальный сайт Python: https://www.python.org/

Скачайте последнюю версию Python для вашей операционной системы.

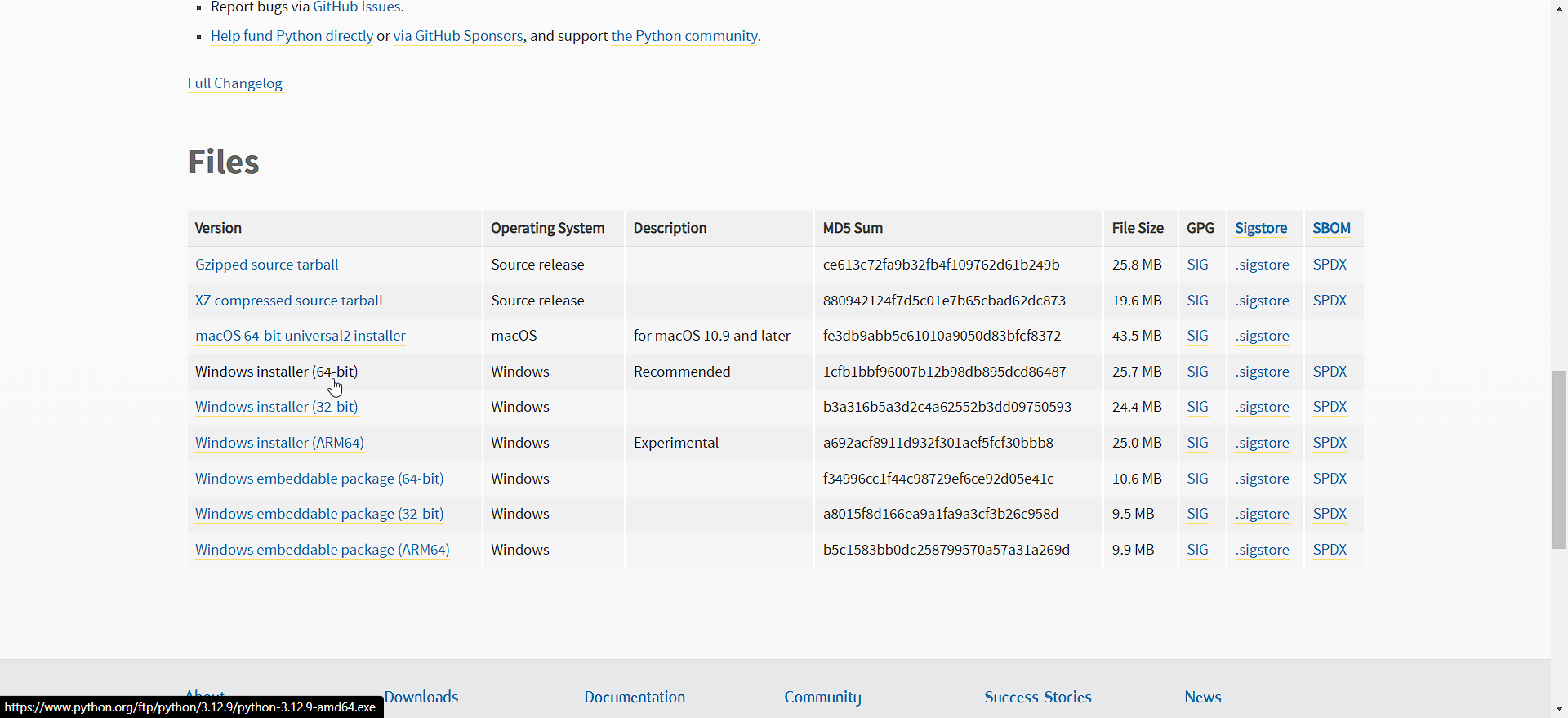


Рисунок 6 – Скриншот веб-сайта python.org

Запустите установочный файл и следуйте инструкциям установщика.

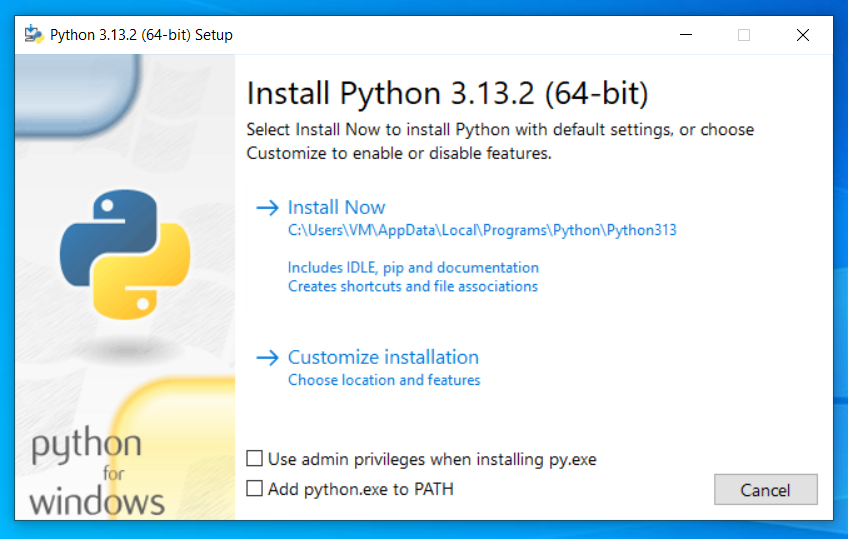


Рисунок 7 – Скриншот окна установщика Python

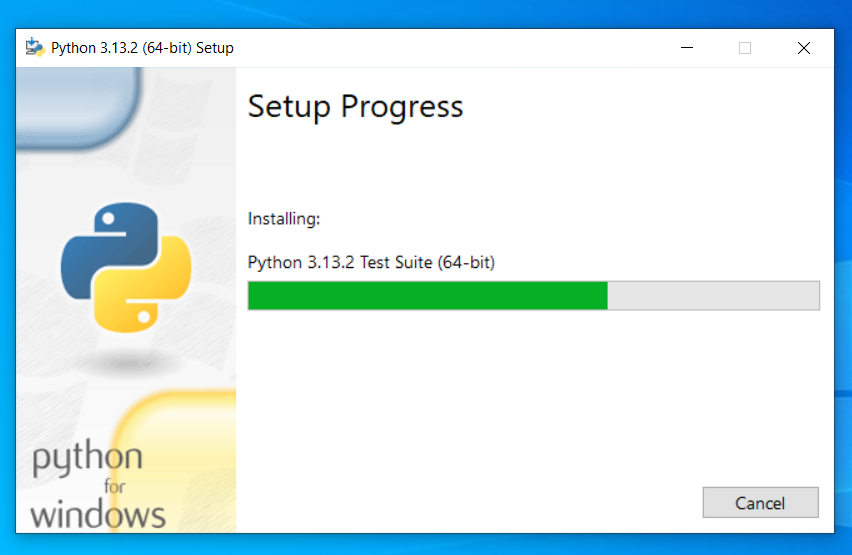


Рисунок 8 – Скриншот окна процесса установки Python

**Шаг 2: Установка Visual Studio Code**

Перейдите на официальный сайт VSCode: https://code.visualstudio.com/

Скачайте версию VSCode для вашей операционной системы.

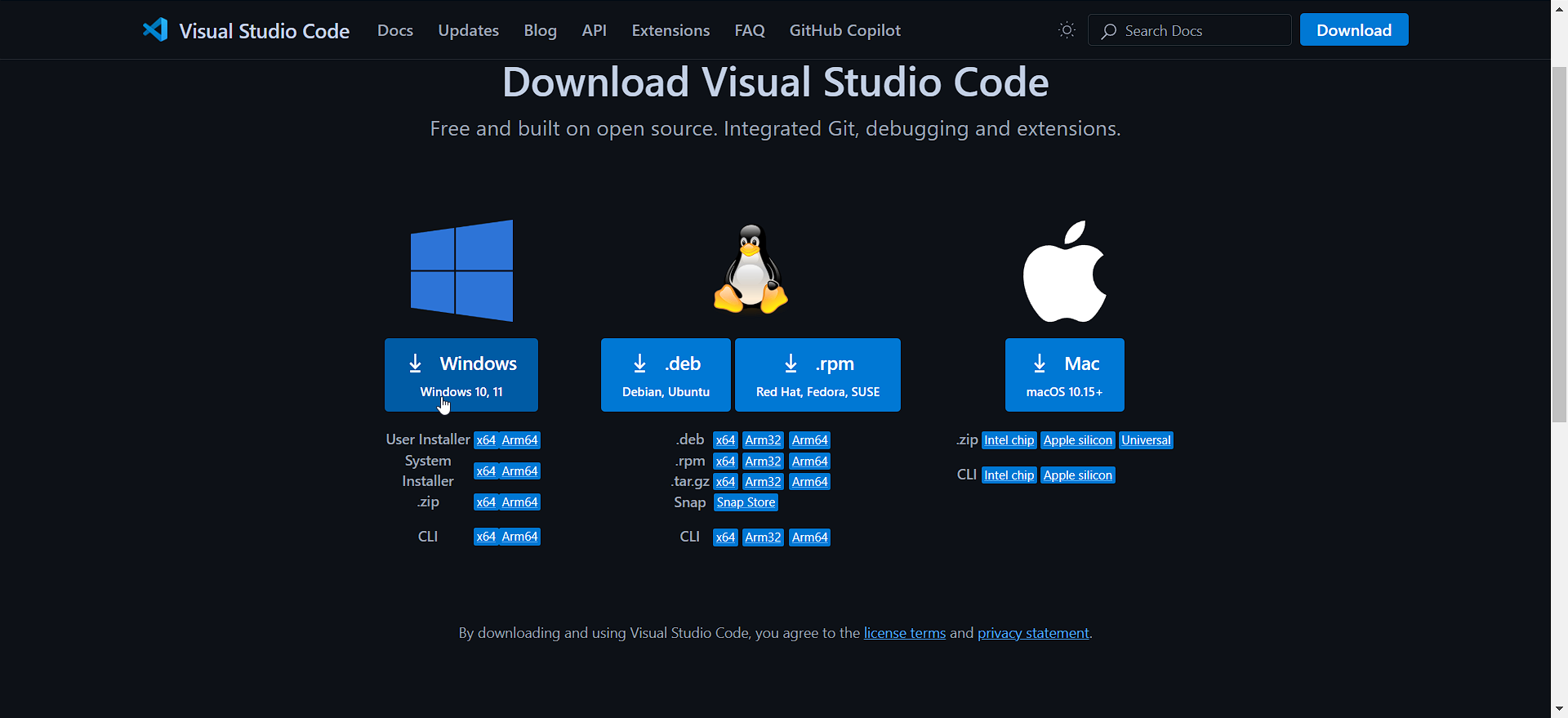


Рисунок 9 – Скриншот веб-сайта code.visualstudio.com

Запустите установочный файл и следуйте инструкциям установщика.

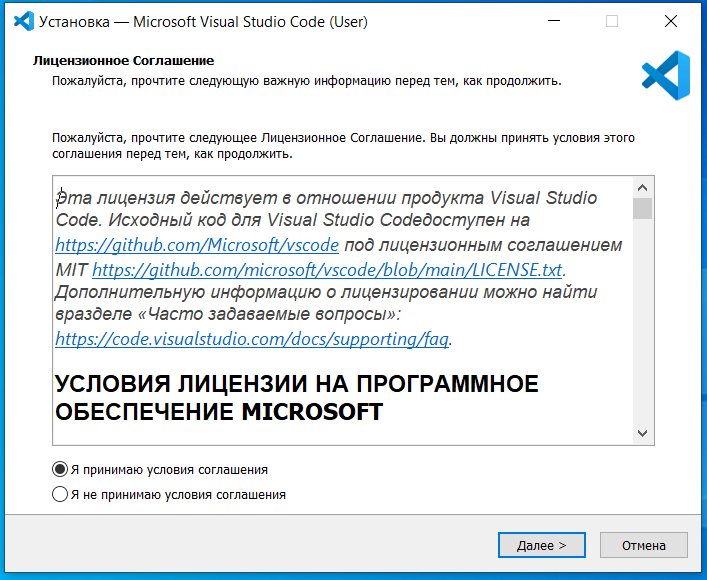


Рисунок 10 – Скриншот окна начала установки Visual Studio Code

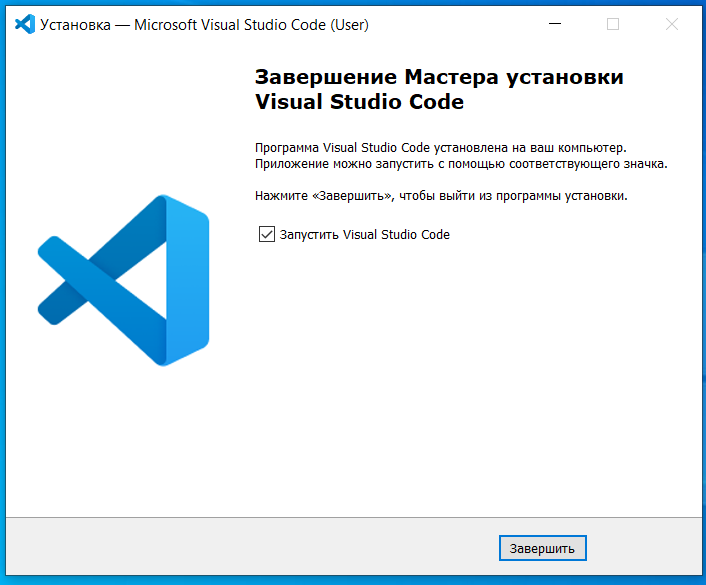


Рисунок 11 – Скриншот завершения процесса установки

**Шаг 3: Настройка VSCode для работы с Python**

Запустите VSCode.

Перейдите в раздел расширений (иконка квадратов в боковой панели) и найдите расширение "Python" от Microsoft.

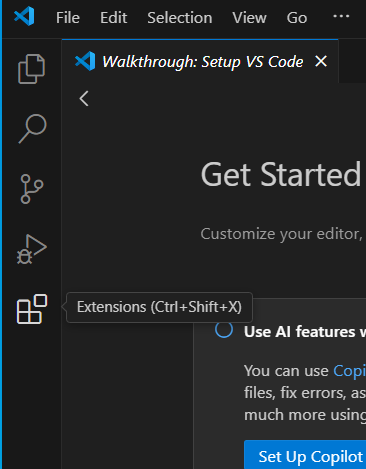


Рисунок 12 – Скриншот боковой панели Visual Studio Code

Установите расширение из списка Popular или введя название Python в поиске.



Рисунок 13 – Скриншот окна Visual Studio Code при установке расширения Python

После установки расширения перезапустите VSCode.

**Шаг 4: Установка Git и Git Bash**

Перейдите на официальный сайт Git: <https://git-scm.com/downloads/win>

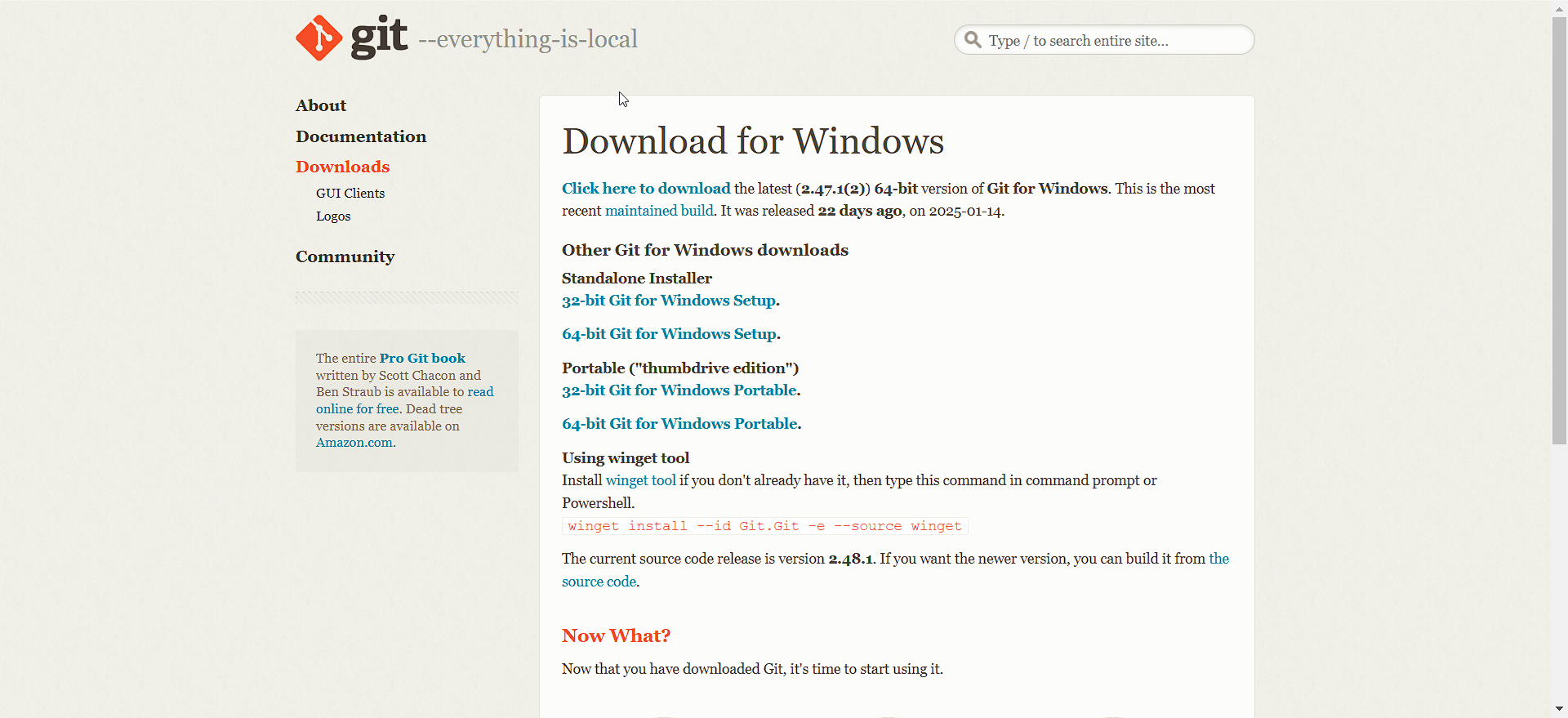


Рисунок 14 – Скриншот веб-сайта git-scm.com

Скачайте установочный файл для вашей операционной системы.

Запустите установочный файл и следуйте инструкциям установщика.

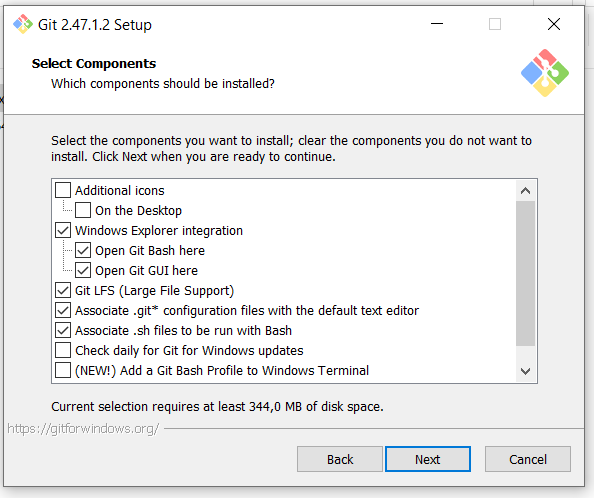


Рисунок 15 – Скриншот окна опций установщика Git

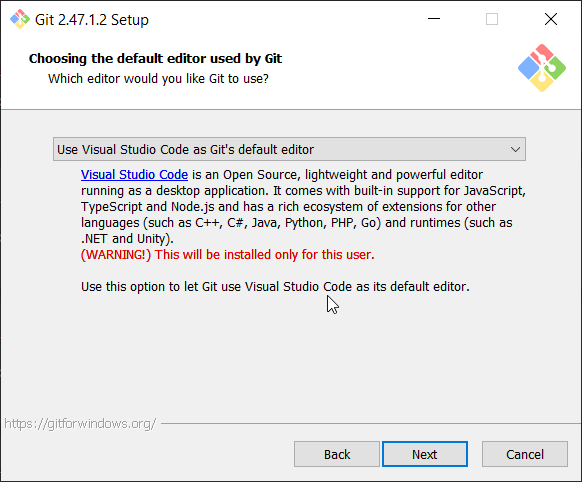


Рисунок 16 – Скриншот окна установщика при выборе стандартного редактора

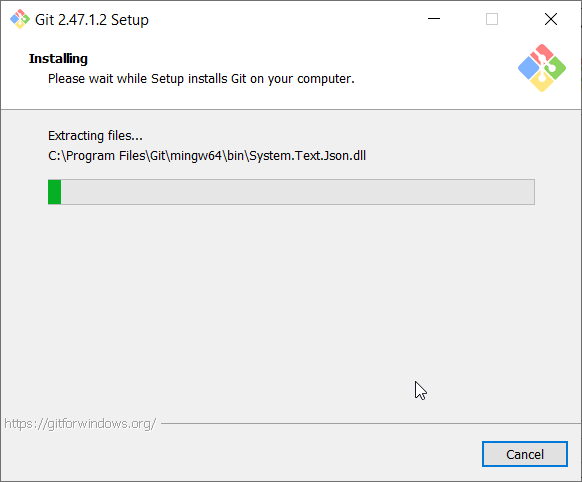


Рисунок 17 – Скриншот окна процесса установки Git

**Шаг 5. Создание GitHub репозитория**

Для получения доступа к сервису GitHub нужно в веб-браузере перейти по адресу https://github.com

Для начала необходимо пройти процесс регистрации с помощью электронной почты, имени пользователя и пароля.

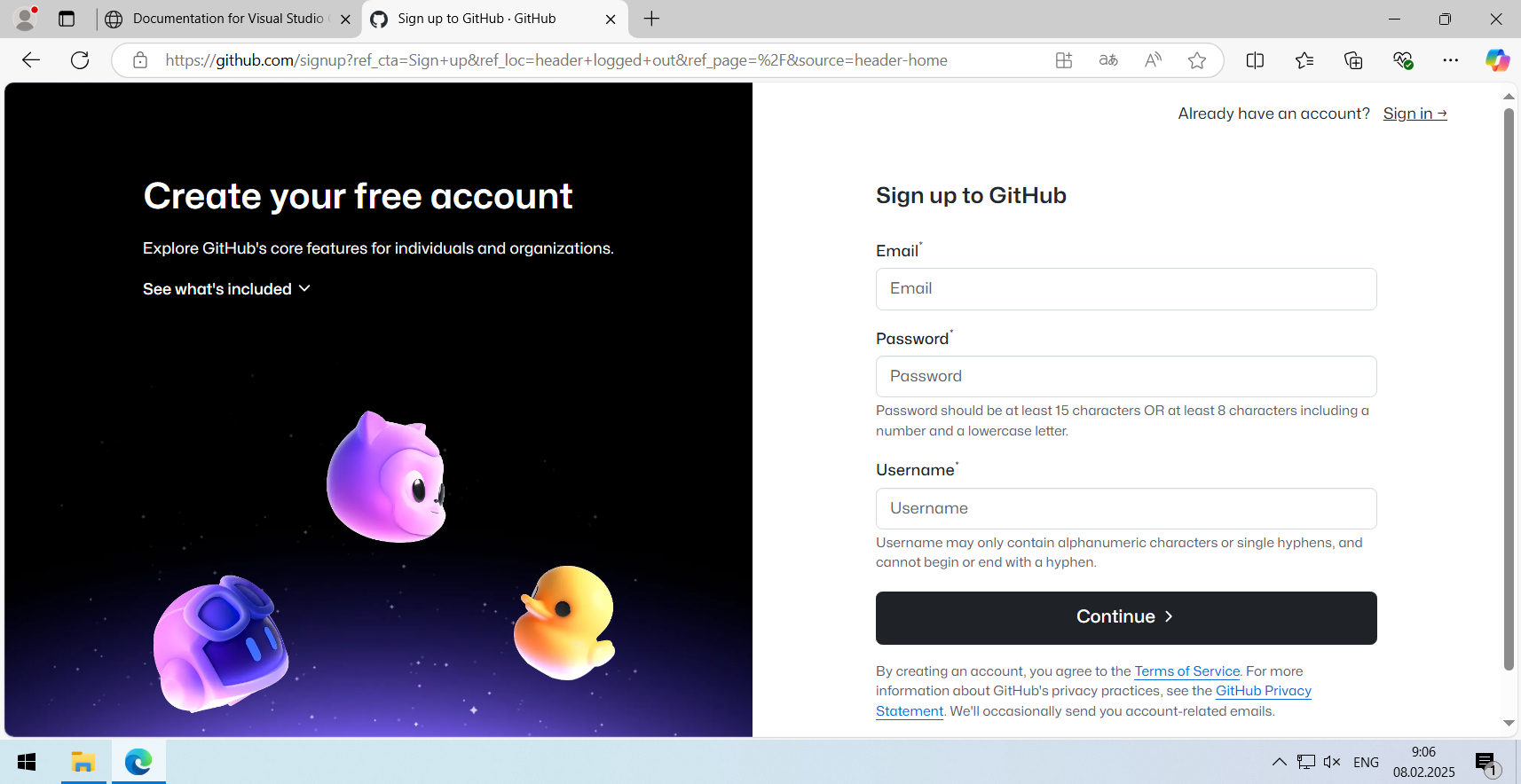


Рисунок 18 – Скриншот процесса регистрации на GitHub

После подтверждения почты и завершения регистрации произойдёт переход на главную страницу GitHub, где есть список всех репозиториев и возможность добавления новых.

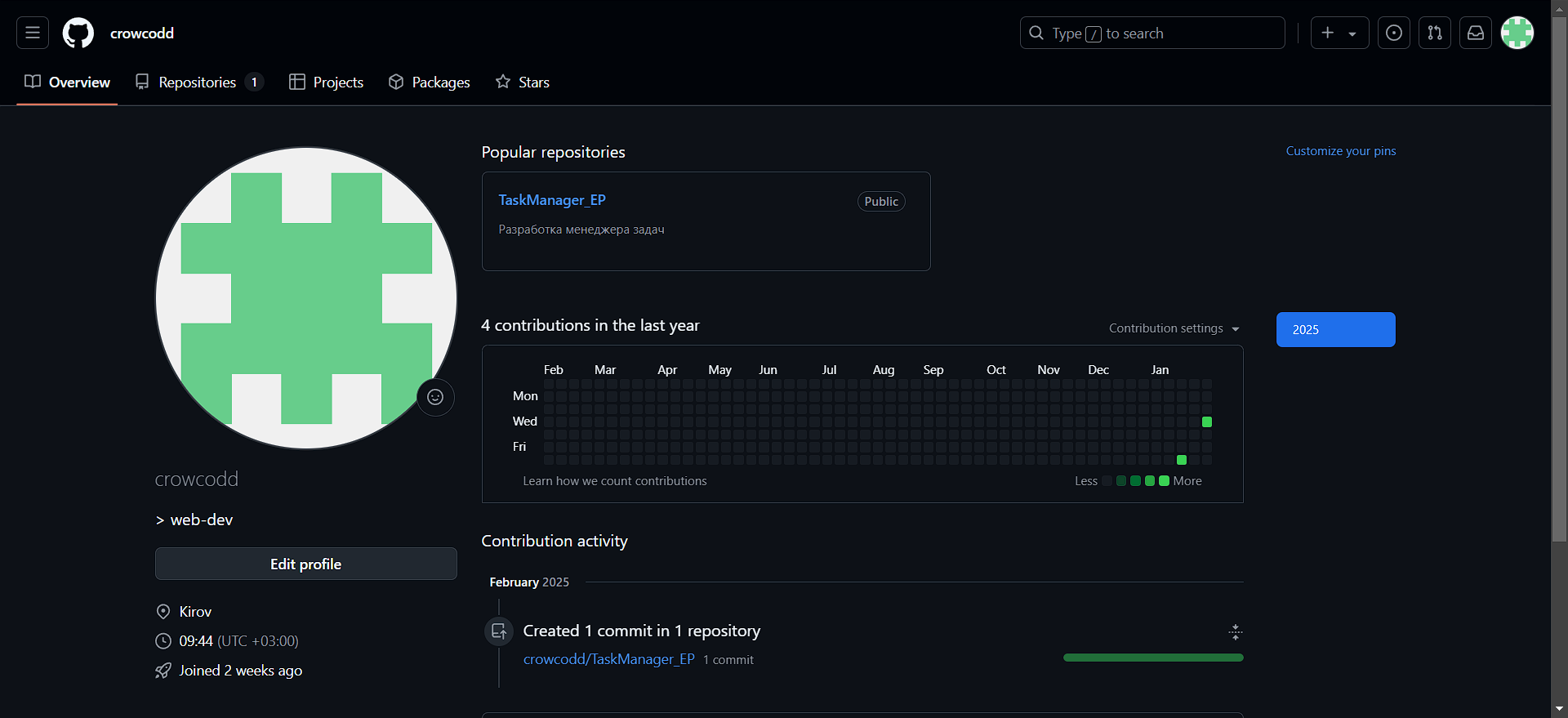


Рисунок 19 – Скриншот профиля на GitHub

Нажав зелёную кнопку New можно перейти к созданию репозитория. При создании требуется указать название на латинице, описание, а также выбрать настройки приватности репозитория и задать другие необязательные опции.

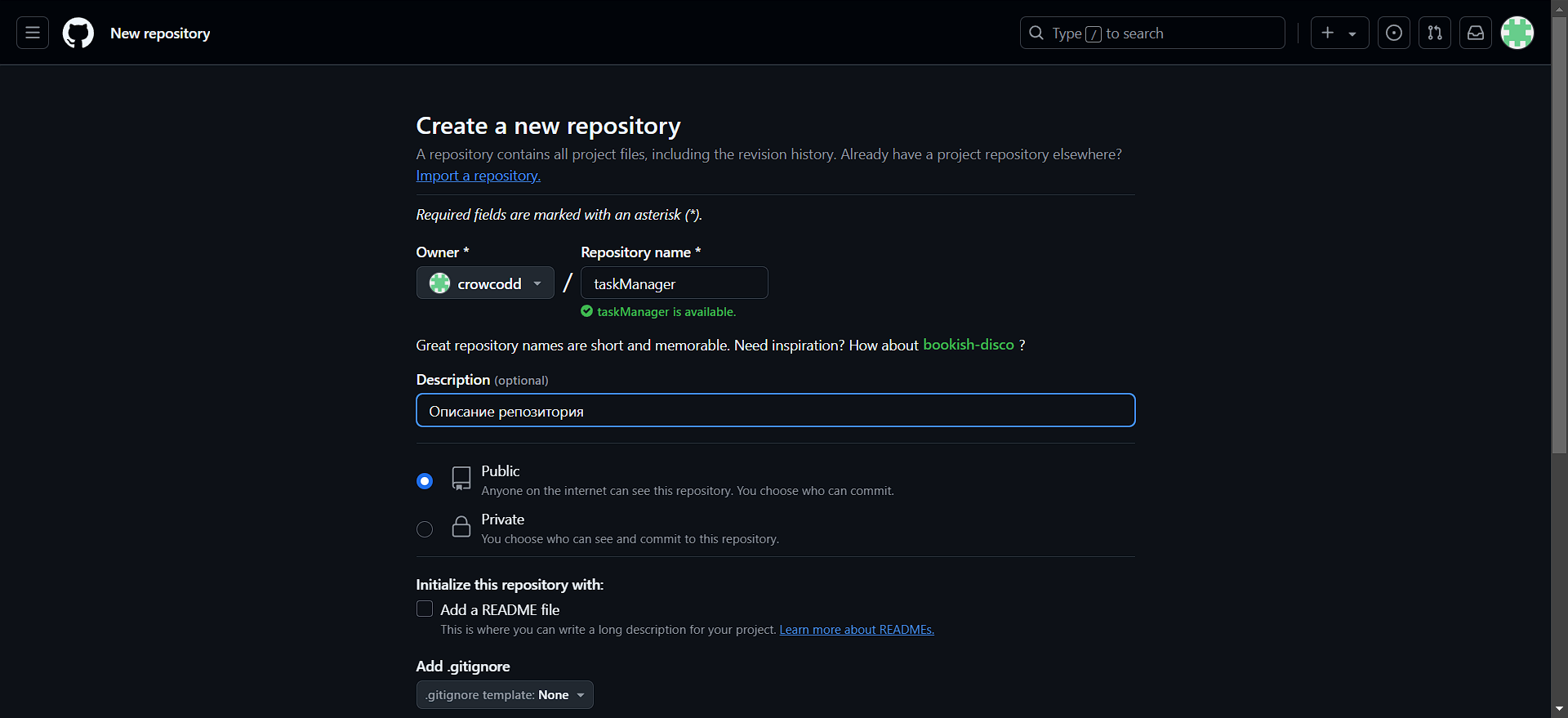


Рисунок 20 – Скриншот страницы создания репозитория GitHub

Ссылка на репозиторий: https://github.com/crowcodd/TaskManager\_EP

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Менеджмент - Википедия: сайт. URL: <https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%BD%D0%B5%D0%B4%D0%B6%D0%BC%D0%B5%D0%BD%D1%82>
2. Почему Python такой популярный. URL: <https://tproger.ru/articles/pochemu-python-takoj-populyarnyj-napisano-chelovekom>
3. Зачем нужен Git и GitBash? URL: <https://qna.habr.com/q/913527>
4. Qt Documentation. URL: <https://doc.qt.io/>