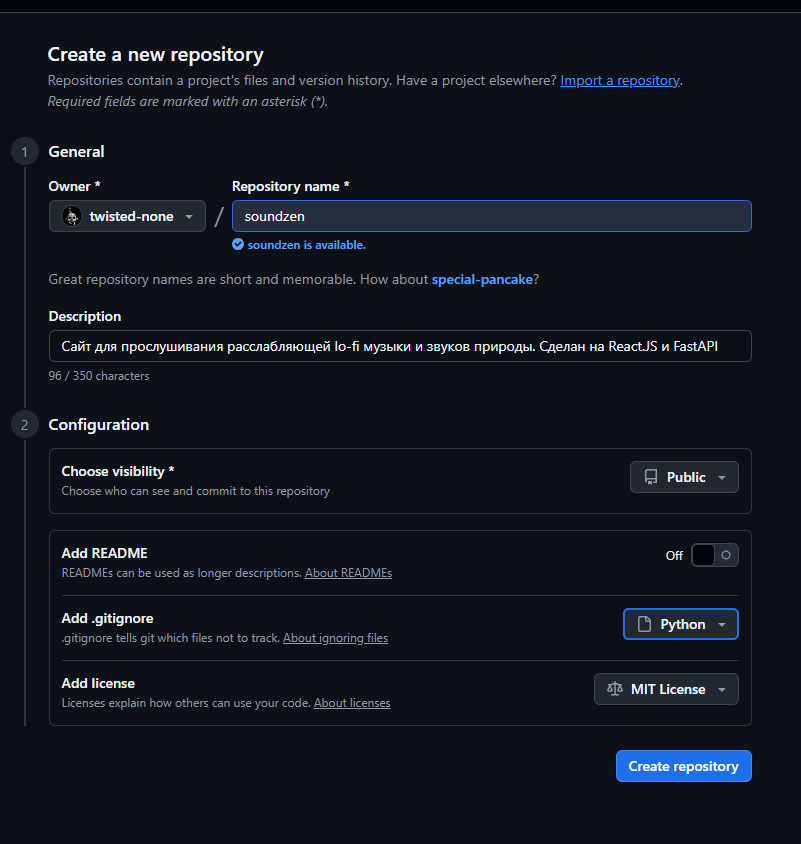
Отчет по практической работе №2: Настройка инфраструктуры проекта «SoundZen»

Ссылка на репозиторий GitHub: https://github.com/twisted-none/soundzen

1. Создание удаленного публичного репозитория на GitHub

Был создан публичный репозиторий на хостинге GitHub для проекта «SoundZen».

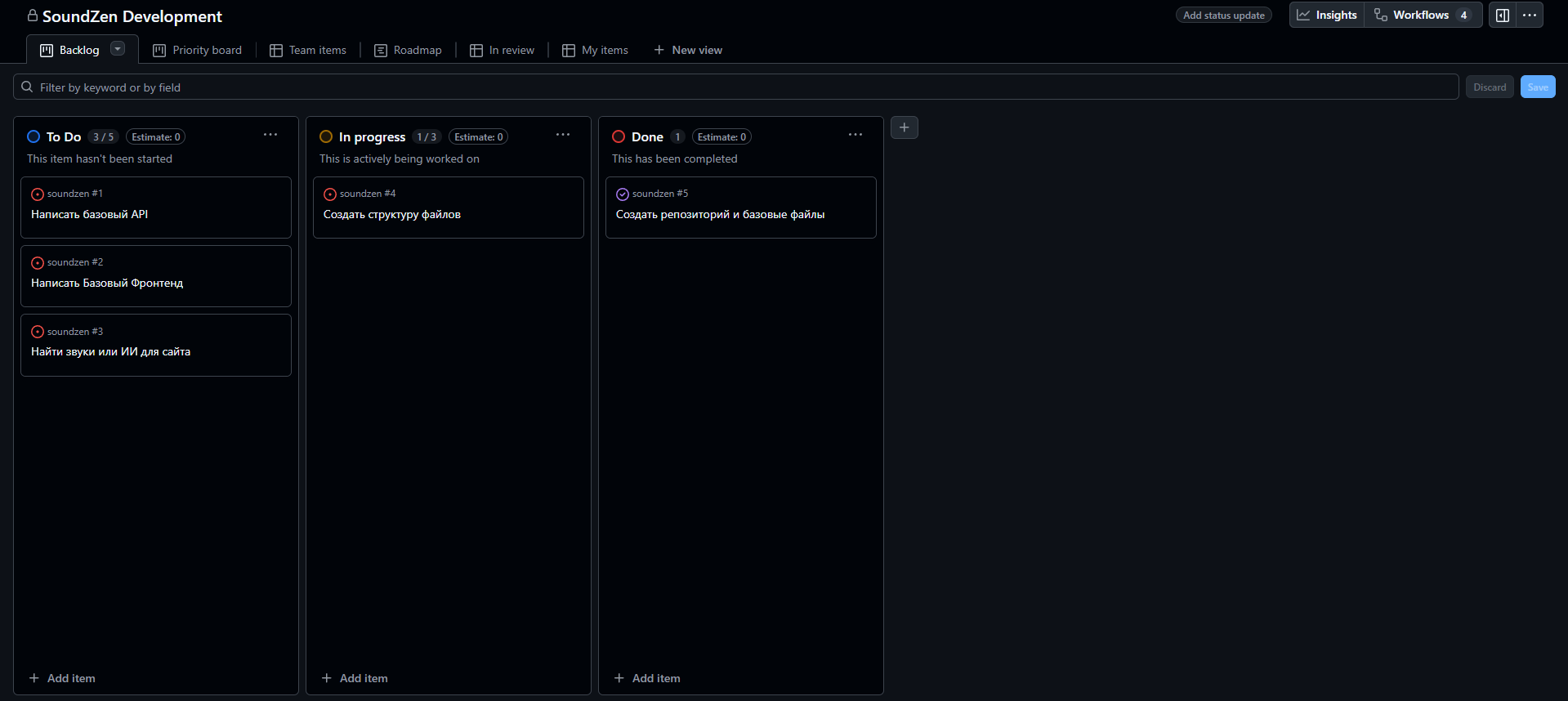
* Название репозитория: soundzen
* Описание: Сайт для прослушивания расслабляющей lo-fi музыки и звуков природы. Сделан на React.JS и FastAPI
* Настройки: Репозиторий был создан публичным (Public). Был добавлен файл .gitignore для Python и лицензия MIT для открытости проекта.



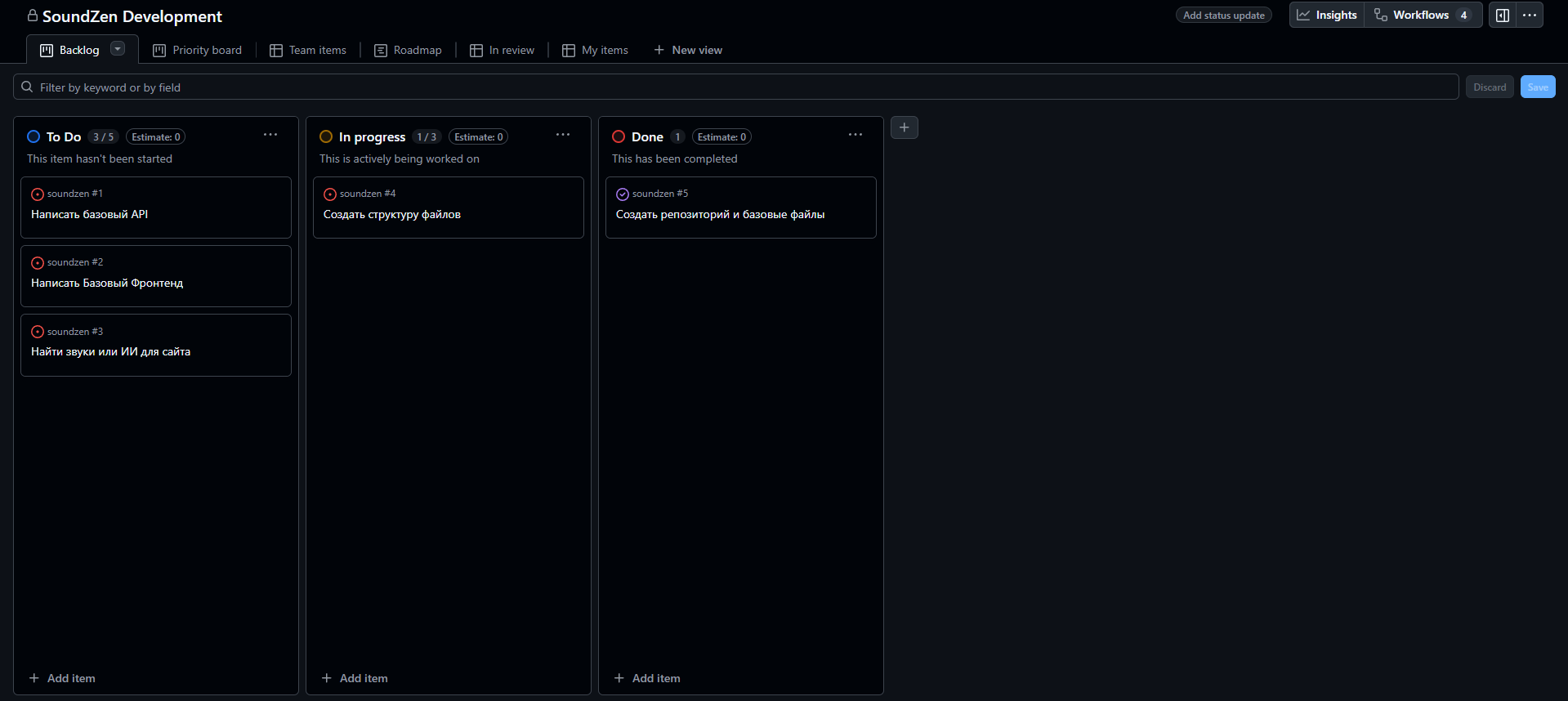
2. Создание доски задач проекта

Для управления задачами по проекту была создана доска Projects прямо на GitHub.

* Название доски: SoundZen Development
* Статус доски: Public
* Шаблон доски: Был выбран шаблон «Automated kanban», который автоматически создает колонки «Todo», «In progress» и «Done», а также автоматически перемещает задачи по ним при создании Pull Requests.



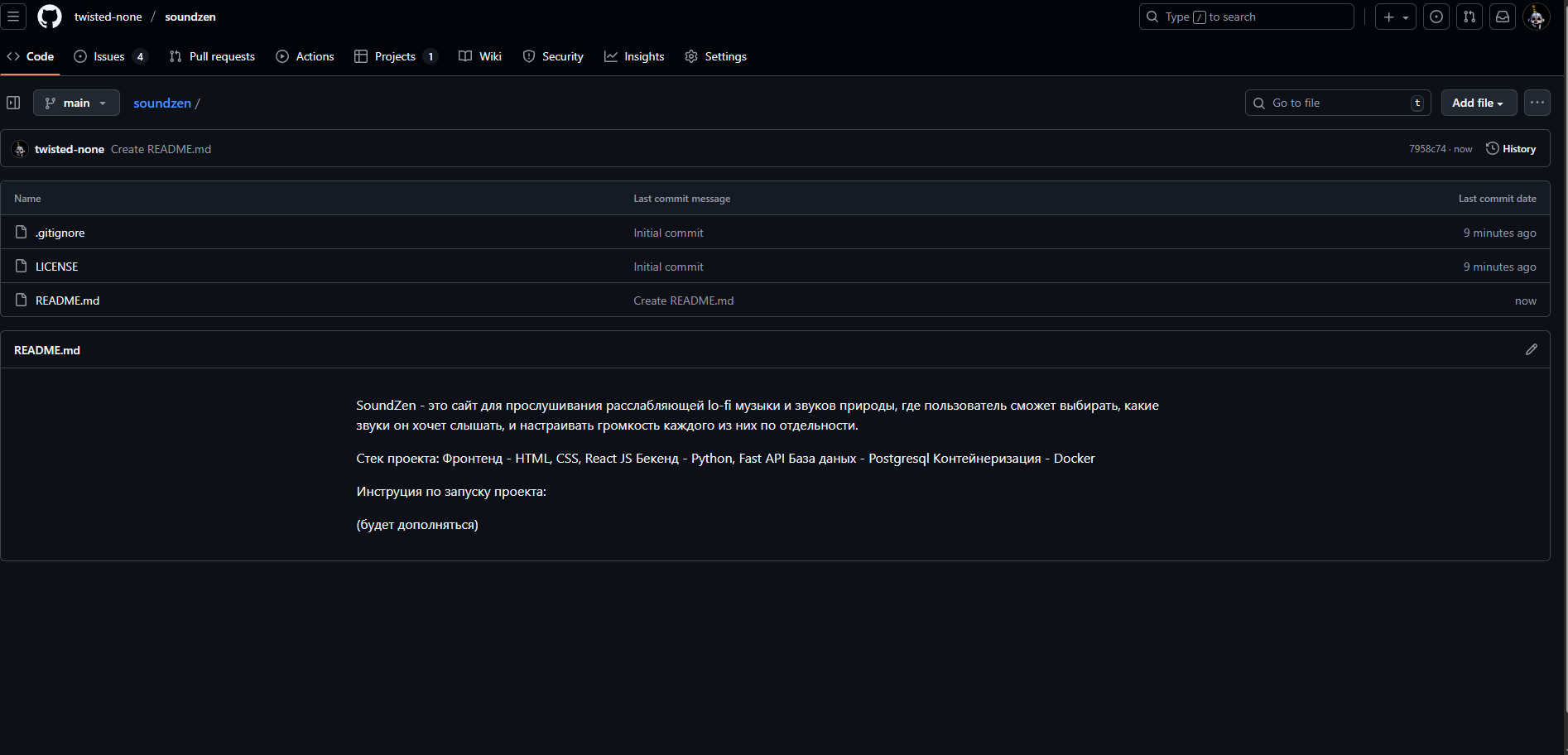
3. Создание задач для проекта

На доску проекта были добавлены первоначальные задачи (issues) для реализации базового функционала. Задачи были размещены в колонке «Todo».  


4. Создание раздела с документацией

В репозитории был создан файл README.md, который является основной точкой входа для документации. В него был перенесен и адаптирован текст из концептуального описания проекта (практическая работа №1).  
Содержание [README.md](https://readme.md/):

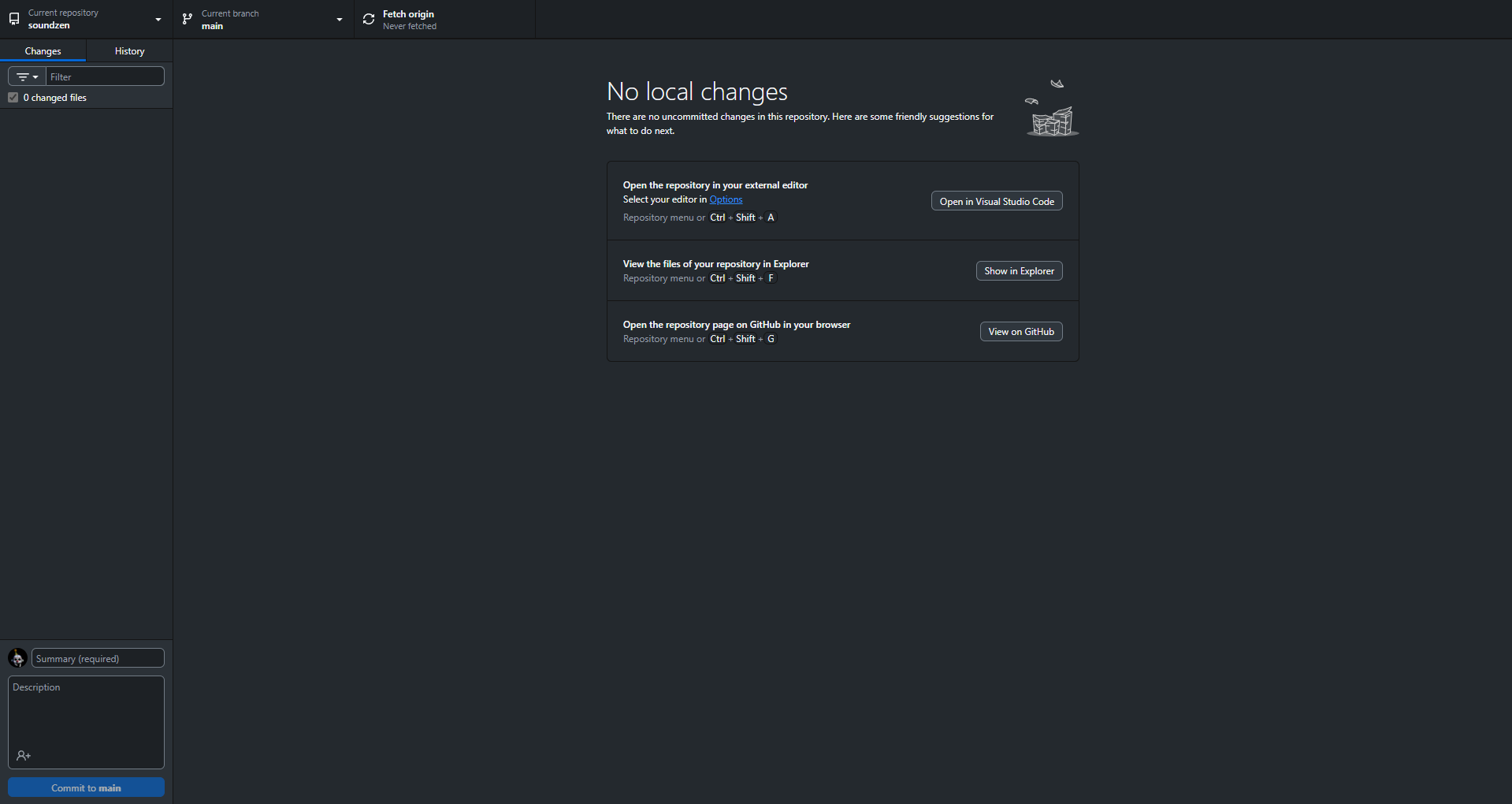
* Описание проекта (SoundZen)
* Выбранная технологическая стека (React, FastAPI, PostgreSQL)
* Инструкция по запуску проекта (будет дополняться в будущем)



5. Создание локального репозитория

Вместо использования командной строки (как в примере), для удобства было использовано приложение GitHub Desktop.

1. В GitHub Desktop был выполнен вход в аккаунт GitHub.
2. Через меню File -> Clone Repository был выбран созданный онлайн-репозиторий soundzen.
3. Была выбрана локальная папка для клонирования проекта.
4. Приложение автоматически проинициализировало локальный репозиторий Git, связало его с удаленным репозиторием (origin) и склонировало все файлы (в данном случае только README.md, .gitignore и LICENSE).

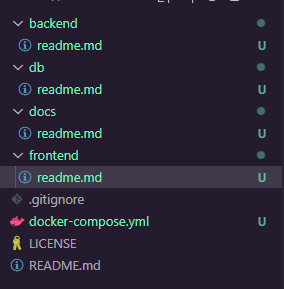


6. Связывание локального и удаленного репозиториев

Данный шаг был выполнен автоматически при клонировании репозитория через GitHub Desktop.

7. Определение структуры проекта и создание основных файлов

Через VS Code была создана первоначальная структура папок и файлов проекта в локальной директории:



8. Отправка изменений на удаленный репозиторий

Все изменения (создание новой структуры папок и файлов) были зафиксированы и отправлены на GitHub с помощью GitHub Desktop.

1. В приложении были выбраны все созданные файлы для коммита.
2. Был написан комментарий к коммиту: создания базовой структуры проекта
3. Изменения были закоммичены в локальную ветку main.
4. Коммит был отправлен на удаленный сервер с помощью кнопки Push origin.

