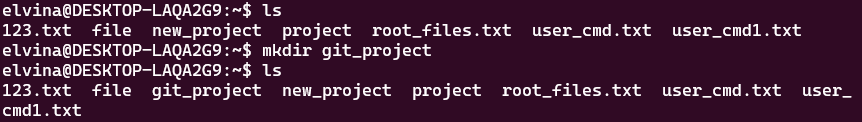
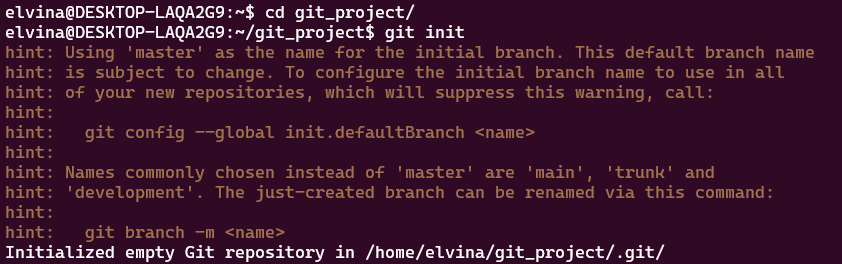
**Задание 1. Git из Linux. Коммиты**

**Выполните следующие шаги:**

1. Создайте в домашней директории новую директорию git\_project.

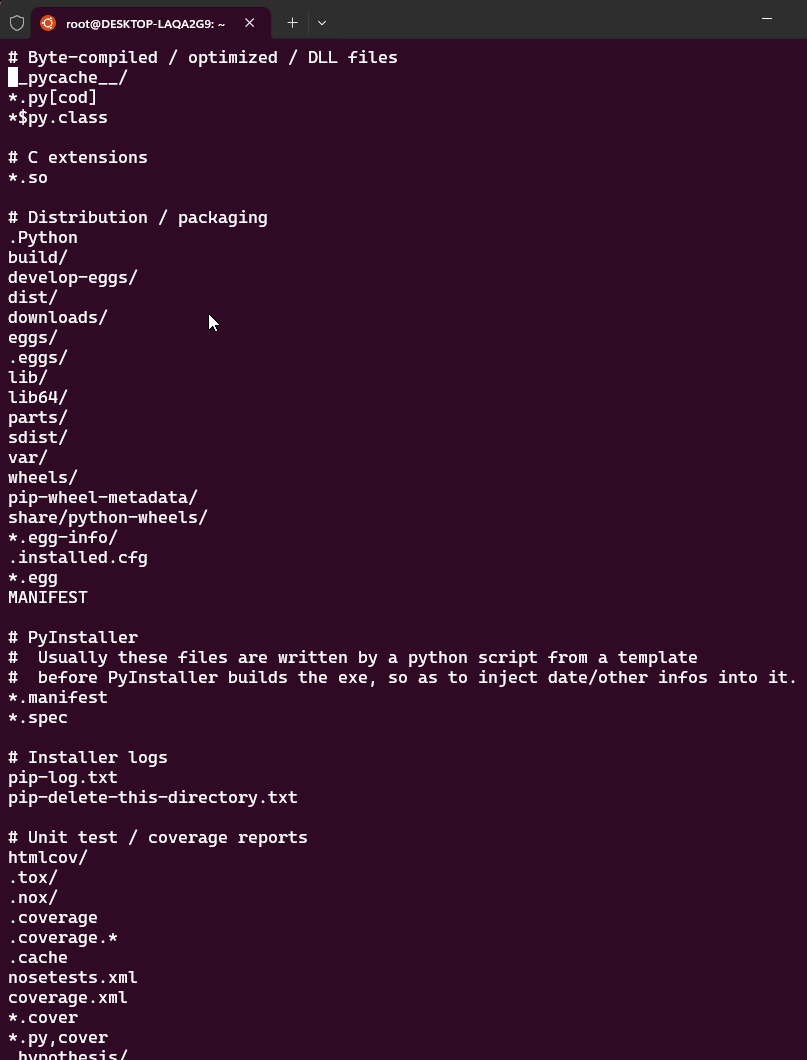


1. Через терминал инициализируйте в git\_project Git-репозиторий.

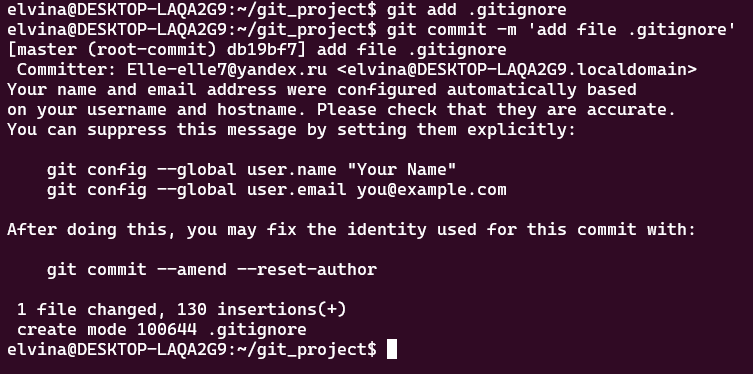


1. Скопируйте приложенный .gitignore-файл в git\_project (или создайте свой и скопируйте содержимое)[.gitignore](https://drive.google.com/file/d/19nXEy3kpUwLOC0x9h_ETNdDmOStMjFgx/view?usp=sharing)



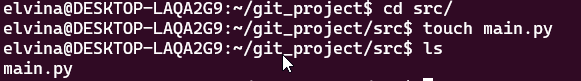


1. Сделайте коммит с добавлением .gitignore.



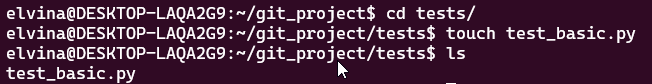
1. Создайте в git\_project директорию src/и в ней файл main.py.





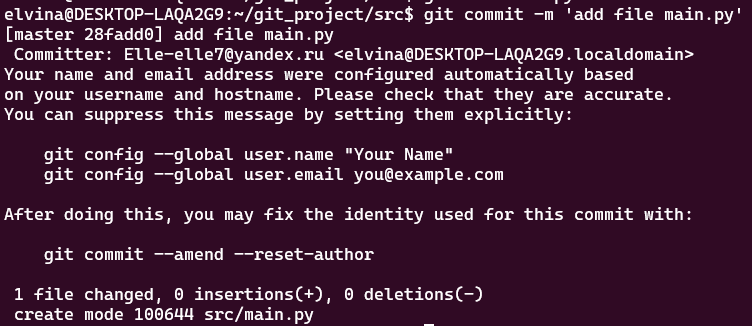
1. Создайте в проекте директорию tests/ и в ней файл test\_basic.py.





1. Сделайте два коммита: один с добавлением main.py, другой с добавлением test\_basic.py.





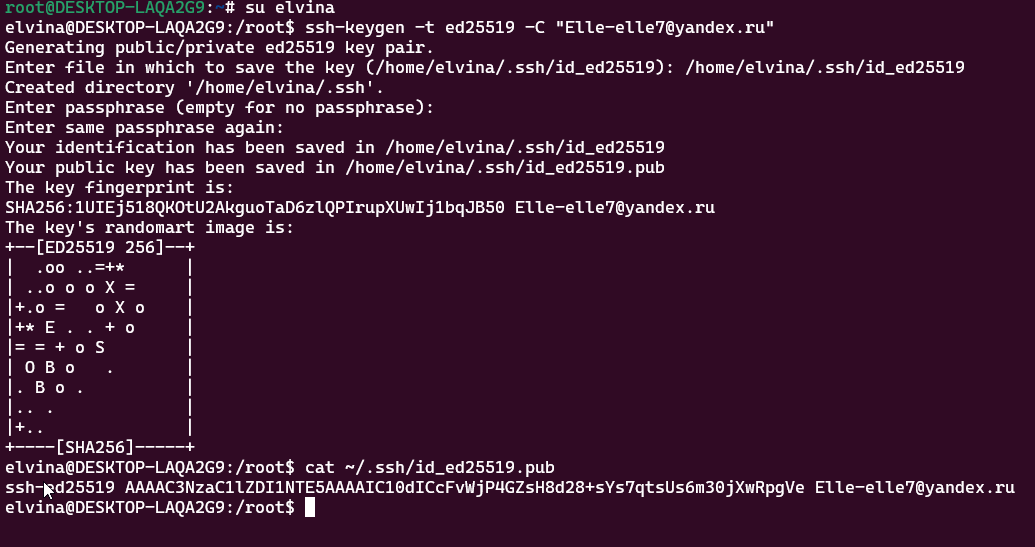


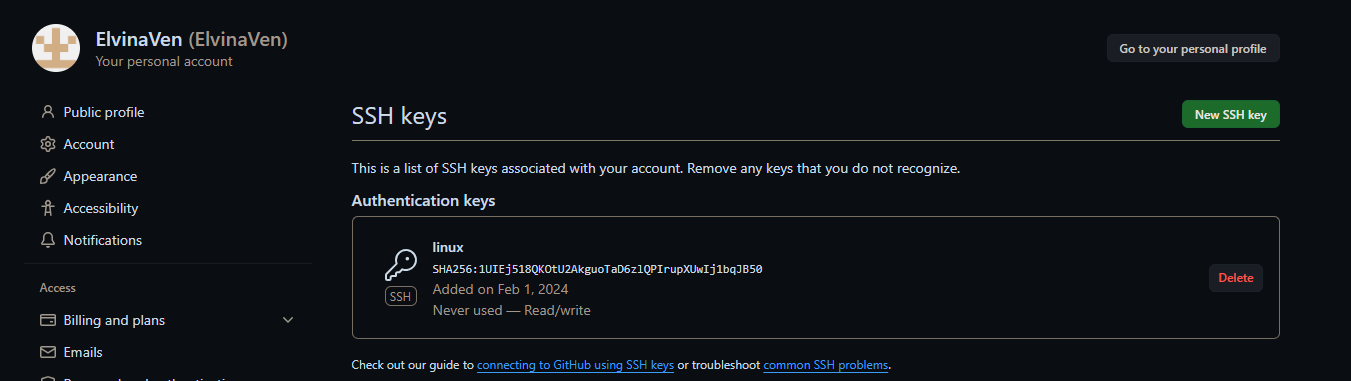


**Задание 2. Git из Linux. GitHub**

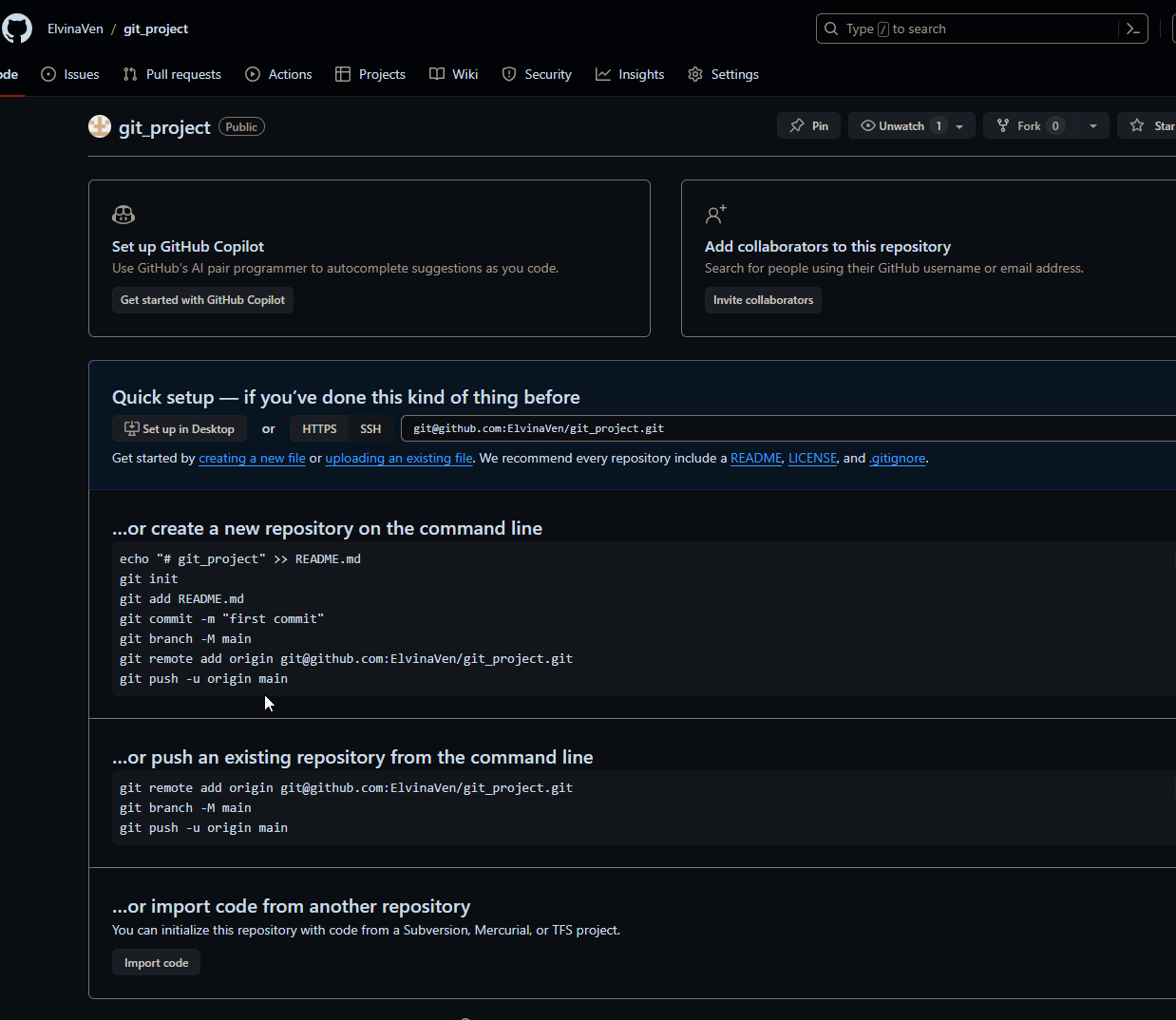
**Выполните следующие шаги:**

1. Привяжите GitHub-аккаунт по SSH-ключу.

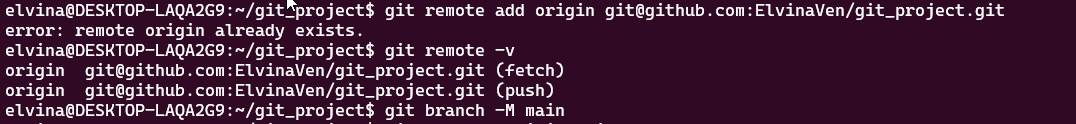




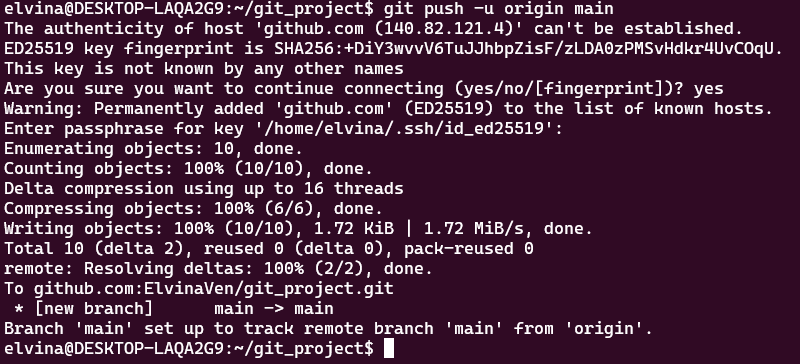
1. Создайте на GitHub пустой репозиторий git\_project.

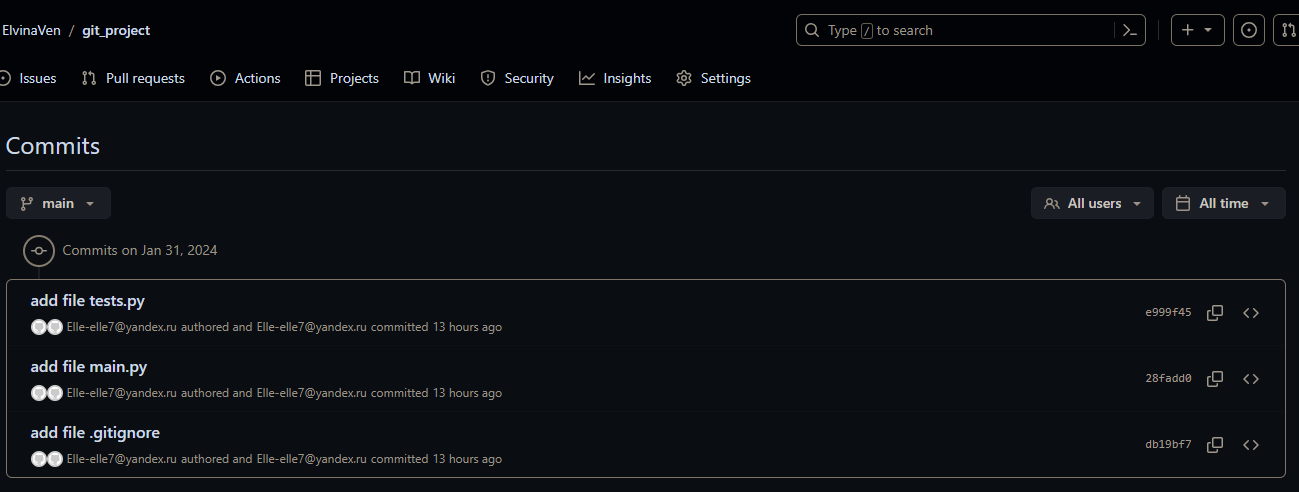


1. Выполните инструкции, которые дает GitHub для связки локального и удаленного репозитория.



1. Запушьте ваш проект на GitHub. На этот момент в нем должно быть минимум 3 коммита.

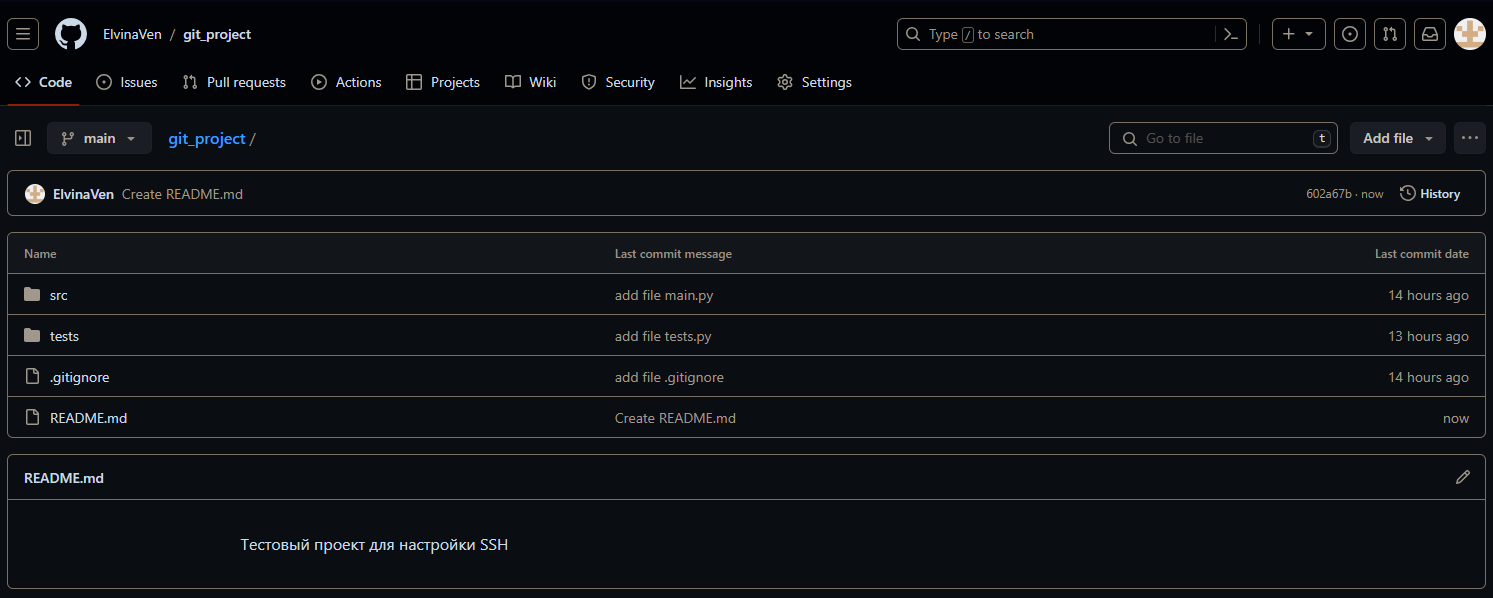




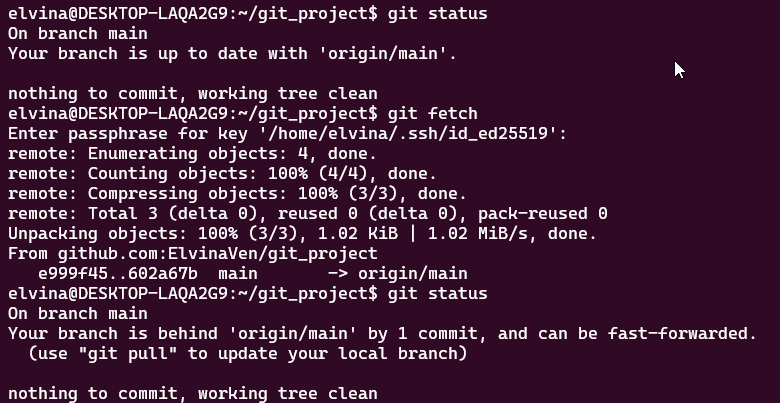
**Задание 3. Git из Linux. Обновляем локальный репозиторий**

**Выполните следующие шаги:**

1. Создайте прямо на GitHub файл README.md и добавьте в него произвольное описание проекта. Сделайте коммит прямо на GitHub.



1. Обновите локальный репозиторий. В нем должен появиться файл README.md.



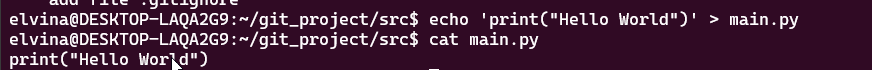




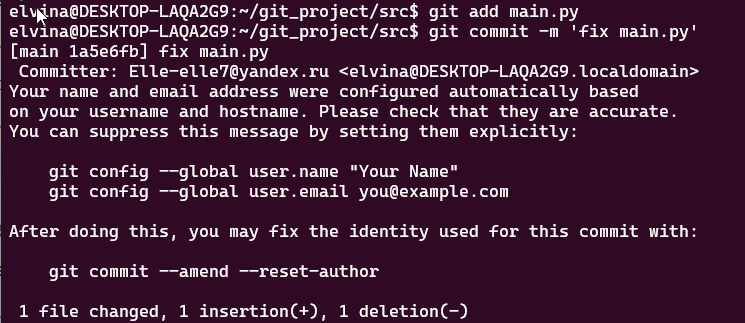
**Задание 4. Git из Linux. Отмена коммитов**

**Выполните следующие шаги:**

1. Добавьте в main.py код, который выводит в консоль Hello World.

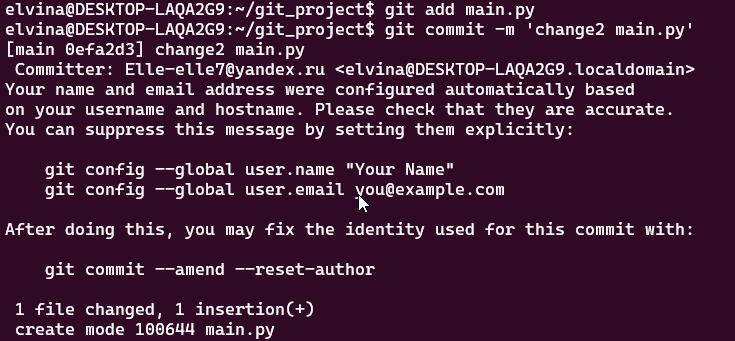


1. Сделайте коммит и push в удаленный репозиторий.

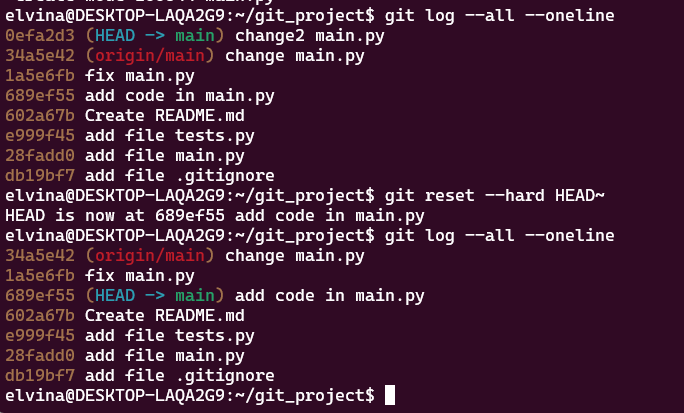


1. Измените Hello World на Hello Linux, сделайте коммит.

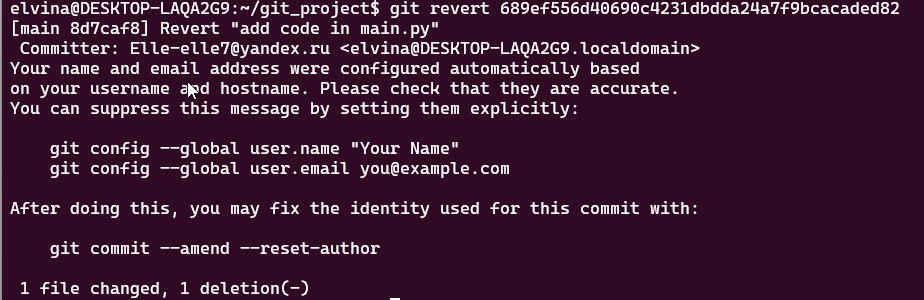


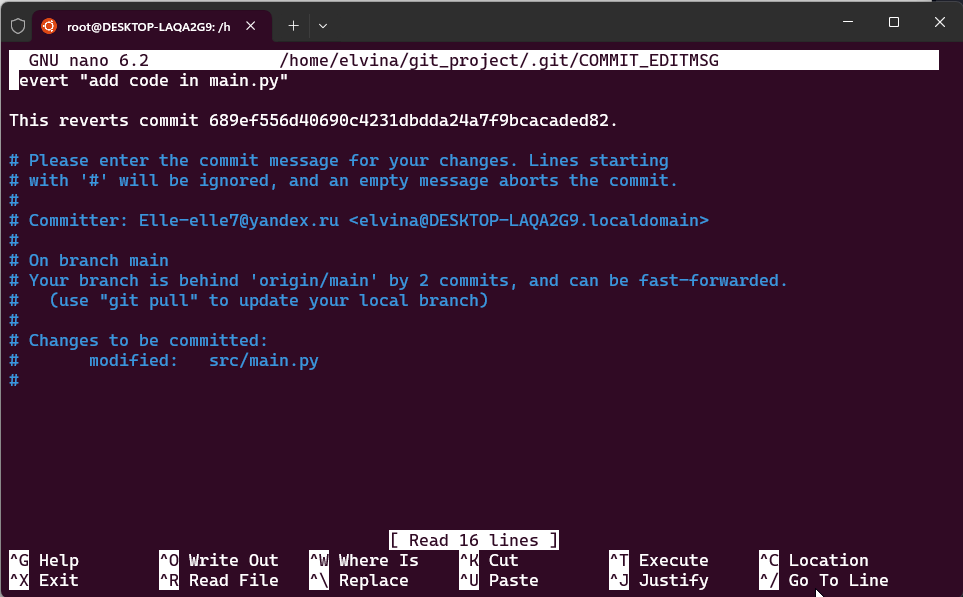


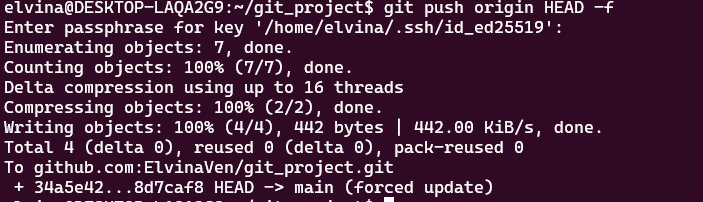
1. Отмените последний локальный коммит через git reset.

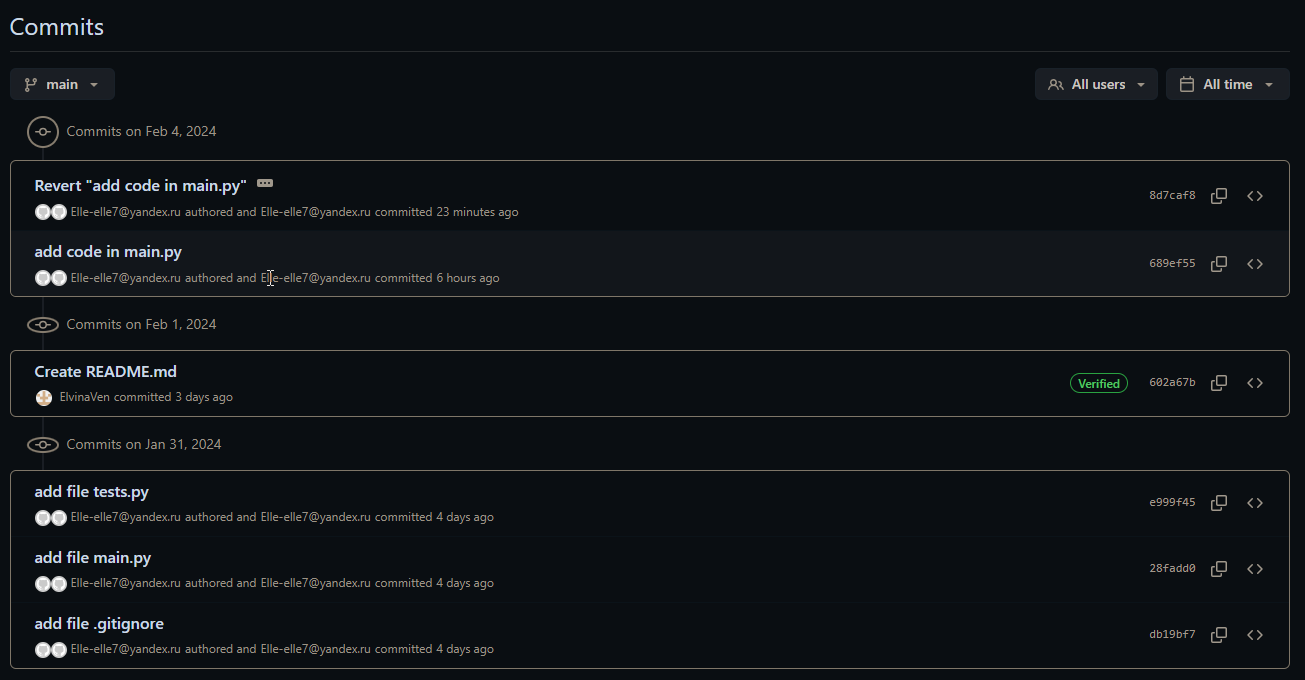


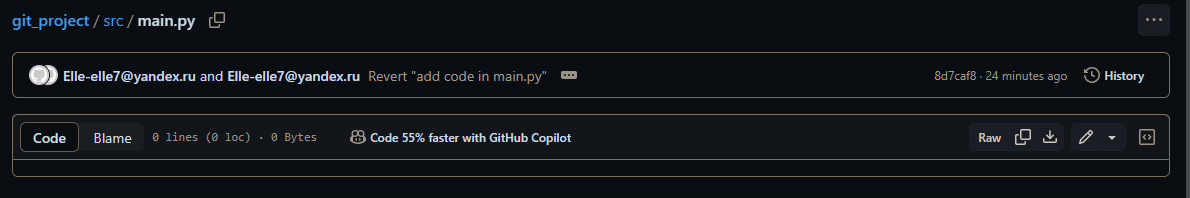
1. Отмените коммит с добавлением Hello World через git revert. Запушьте изменения.







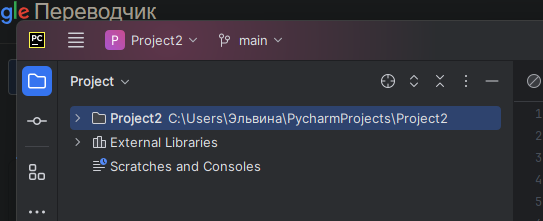


1. На GitHub в итоге должен быть пустой main.py. 

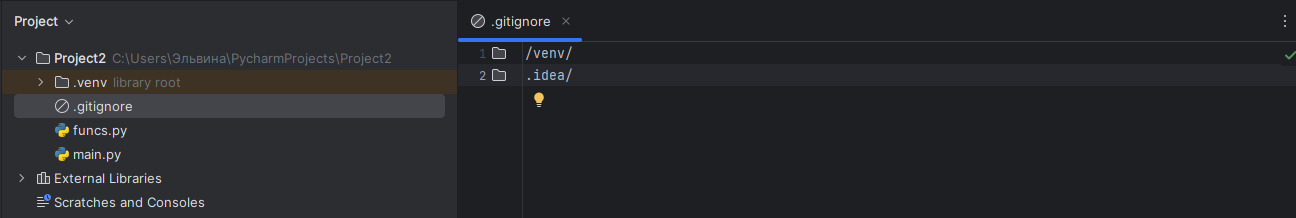
**Задание 5. Git из PyCharm. Коммиты и их история**

**Выполните следующие шаги:**

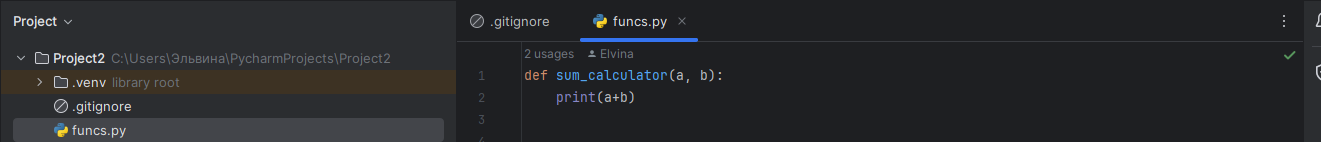
1. Создайте в PyCharm новый проект в директории py\_project.



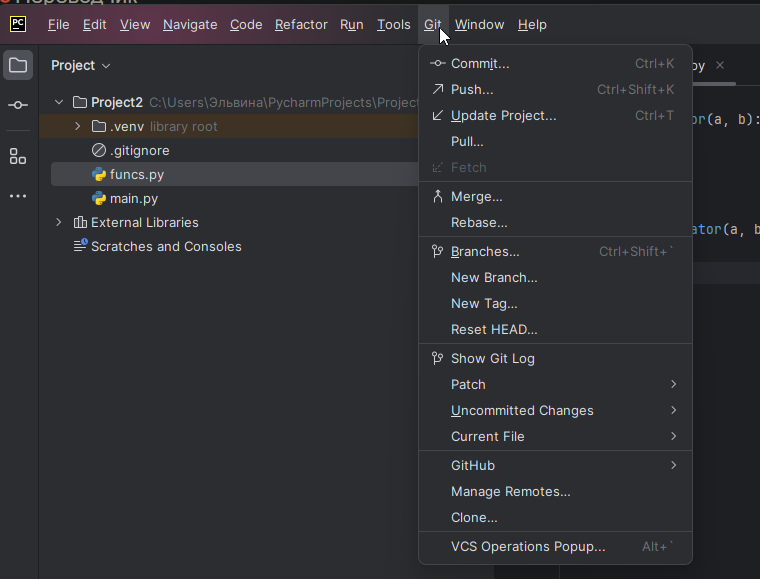
1. Создайте в проекте файл .gitignore и пропишите в нем игнорирование:/venv/.idea/



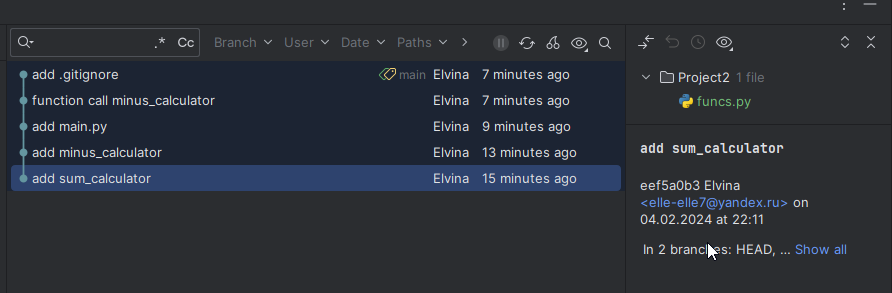
1. Создайте файл funcs.py и напишите в нем функцию, которая складывает два числа.



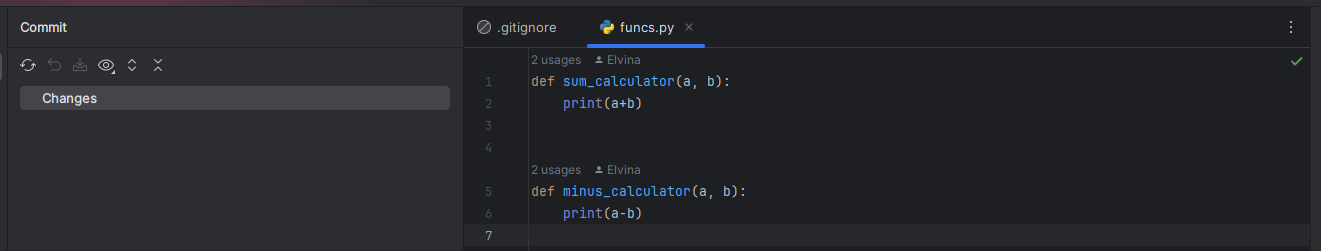
1. Подключите из PyCharm систему контроля версия Git.



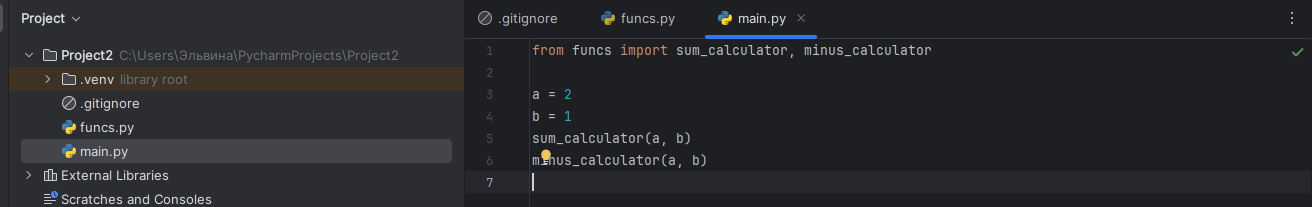
1. Сделайте первый коммит изменений в funcs.py.



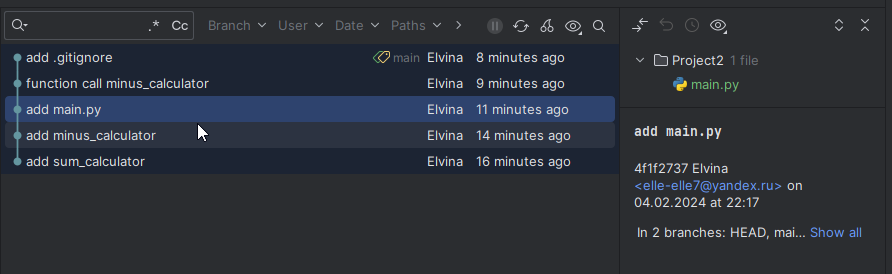
1. Добавьте в funcs.py вторую функцию, реализующую вычитание, и сделайте коммит.



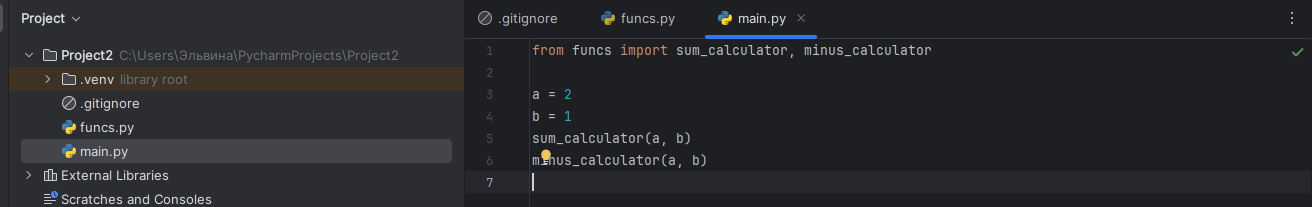
1. Создайте файл main.py и сразу добавьте его в отслеживаемый.
2. Импортируйте в main.py функцию сложения из funcs.py, вызовите ее.



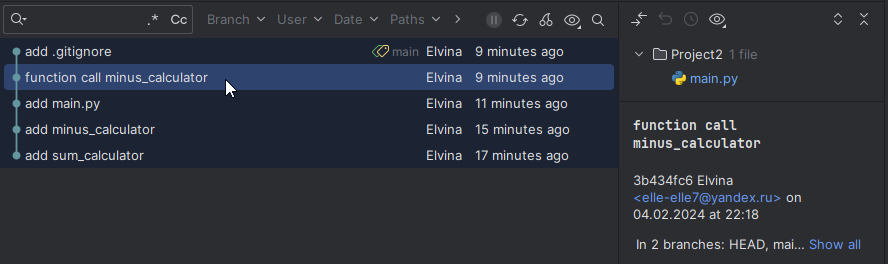
1. Сделайте коммит.



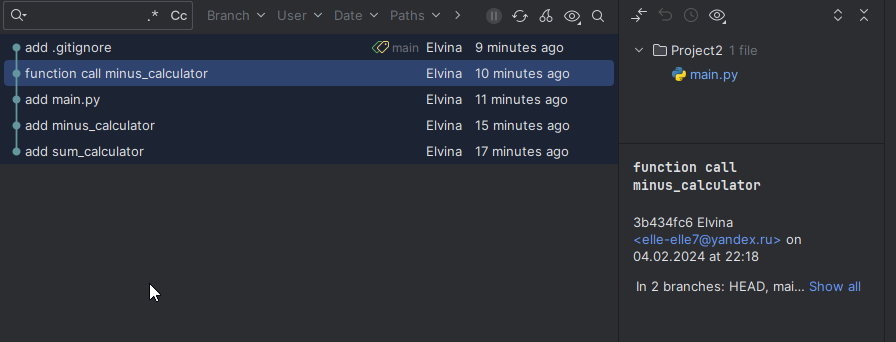
1. Импортируйте в main.py функцию вычитания из funcs.py и вызовите ее.



1. Сделайте коммит.

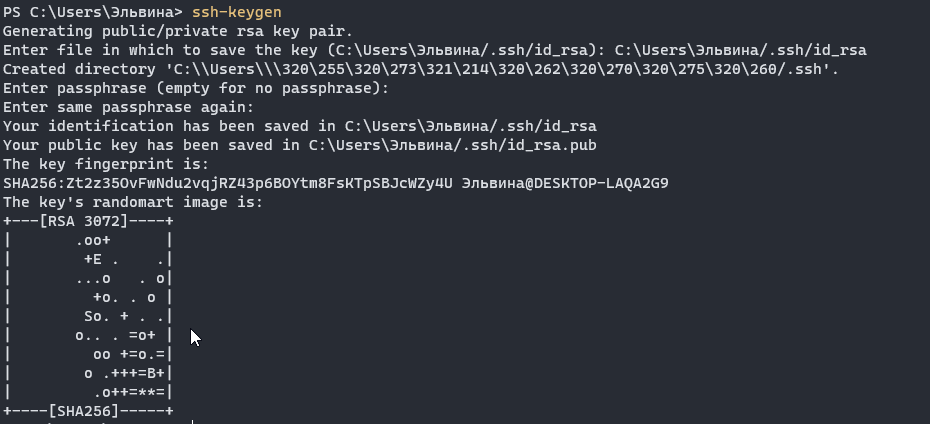


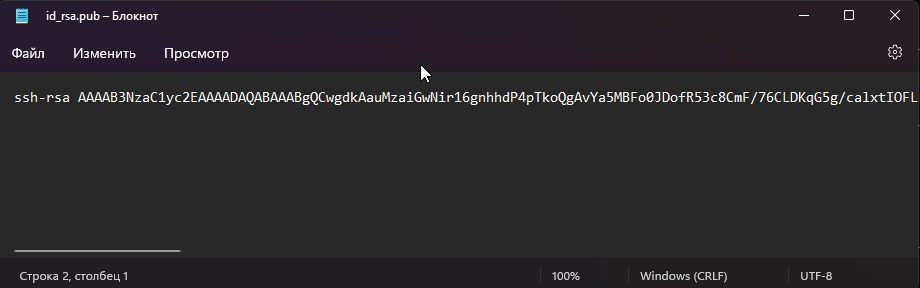
1. Проанализируйте историю изменений на вкладке Git.

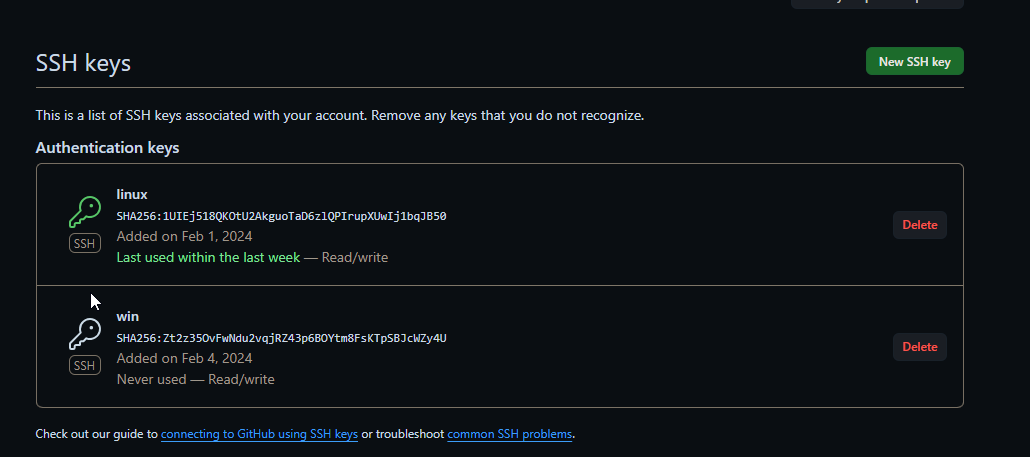


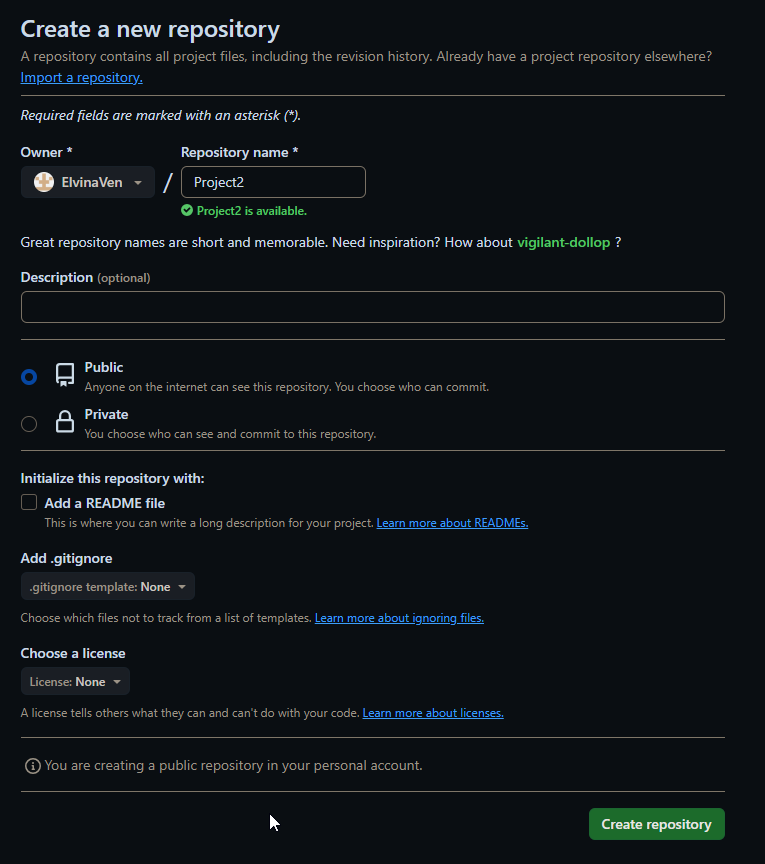
## Задание 6. Git из PyCharm. GitHub

1. Выполните следующие шаги:
2. Добавьте в GitHub SSH-ключ (если еще нет) и создайте там репозиторий.

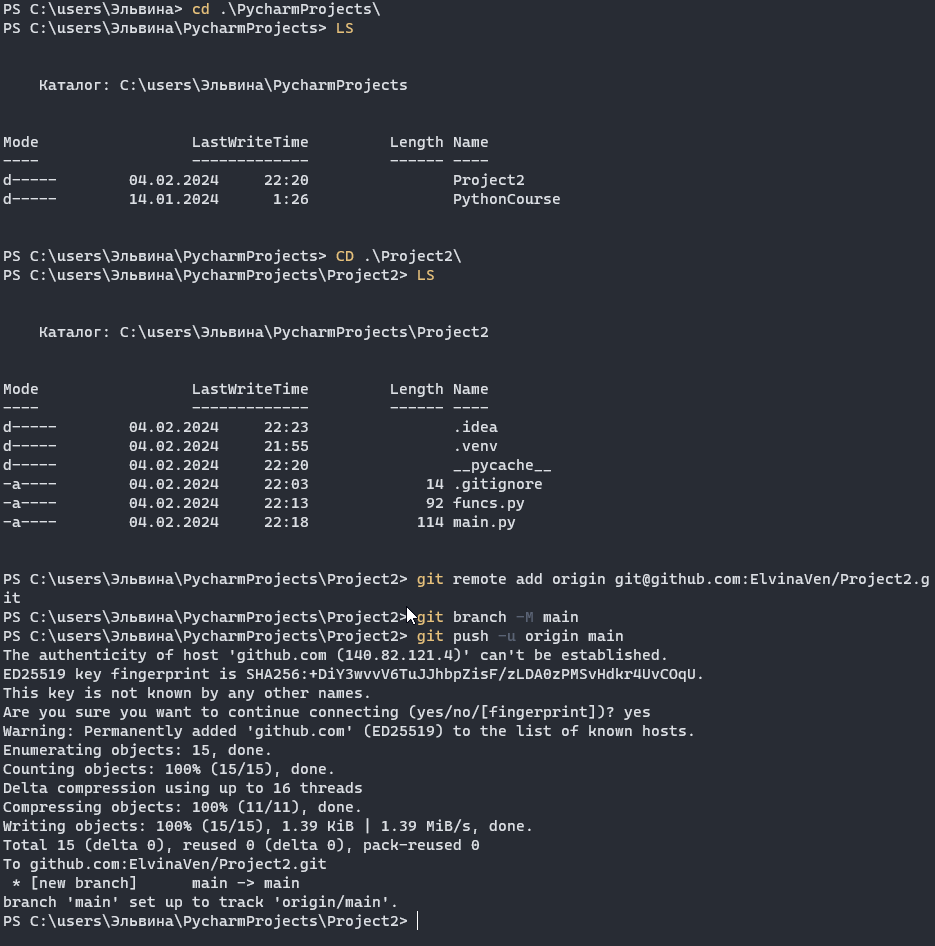


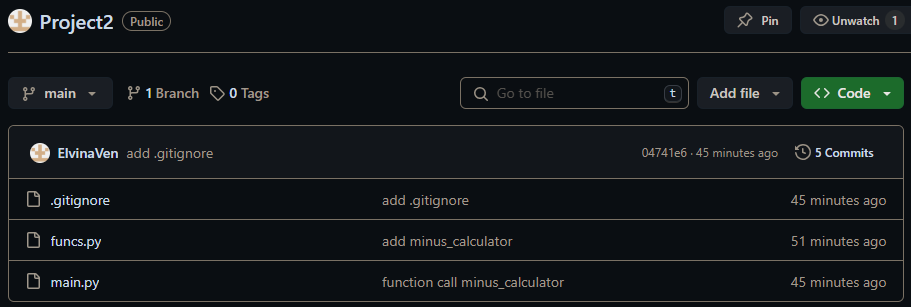


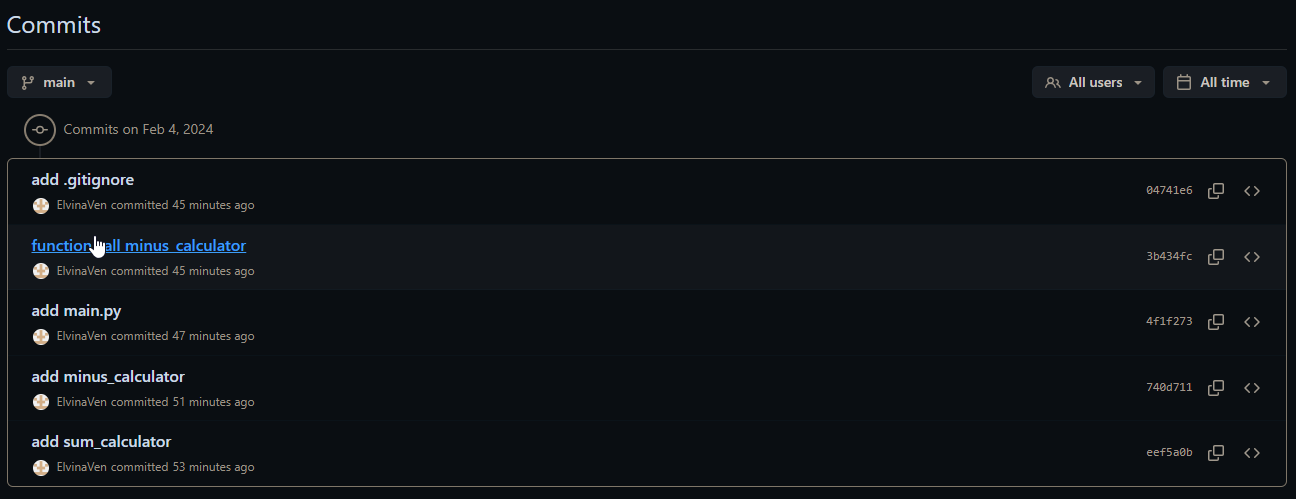




1. Привяжите удаленный репозиторий через PyCharm и запушьте проект.



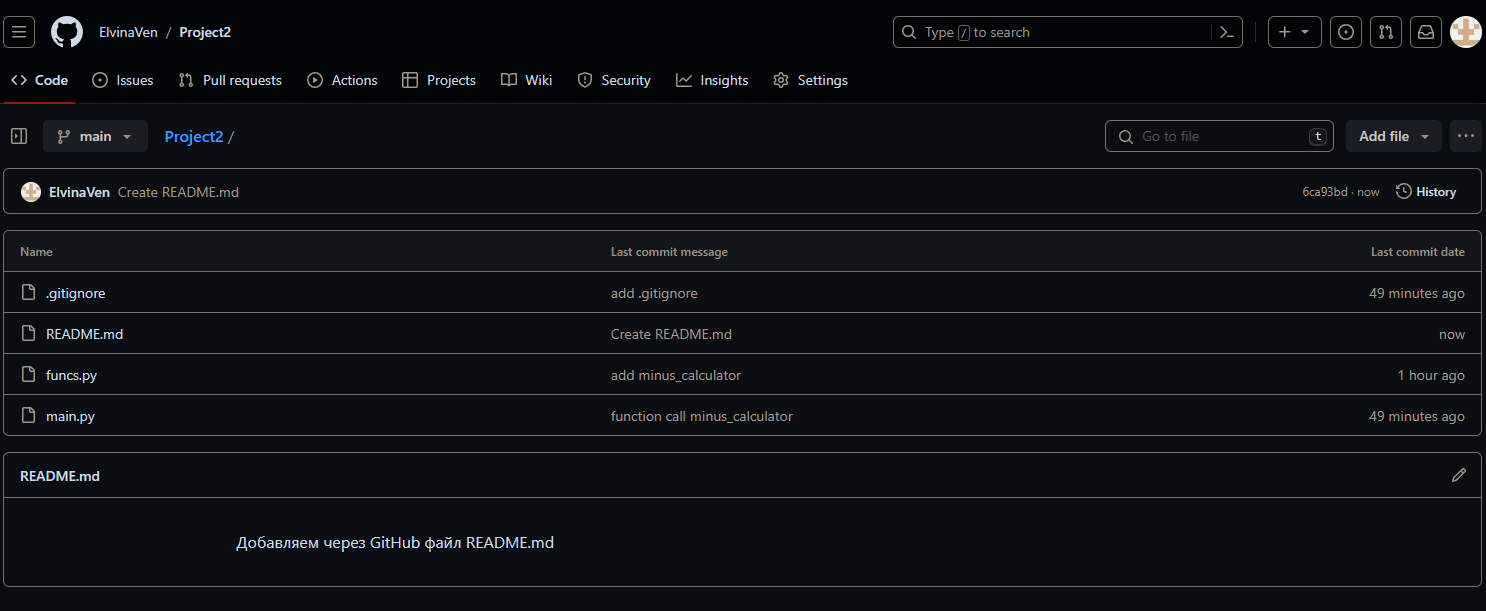




**Задание 7. Git из PyCharm. Обновляем локальный репозиторий**

**Выполните следующие шаги:**

1. Добавьте через GitHub файл README.md.



1. Через PyCharm сделайте обновление проекта и получите README.md.

