**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования «Заполярный государственный университет имени Н. М. Федоровского»**

**Отчет**

по дисциплине:

«Информационные технологии»

Лабораторная работа №2

Выполнила: студентка 3 курса

очной формы обучения

Группы: ИС-23

Якуш Елизавета Анатольевна

Проверил: Сидельников М. Э.

Лабораторная работа №2 - Работа с Git на GitHub

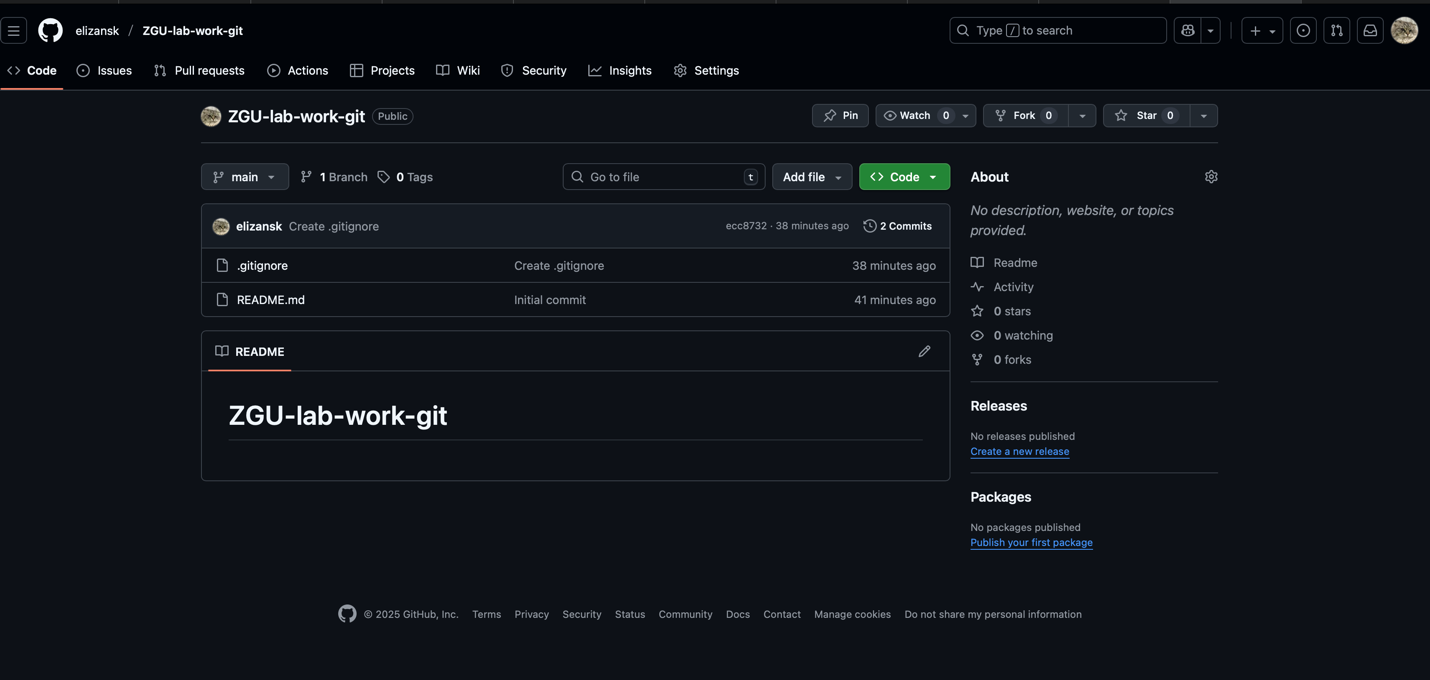
**Цель лабораторной работы**

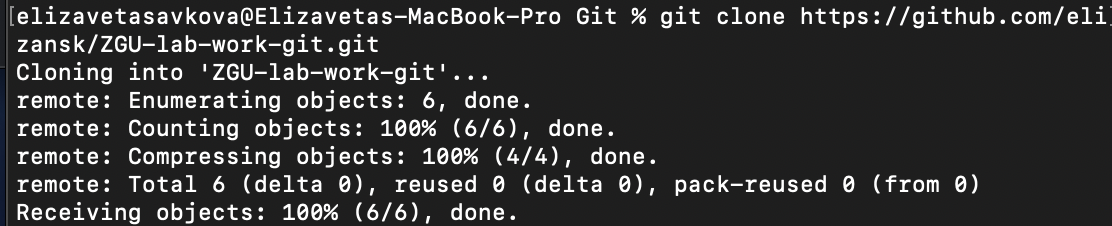
* Изучить основные команды и функции Git.
* Освоить практические навыки работы с GitHub.
* Научиться работать с удалёнными репозиториями, ветками и основными операциями Git.

**Этапы выполнения работы**

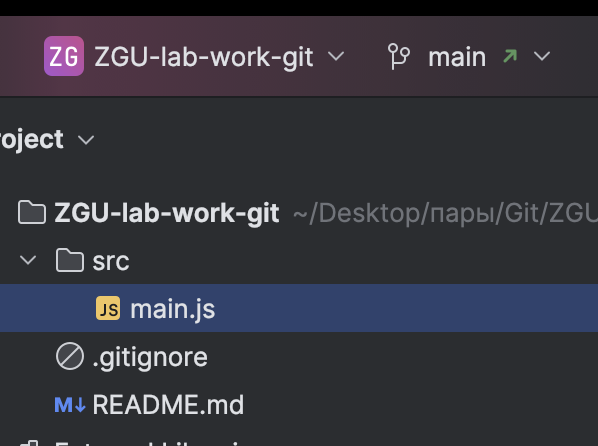
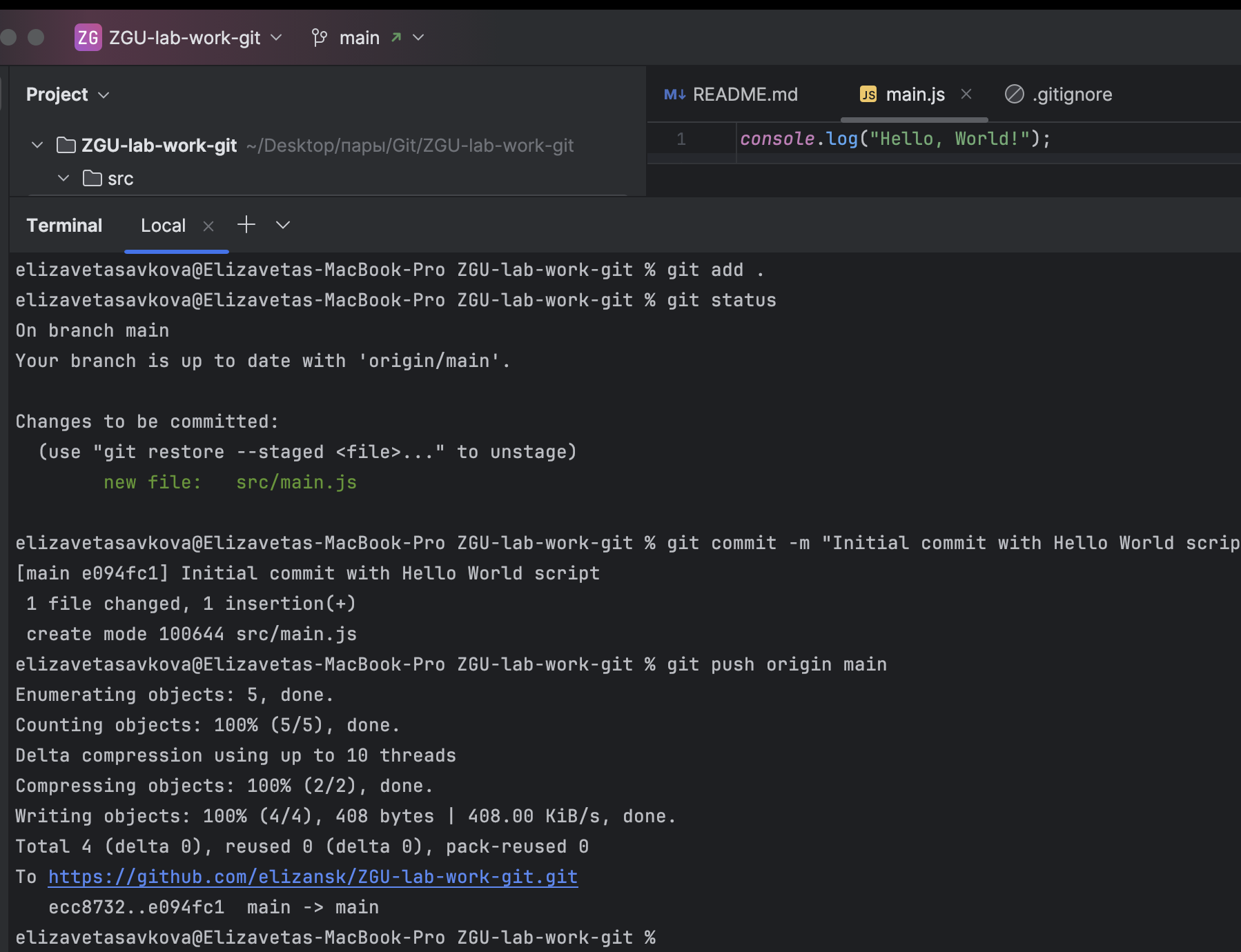
Шаг 1: Создание репозитория на GitHub

1. **Зарегистрировалась на GitHub** .
2. **Создала новый репозиторий**:
   * ZGU-lab-work-git.
   * Добавила README.md файл для описания проекта.
   * Добавила .gitignore файл для языка, JavaScript

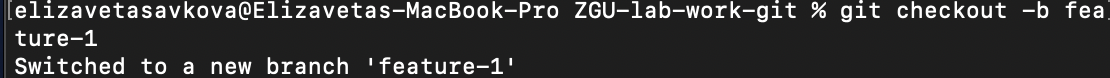
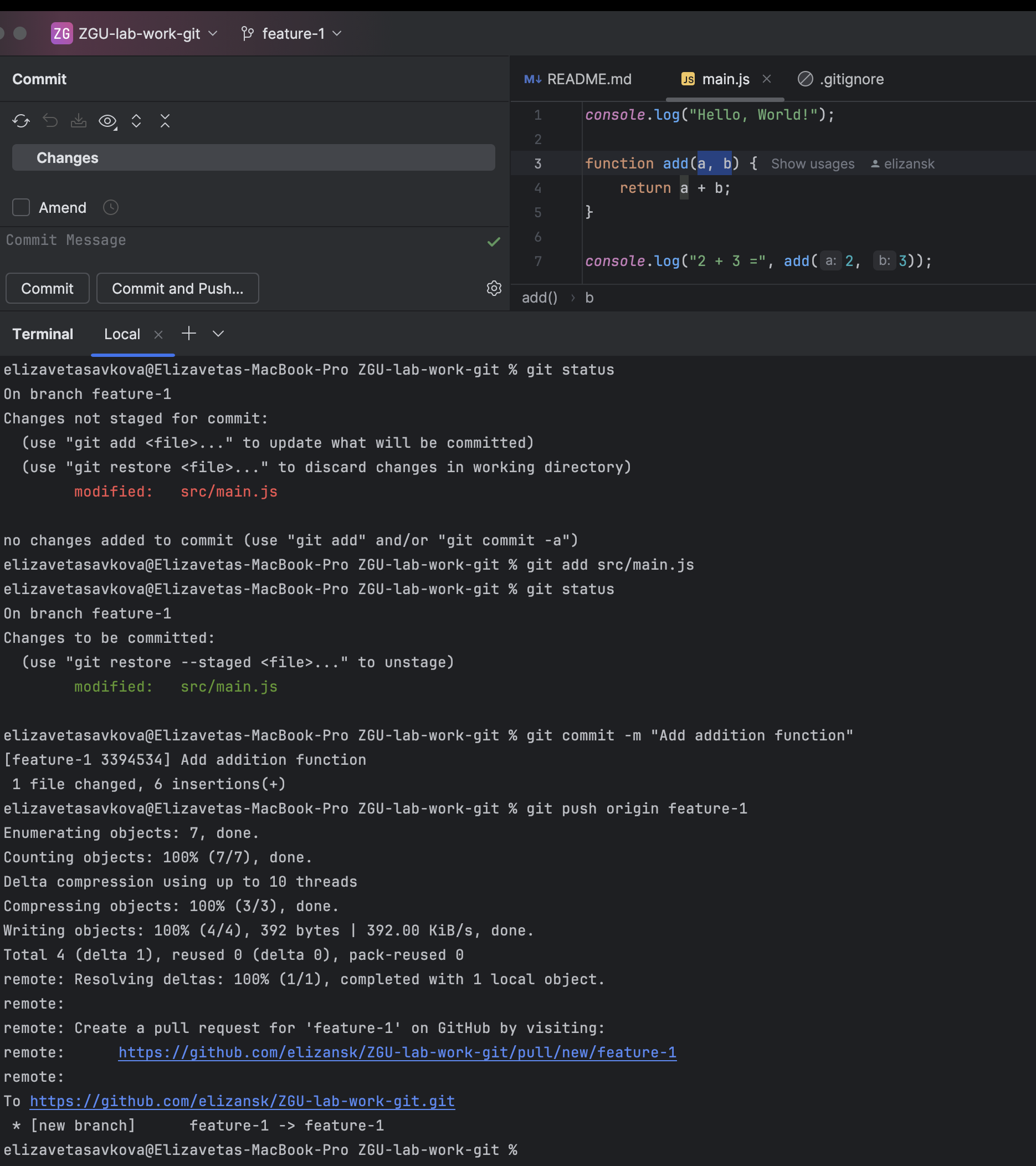


1. **Клонировала репозиторий** на локальный компьютер, используя команду:
2. git clone

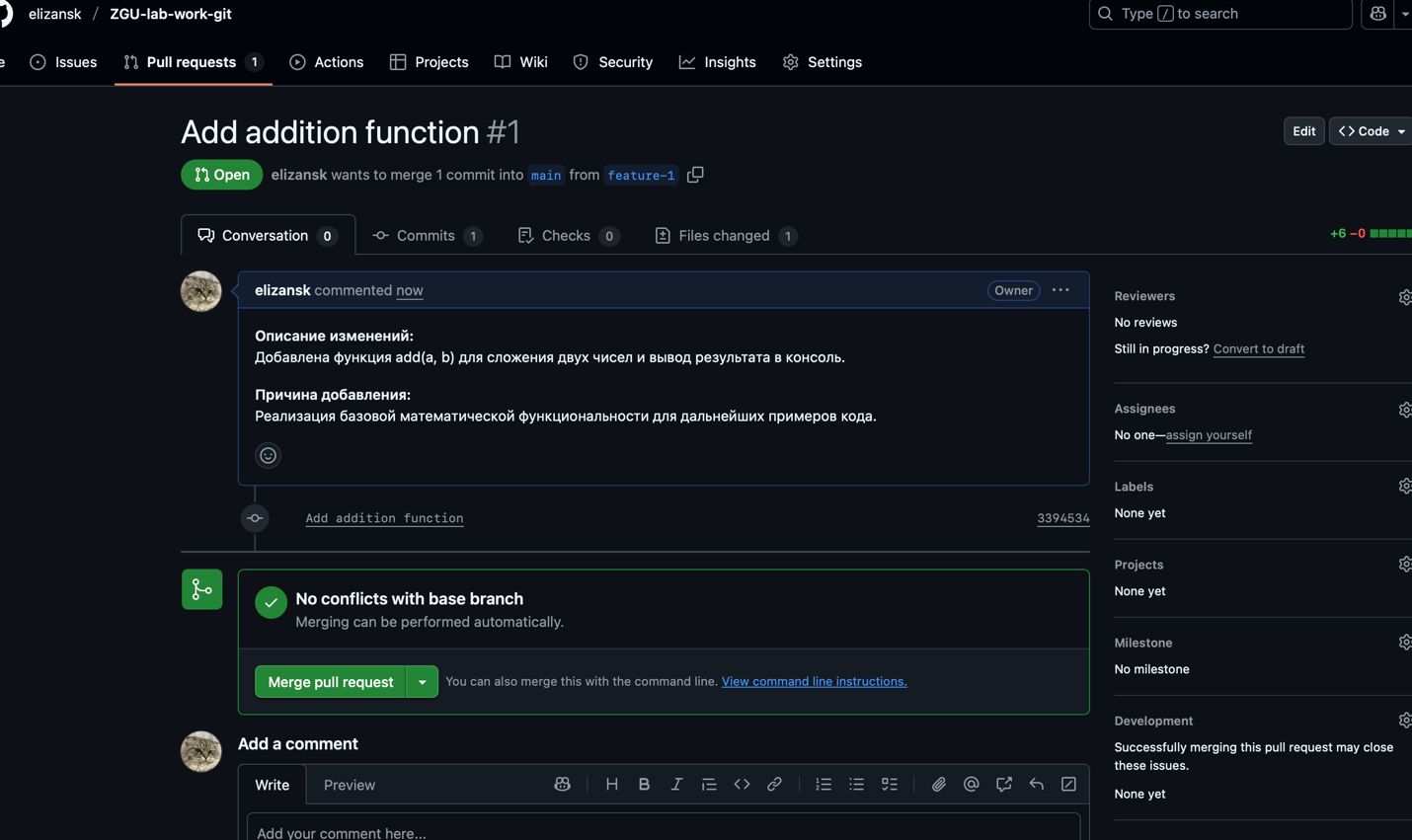
Шаг 2: Настройка репозитория и начальные коммиты

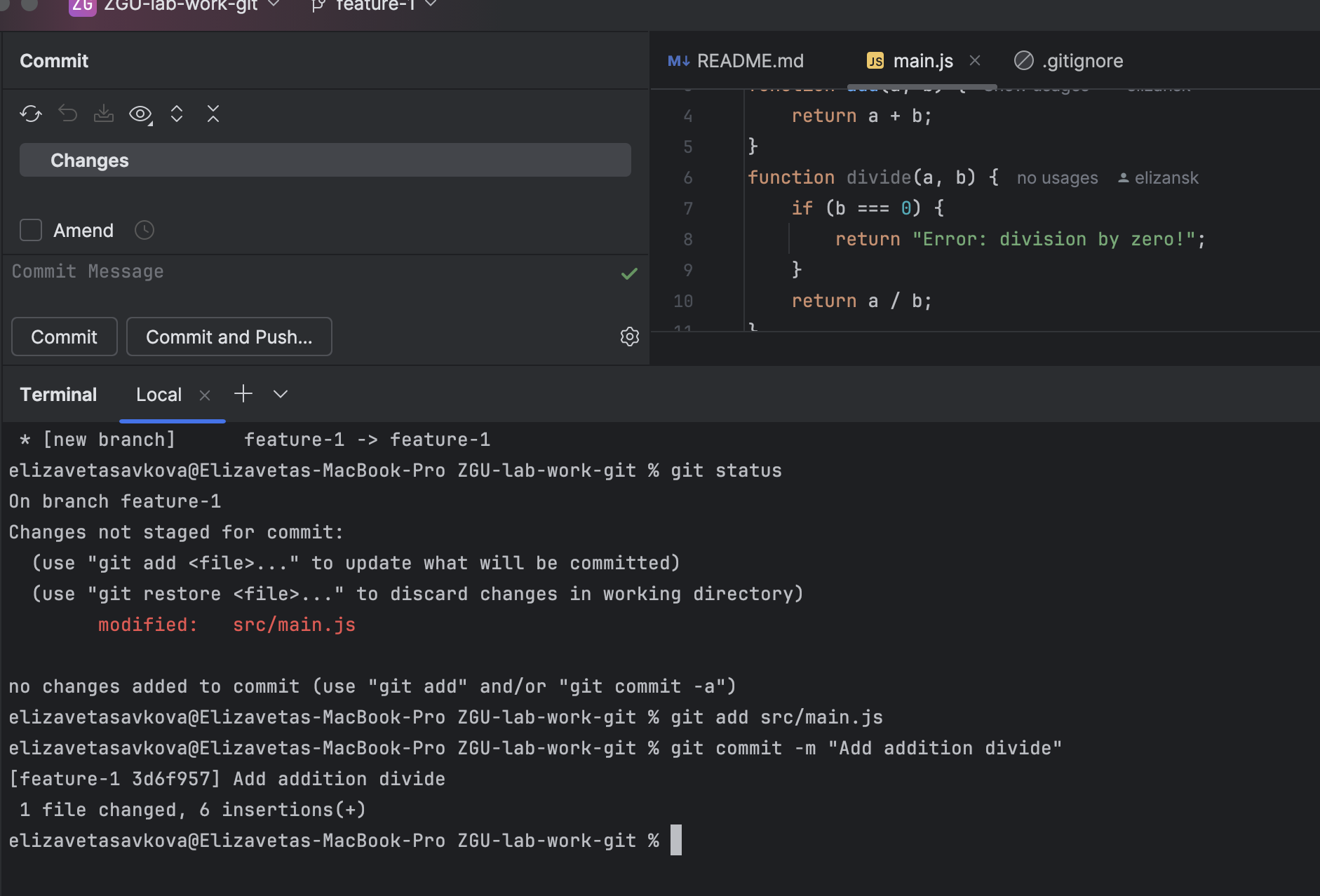
1. В локальном репозитории создала папку src и добавила в неё файл  main.js.
2. Написала небольшой код для вывода фразы "Hello, World!".
3. Добавилв изменения в коммит: git add . и git commit -m "Initial commit with Hello World script"
4. Запушила изменения на GitHub: git push origin main

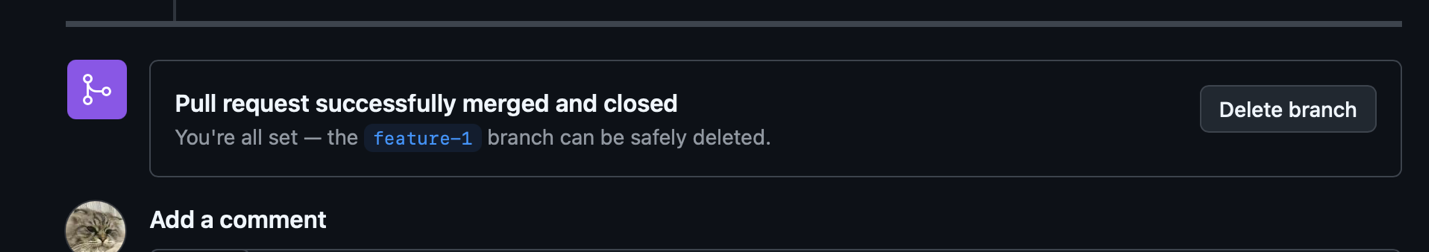
Шаг 3: Создание новой ветки и работа с ней

1. **Создала ветку** feature-1 для добавления новой функциональности. git checkout -b feature-1
2. В этой ветке добавила новую функцию для сложения двух чисел.
3. Зафиксировала изменения и сделала коммит с описанием: git add . git commit -m "Add addition function"
4. Запушила ветку feature-1 на GitHub: git push origin feature-1

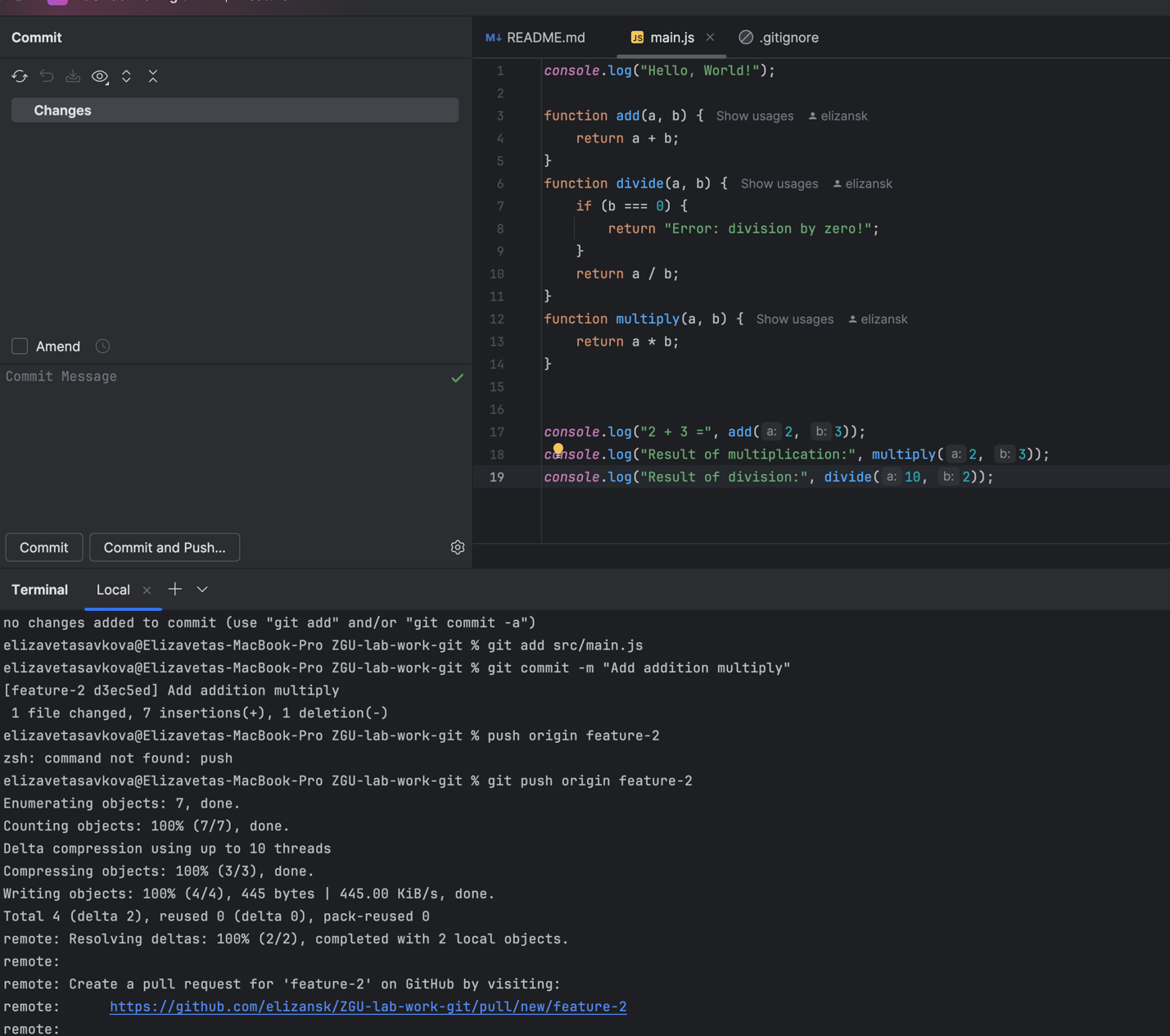
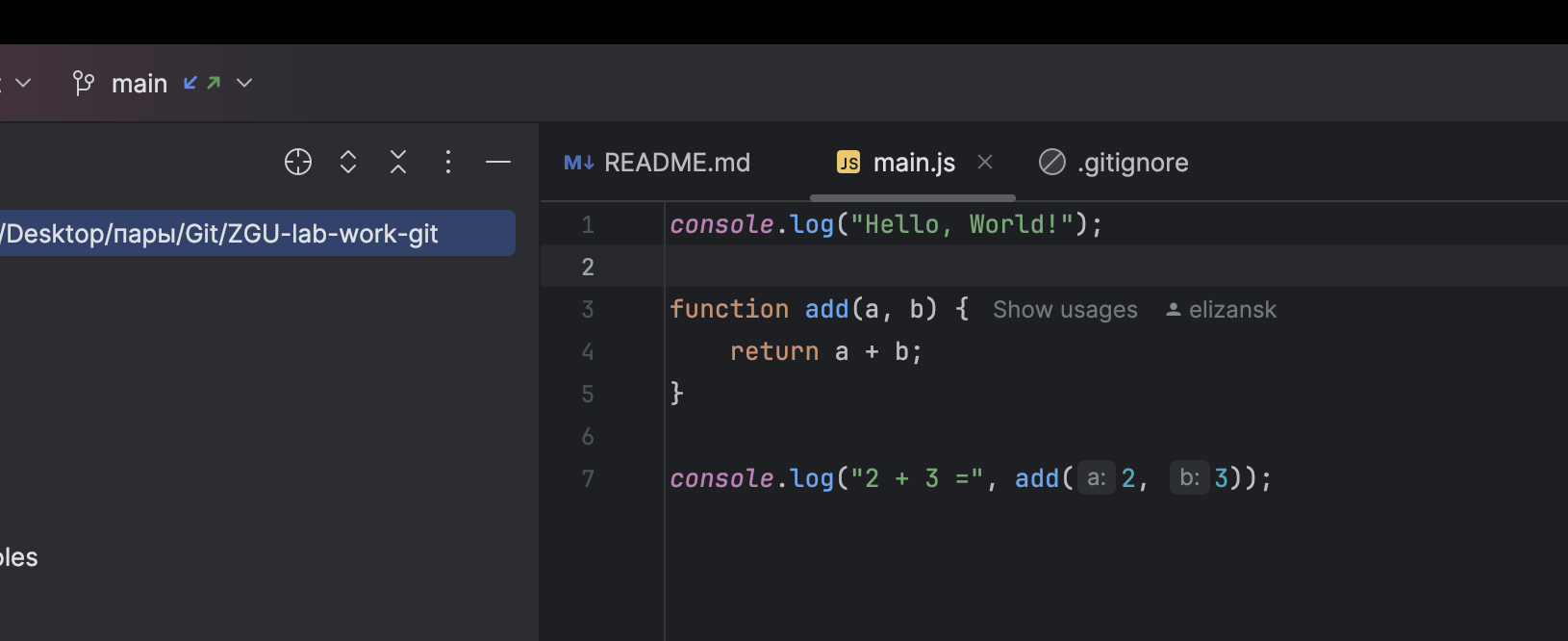
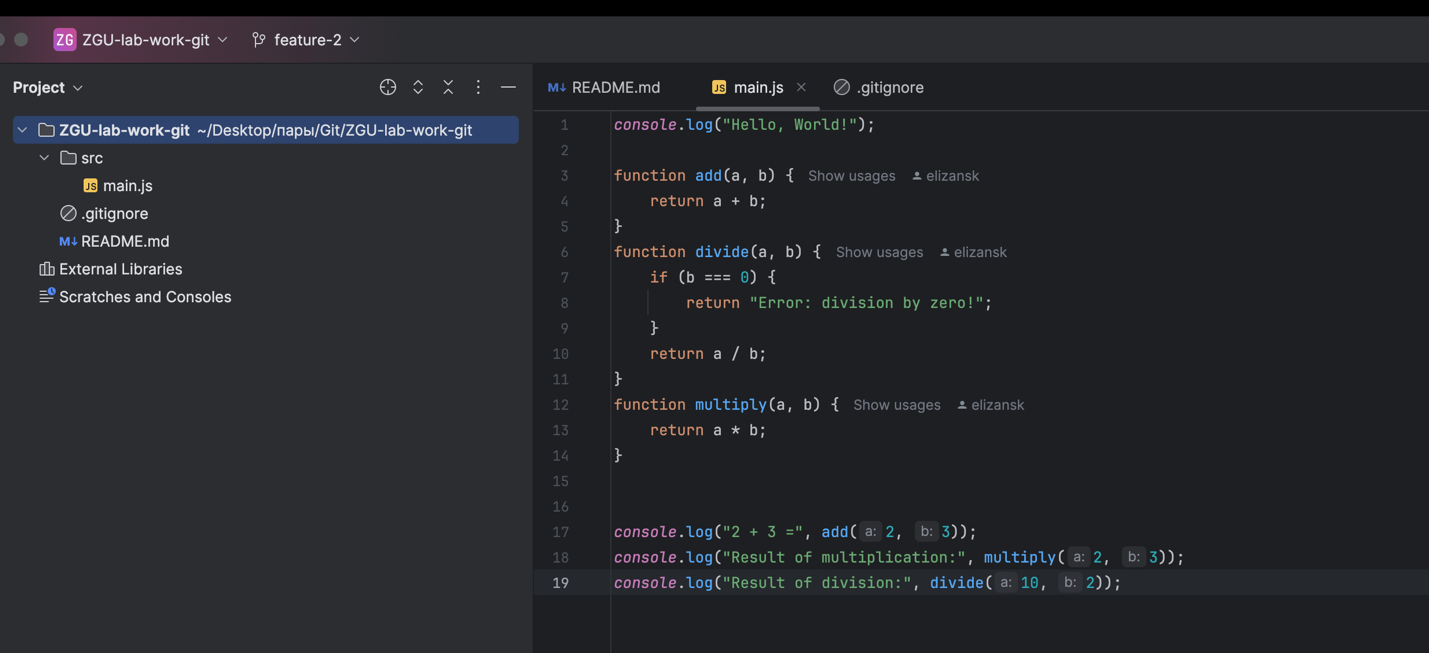
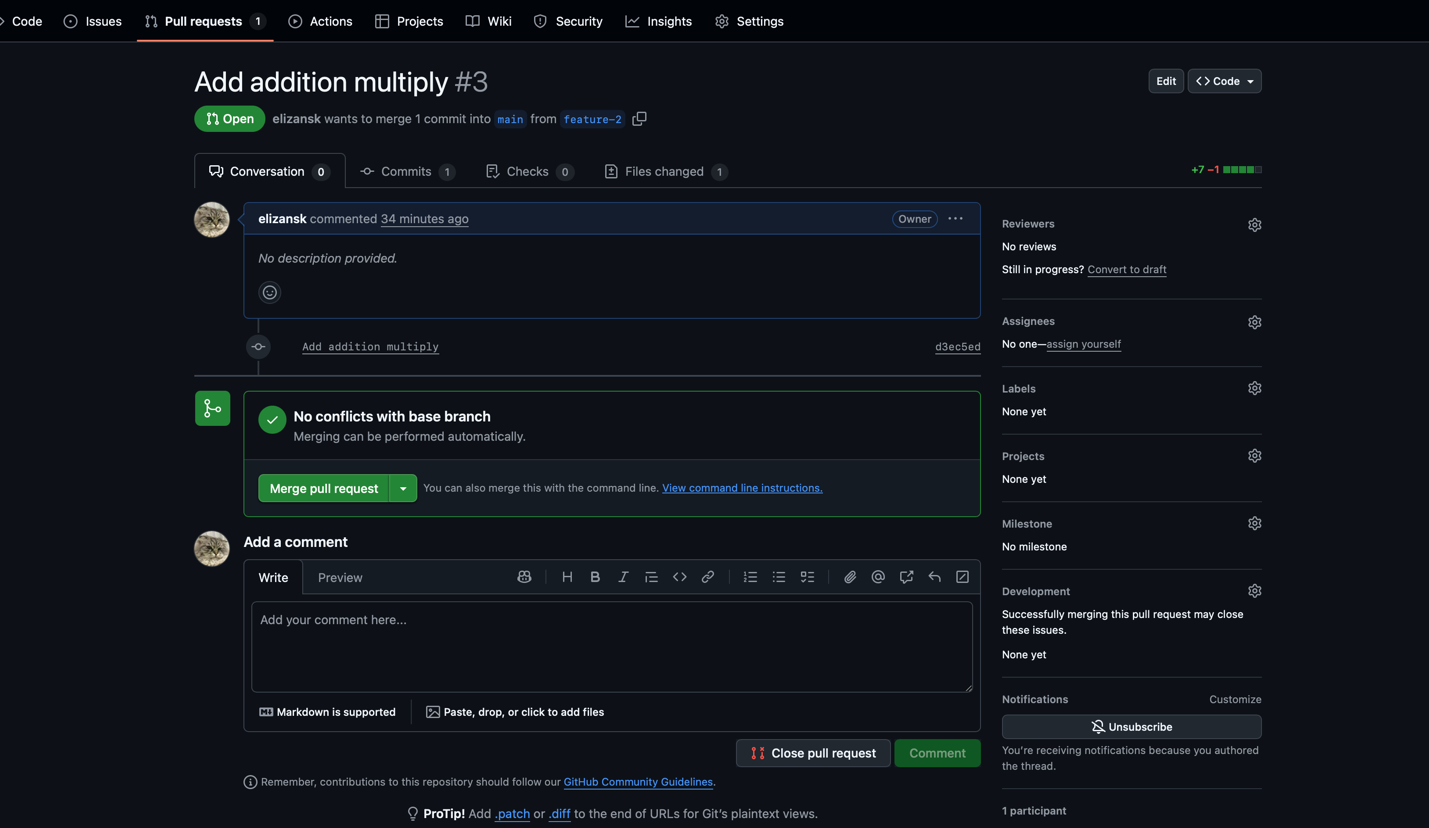
Шаг 4: Работа с Pull Request (PR)

1. **Создала Pull Request** на GitHub из ветки feature-1 в ветку main.
2. В описании PR добавила:
   * Краткое описание изменений.
   * Причину, по которой добавлена новая функциональность.
3. Смоделировала проверку PR. Внесла корректировки в ветку feature-1, затем снова сделала коммит и запушила изменения на GitHub. Обновления автоматически попали в PR.



1. **Приминила PR** и объединла его с веткой main. 

Шаг 5: Управление конфликтами и их разрешение

1. Создала новую ветку feature-2.
2. Внесла изменения в файл main.js добавила функцию умножения двух чисел.
3. Сделала коммит и запушила изменения на GitHub.
4. Вернулась в ветку main и создала еще одну новую ветку hotfix.
5. Внесла изменения в тот же участок кода.
6. Зафиксируйте изменения и объедините ветку hotfix с веткой main.
7. **Создала PR из feature-2** в main, чтобы смоделировать конфликт.
8. Конфликт не был создан, он автоматически был разрешен github. 
9. На скриншотах прилагаю код , на них видно что код у веток main и feature -2 отличается, но конфликт не был создан
10. Был создан конфликт с помощью редактирования/удаления файла в разных ветках.

В проекте существовал файл main1 в ветке main.

Из этой ветки была создана новая ветка feature-2, в которой файл main1 был изменён — например, добавлен новый код или исправлены строки.

Тем временем, в основной ветке main этот же файл main1 был удалён (возможно, как часть рефакторинга или чистки проекта).

После завершения работы над новой функциональностью из ветки feature-2 был создан Pull Request (PR) для объединения изменений обратно в main.

Однако при попытке слияния Git выявил конфликт:

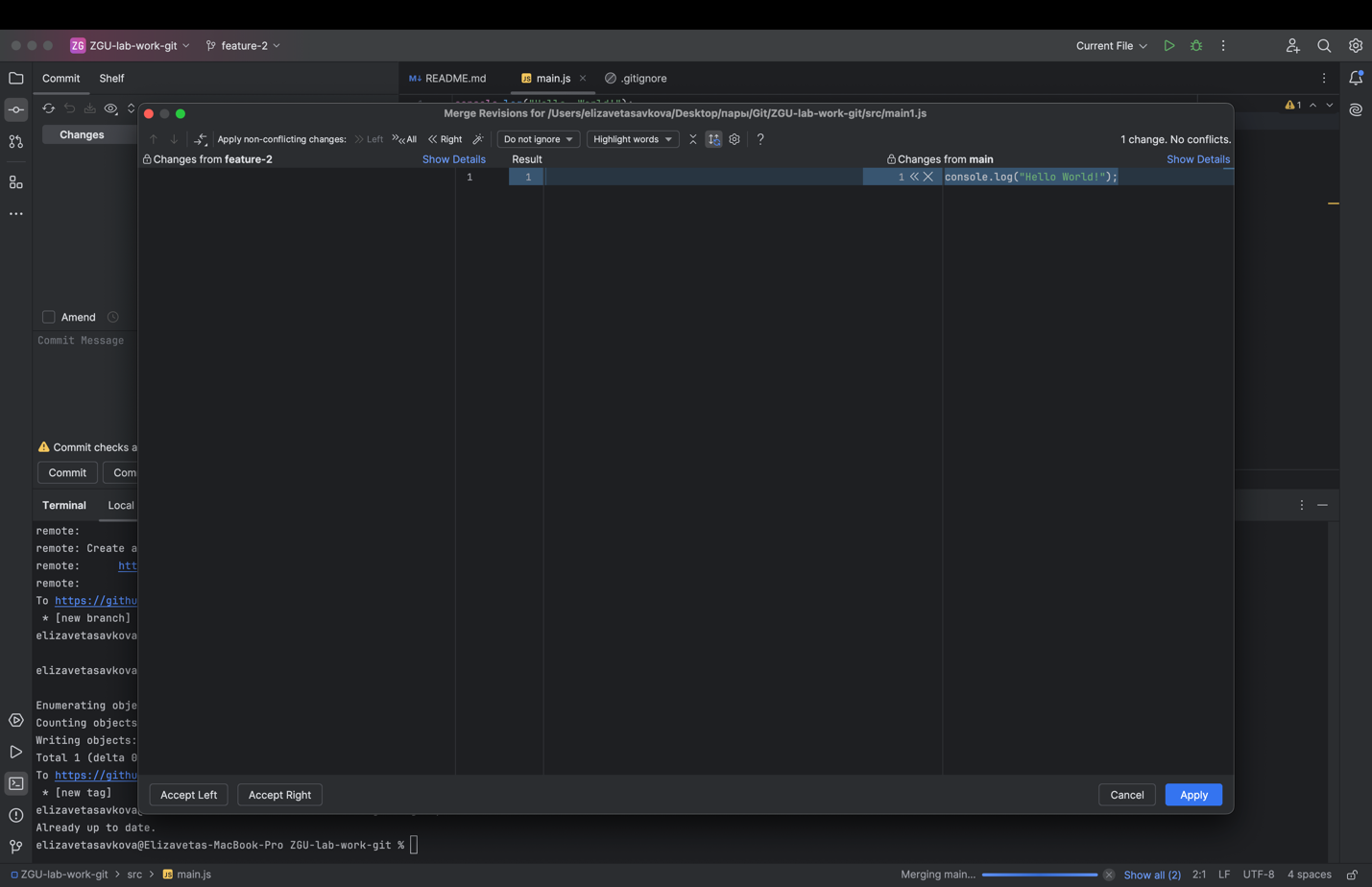
в ветке main — файла main1 уже нет,

в ветке feature-2 — файл main1 всё ещё существует и содержит изменения.

Git не может автоматически решить, что нужно сделать:

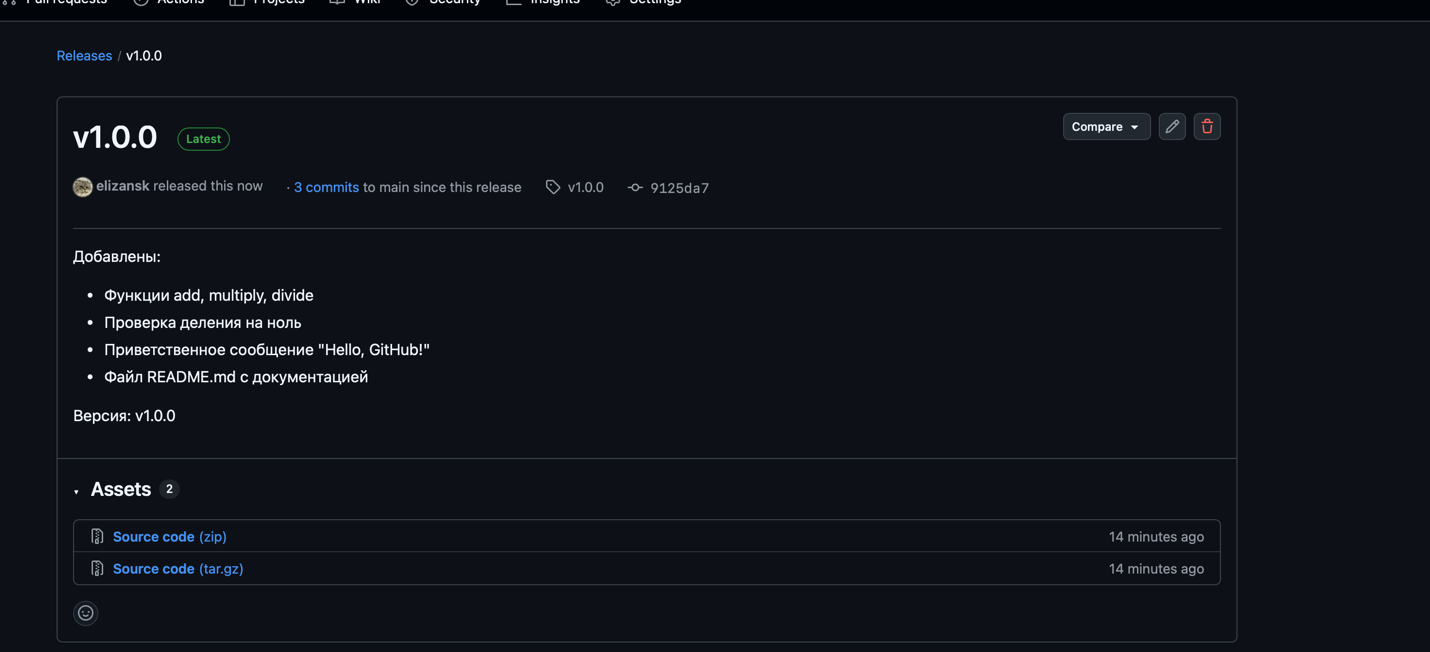
восстановить файл из feature-2 или оставить его удалённым, как в main.

Поэтому PR оказался в состоянии конфликта, требующего ручного разрешения.



Через ide решен конфликт.

Шаг 6: Работа с тегами и релизами

1. Создайте тег для новой версии v1.0.0:  git tag -a v1.0.0 -m "First release"
2. Запушила тег на GitHub: git push origin v1.0.0
3. В разделе "Releases" на GitHub создала новый релиз, привязав его к тегу v1.0.0. Добавила краткое описание обновлений.

Шаг 7: Создание файла документации

Я отредактировала файл README.md , добавив туда описание проекта и его установку

