**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
імені ІГОРЯ СІКОРСЬКОГО”**

Інститут прикладного системного аналізу

кафедра системного проектування

**Контрольна робота №1**

з дисципліни «Проектування інформаційних систем»

на тему *«Системи контролю версій SVN, GIT»*

Виконала:

студентка 4 курсу

групи ДА-72

Грищенко О.Ю.

Київ – 2020

**Мета роботи:** за допомогою системи контролю версій завантажити коди програми у репозіторій. Відтворити типовий цикл розробки програмного забезпечення з використанням системи контролю версій.

**Завдання:**

1. Обрати безкоштовну систему репозиторія для системи контролю версіями, наприклад projectlocker, або інш.

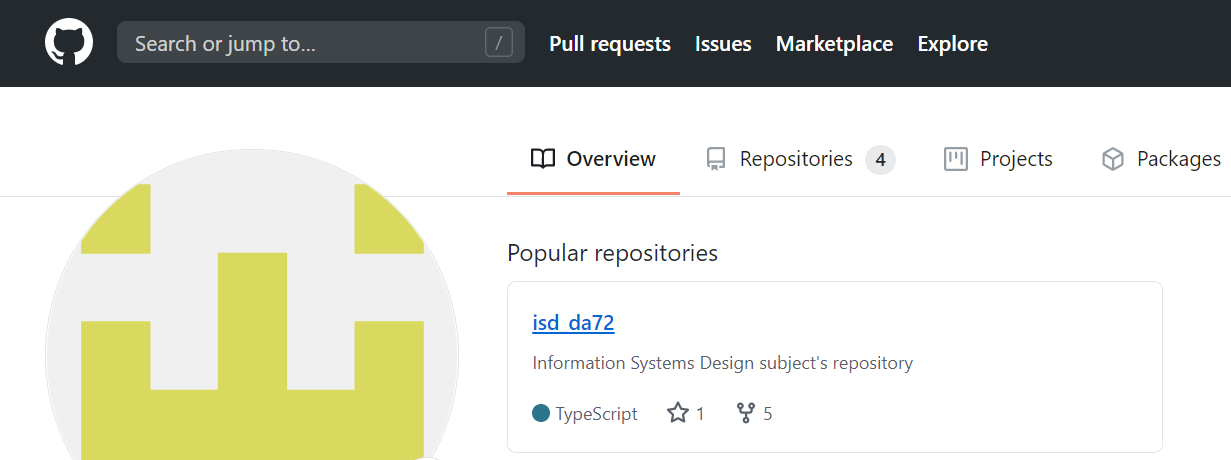
2. Встановити клієнтське безкоштовне програмне забезпечення для роботи с системою контролю версій (GIT, SVN clients).

3. Протягом роботи над лабораторними роботами 2-6 використовувати систему контролю версіями.

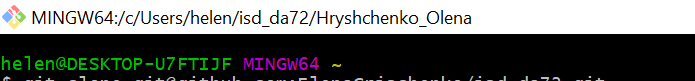
4. Описати цикл розробки програмного забезпечення з використанням системи контрою версій.

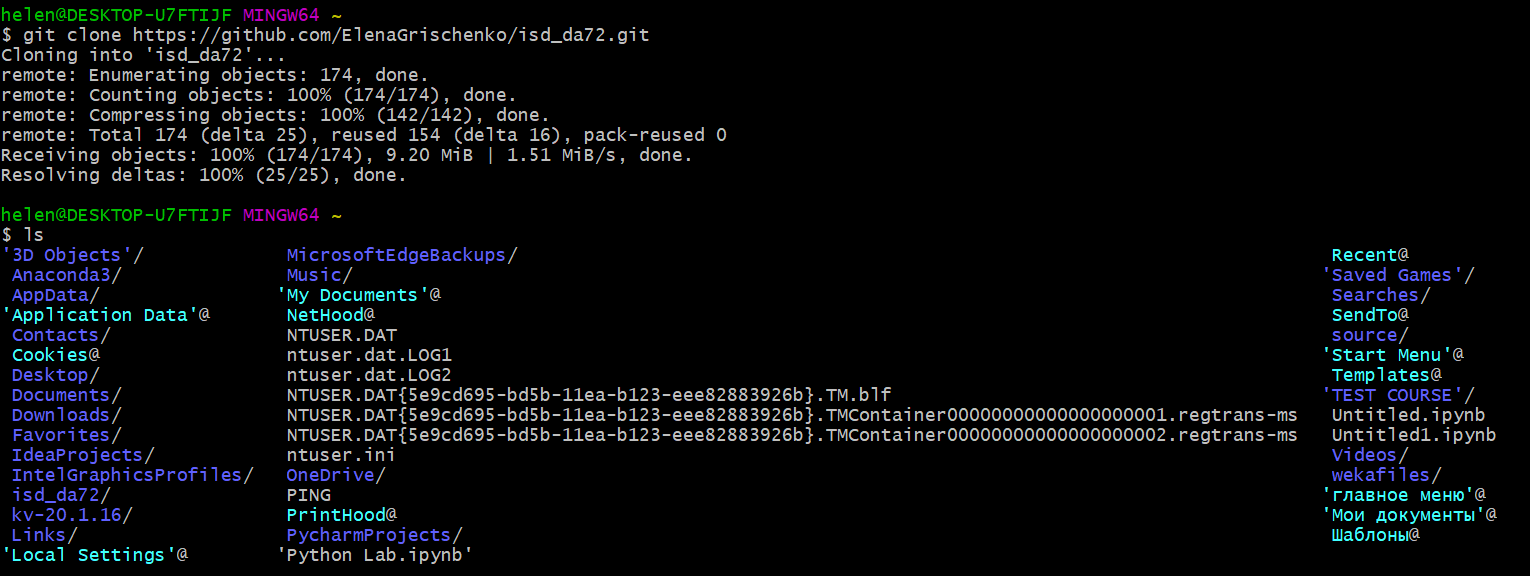
**Хід виконання лабораторної роботи:**

1. Було обрано безкоштовною системою репозиторія для системи контролю версіями GitHub.

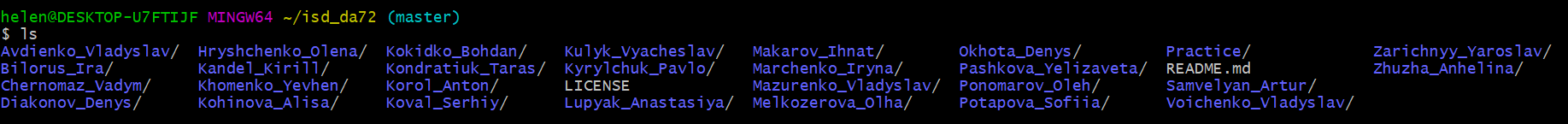


1. Було встановлено безкоштовне програмне забезпечення для роботи з системою контролю версій Git Bash.

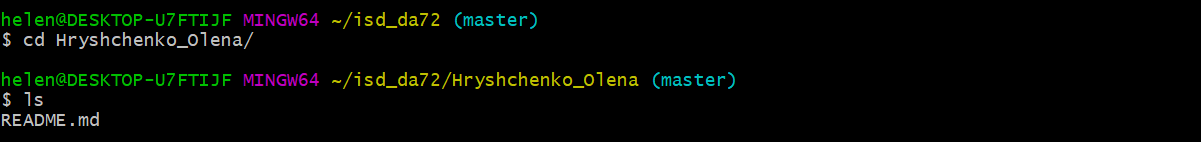


1. Клонуємо репозиторій на локальний комп’ютер за допомогою команди git clone <url> .

Переглянемо вміст папки після завантаження. На сріншоті видно, що команда git clone <url> завантажує всі данні, що знаходяться в репозиторії на момент клонування

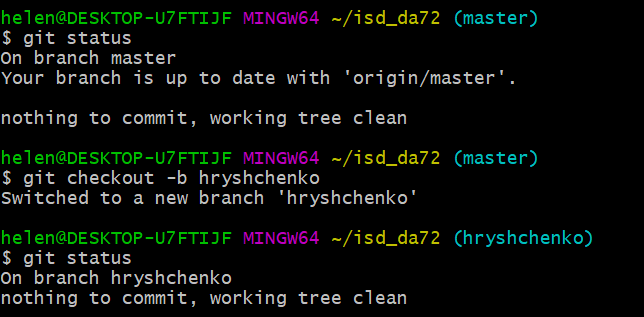


Переглянемо вміст власної папки:

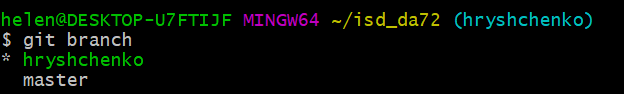


Створимо окремі гілку та перейдемо на неї за допомогою команди

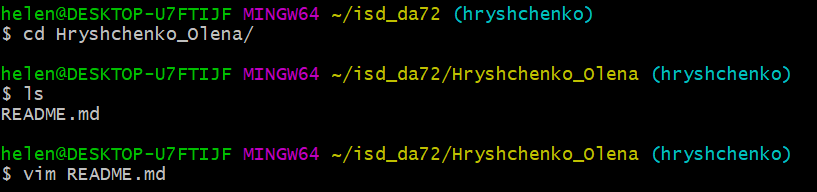
git checkout –b <branch\_name>.

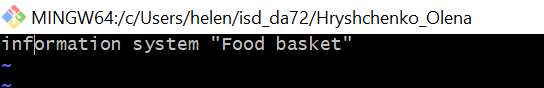


Переглянемо список існуючих гілок.

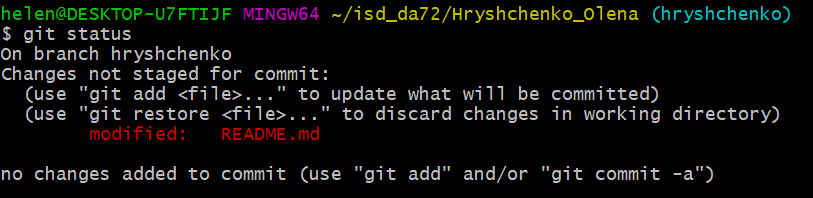


Змінимо вміст README.md та додамо туди назву проекту.

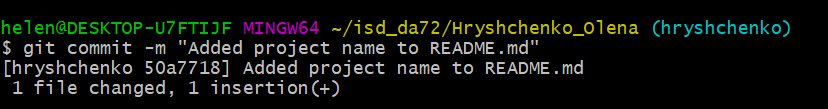
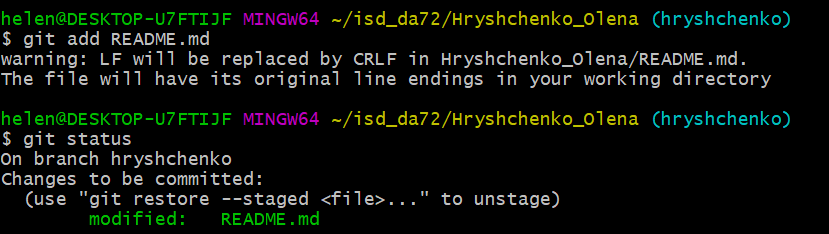




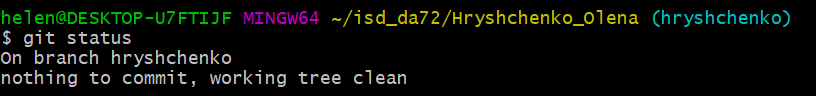
Перевіримо в якому стані, які файли знаходяться у клонованій папці за допомогою git status. На скріншоті видно, що README.md був змінений.



Для того, щоб зробити коміт, спочатку треба додати файл під версій ний контроль за допомогою команди git add, а потім вже зробити комміт git commit із флагом –m для фіксування змін із поясненням.

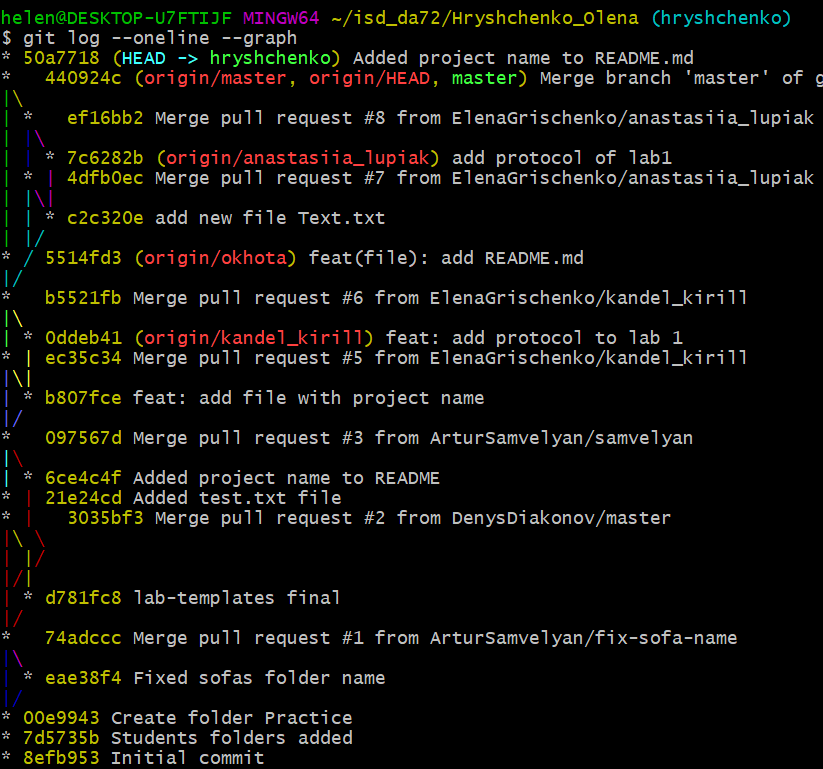


Після коміту переглянемо статус. На скріншоті видно, що немає змін у папці.

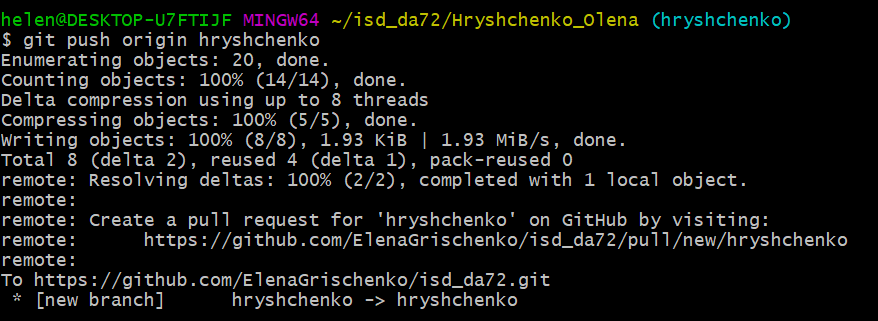


Переглянемо вивід графічного журналі Git за допомогою команди

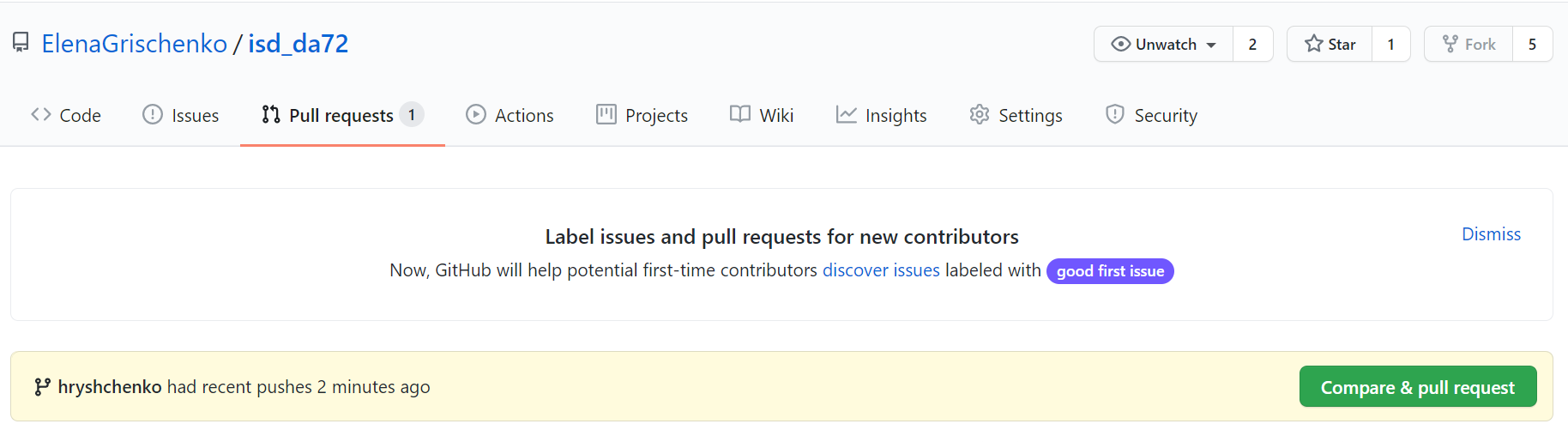
git log –oneline –graph .

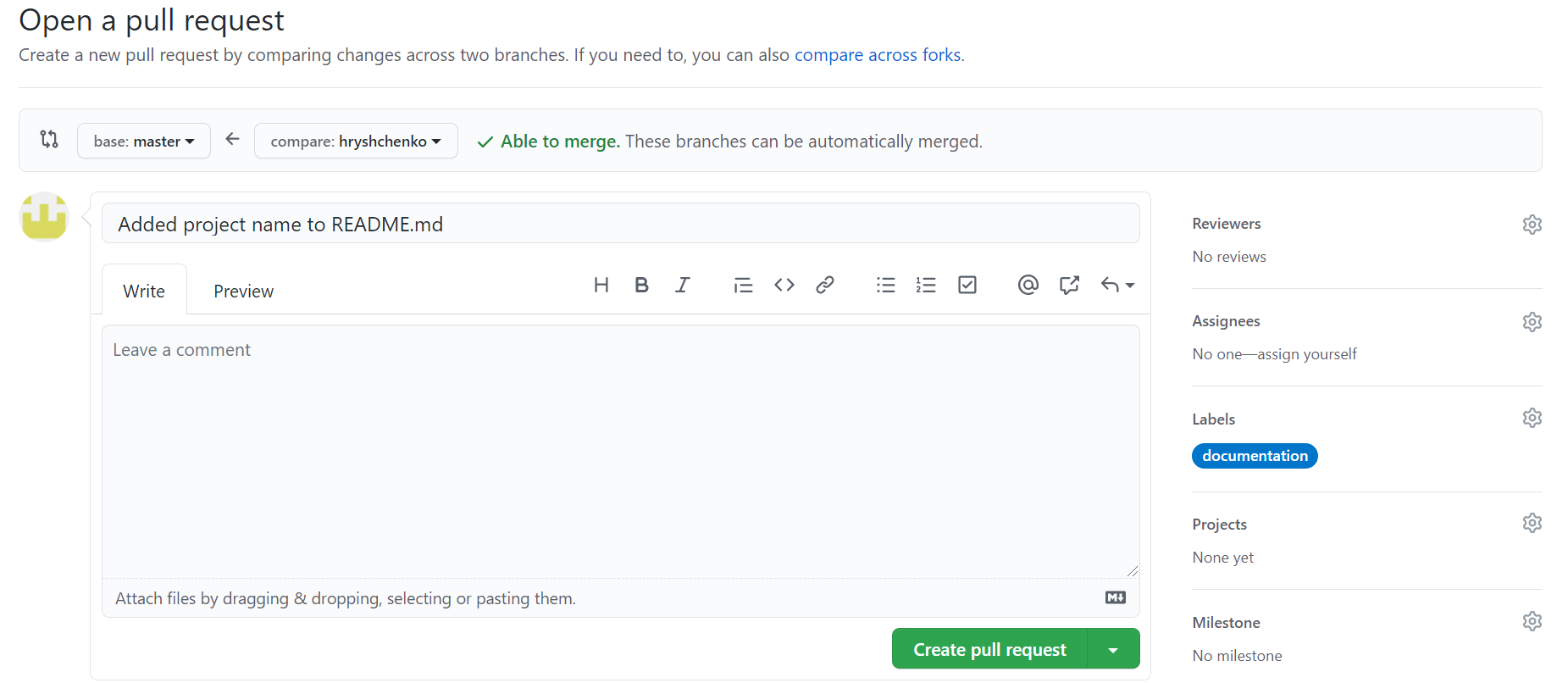


Вигрузимо зміненні файли із локального репозиторія на віддалений на особисту гілку за допомогою команди git push origin <branch\_name>

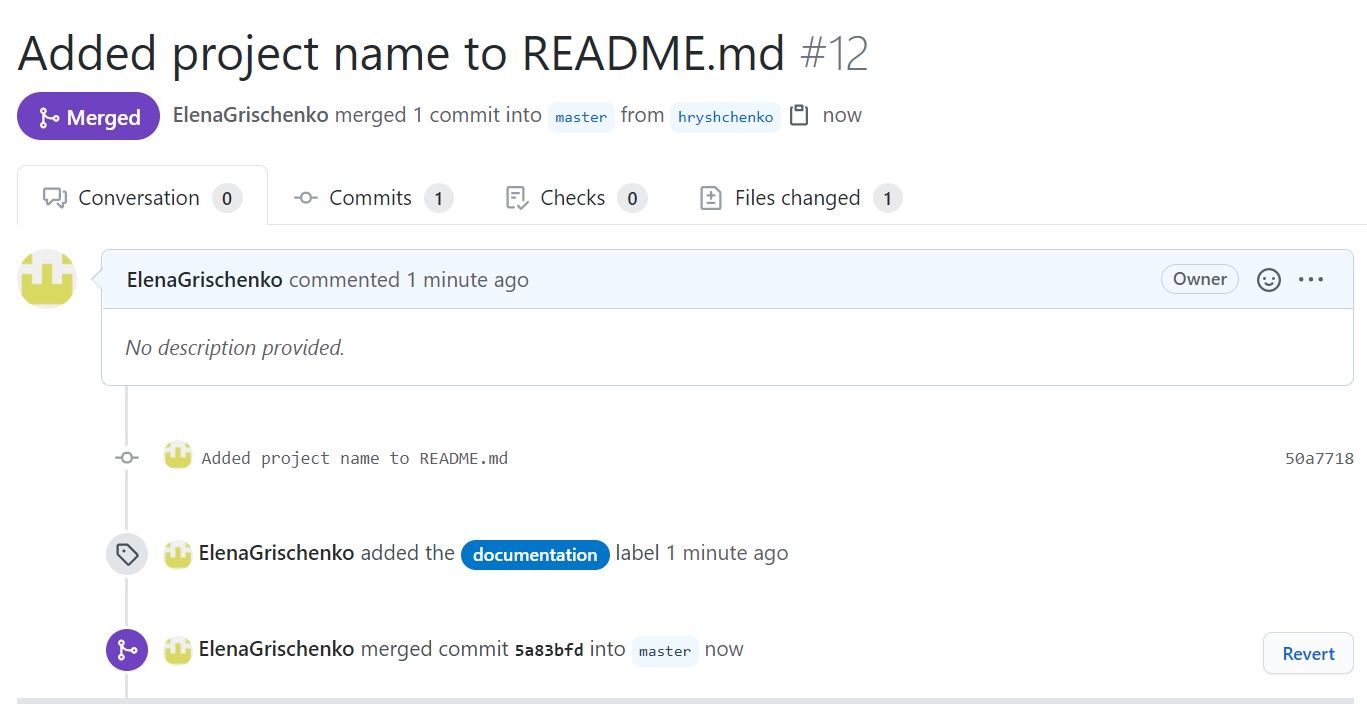


Після пророблених кроків в веб інтерфейсі зливаємо гілки hryshchenko та master для фінального кроку у розробці проекту. Для цього спочатку треба створити пулл реквест, де ми додамо мітку документація.





Пулл реквест було створено.



**Висновок**

Виконуючи лабораторну роботу, було ознайомлено із системою контролю версій GitHub та імітовано розробку програмного забезпечення за допомогою обраної системи. Під час роботи з системою контролю версій я зіштовхнулась із проблемою, коли потрібно весь час розуміти та тримати в голові на якій гілці зараз відбуваються зміни. Також труднощі може визвати перехід між комітами, де потрібно чітко розуміти структуру.