Міністерство освіти і науки України

Національний університет «Львівська політехніка»



**Лабораторна робота № 2**

з дисципліни: «Архітектура спеціалізованих комп’ютерних систем»,

На тему: «Робота з GIT клієнтом та порталом GitHub»

Виконав: ст. гр. КІ-47

Шваліковський Дмитро

Прийняв:

Ткачук Т.І.

Львів - 2023

**Мета:** Ознайомитися з роботою системи контролю версій Git та порталу GitHub. Навчитися створювати репозиторії, клонувати їх, вносити зміни до проекту та синхронізувати їх між локальною та віддаленою версіями репозиторію.

**Завдання:**

**1.** Встановити на своїй робочій машині Git клієнт з можливістю запуску із командного рядка ОС

**2.** Зареєструватись на порталі GitHub

**2.1.** Створити новий репозиторій на GitHub

**2.2.** Зробити опис файлів які присутні на вашому репозиторії

**2.3.** Зробити опис функціоналу (кнопок), які присутні у вашому репозиторії

**3.** Налаштувати файл “.gitconfig”, який знаходиться у каталозі з назвою вашого користувача на вашій локальній робочій машині.

**4.** Створити на робочій машині каталог та клонувати репозиторій

**4.1.** Перевірити гілку проекту, яка в даний момент вибрана

**4.2.** дізнатись alias та та url проекту задля виконання команд push та fetch

**5.** Створити ще один каталог та клонувати репозиторій іншим способом

**5.1.** Перевірити alias та та url проекту задля виконання команд push та fetch

**5.2.** Виконати команду “git pull” для клонування проекту (попередньо ініціалізувавши порожній репозиторій)

**5.3.** Перевірити гілку проекту, яка в даний момент вибрана. Якщо назва гілки відрізняється від очікуваної, пояснити та вирішити цю проблему

**6.** Зробити зміни у одному із файлів репозиторію, додати новий тестовий файл. Результат зафіксувати за допомогою команд

**7.** Створити токен (Personal access tokens) на порталі GitHub та внести зміни у віддалений репозиторій (зміни відобразити у вигляді скріншотів)

**8.** Ознайомитись з консольними командами GIT:

**8.1.** Створити нову гілку проекту та виконати від її імені зміни в проекті

**8.2.** Перемкнутись між різними гілками проекту

**8.3.** Перемкнутись на один з попередніх commit’ів

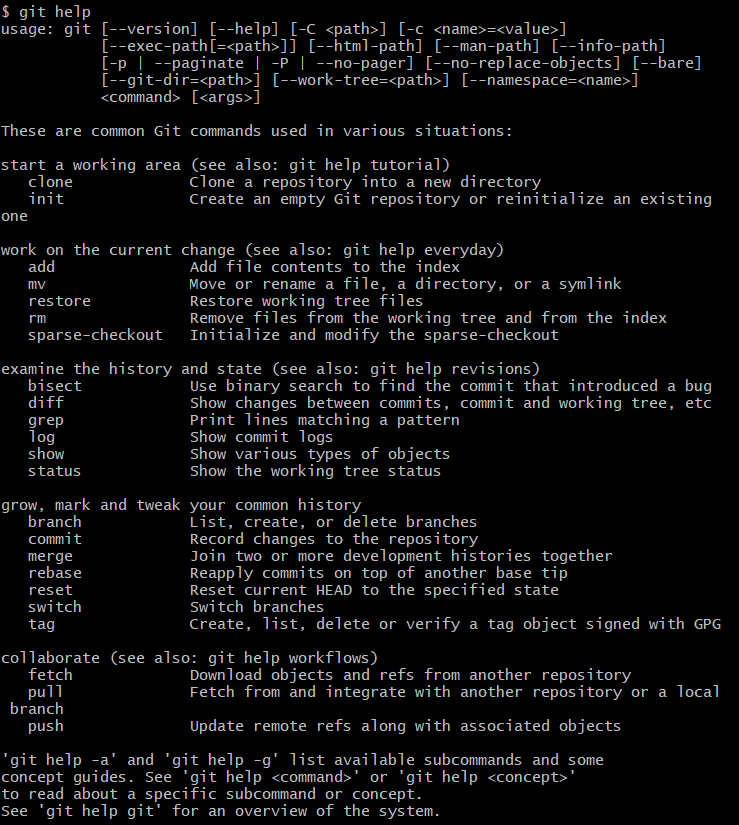
**Хід Роботи:**

**1.** Враховуючи особливість Linux дистрибутиву Debian 12, GIT є встановленим за замовчуванням

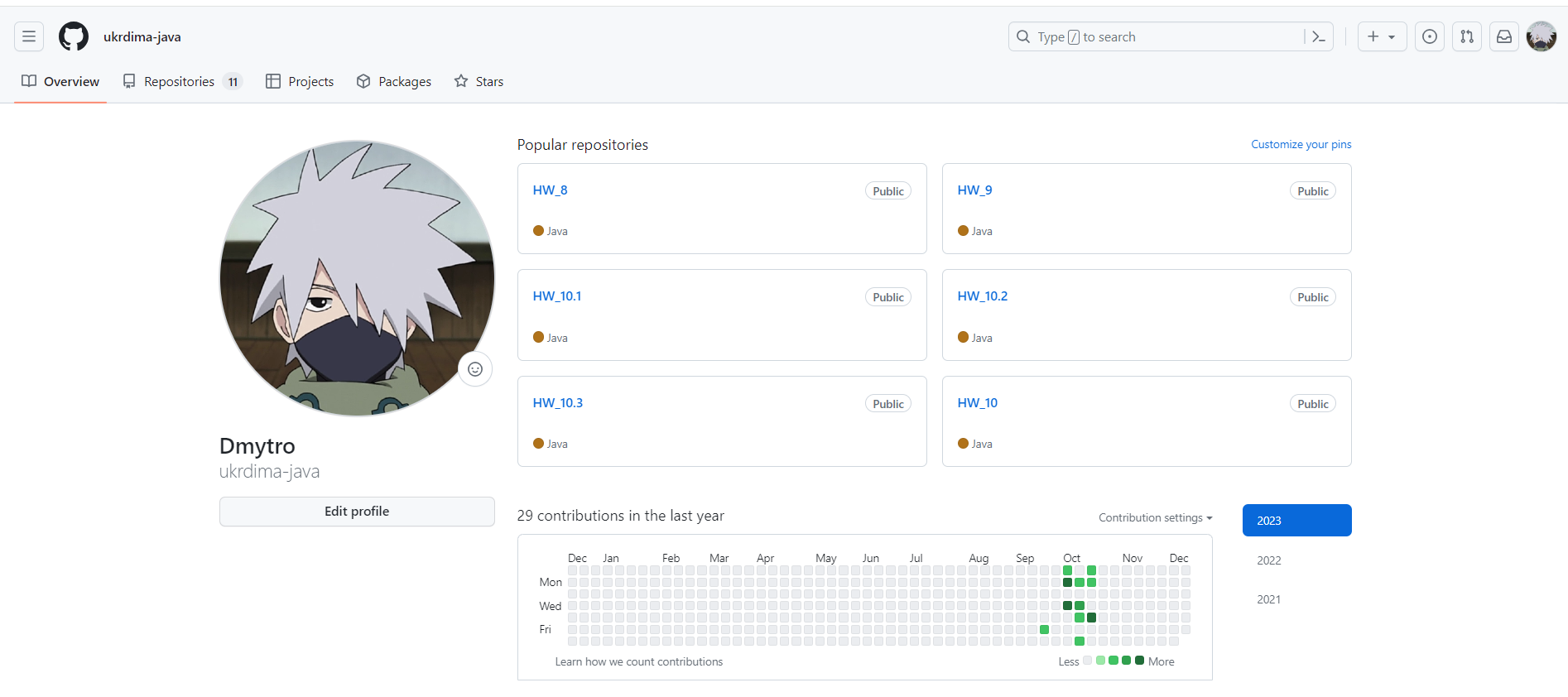
**1.1.** Перевіряємо версію попередньо встановленого GIT клієнту:



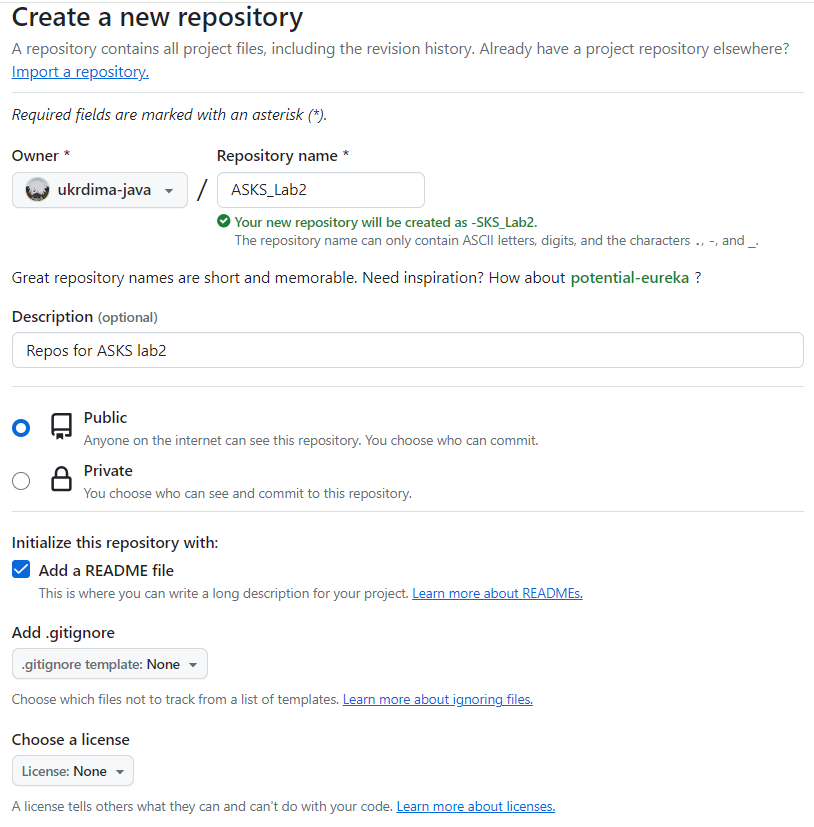
**1.2.** Команда git help для ознайомлення з командами:

****

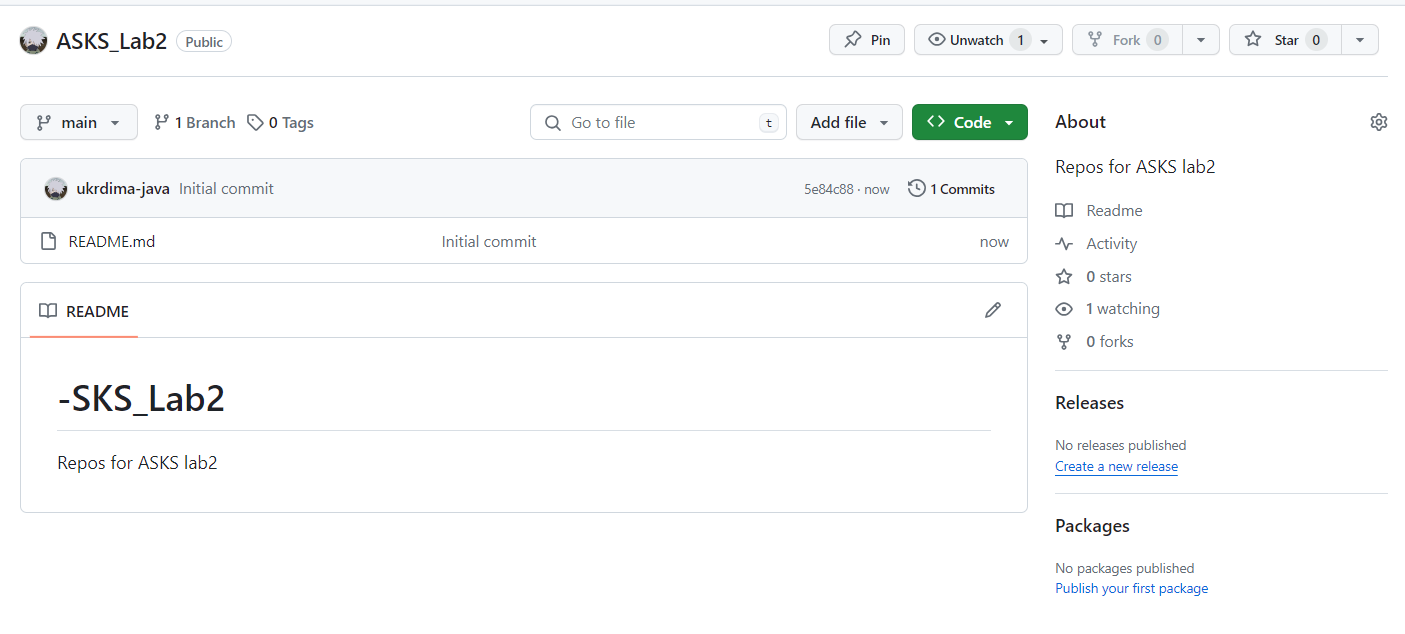
**2.** Реєструємось на порталі GitHub:

****

**2.1.** Створюємо новий репозиторій:

****

**2.2.** Після створення репозиторію нас перенаправить на саму сторінку репозиторію, де можна побачити вказаний при створенні README файл:



README файл є короткою документацією про сам репозиторій, містить короткий опис файлів, що присутні в репозиторії, а також може надавати інструкції зі встановлення чи компіляції (у випадку якщо репозиторій містить код програмного забезпечення)

**2.3.** Також для роботи з репозиторієм GitHub пропонує ряд функцій, для взаємодії з якими у верхній частині сторінки розміщені кнопки:

**2.3.1.** Секція “Code” надає доступ до самого репозиторію та взаємодію з кодом

**2.3.2.** Секція “Issues“ дозволяє стороннім розробникам вказувати на помилки у коді для їх виправлення власником чи групою власників репозиторію

**2.3.3.** Секція “Pull request” надає можливість зручно взаємодіяти розробникам, якщо над репозиторієм працює група людей. Кожен розробник виконує зміни над кодом або файлами репозиторію, після чого за допомогою команди “git push” вносить їх у віддалений репозиторій у GitHub. Для того, щоби зміни збереглись не тільки у гілці якогось розробника, а й у основній гілці, необхідно створити запит на злиття (merge), після чого відповідальний за перевірку та підтвердження учасник (contributor) дозволяє (або відхиляє) запропоновані зміни з другорядної гілки. Після підтвердження зміни вносяться до основої гілки у репозиторії

**2.3.4.** Секція “Actions” дозволяє розробникам створити робочий простір на порталі GitHub для швидкої розробки та розгортання програмного забезпечення. GitHub пропонує готові пресети робочих просторів,а також можливість створити порожній workflow для власного налаштування

**2.3.5.** Секція “Projects” дозволяє зручніше взаємодіяти команді розробників, надаючи можливість створювати, групувати та сортувати завдання, які розподіляються між розробниками для більшої продуктивності в команді

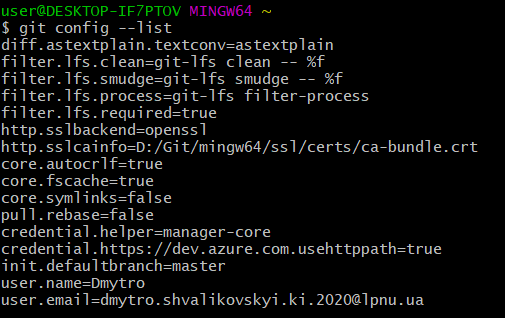
**2.3.6.** Секція “Wiki” надає можливість детально документувати репозиторій (або проект) для сторонніх розробників та кращого розуміння самої програми або коду

**2.3.7.** Секція “Security” надає можливість великим організаціям захистити код і секрети від сторонніх людей або хакерів

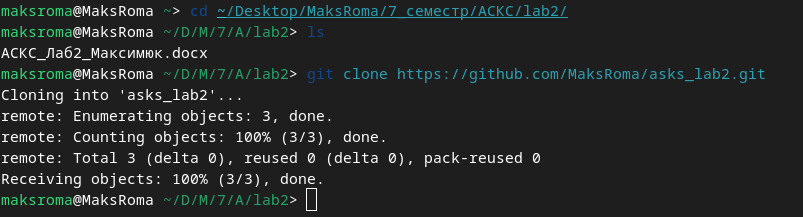
**2.3.8.** Секція “Insights” надає доступ до перегляду змін репозиторію, кількості pull request, закритих та відкритих проблем (issues), співавторів (contributors) тощо

**2.3.9.** Секція “Settings” дозволяє налаштувати різноманітні опції репозиторію, починаючи від додавання контриб’юторів і закінчуючи видаленням репозиторію.

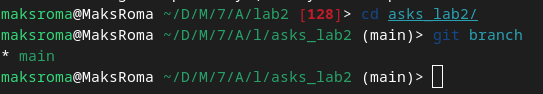
**3.** Виконані налаштування файлу .gitconfig:

****

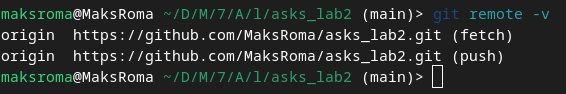
**4.** Створюємо на локальній робочій машині каталог, куди клонуємо репозиторій за допомогою команди “git clone”:



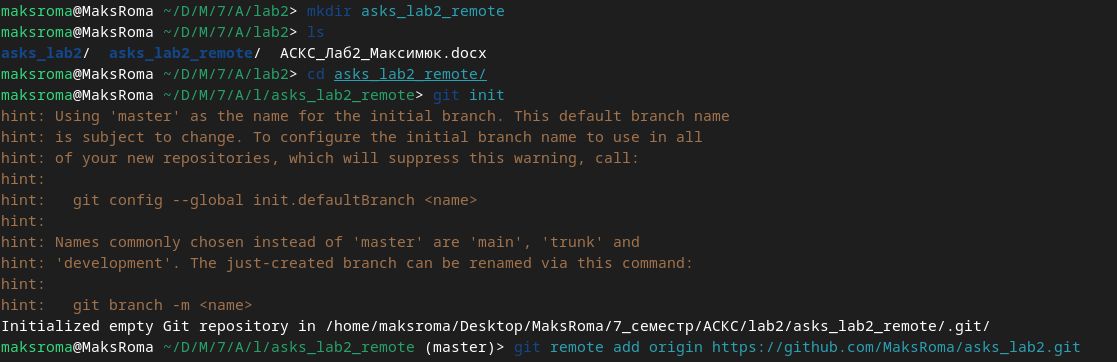
**4.1.** Перевірка гілки, на якій ми знаходимось за допомогою команди “git branch”



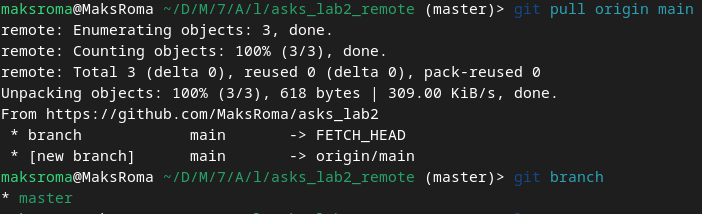
**4.2.** Перевірка alias та url проекту за допомогою команди “git remote -v”



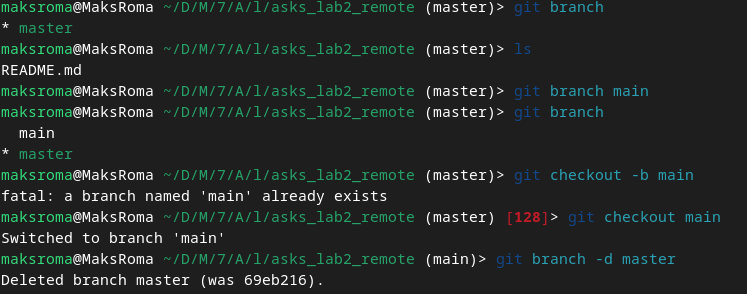
**5.** Створюємо інший каталог для клонування репозиторію іншим способом:



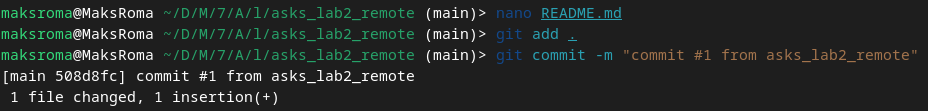
**5.1.** Клонуємо проект та перевіряємо гілку,яка вибрана в даний момент:



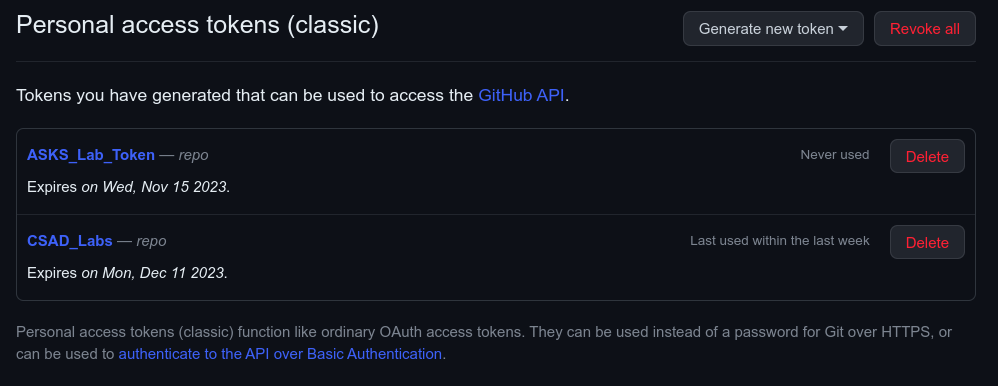
Як можна помітити, гілка залишилась master, хоча при створенні репозиторію основною гілкою була main. Це баг консольної версії GIT, виправити який можна створенням гілки main (тобто основної) та видалення гілки master:

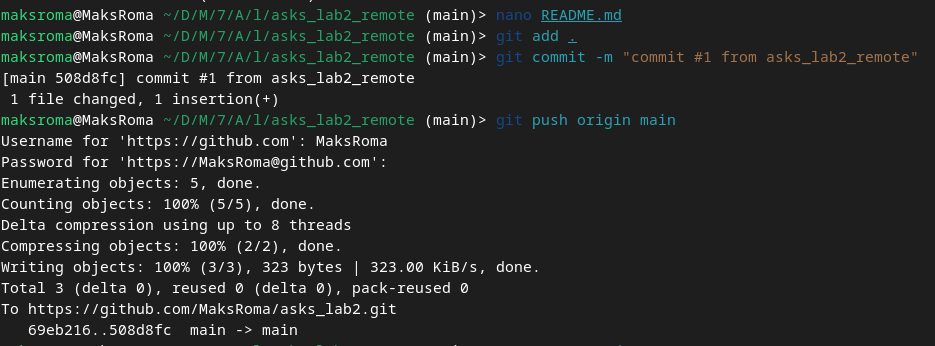


**6.** Робимо зміни у одному з файлів репозиторію, додаємо тестовий файл:



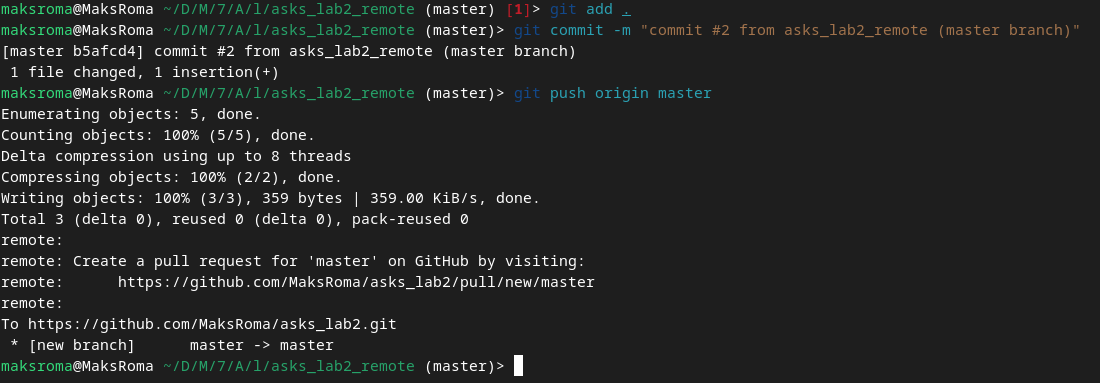
**7.** Створюємо токен (Personal access token) та вносимо зміни у віддалений репозиторій за допомогою команди “git push”:



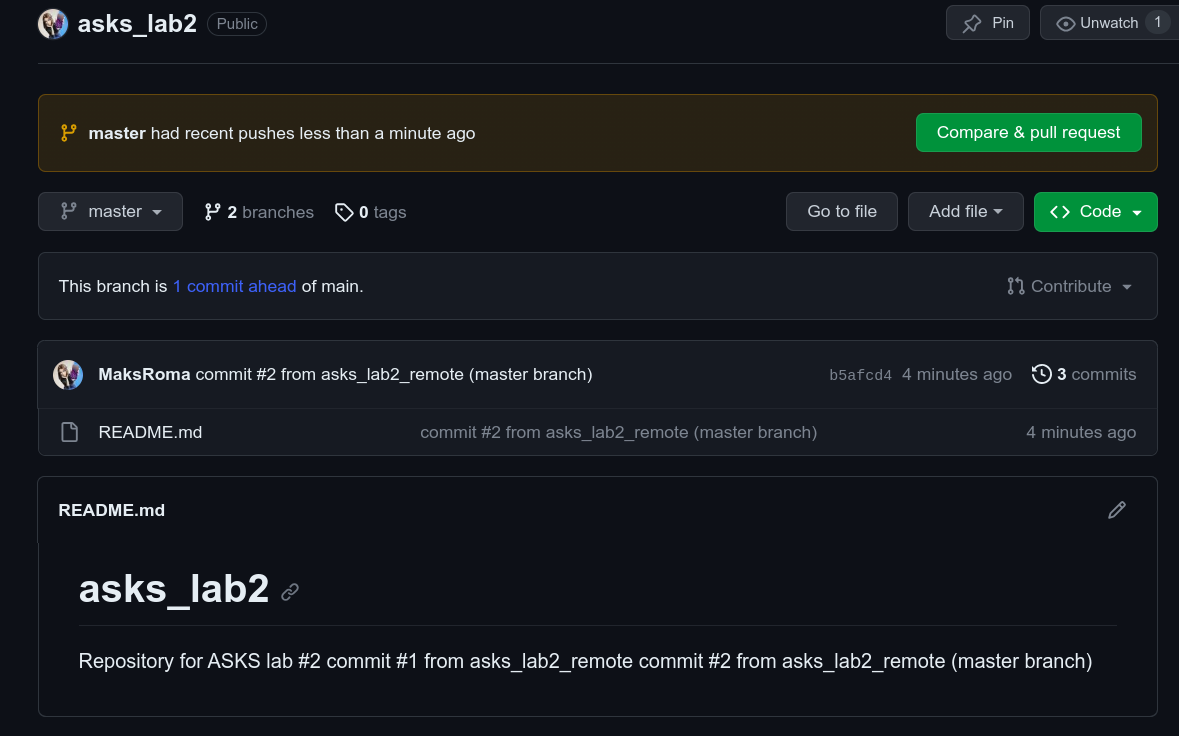


**8.** Ознайомлення з консольними командами GIT:

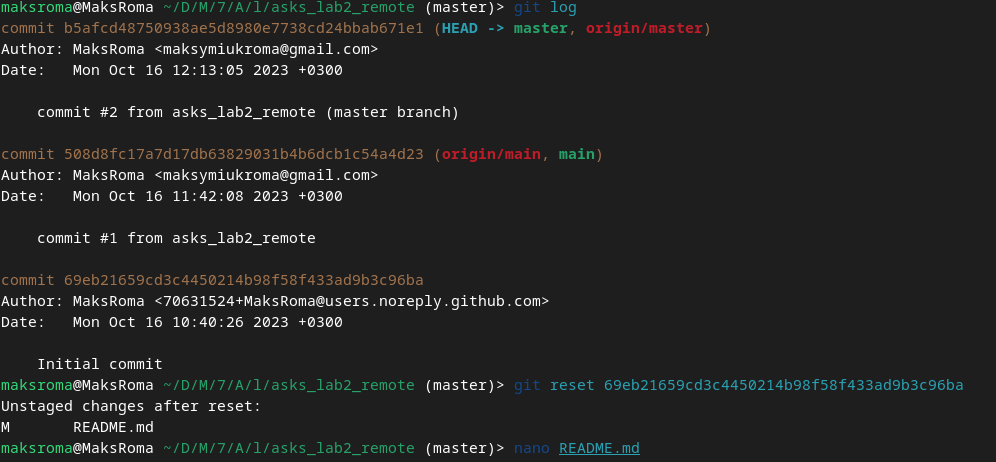
**8.1.** Створення нової гілки та внесення змін від її імені



Як можна помітити на скріншоті нижче, у віддаленому репозиторії з’явилась нова гілка master, в якій збереглись зміни файлу. Для подальшого збереження у основній гілці main необхідно створити pull request для злиття гілок master та main



**8.2.** використовуємо команду “git reset” для повернення до попереднього коміту:



**Висновок:**

У ході виконання лабораторної роботи було встановлено Git клієнт, зареєстровано акаунт на GitHub та створено репозиторій. Виконано налаштування локального середовища Git та клонування віддаленого репозиторію двома способами. Внесено зміни у локальний репозиторій, зафіксовано їх за допомогою команд commit та push. Створено токен доступу до GitHub та оновлено віддалений репозиторій. Було ознайомлено з основними командами GIT для роботи з гілками та фіксацією змін. Таким чином, отримано практичні навички використання системи контролю версій Git спільно з GitHub для управління проектами.