

Вступ

Сервіс Github є ефективним інструментом для спільної командної роботи над проектами. Володіння навичками використання сервісу дозволяє розробнику ефективніше працювати. Така компетенція є вимогою роботодавців до сучасних розробників програмного забезпечення.

Даний методичний посібник містить теоретичний матеріал в інструктивній формі, доповнений ілюстраціями. Для закріплення вивченого матеріалу в посібнику наведено запитання і практичні вправи.

1. Робота з github

1.1 Сутність поняття Git

1.1.1 Збереження файлів

Git – це розподілена система керування версіями файлів та спільної роботи. Проект створив Лінус Торвальдс для управління розробкою ядра Linux, сьогодні проект підтримується Джуніо Хамано.

Git є однією з найефективніших, надійних і високопродуктивних систем керування версіями, яка надає гнучкі засоби нелінійної розробки, що базуються на відгалуженні і злитті гілок. Для забезпечення цілісності історії та стійкості до змін заднім числом використовуються криптографічні методи, можлива прив'язка цифрових підписів розробників до тегів і комітів.

Система спроектована як набір програм, спеціально розроблених з урахуванням їхнього використання у скриптах. Це дозволяє зручно створювати спеціалізовані системи управління версіями на базі Git. Наприклад, Cogito є саме таким прикладом фронтенда до репозиторіїв Git. А StGit використовує Git для управління колекцією патчів.

1.1.2 Збереження файлів

Git, на відміну від Subversion і подібних до неї систем, не зберігає інформацію як список змін (патчів) для файлів. Замість цього, Git зберігає дані набором зліпків.

Кожного разу при фіксації поточної версії проекту Git зберігає зліпок того, як виглядають всі файли проекту. Але, якщо файл не змінювався, то дається посилання на раніше збережений файл. Git схожий на своєрідну файлову систему з інструментами, які працюють поверх неї.

Уся база даних Git зберігається в теці з назвою `.git`. Для кожного файлу, що відстежується, Git зберігає розмір, час створення і час останньої зміни. Ці дані зберігаються у файлі `index`.

У Git файли можуть знаходитися в одному із 3-х станів: *зафіксованому* (файл уже збережено в локальній базі даних), *зміненому* (файл було змінено, але зміни не зафіксовано) і *підготовленому* (файли було змінено і відмічено для фіксації).

1.1.3 Локальні операції

У Git більшість дій можна виконувати без використання Інтернет-підключення на локальній файловій системі. Будь-який коміт спочатку робиться локально, потім вивантажується у віддалений репозиторій. Уся історія змін локально зберігається і, за необхідності, вивантажується у віддалений репозиторій.

1.1.4 Цілісність даних

У своїй базі Git зберігає дані по хешах файлів. Як хешувальна функція використовується SHA-1. Перед кожним збереженням файлів Git обчислює SHA-1 хеш файлу, і отриманий хеш стає індексом файлу в Git. Використовуючи хеш, Git легко відслідковує зміни в файлах.

1.1.5 Галуження (гілки)

Галуження – це розмежування від основної лінії розробки. Git дозволяє створити декілька гілок і перемикатися між ними.

Гілка в Git просто являє собою вказівник на одну із фіксацій. Під час кожної нової фіксації гілка в Git рухається автоматично (тобто перемикається на фіксацію). Гілка є простим файлом, який містить 40 символів контрольної суми SHA-1 фіксації. Створення нової гілки дуже швидке, оскільки це дорівнює запису в файл 41 байта (40 символів + символ нового рядка).

Галуження корисне, бо дозволяє працювати декільком розробникам над своїм функціоналом, не заважаючи іншим і не псуючи основну гілку.

За замовчуванням Git створює гілку з назвою *master*.

1.1.6 Зливання та перебазовування даних

Git підтримує два способи для інтеграції змін з гілки в гілку: *merge* (зливання) та *rebase* (перебазування). Основна різниця полягає в тому, що *rebase* запам'ятовує фіксації у вигляді патчів, перемотує гілку і застосовує патчі у вигляді фіксацій, *merge* - зливає дві гілки в одну.

1.2 Реєстрація на сайті github.com

Для того щоб почати працювати з системою Git, потрібно зареєструватися на сайті github.com та створити репозиторій.

Репозиторій – це контейнер для зберігання ваших файлів.

Розглянемо процес реєстрації на сайті github.com:

У браузері перейдіть до сайту github.com. Для цього у адресному рядку введіть <https://github.com/>. Відкриється сторінка з формою реєстрації.

Заповніть форму (рисунок 1.1) та натисніть на кнопку *Sign up for GitHub*.

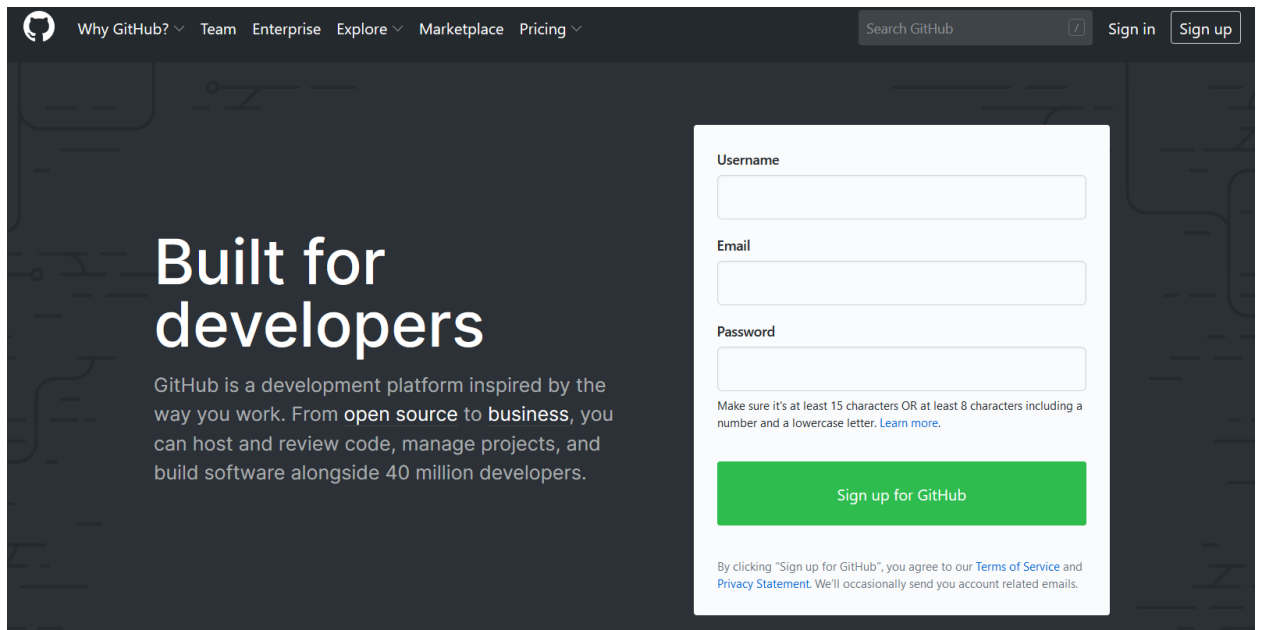


Рисунок 1.1 – Форма реєстрації на сайті github.com

1.3 Створення репозиторію

Після успішної реєстрації та входу на сайт github.com під своїм обліковим записом, натисніть на кнопку «+» та з меню, що випадає, оберіть *New repository* (рис. 1.2).

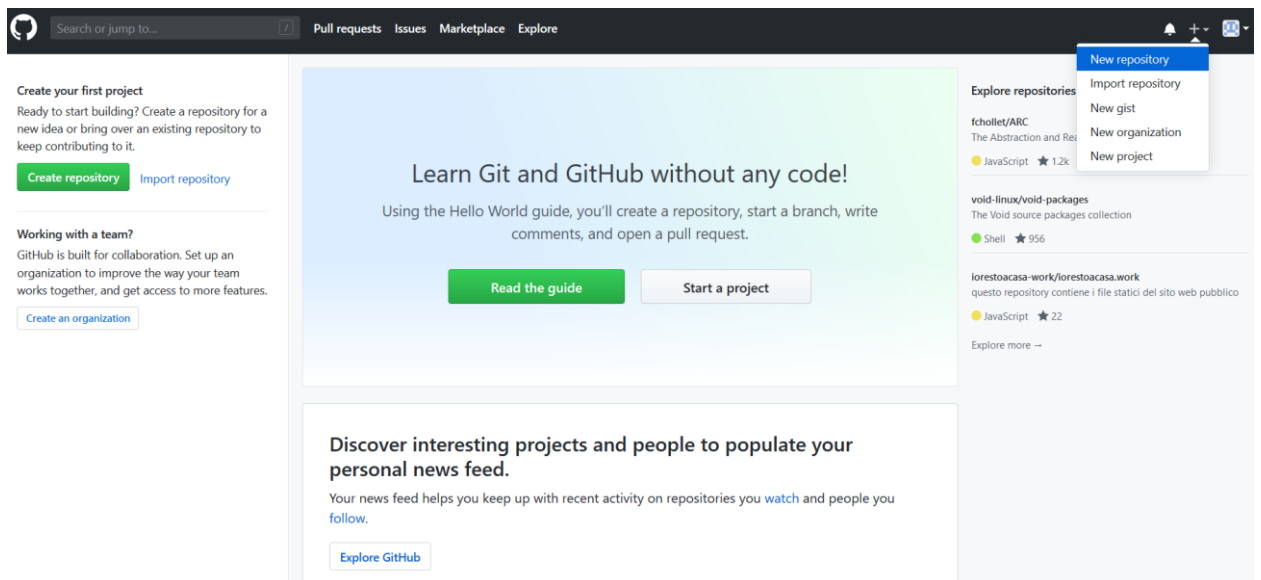



Рисунок 1.2 – Створення нового репозиторію

На сторінці створення репозиторію (рисунок 1.3) треба задати його ім'я, задати опис та ініціалізувати репозиторій README файлом. Потім натиснути на кнопку *Create repository*.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner

 ebabeshko ▾

Repository name *

/ test ✓

Great repository names are short and memorable. Need inspiration? How about **silver-lamp**?

Description (optional)

Тестовий репозиторій

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾

Add a license: **None** ▾



Create repository

Рисунок 1.3 – Створення нового репозиторію

Після створення репозиторію ви потрапите на сторінку репозиторію. Поки що тут є один файл `Readme.md`, котрий було створено разом з репозиторієм (рис. 1.4).

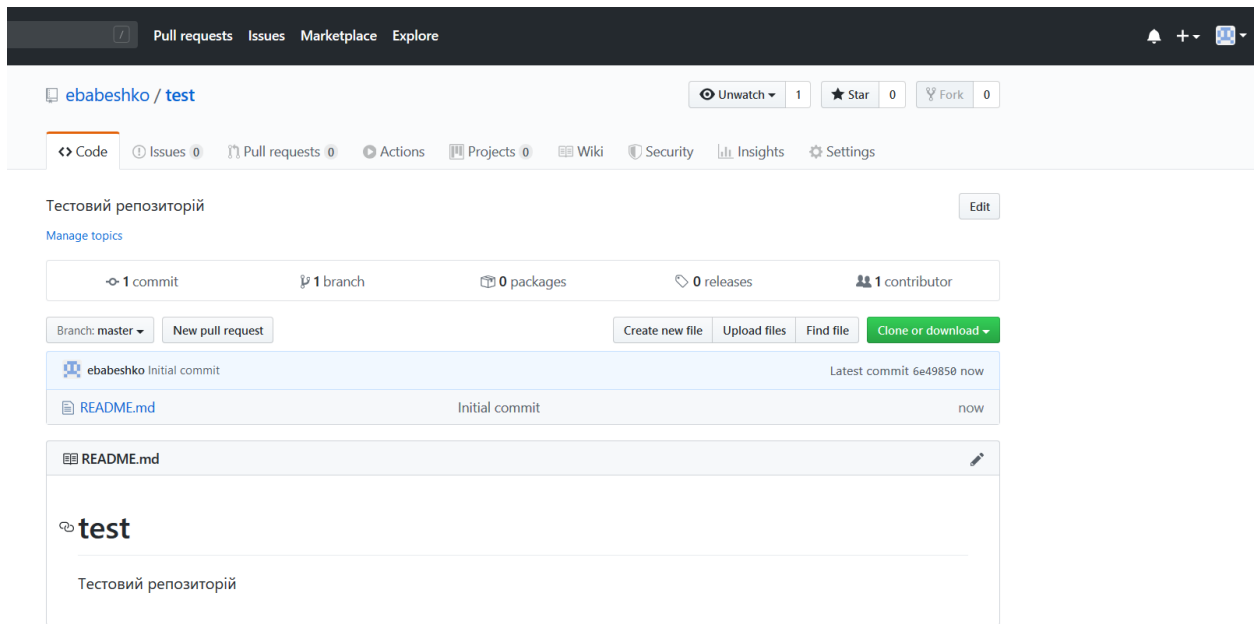


Рисунок 1.4 – Сторінка репозиторію

2. Робота з репозиторієм

2.1 Створення нового файлу

Для додавання нового файлу у репозиторій натисніть кнопку *Create new file* (рис. 2.1).

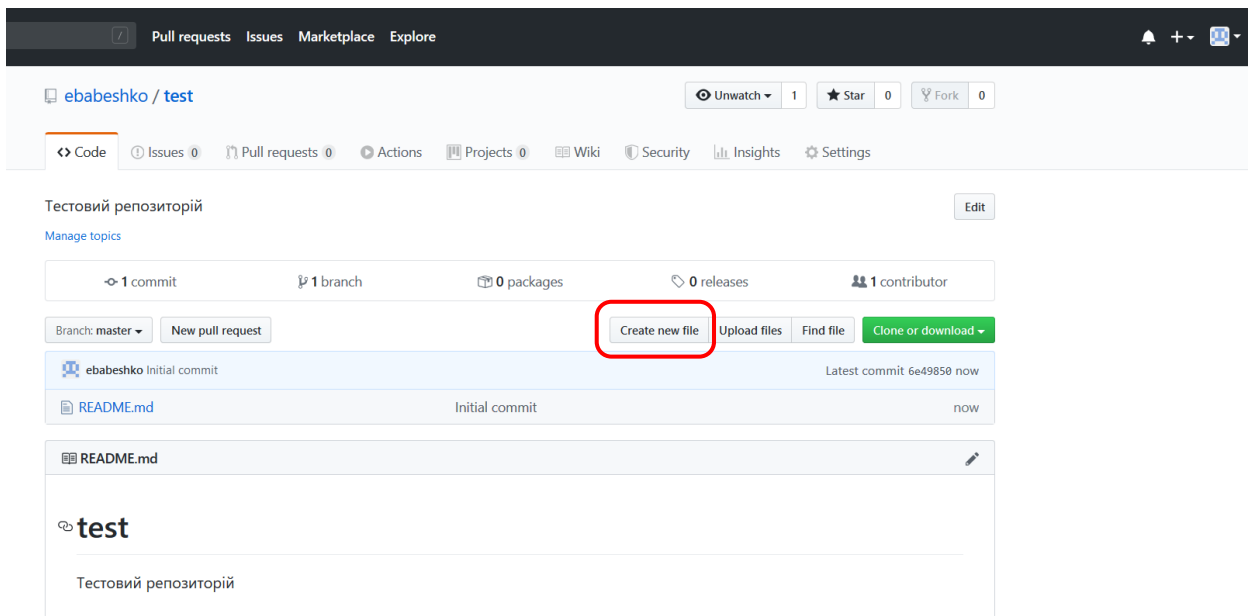


Рисунок 2.1 – Створення нового файлу у репозиторії

На сторінці створення файла введіть назву файла та відредагуйте його зміст (рис. 2.2).

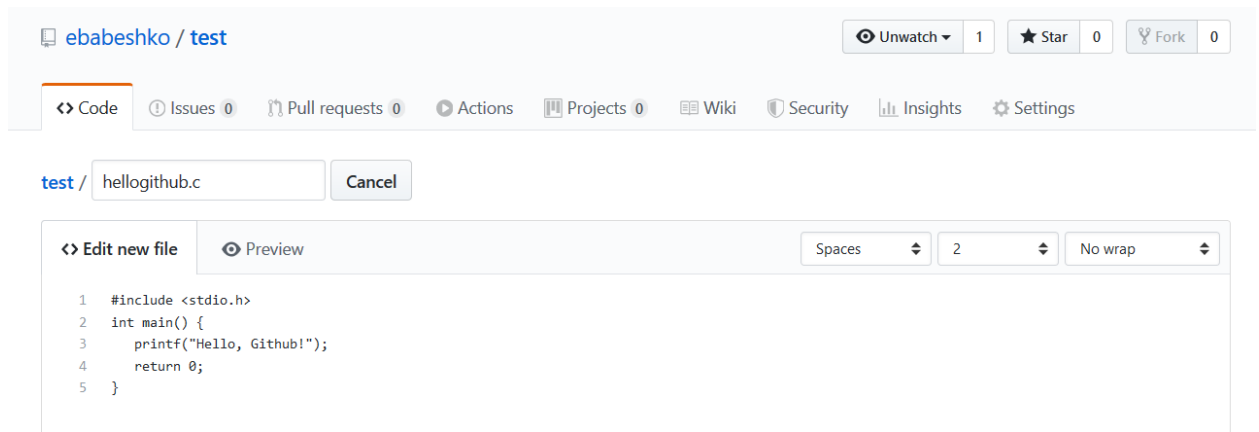


Рисунок 2.2 – Редагування файлу

Після чого натисніть на кнопку *Commit new file*, при цьому новий файл буде створено (рис. 2.3).

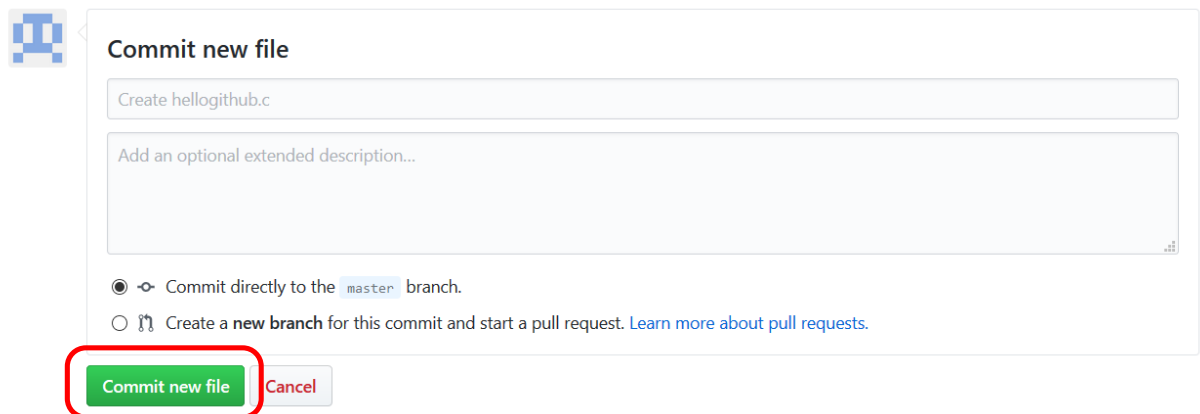


Рисунок 2.3 – Підтвердження змін у файлі

2.2 Завантаження файлів до репозиторію

Для завантаження файлів до репозиторію потрібно натиснути кнопку *Upload files* (рис. 2.4).

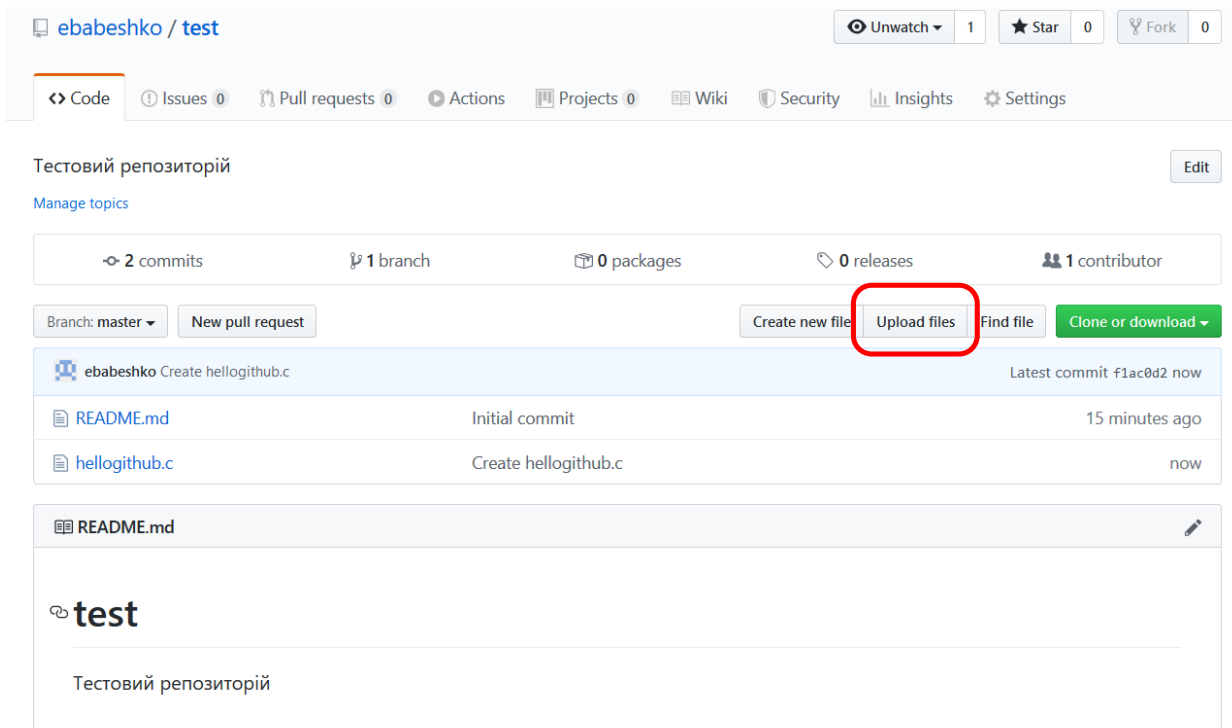


Рисунок 2.4 – Завантаження файлів до репозиторію

На наступній сторінці натисніть на посилання *Choose your files* (рис. 2.5).

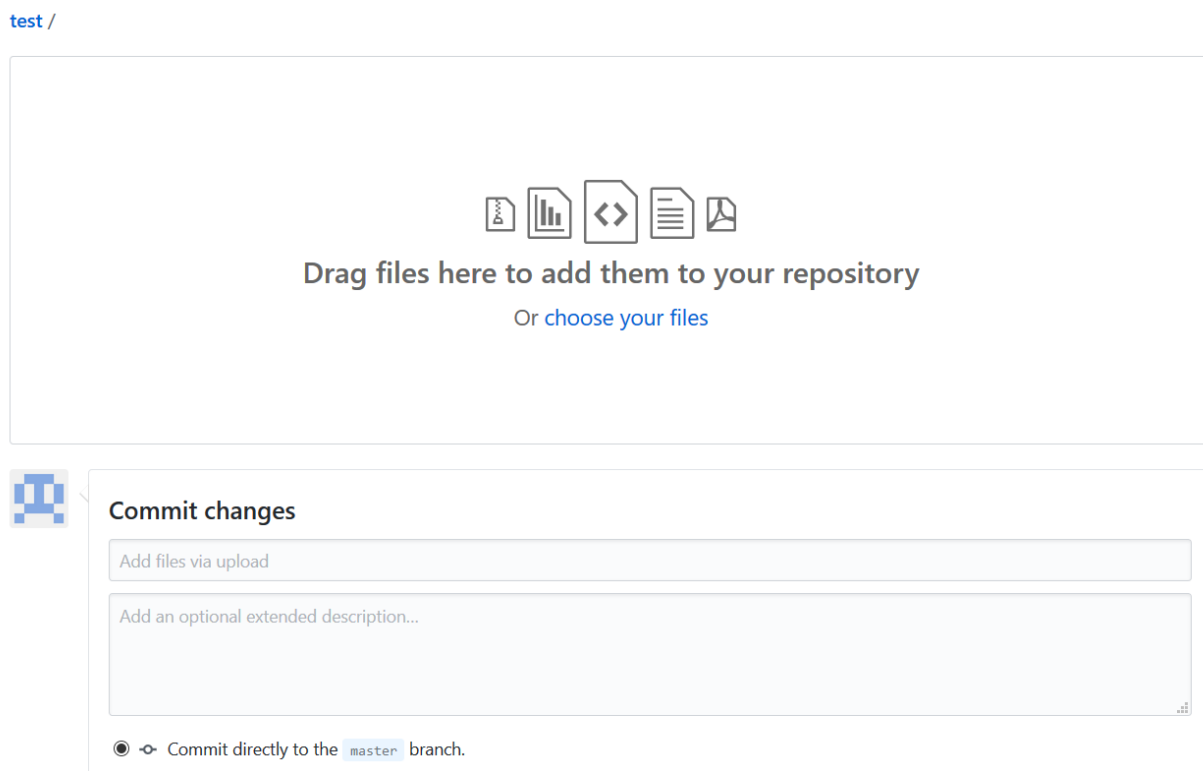


Рисунок 2.5 – Обрання файлів для завантаження

У вікні, що відкриється, оберіть файли для завантаження на Github і

натисніть на кнопку *Відкрити*, при цьому файли завантажаться до репозиторію (рисунок 2.6).

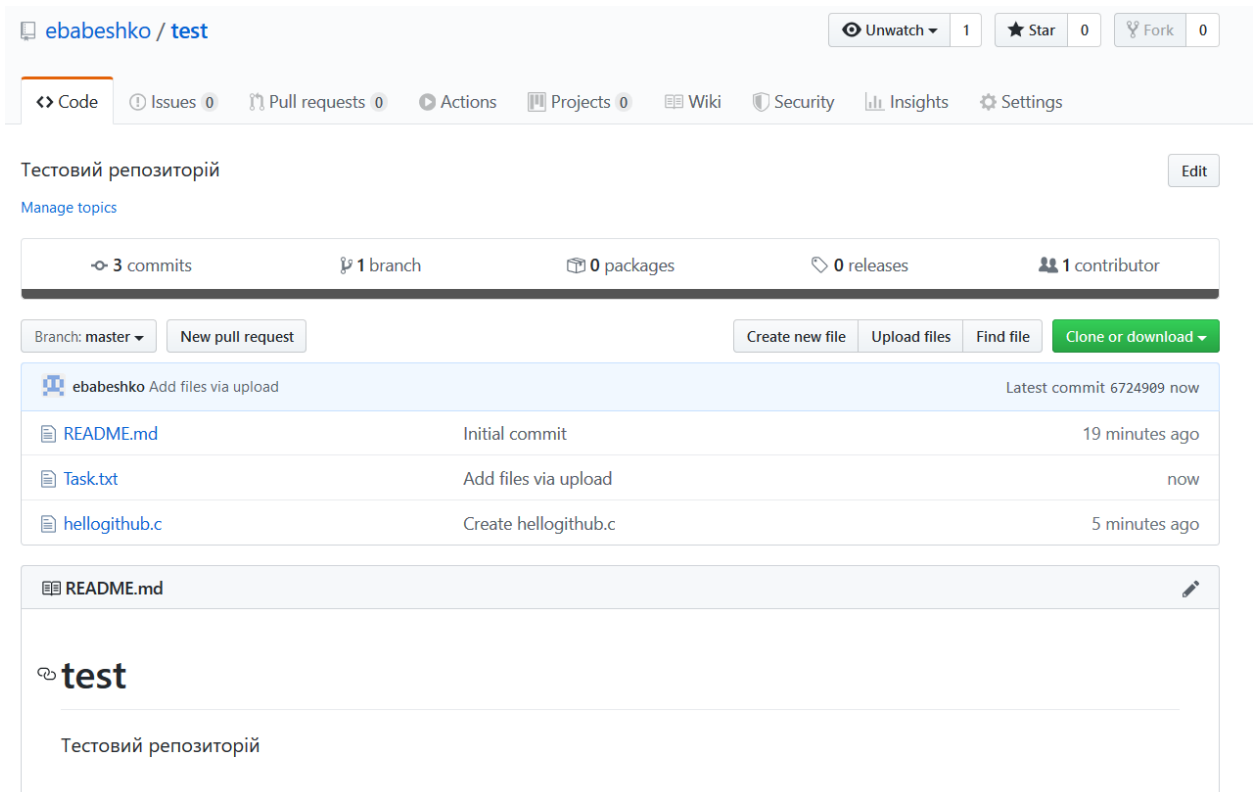


Рисунок 2.6 – Завантажений файл

2.3 Створення нової гілки

Гілки необхідні для створення різних версій проекту. Таким чином, за допомогою гілки Ви, нібито заморожуєте проект на якійсь стадії, і можете над ним працювати окремо, не втручаючись в основну гілку, котра називається master-гілка. Таких гілок можна створювати скільки завгодно, обмежень немає.

Для створення гілки потрібно натиснути на кнопку *Branch master* та у списку, що випадає, ввести ім'я нової гілки (рис. 2.7).

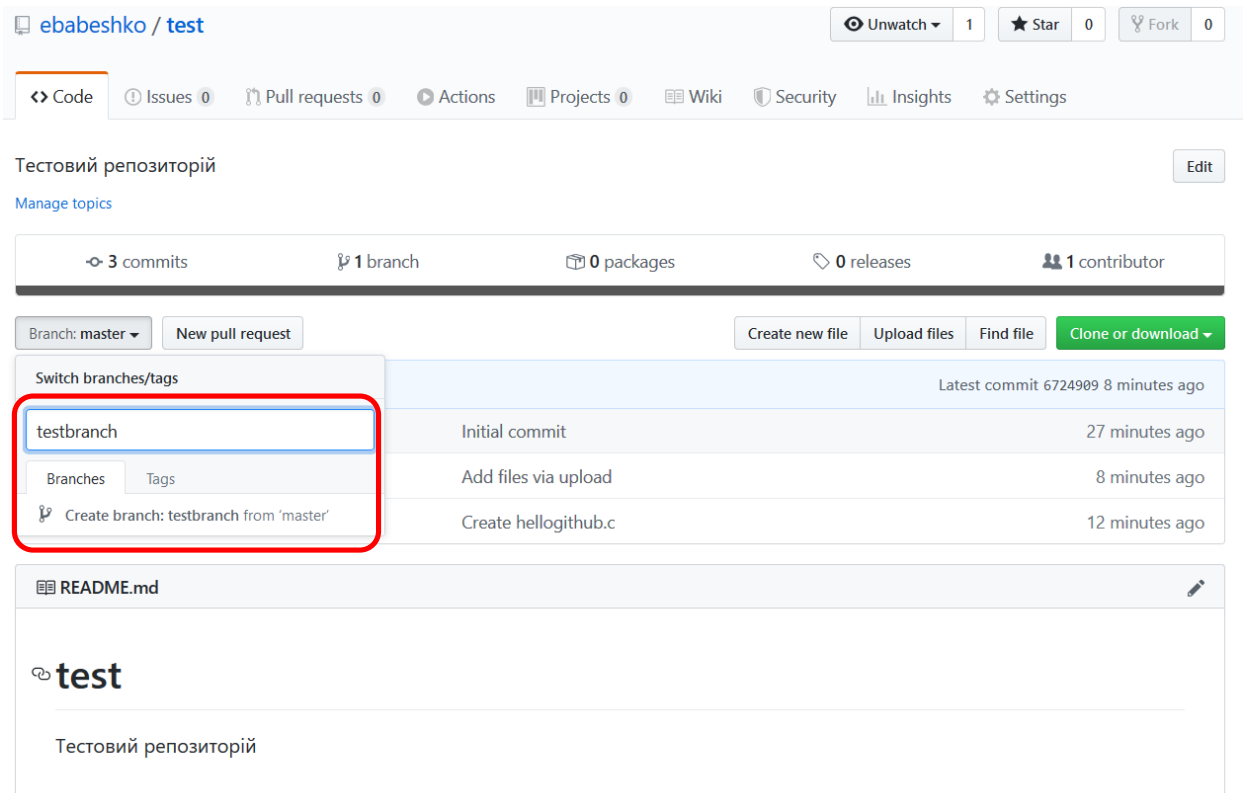


Рисунок 2.7 – Створення нової гілки

За такої умови ви отримаєте копію головної гілки і будете працювати вже у цій гілці. Отже, будь-які зміни, що ви зробите, не вплинуть на основну гілку доти, поки ви не розпочнете злиття гілок з основною гілкою.

2.4 Злиття гілок

Для того щоб зміни, зроблені у будь-якій гілці, синхронізувались з основною, потрібно синхронізувати або злити ці гілки. Для цього натисніть на кнопку *New pull request* (рис. 2.8).

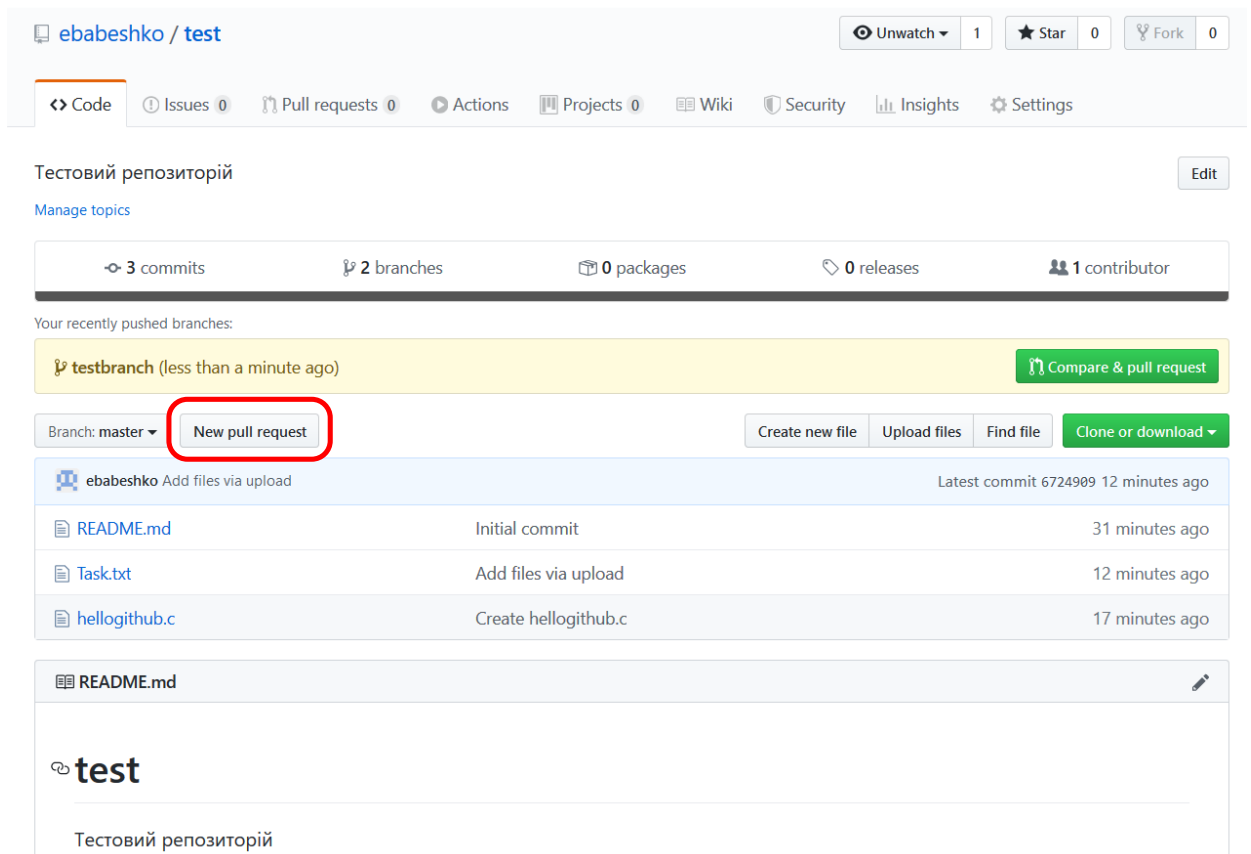


Рисунок 2.8 – Створення нової гілки

На сторінці запиту на злиття буде спочатку перевірено можливість злиття гілок. Після перевірки та можливості злиття натисніть на кнопку *Create pull request* (рис. 2.9).

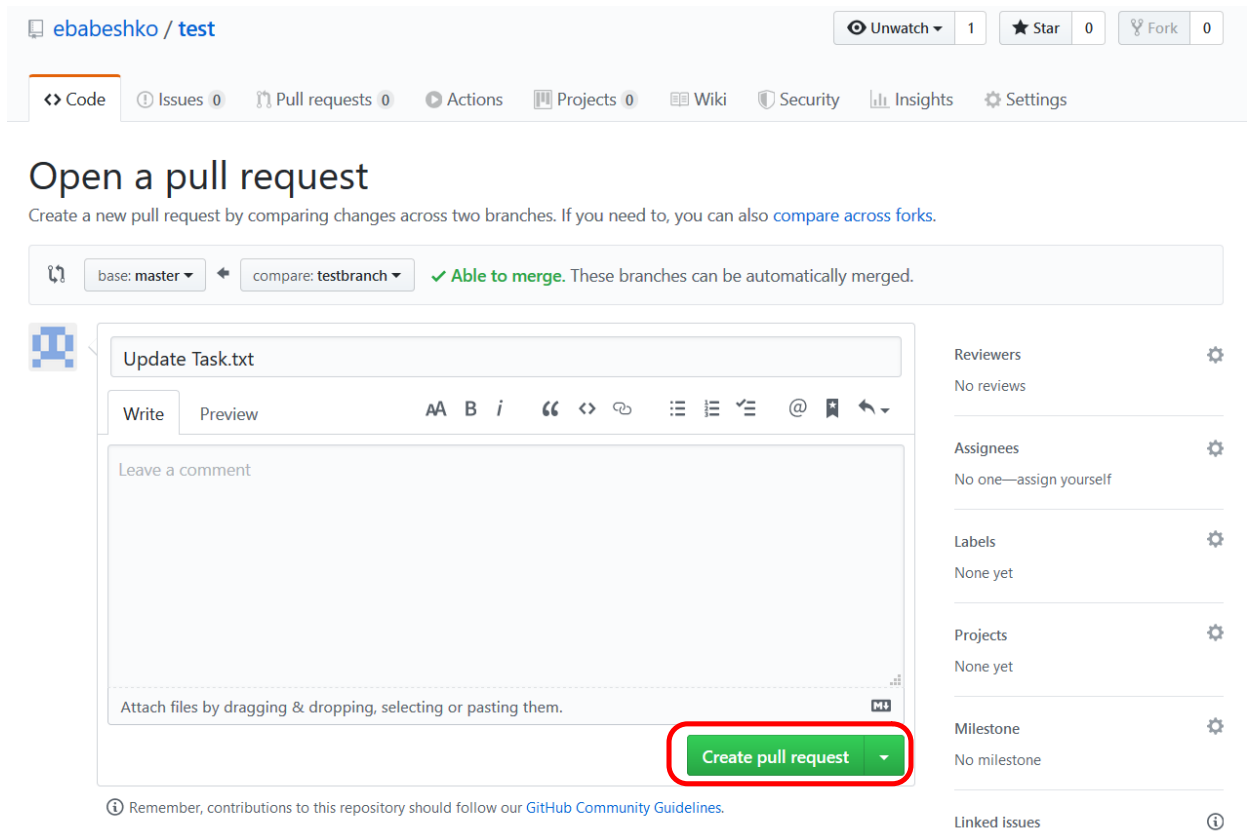


Рисунок 2.9 – Злиття гілок

На сторінці підтвердження злиття гілок буде також задіяно перевірку можливості злиття гілок і, якщо перевірка успішна, натисніть на кнопку *Merge pull request* (рис. 2.10).

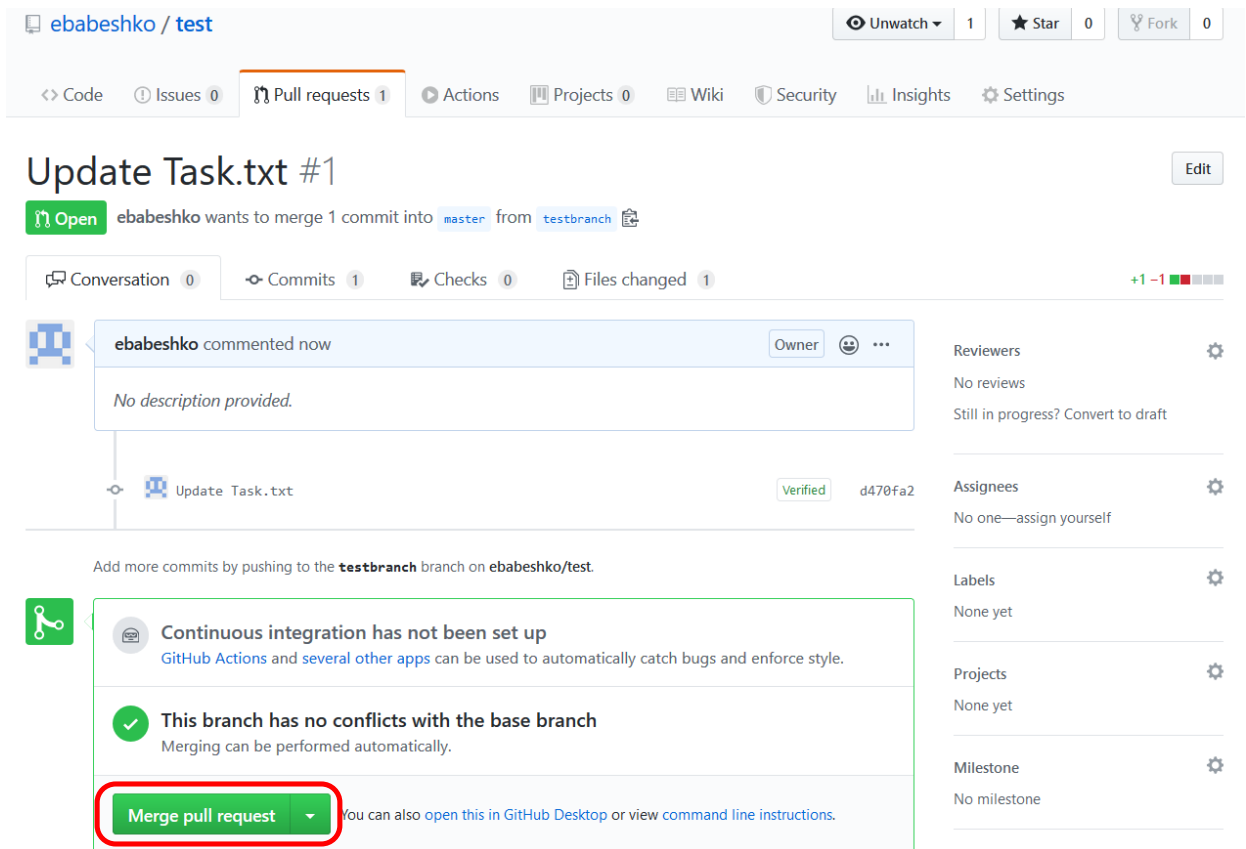


Рисунок 2.10 – Злиття гілок

На наступній сторінці натисніть на кнопку *Confirm merge* (рис. 2.11). Після цього розпочнеться механізм злиття гілок.

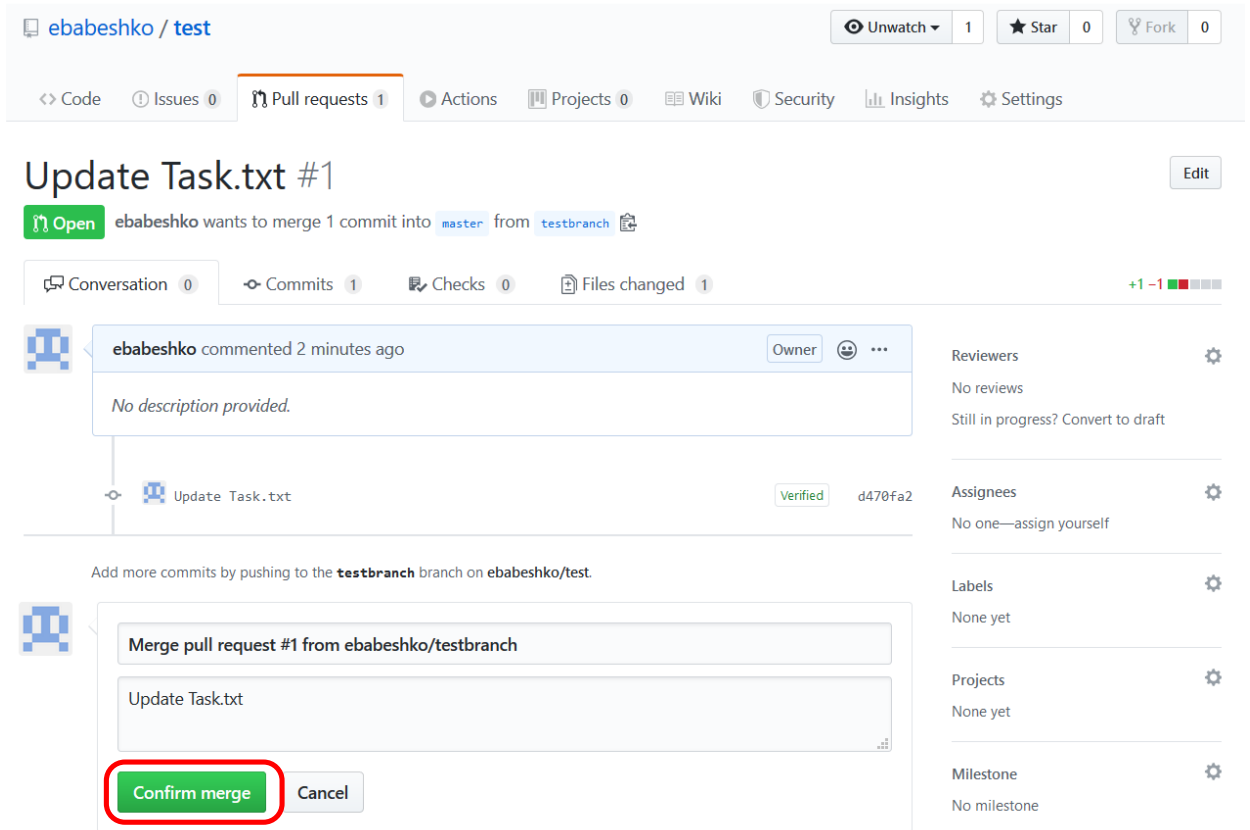


Рисунок 2.11 – Підтвердження злиття гілок

Після закінчення злиття є можливість видалити або залишити гілку, якщо вона ще потрібна (рис. 2.12).

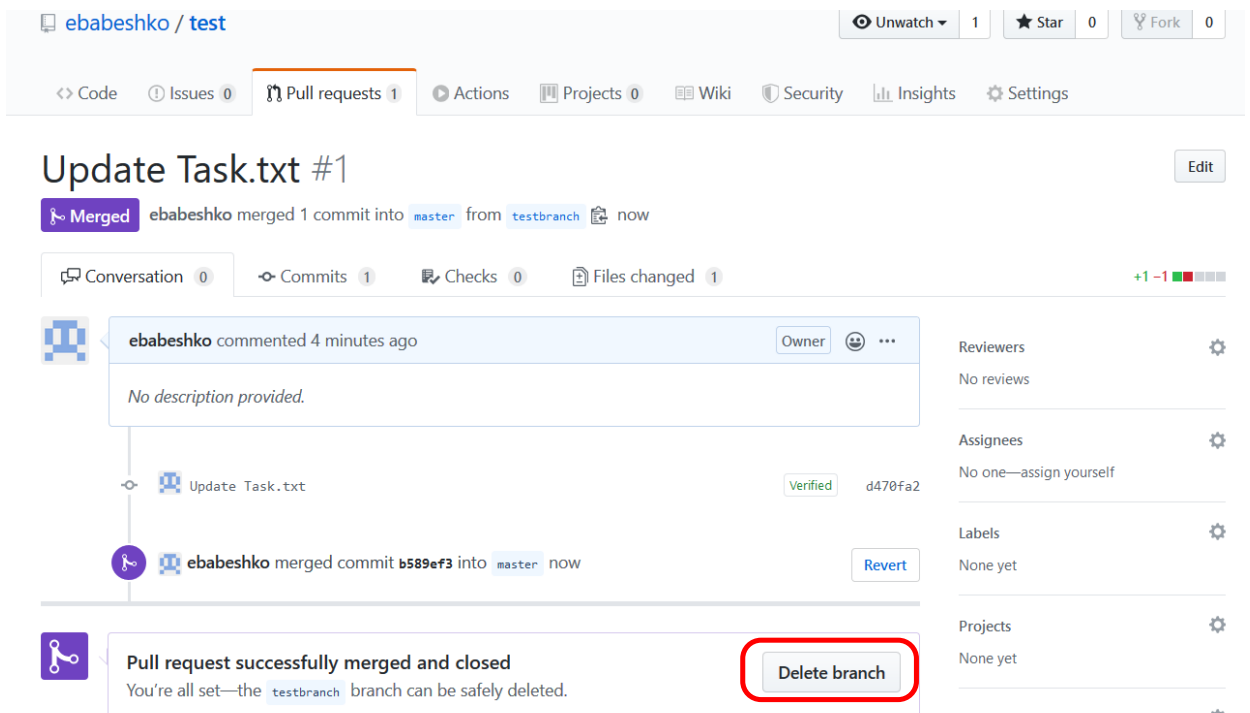


Рисунок 2.12 – Видалення гілки

2.5 Завантаження репозиторію

Для завантаження репозиторію на комп'ютер необхідно натиснути на кнопку *Clone or Download*, потім *Download ZIP* (рис. 2.13).

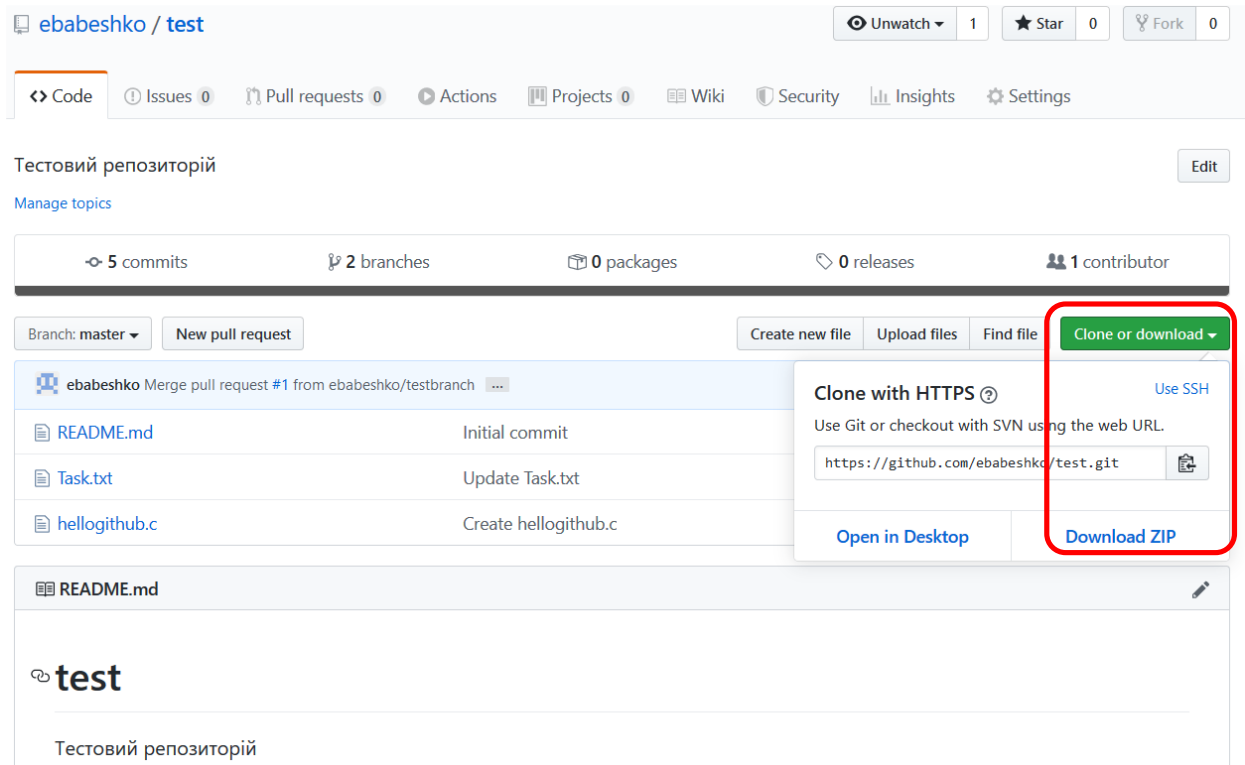


Рисунок 2.13 – Завантаження репозиторію

Водночас завантажиться архів у форматі Zip, який необхідно буде розархівувати.

2.6 Питання до розділу

1. Як створити новий репозиторій?
2. Для чого призначен репозиторій?
3. Як створити нову гілку у репозиторії?
4. Як злити гілки з основною гілкою master?
5. Як завантажити репозиторій до себе на комп'ютер?

2.7 Самостійна робота

1. Зареєструйтеся на сайті github.com.

2. Створіть репозиторій. Дайте назву репозиторію (наприклад My repository).
3. У новому репозиторії створіть новий файл.
4. Створіть нову гілку.
5. У новій гілці створіть файл.
6. Синхронізуйте нову гілку з основною гілкою master.
7. Завантажте репозиторій на комп'ютер.

3. Робота з консоллю git

3.1 Створення локального git-репозиторію

Для швидкої роботи з репозиторієм у системі Git можна використовувати один з багатьох доступних графічних інструментів або консольну утиліту Git. Розглянемо, як працювати з Git у консолі.

Оскільки вбудована консоль у систему Windows не підтримує утиліту Git, необхідно її встановити, завантаживши за посиланням:

<https://git-scm.com/download/win>

Послідовність встановлення показана на рисунках 3.1-3.2.

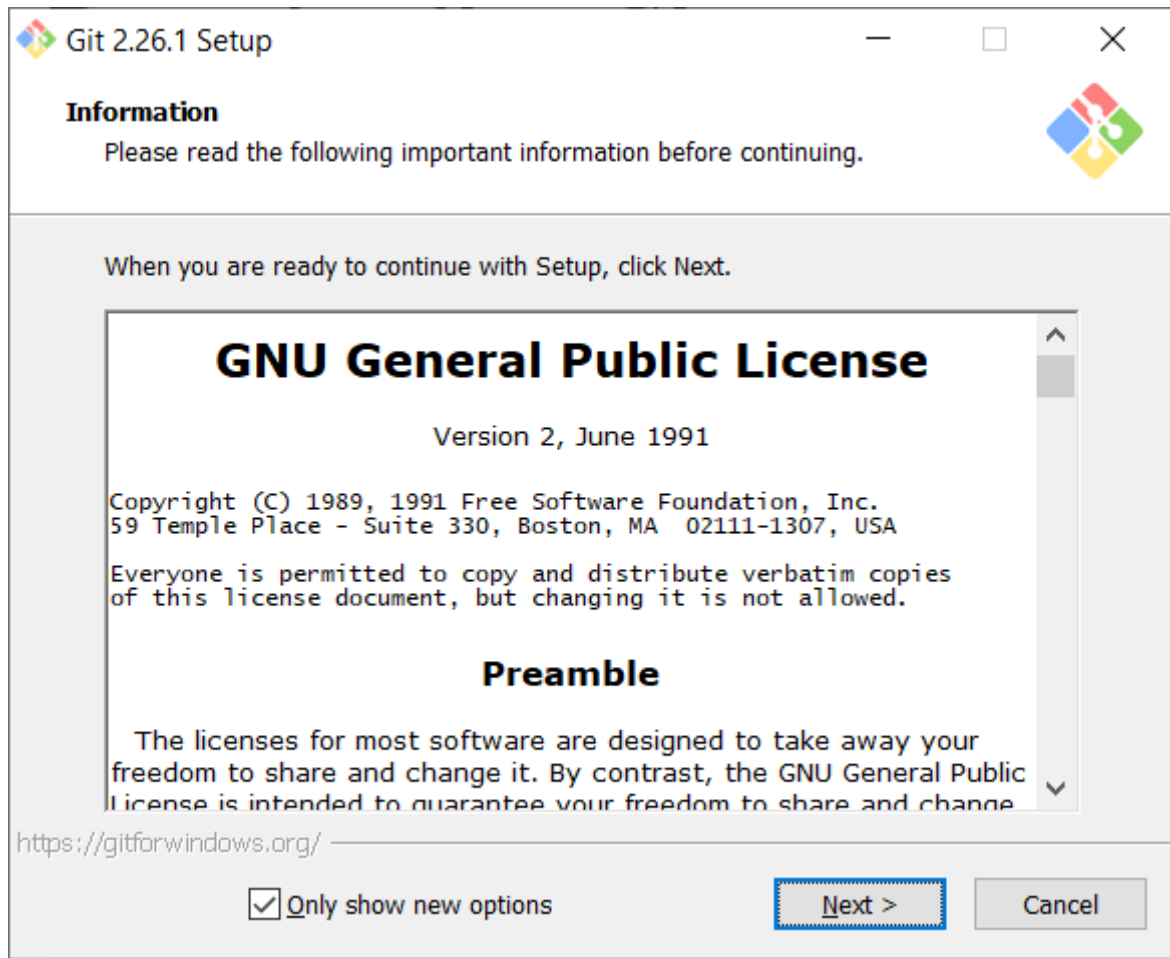


Рисунок 3.1 – Встановлення Git для Windows

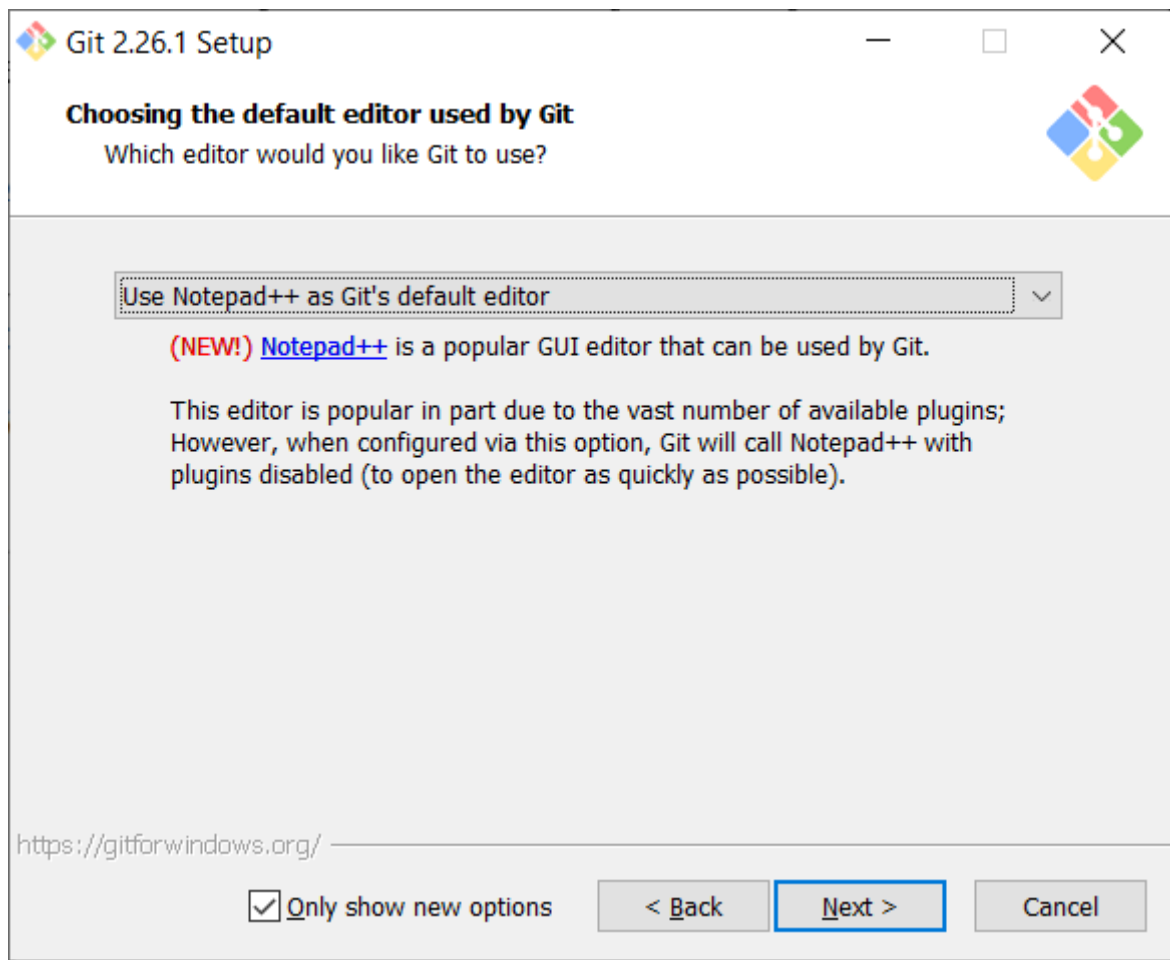


Рисунок 3.2 – Встановлення Git для Windows

Після завершення встановлення git можна перевірити встановлену версію, запустивши командний рядок та виконав команду (рис. 3.3):

git --version

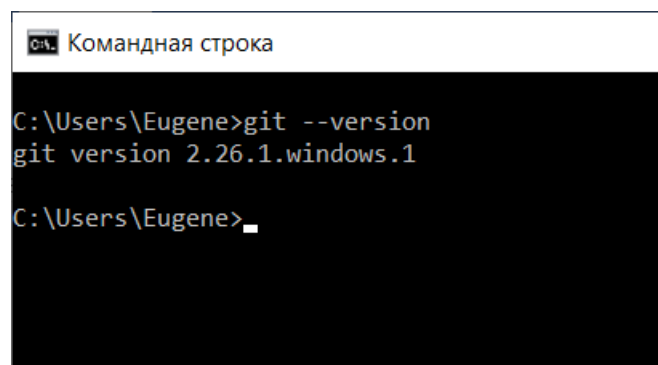


Рисунок 3.3 – Перевірка версії встановленого Git

Спочатку потрібно ввести дані для входу на сайт github.com. Для цього у консолі необхідно ввести команди:

git config --global user.name «Ваше ім'я»

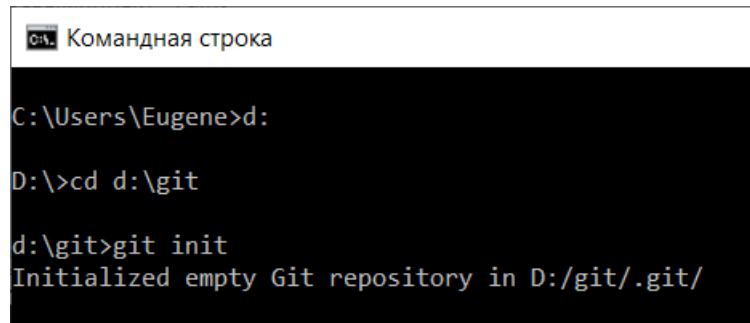
git config --global user.email«Ваш email».

Замість *Ваше ім'я* та *Ваш email* введіть відповідно своє ім'я та свій e-mail на сайті github.com.

Перейдіть до папки з проектом, для чого введіть команду:

cd «Папка з проектом»

Знаходячись у папці з проектом, введіть команду *git init*, таким чином у папці з проектом буде створено приховану папку *.git* (рис. 3.4).

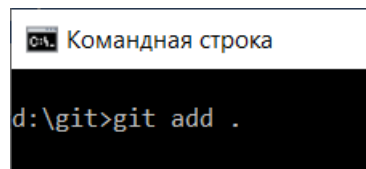


```
C:\Users\Eugene>d:
D:\>cd d:\git
d:\git>git init
Initialized empty Git repository in D:/git/.git/
```

Рисунок 3.4 – Створення репозиторію

Додайте усі файли проекту до репозиторію (рис. 3.5). Для цього введіть команду

git add .

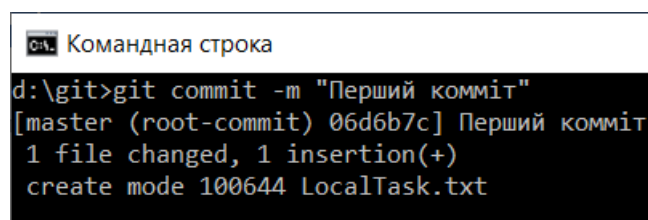


```
d:\git>git add .
```

Рисунок 3.5 – Додавання усіх файлів поточного каталогу до репозиторію

Підтвердіть зміни командою

git commit -m «Перший комміт».



```
d:\git>git commit -m "Перший комміт"
[master (root-commit) 06d6b7c] Перший комміт
1 file changed, 1 insertion(+)
create mode 100644 LocalTask.txt
```

Рисунок 3.6 – Підтвердження змін (комміт)

3.2 Завантаження локального репозиторію до віддаленого

Для такого завантаження необхідно додати віддалений репозиторій, для чого введіть команду

```
git remote add rep https://github.com/ebabeshko/test2.git,
```

де замість адреси `https://github.com/ebabeshko/test2.git`, потрібно ввести свою, подивитись її можна на сторінці репозиторію (рис. 3.7).

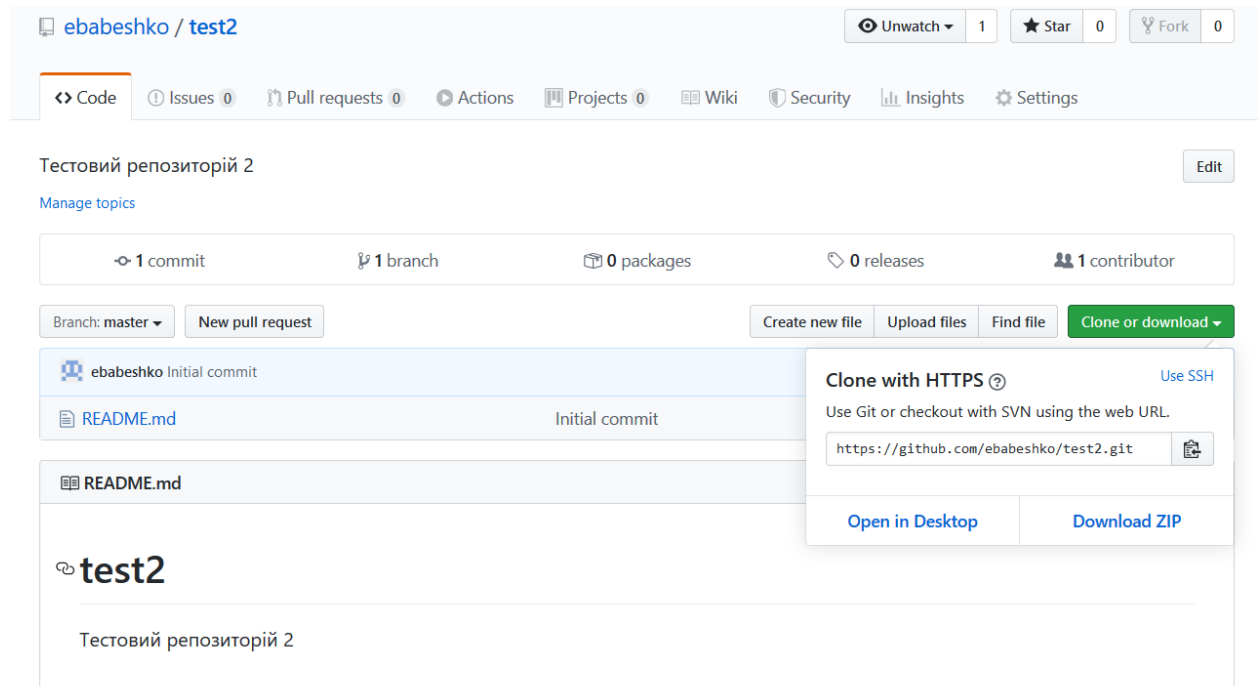


Рисунок 3.7 – Перегляд адреси репозиторію

Перевірити усі віддалені репозиторії можна командою (рис. 3.8)

```
git remote -v
```

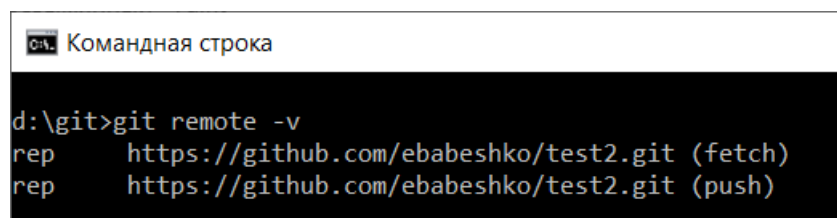
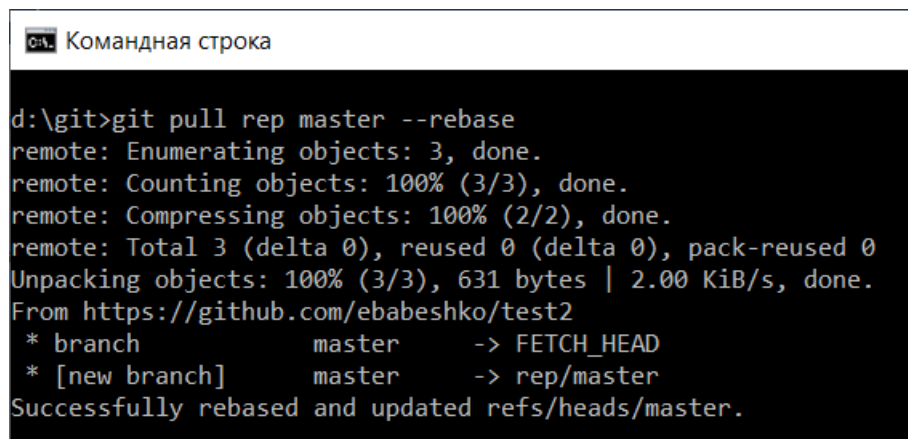


Рисунок 3.8 – Перевірка віддалених репозиторіїв

Далі завантажте усі зміни з віддаленого репозиторію командою (рис. 3.9)

```
git pull rep master -rebase
```



```
C:\> Командная строка

d:\git>git pull rep master --rebase
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 631 bytes | 2.00 KiB/s, done.
From https://github.com/ebabeshko/test2
* branch          master      -> FETCH_HEAD
* [new branch]     master      -> rep/master
Successfully rebased and updated refs/heads/master.
```

Рисунок 3.9 – Завантаження змін до локального репозиторію

Після цієї команди можна завантажити усі зміни до віддаленого репозиторію командою

git push rep master

У вікні, що з'явиться, необхідно ввести авторизаційні дані (рис. 3.10, 3.11)

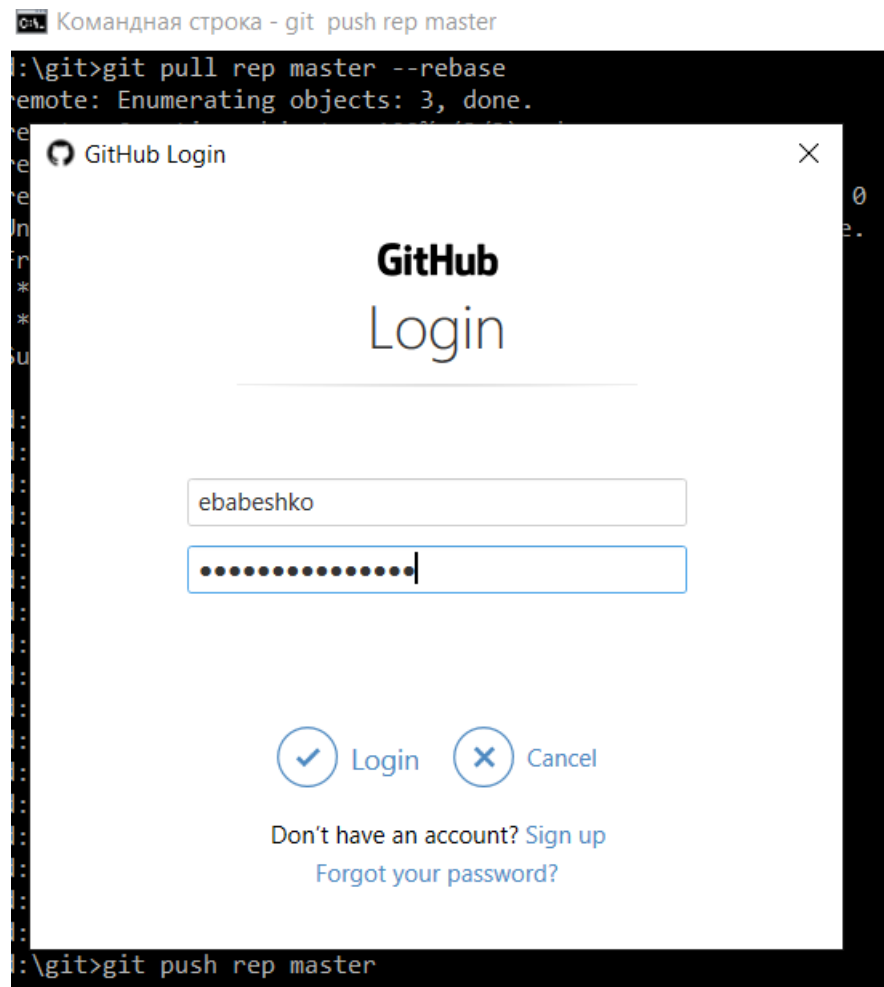


Рисунок 3.10 – Завантаження змін до віддаленого репозиторію

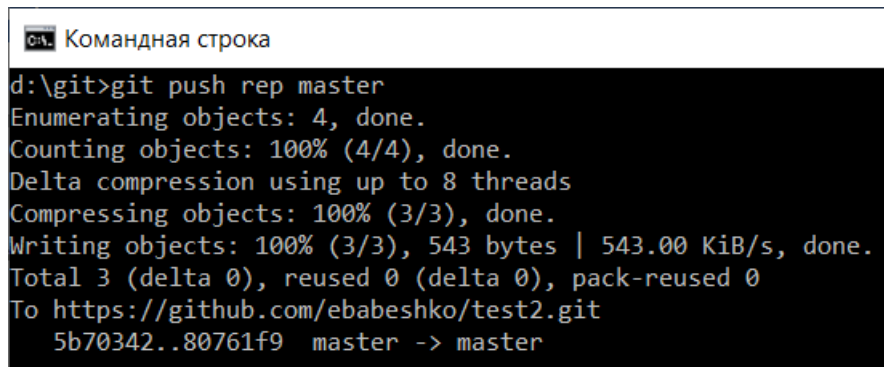


Рисунок 3.11 – Завантаження змін з локального до віддаленого репозиторію

3.3 Додавання іншого користувача до репозиторію

Для того щоб додати іншого користувача для спільної роботи над проектом, необхідно обрати репозиторій, до якого ви бажаєте запросити іншого користувача, перейти на вкладку *Settings* та з панелі ліворуч обрати *Manage access* (рис. 3.12).

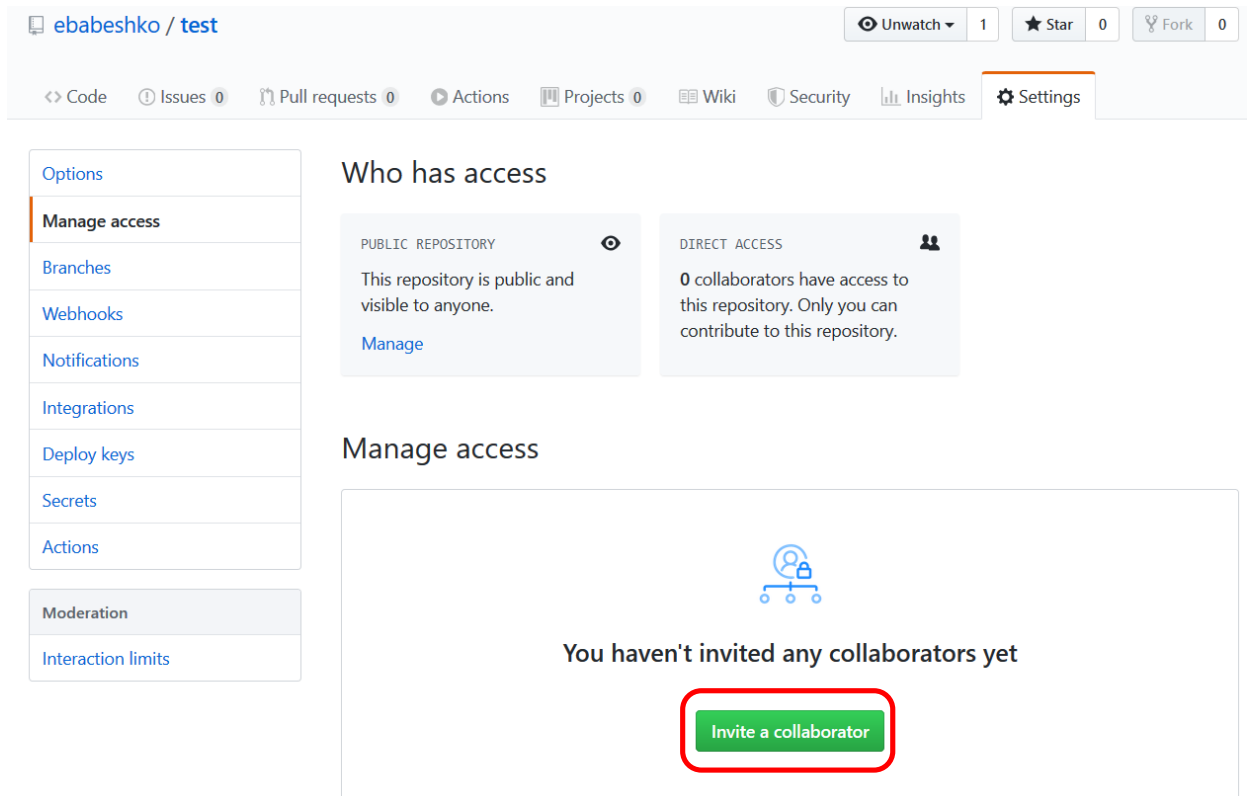


Рисунок 3.12 – Налаштування доступу до репозиторію

У текстовому полі введіть логін (рис. 3.13) або e-mail (рис. 3.14) користувача, який залучається до участі у проекті, та натисніть на кнопку *Add* (рис. 3.15).

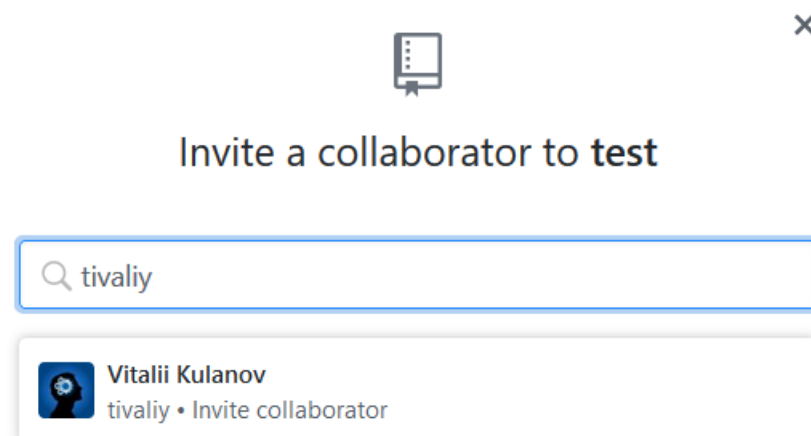


Рисунок 3.13 – Додавання користувача за логіном

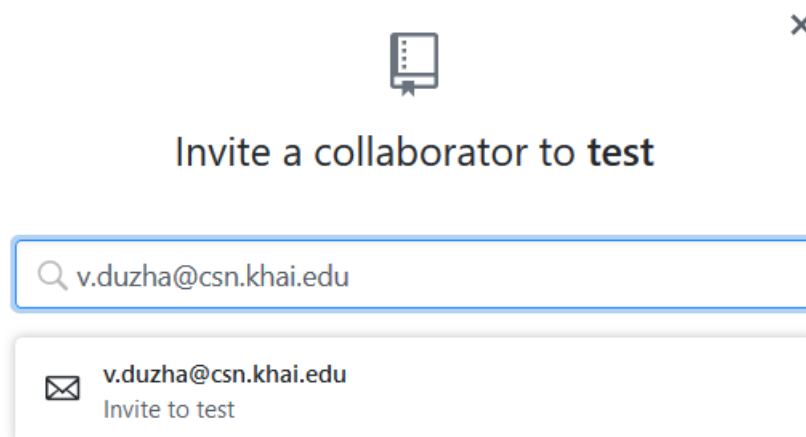


Рисунок 3.14 – Додавання користувача за e-mail

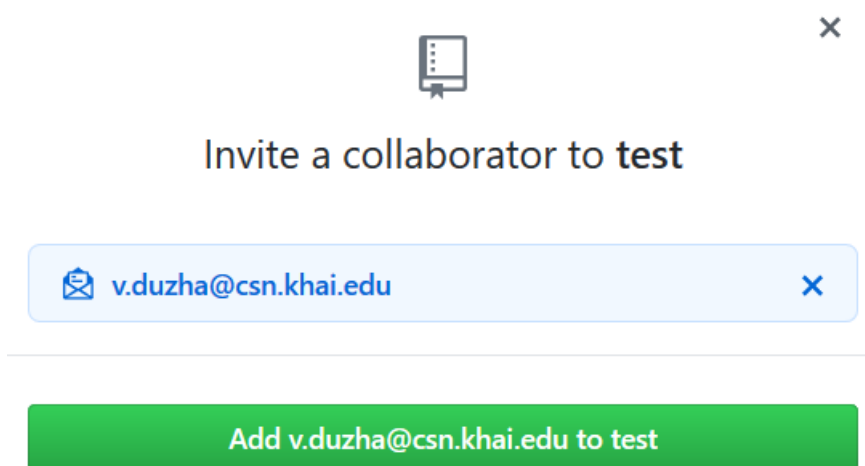


Рисунок 3.15 – Підтвердження додавання користувача

Користувачеві, якого ви запросили, на електрону пошту буде надіслано лист із запрошенням (рис. 3.16).

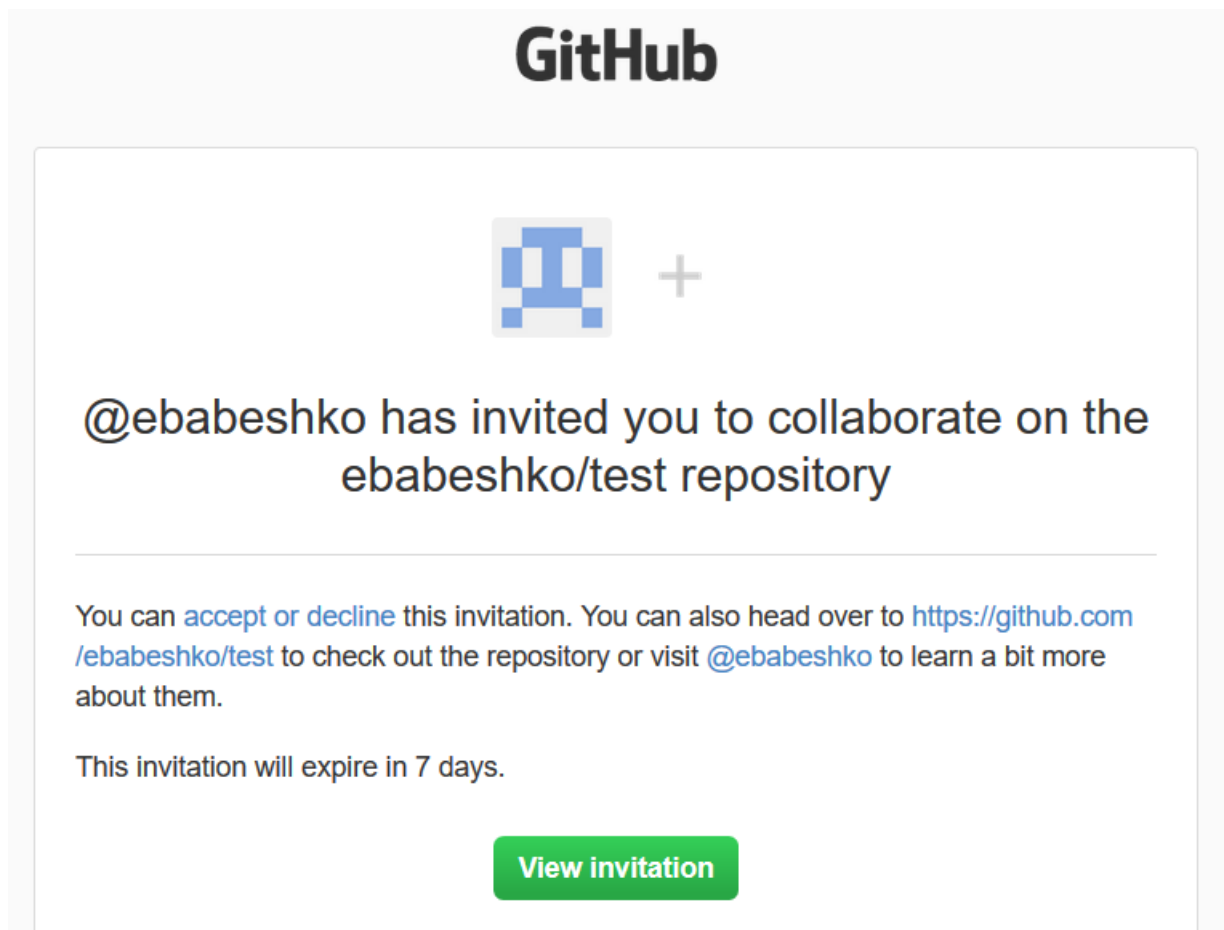


Рисунок 3.16 – Запрошення

У листі необхідно перейти за посиланням *accept or decline* та прийняти або відхилити запрошення (рис. 3.17).

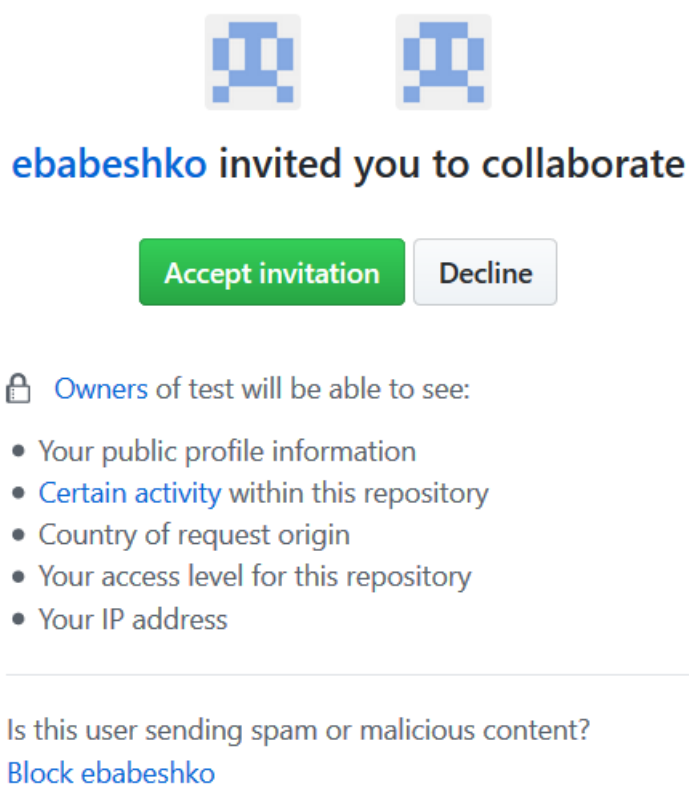


Рисунок 3.17 – Підтвердження або відхилення запрошення

Після натискання кнопки *Accept invitation* відбудеться перенаправлення на сторінку з репозиторієм і з'явиться можливість редагувати файли та папки цього репозиторію (рис. 3.18).

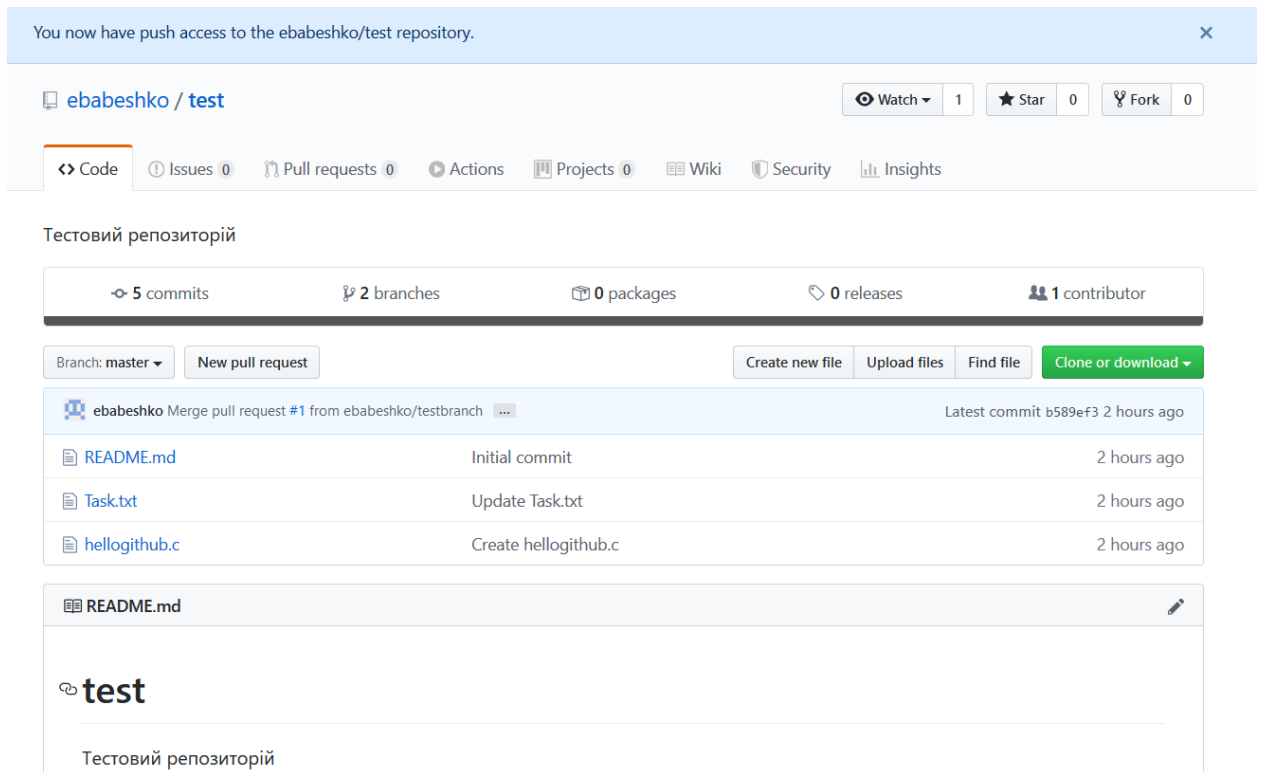
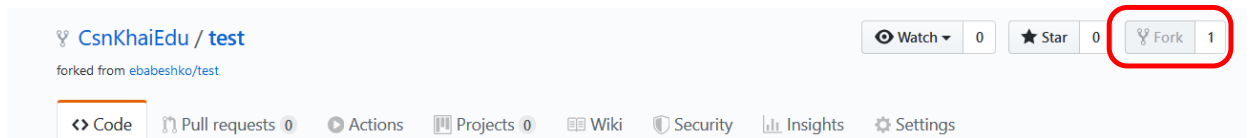


Рисунок 3.18 – Доступ до репозиторію

Для додавання цього репозиторію до свого профілю є можливість його скопіювати або форкнути, для чого необхідно натиснути кнопку *Fork* на панелі вгорі. Доступна копія цього репозиторію буде у власному профілі.



Forking ebabeshko/test

It should only take a few seconds.



Рисунок 3.19 – Створення форку

3.4 Клонування репозиторіїв

Для клонування репозиторію необхідно створити каталог, куди його буде клоновано. Наприклад, створюємо каталог *LocalRepo* у каталозі *D:\Git* і туди клонуємо репозиторій.

Запустіть консоль. Перейдіть у каталог *LocalRepo*.

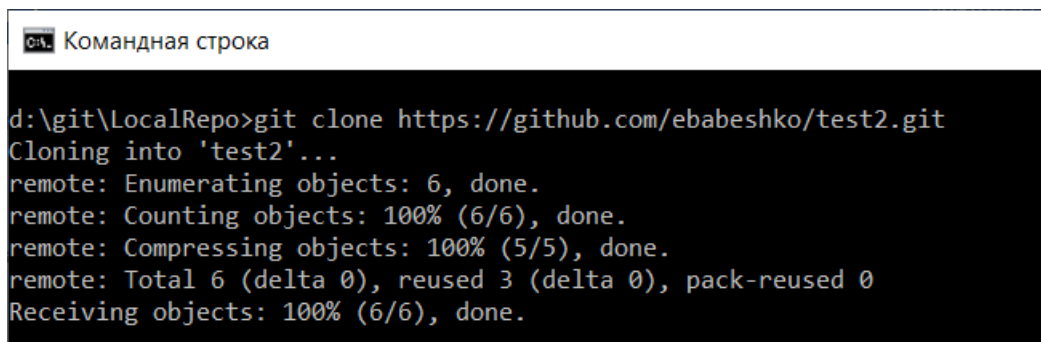
Для цього у консолі введіть команду

```
cd d:/Git/LocalRepo
```

Знаходячись у каталозі *LocalRepo*, введіть команду

```
git clone https://github.com/ebabeshko/test2.git
```

(замість <https://github.com/ebabeshko/test2.git> вкажіть адресу репозиторію, який бажаєте клонувати).



```
d:\git\LocalRepo>git clone https://github.com/ebabeshko/test2.git
Cloning into 'test2'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
```

Рисунок 3.20 – Клонування репозиторію

3.5 Питання для закріплення

1. Як у консолі створити локальний git-репозиторій?
2. Як додати віддалений репозиторій?
3. Як завантажити файли до віддаленого репозиторію?
4. Чим клонування репозиторію відрізняється від форка репозиторію?

3.6 Вправи для самостійної роботи

1. Запустіть консоль git. Введіть дані для доступу на сайт github.com.

2. Перейдіть у папку з проектом та ініціалізуйте локальний репозиторій git
3. Додайте файли проекту до репозиторію.
4. Підтвердіть зміни, зробивши commit.
5. Додайте свій видалений репозиторій, котрий був створений у попередній вправі.
6. Завантажте усі файли проекту до віддаленого репозиторія.
7. Обміняйтесь своїми логіном з іншим користувачем, додайте його до вашого репозиторію.
8. Клонуйте репозиторій, до якого вас додав інший користувач