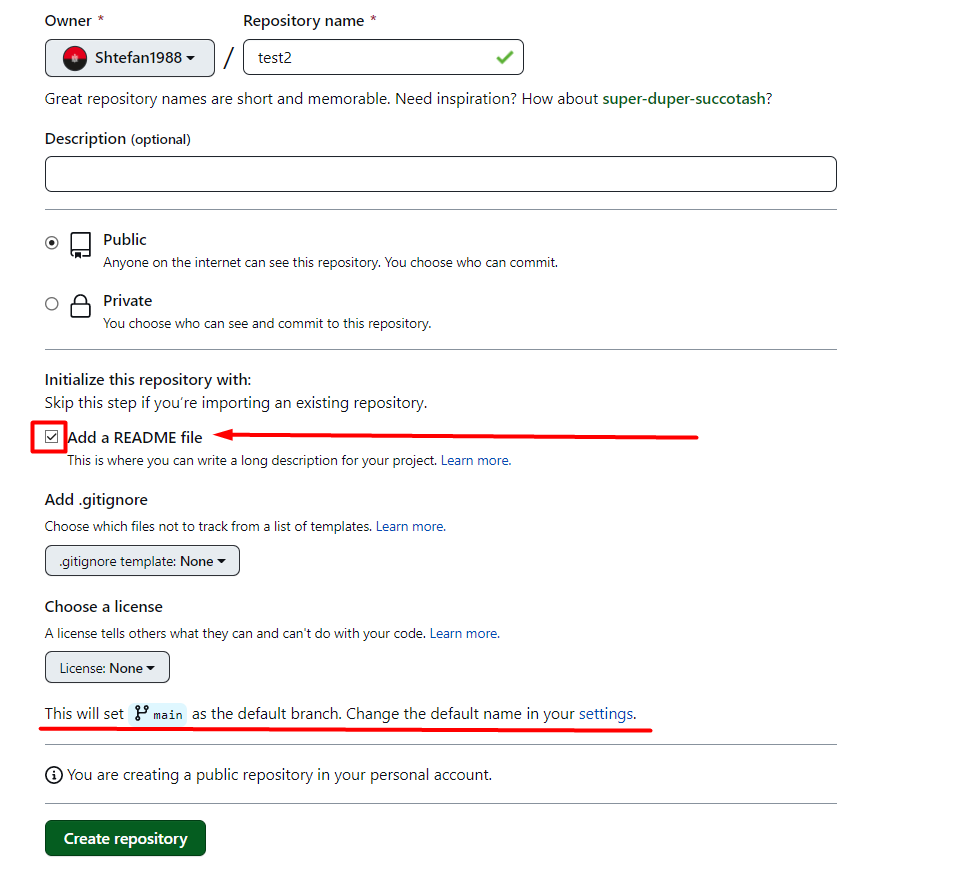
**Робота з репозиторієм**

**(для власного репозиторію)**

**GitGui (візуальна)**

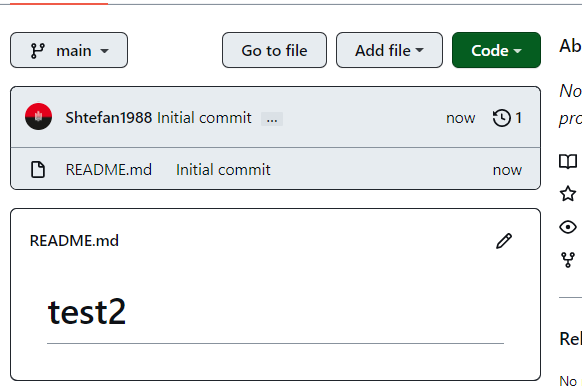
**Крок 1й – створити віддалений репозиторій на GitHub**



Вже цього разу ми додаємо **REAME.MD**.

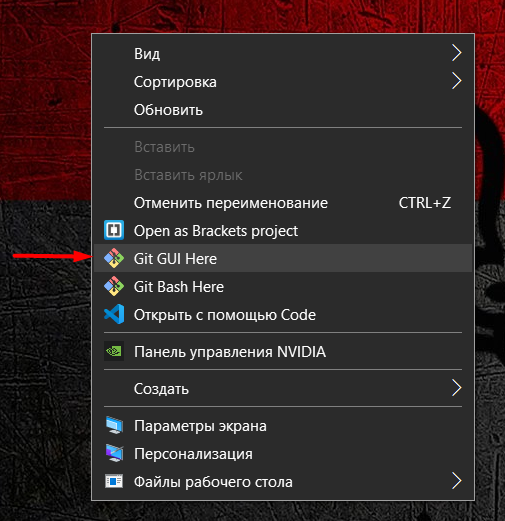
**!!!**У віддаленому репозиторію GitHub вже є гілка **main** , тому при копіюванні віддаленого репозиторію на ПК, нам потрібно буде у **Git Gui** одразу перейменувати гілку **master** на **main** .

Створюємо:



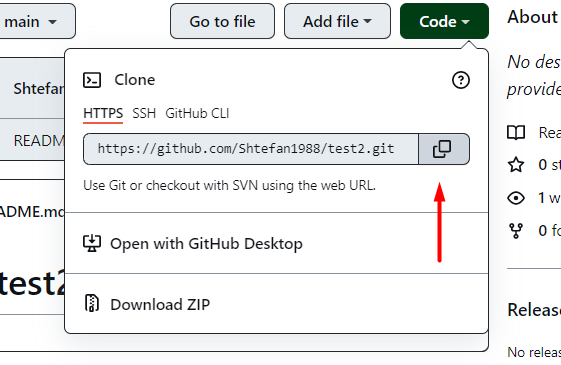
**Крок 2й – створити клон віддаленого репозиторію на ПК**

У будь-якій папці клацнути ПКМ та обрати **Git Gui Here** :



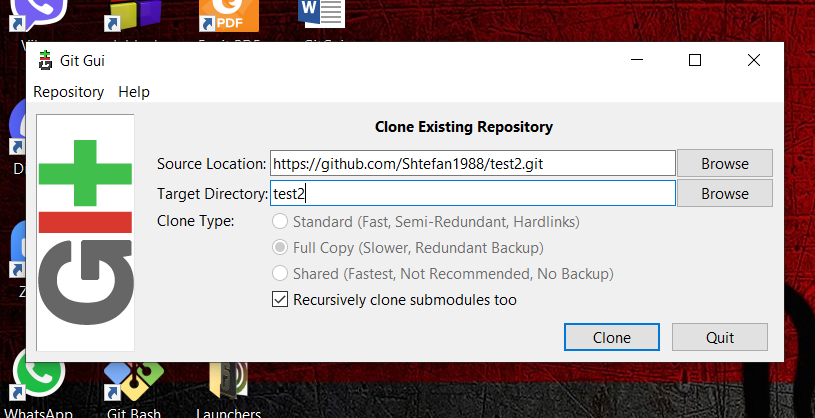
У вікні Git Gui обрати **Clone Existing Repository**.

Переходимо у GitHub та копіюємо посилання на наш репозиторій:

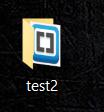


У відкритому вікні **Git Gui** у **Source Location** вставляємо наше посилання та у **Target Directory** створюємо назву папки для локального репозиторію:

**!!!** Якщо посилання не вставляється через **Ctrl+V**, тоді перейдіть на англійську мову і спробуйте ще раз. У **Git Gui Ctrl+V** не працює, якщо ви на російській або на українській розкладці.

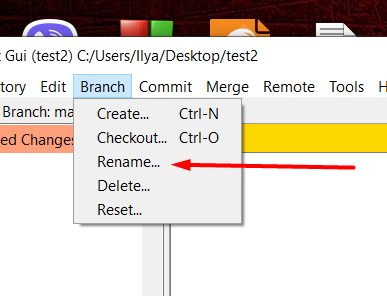


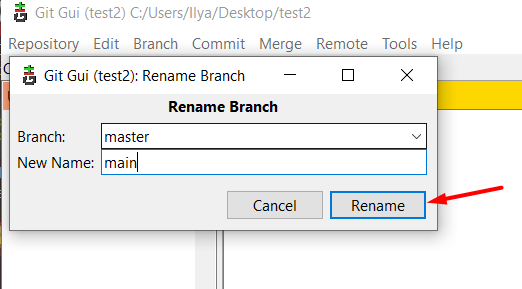
Натискаємо **Clone**:



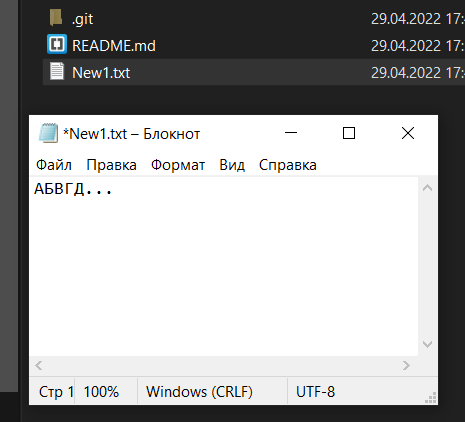
**Крок 3й – робота з гілками в Git Gui**

Перейменовуємо у Git Gui гілку **master** на **main** .



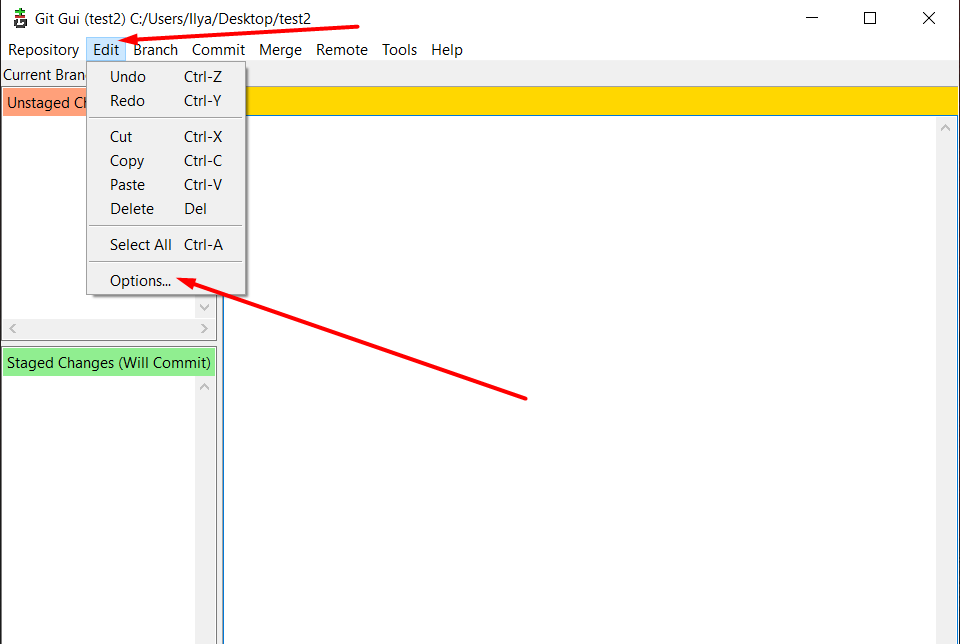


Створюємо новий текстовий файл і в ньому прописуємо будь-який текст **кирилицею**:

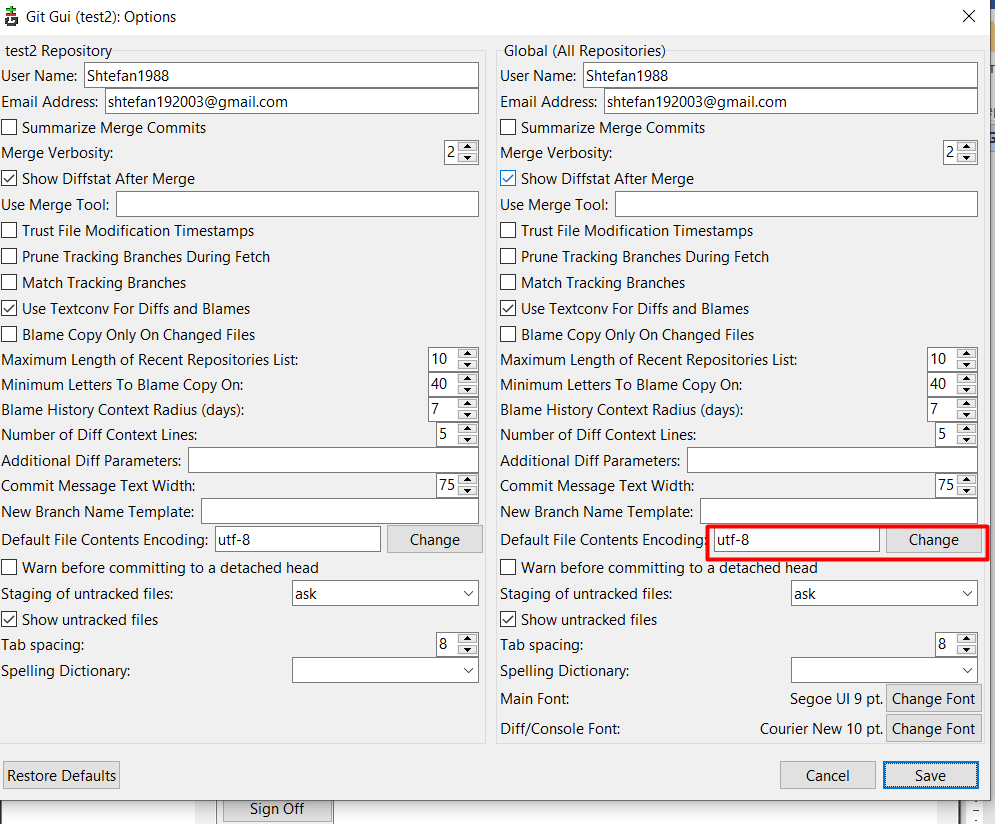


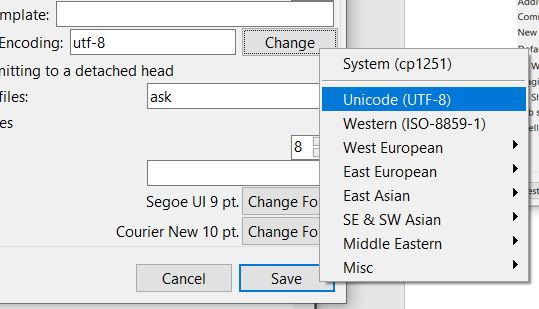
Зберігаємо зміни.

Переходимо у налаштування Git Gui:



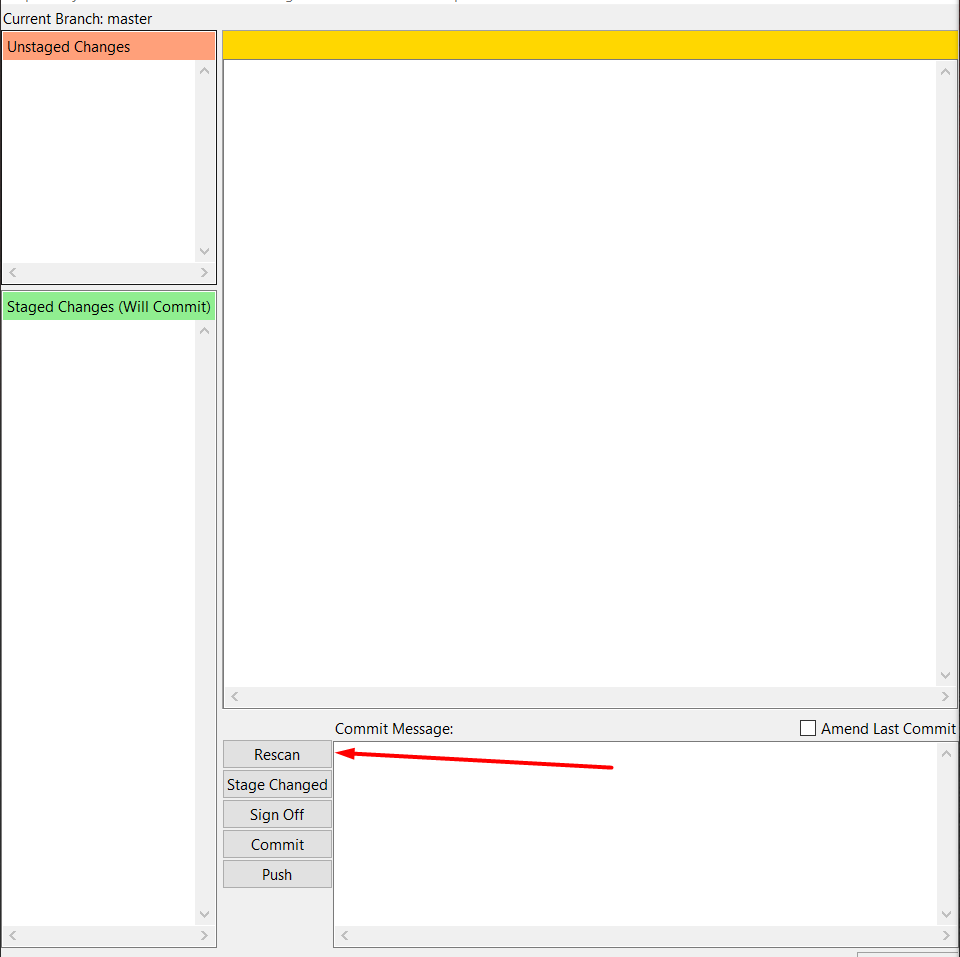
Переконайтеся, що у вас формат **utf-8**:



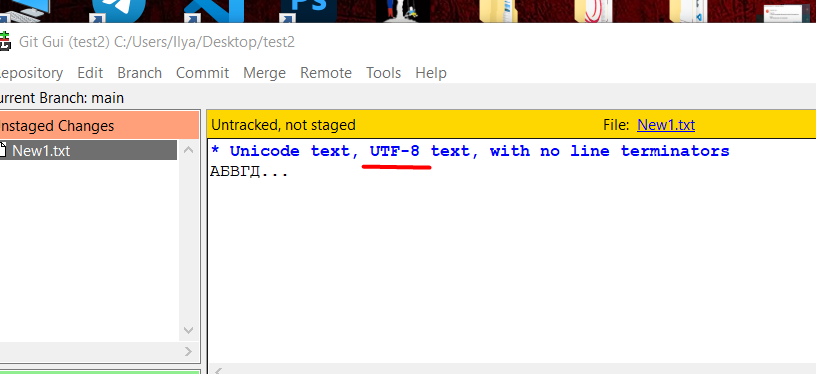


Зберігаємо налаштування.

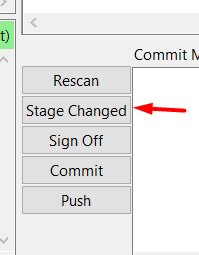
Щоб у **Git Gui** додати наш текстовий файл, потрібно спочатку натиснути на кнопку “**Rescan**”:

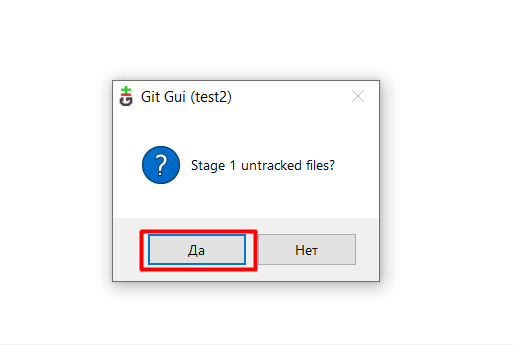


Після чого він з’явиться у відповідному вікні:

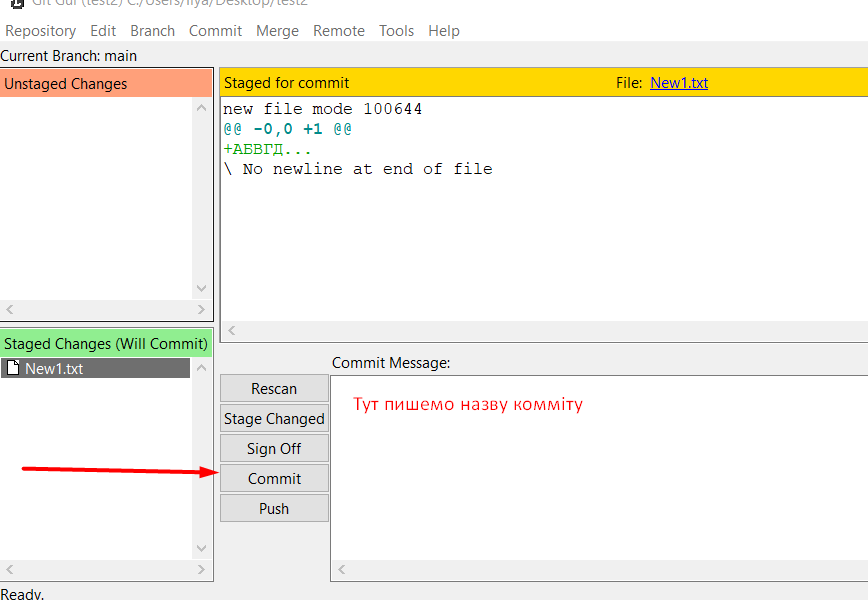


Тепер додаємо його. Потрібно або натиснути кнопку “**Stage Changed**”або двічі по самому файлу. Це така ж сама дія, як **git add** у консолі:

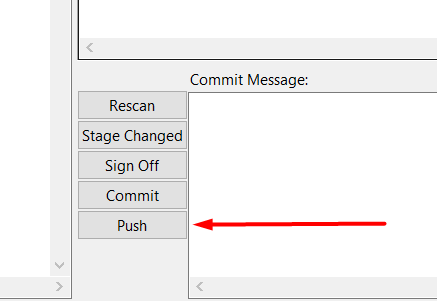




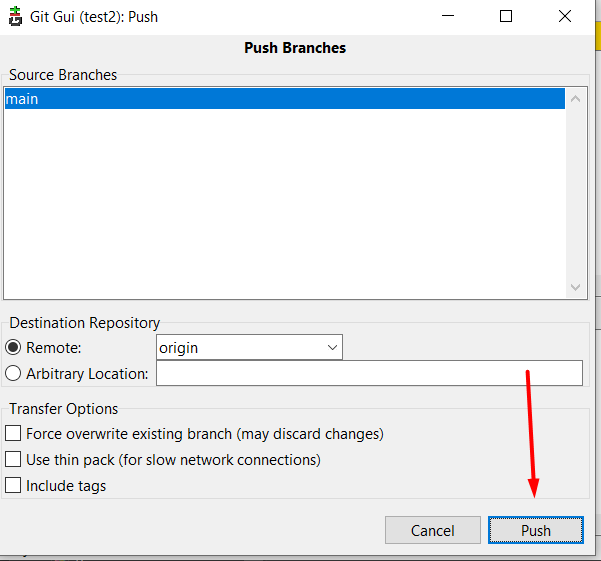
Тепер пишемо назву комміту та робимо комміт. Потрібно натиснути кнопку “**Commit**” (**git commit -m** у консолі):



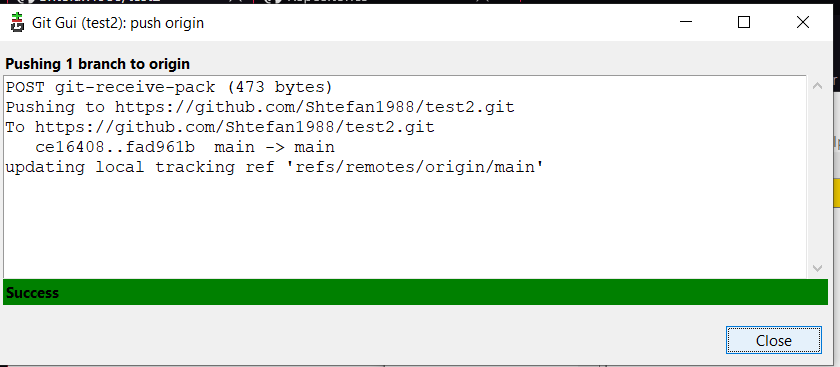
Після того як закомітили робимо пуш (**git push origin –u** у консолі):



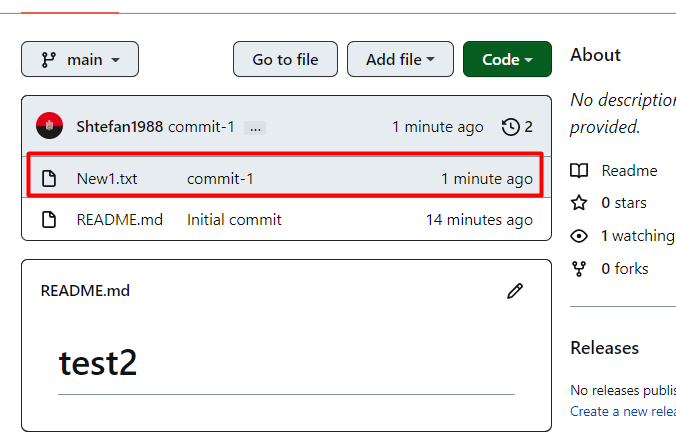
Відкриється вікно вибору гілок ( але в нас поки що одна – **main**):



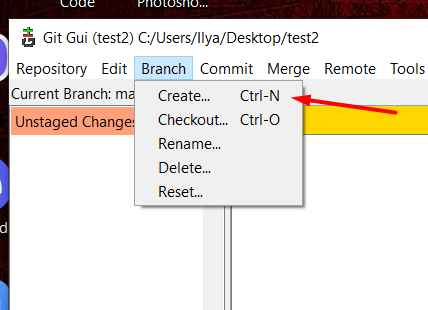
Тиснемо “**Push**”:

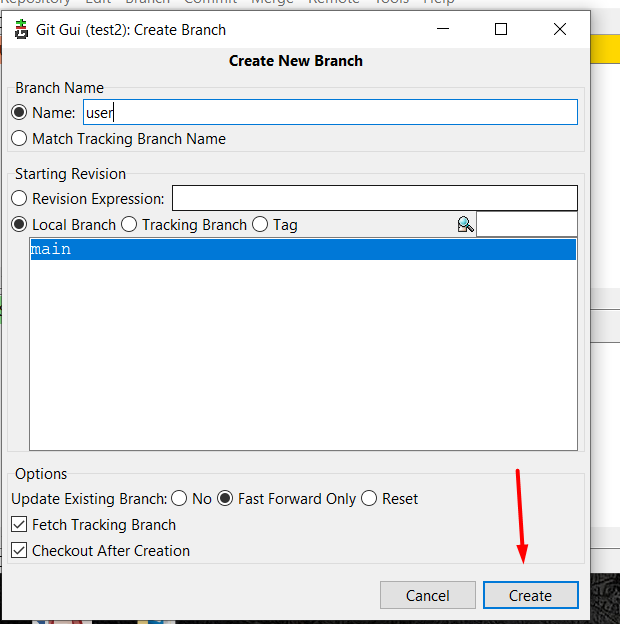


Успіх! Оновлюємо сторінку GitHub і бачимо наступне:

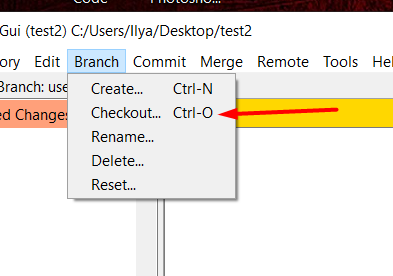


Тепер створимо нову гілку в Git Gui під назвою **user** (**git branch user** – у консолі) .

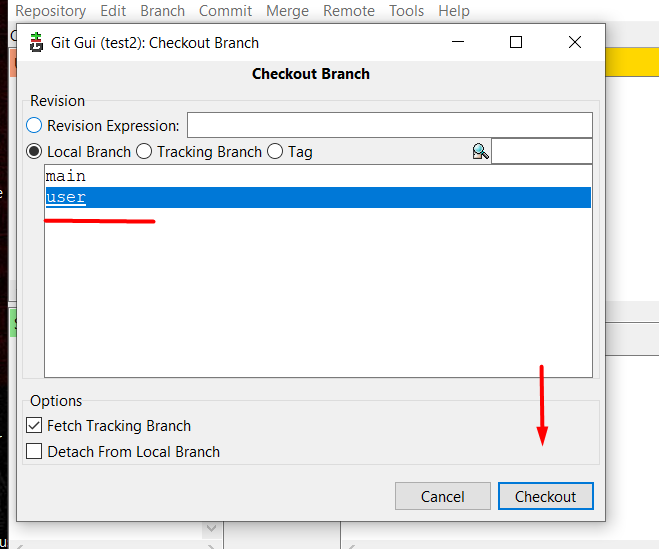




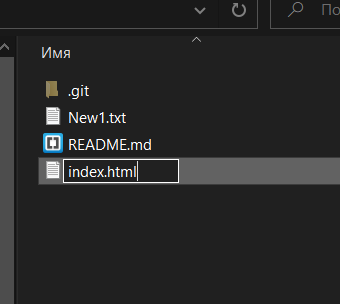
Щоб перейти на нову гілку **user**, потрібно натиснути “**Checkout**”:



(у консолі **git checkout user**)



Тепер створимо у нашій папці якийсь файл (**будь-який**), наприклад, **index.html** з уже готовим DOCTYPE :



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

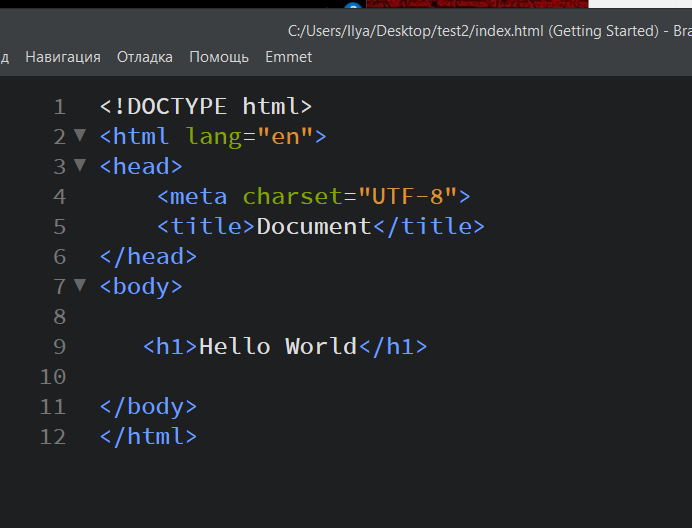
</head>

<body>

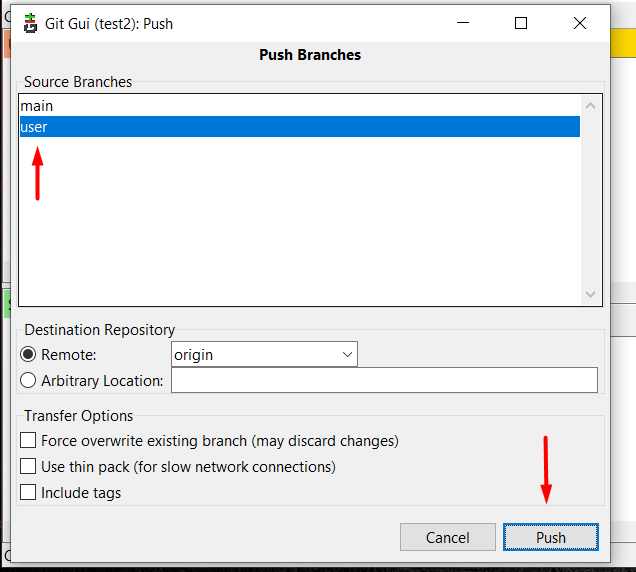
<h1>Hello World</h1>

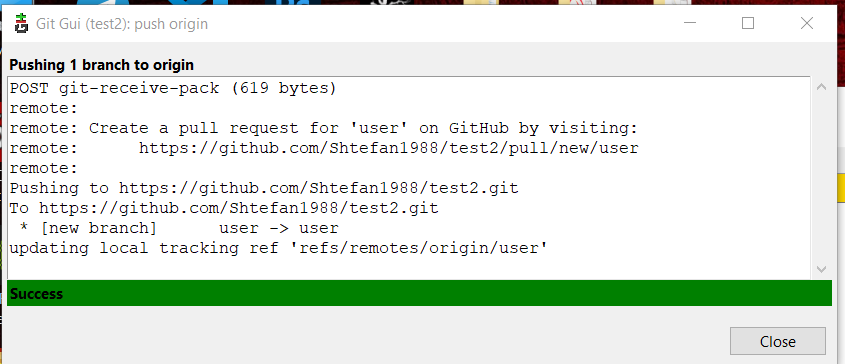
</body>

</html>

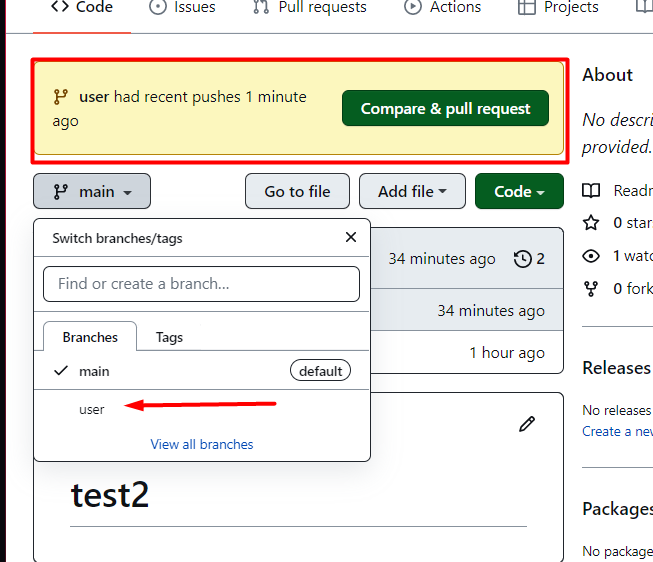


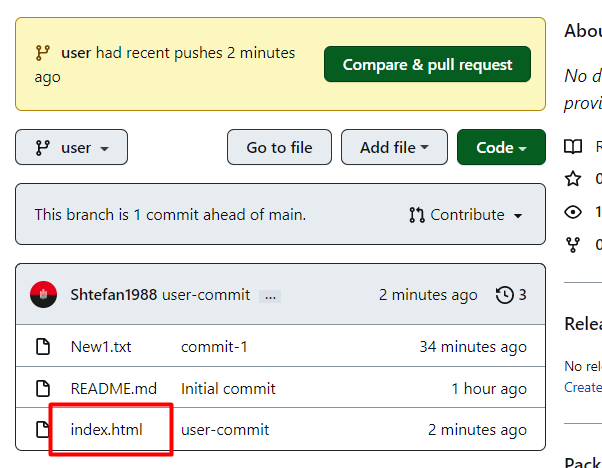
Зберігаємо. У Git Gui тиснемо “**Rescan**”, коммітимо “**Commit**” та пушимо “**Push**” у гілку **user** :





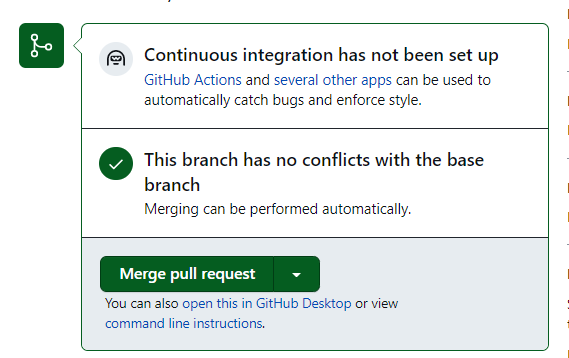
Заходимо на GitHub та оновлюємо сторінку:





Тиснемо на кнопку “**Compare & pull request**”

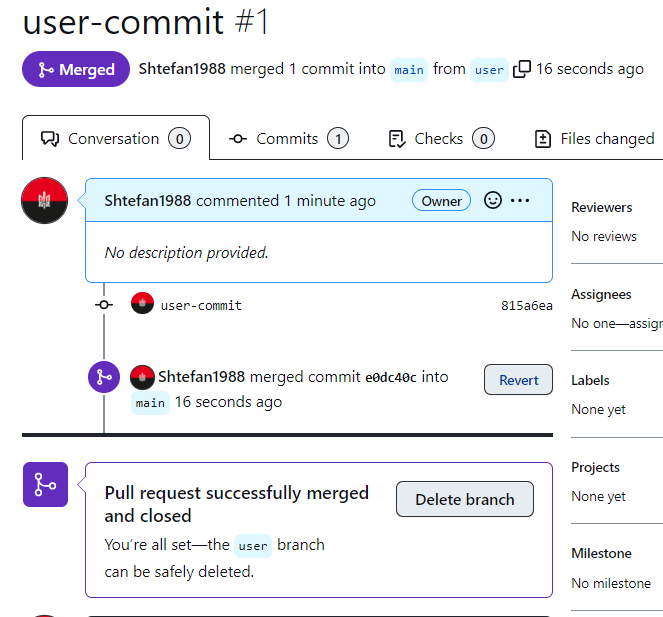
**Pull request'и** дозволяють вам розповісти іншим про ті зміни, які ви розмістили у своєму GitHub-репозиторії. Як тільки **pull request** надіслано, зацікавлені сторони розглядають ваші зміни, обговорюють можливі редагування або навіть додають додаткові комміти, якщо потрібно.



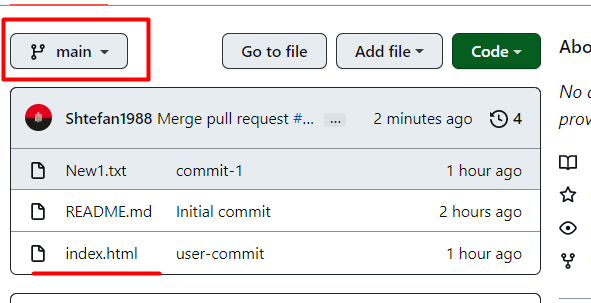
Тепер якщо ми хочемо зробити злиття двох гілок в одну, тоді нам потрібно їх змерджети (**merge**)

Натискаємо на “**Merge pull request**”.

Результат мерджу:



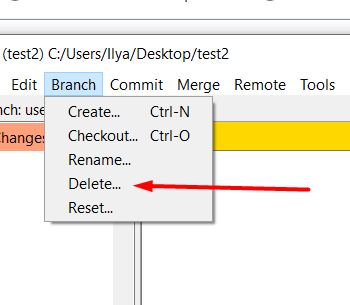
Тепер файл **index.html**  не тільки у гілці **user,** а і **main** також:

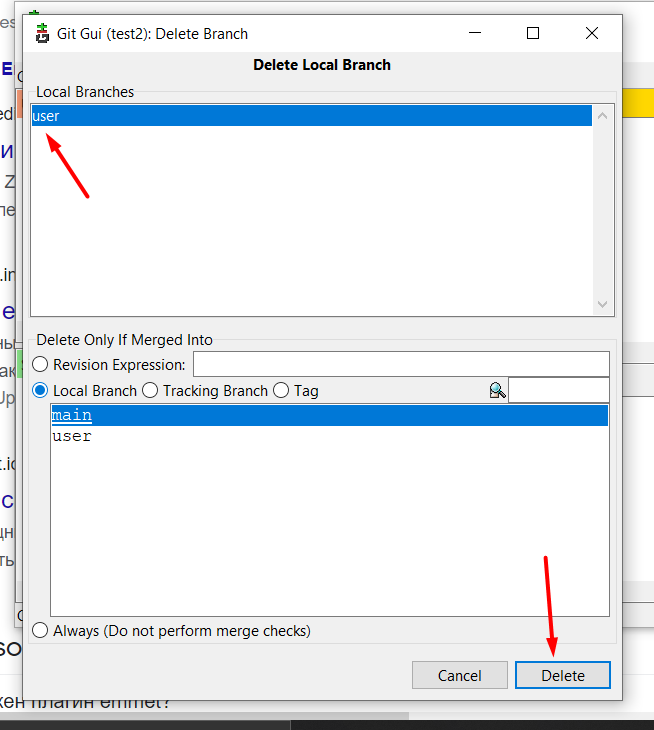


Тепер давайте видалимо гілку **user:**

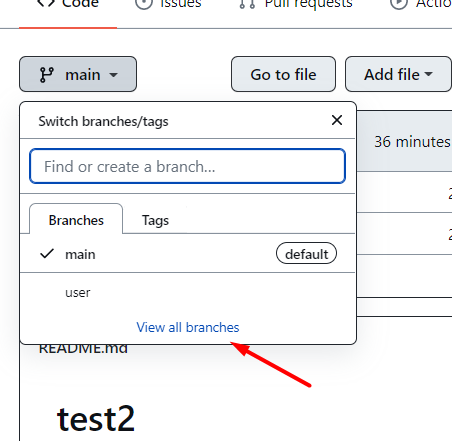
Перед тим, як її видалити, ми повинні перейти на іншу гілку, тобто на **main** “**Checkout**”.

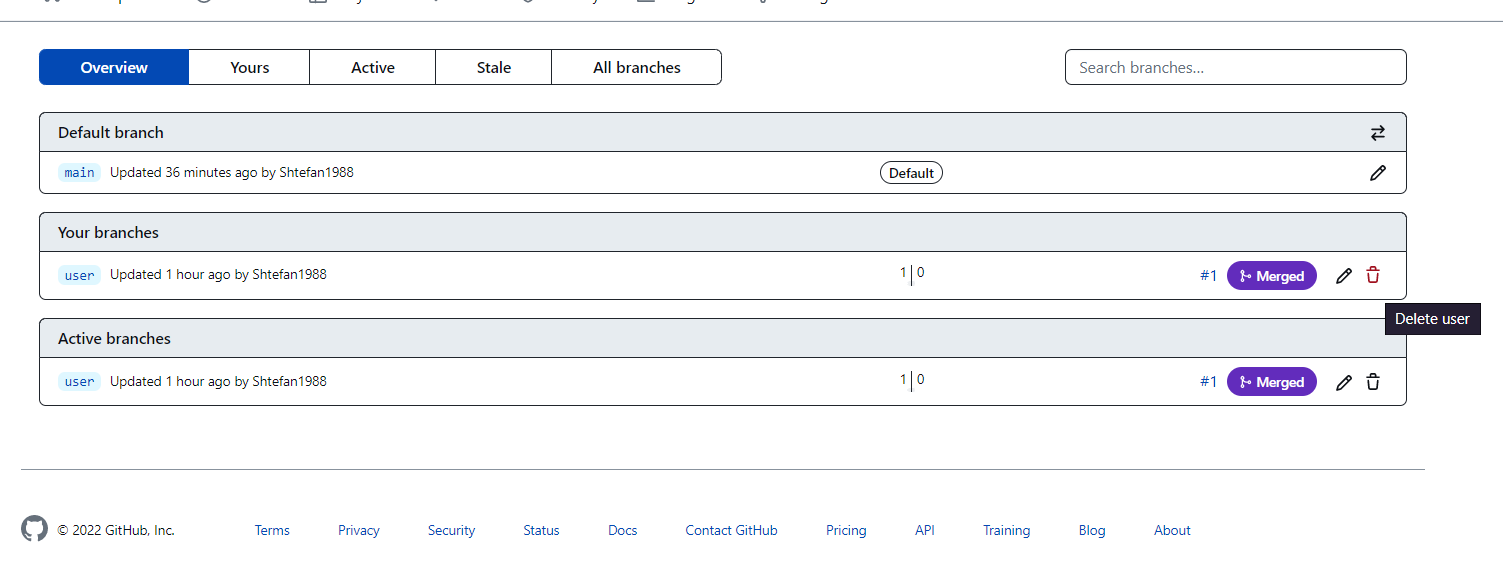
!!! Пам’ятайте, що неможливо видалити гілку, на якій ви знаходитесь.

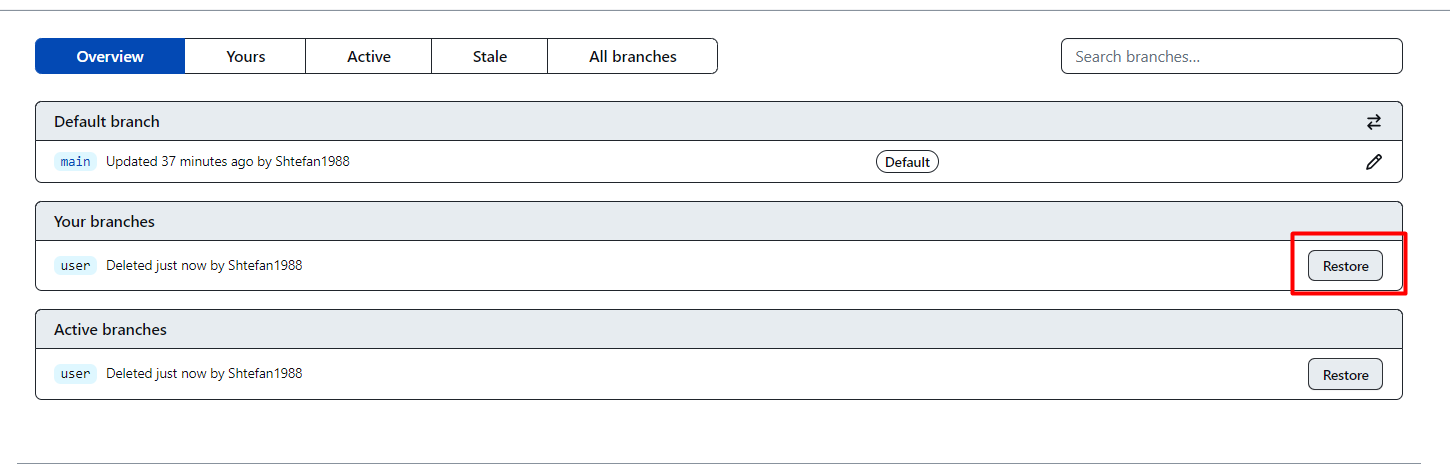




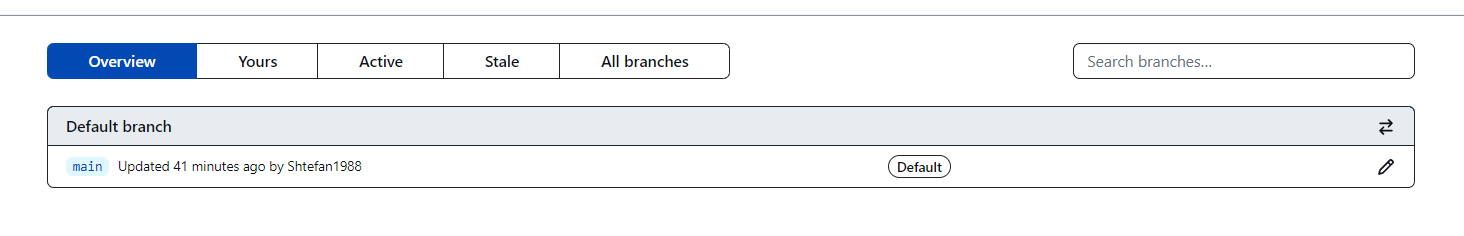
Також видаляємо гілку на віддаленому репозиторії GitHub :



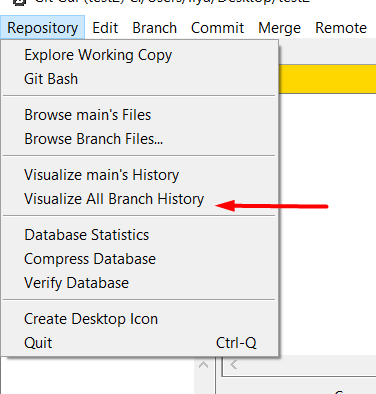


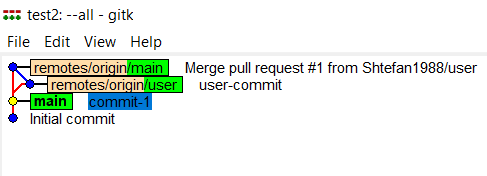


Як ви вже знаєте, у нас є можливість відмінити видалення натиснувши кнопку “**Restore**” , але нам треба її видалити, тому просто оновлюємо сторінку:



Також у Git Gui є така функція, як “**Visualize All Branch History**” , яка дає змогу побачити нам, як саме вибудовуються, змінюються та видаляються гілки:





**Робота з репозиторієм**

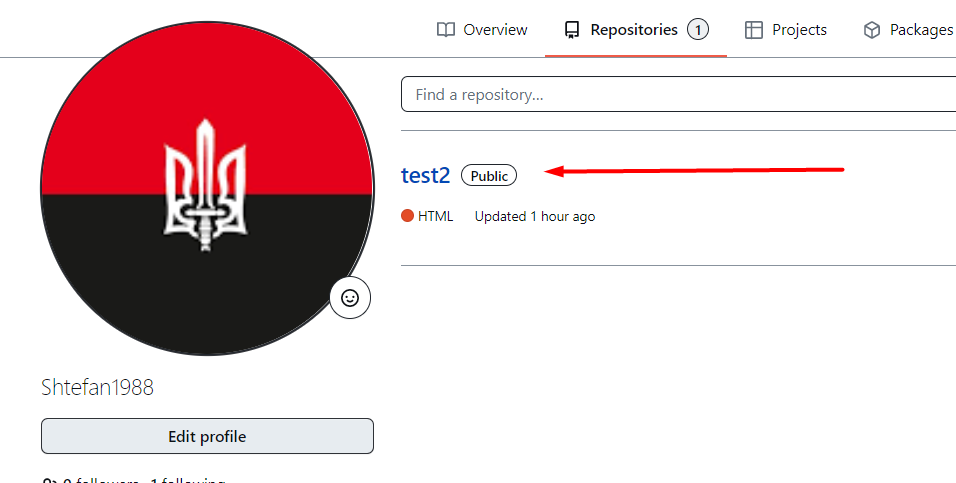
**(для співавторства)**

**Крок 4й – стати співавтором або автором проєкту в GitHub (робота в команді)**

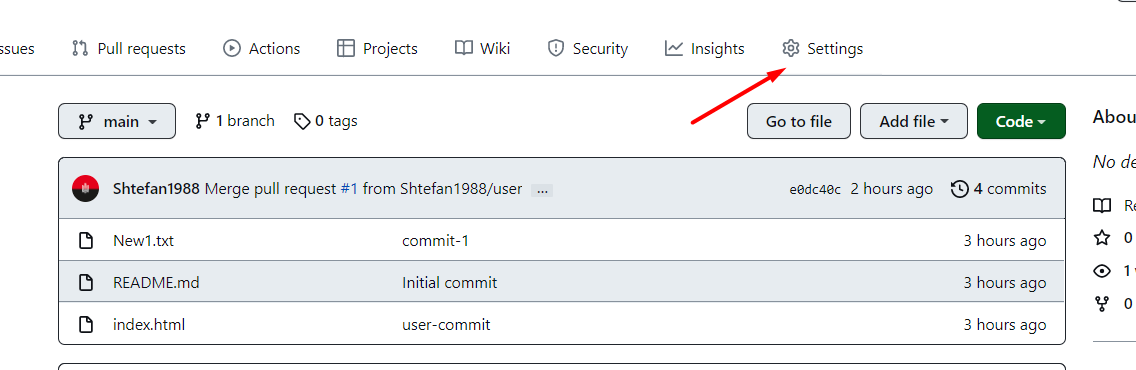
Для того, щоб стати співавтором проєкту, спочатку вас мають запросити на участь у співавторстві. Автор проєкту робить запрошення, після чого вам на електрону пошту приходить повідомлення, яке ви можете прийняти або відхилити.

Якщо ви **автор**, то запрошення робляться так:

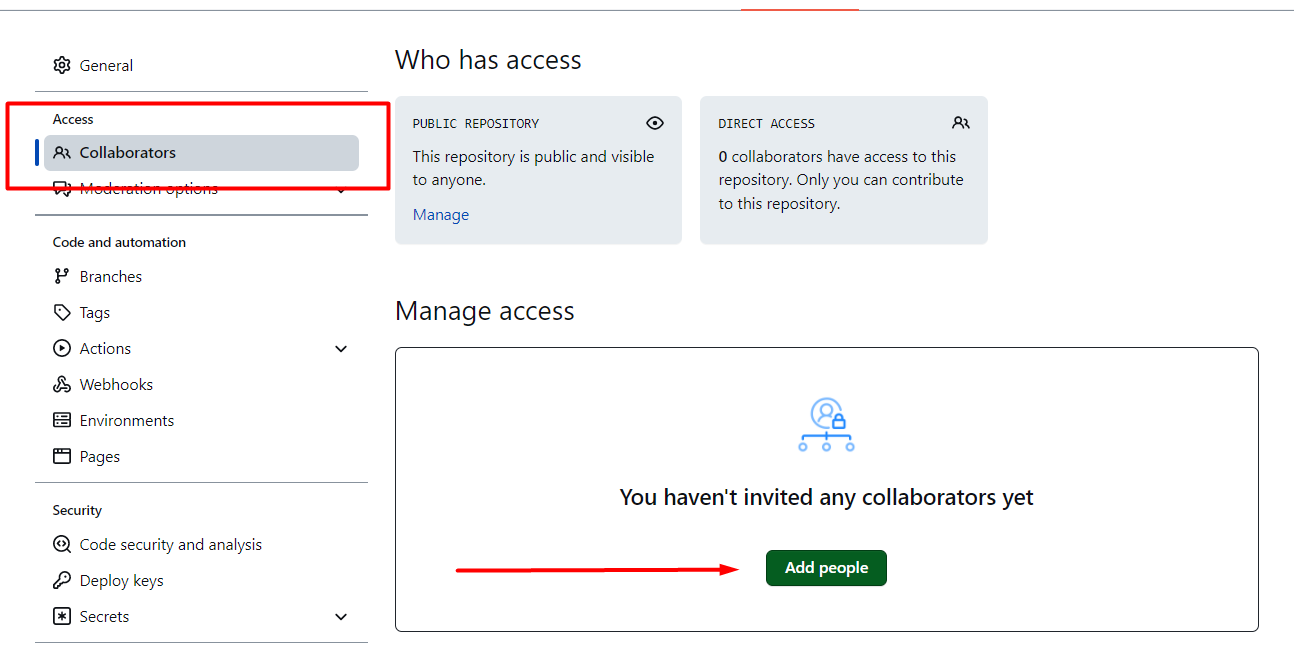
1. Заходимо у свій віддалений репозиторій



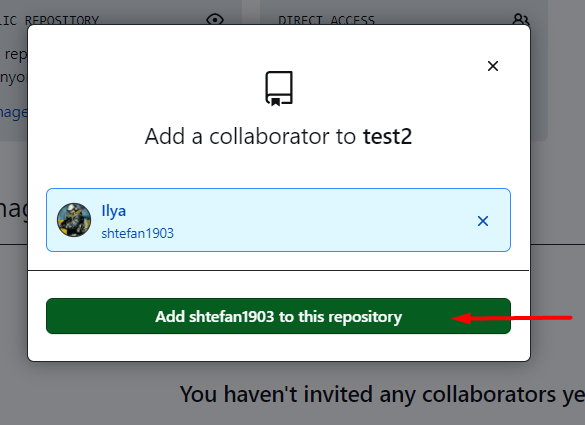
2. **Settings**



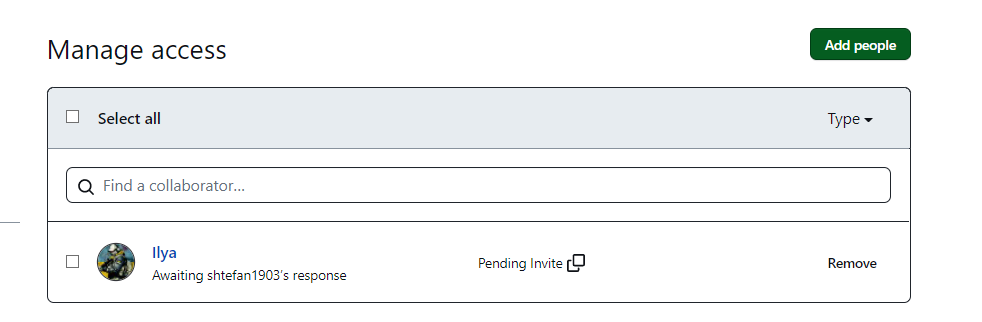
3. Collaborators



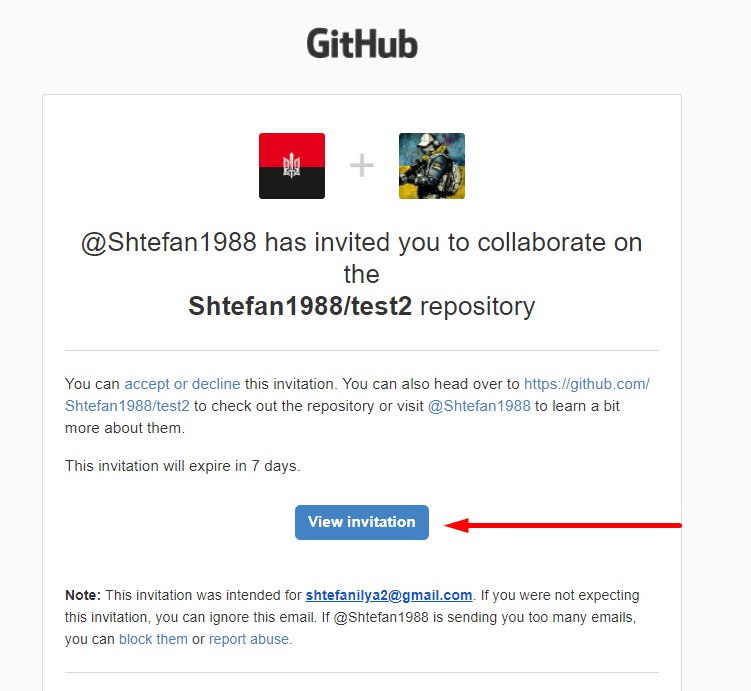
Натискаємо “**Add people**”. У пошуку вводимо **нікнейм** того, кого ви хочете додати до співавторів:



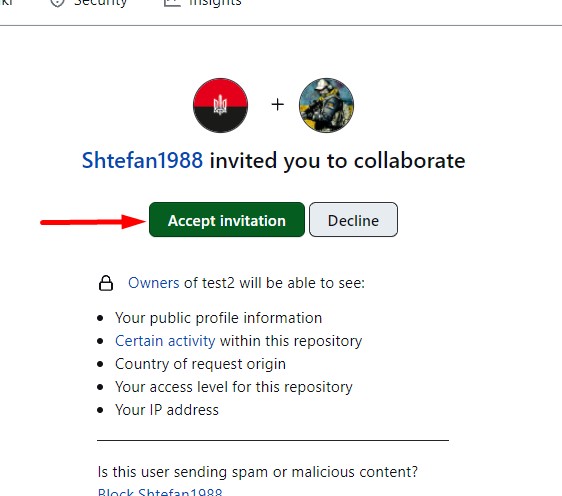
Натискаємо “**Add** user **to this repository**”:



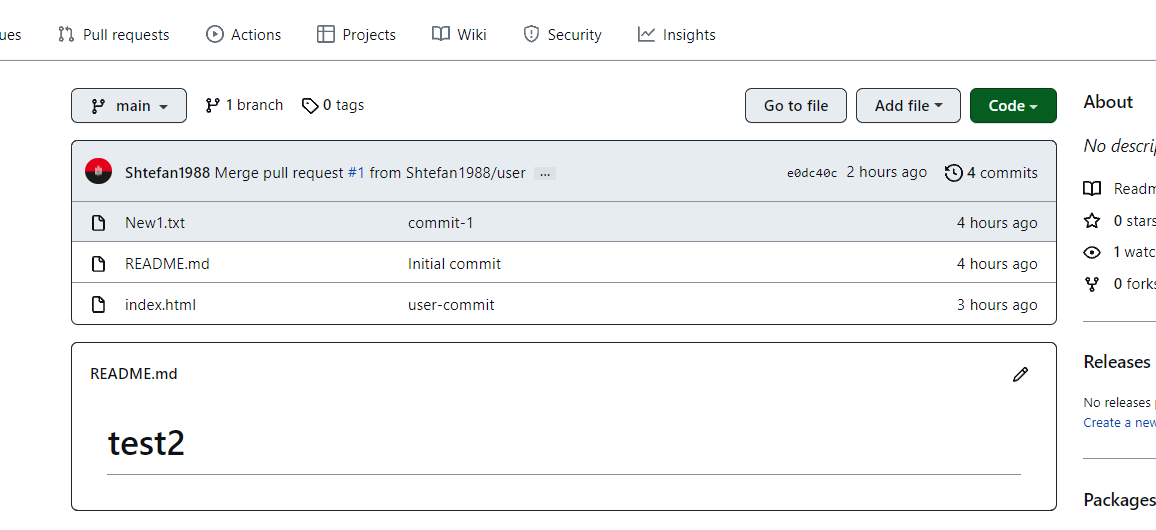
А ось так виглядає запрошення у того, кому ви його надіслали:



Якщо ви **співавтор**, то натискаємо “**View invitation**”:

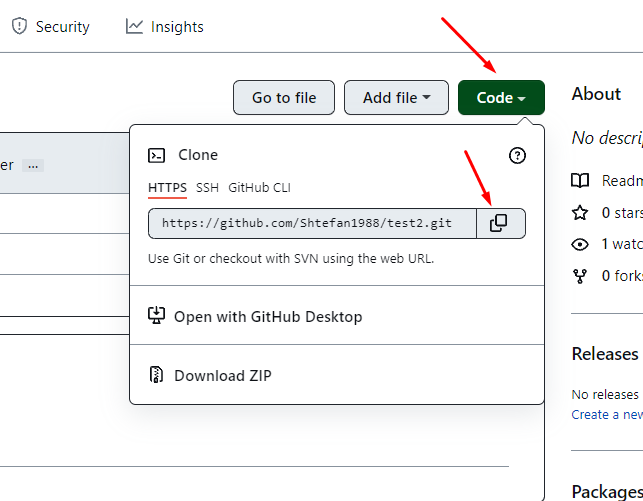


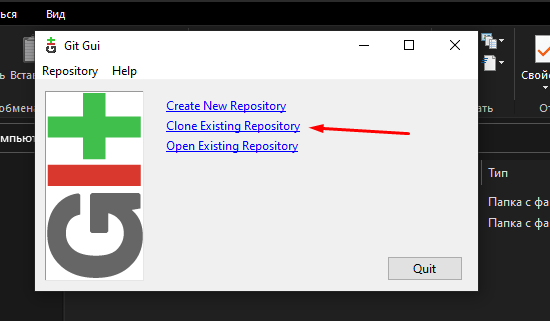
Приймаємо запрошення:

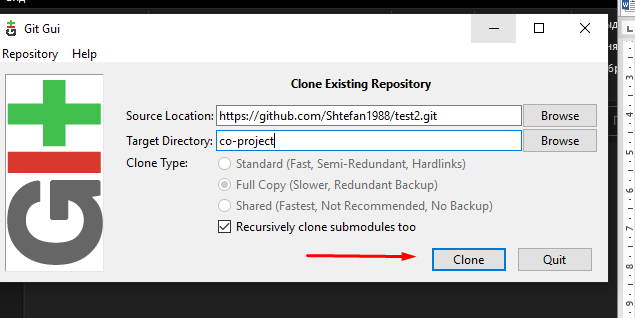


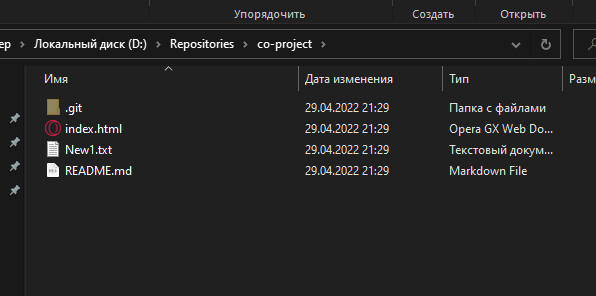
Успіх, тепер ви **співавтор** !

Щоб нормально працювати с репозиторієм **автора**, вам потрібно створити клон його репозиторію на вашому ПК:

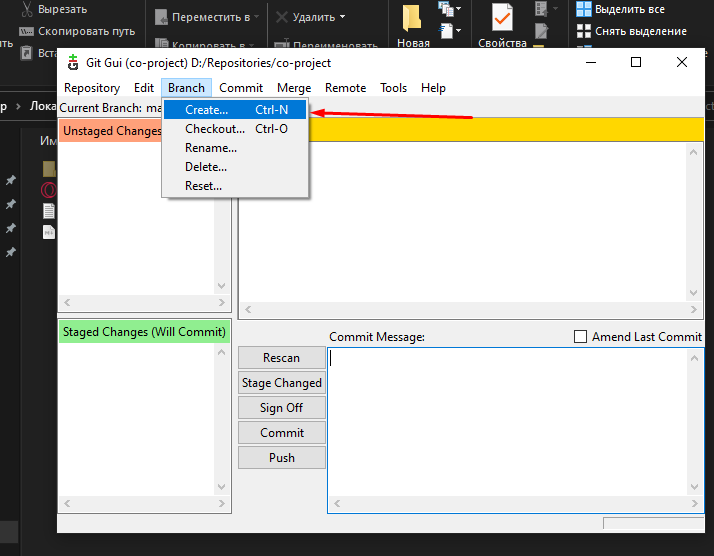


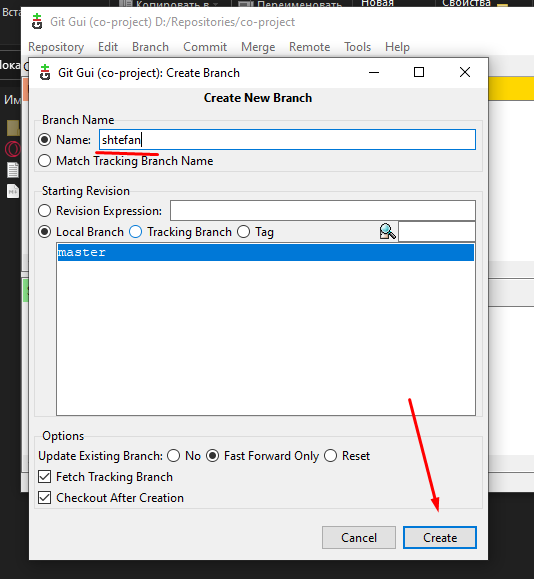






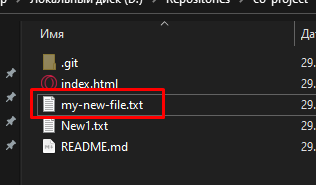
Для нормальної і продуктивної роботи, ви, як **співавтор** створюєте **окрему гілку**, в якій ви будете працювати:



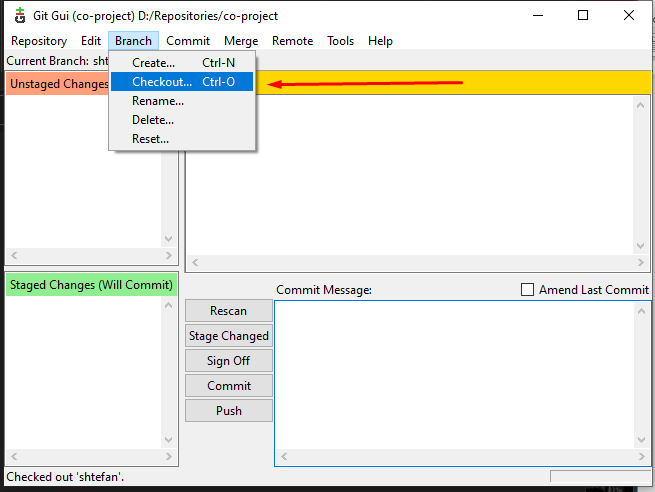


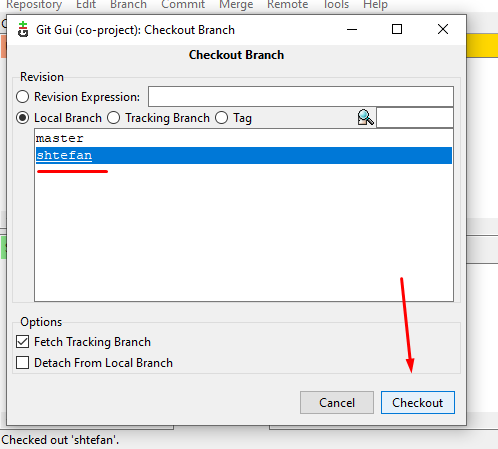
**!!! НІ В ЯКОМУ РАЗІ НЕ МОЖНА ЗМІНЮВАТИ, АБО ЩЕ ЯКОСЬ РЕДАГУВАТИ ТА ВНОСИТИ ЗМІНИ У ГІЛКУ АВТОРА. ПРАЦЮЄТЕ ТІЛЬКИ У СВОЇЙ ГІЛЦІ!!!**

Додаємо текстовий файл (типу ми щось зробили) :

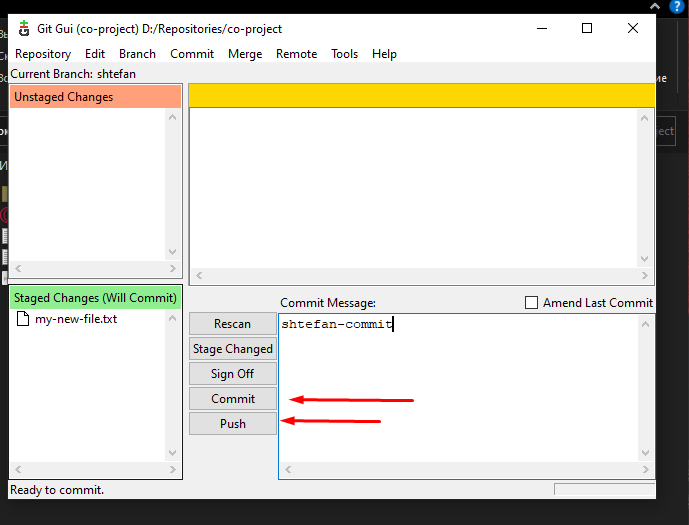


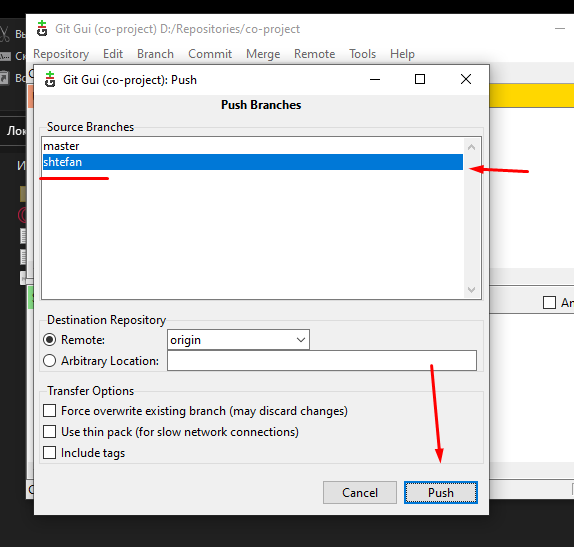
Переконаємося, що ми на **своїй** гілці, а не на гілці **автора**:

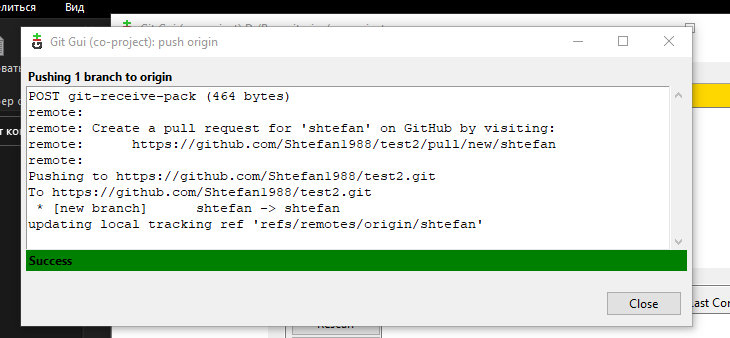




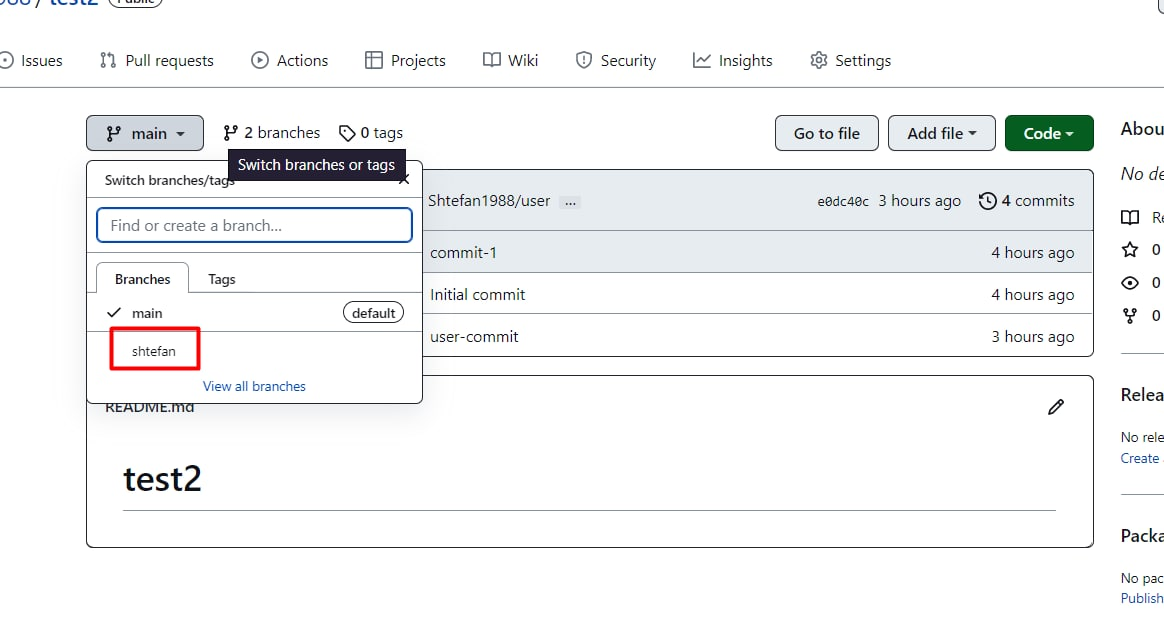
Тепер додаємо цей файл, закоммітимо та запушимо його:



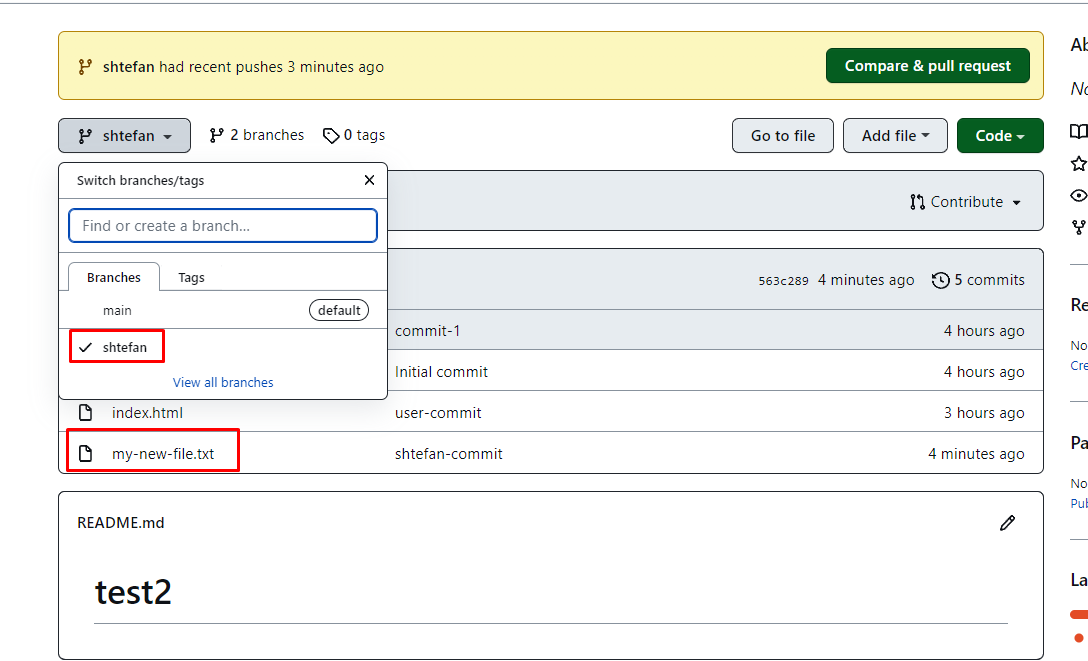




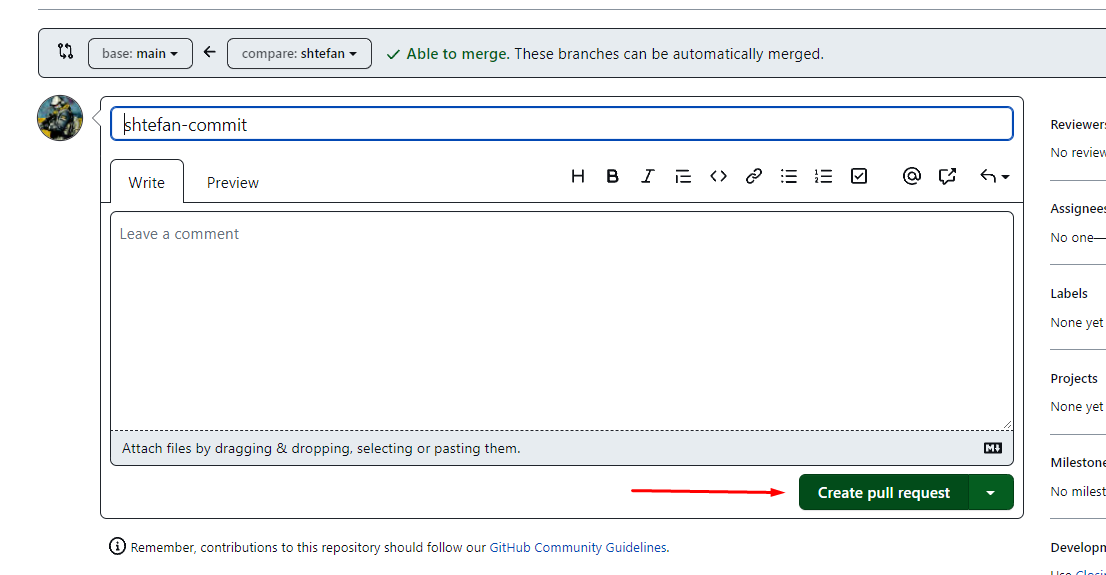
Ось так виглядає сторінка **автора** після пушу:



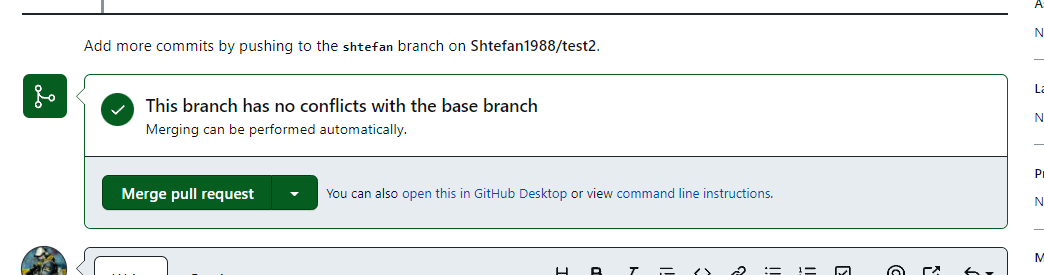
А ось так у нас:



Тиснемо “Compare & pull request”:



**!!!АЛЕ НЕ РОБИМО** “**Merge**” **БЕЗ ДОЗВОЛУ АВТРОРА!!!**



Вітаю, тепер ви знаєте трішки більше ніж базові знання!)