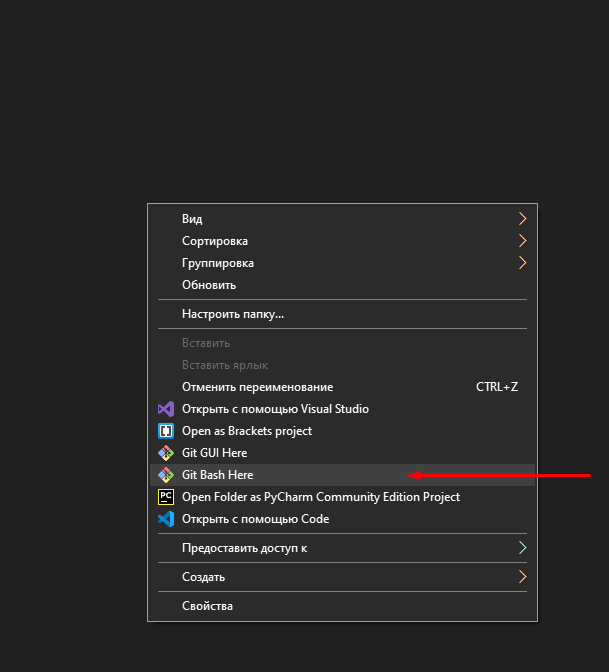
**Робота з репозиторієм**

**(для власного репозиторію)**

**GitBash(консоль)**

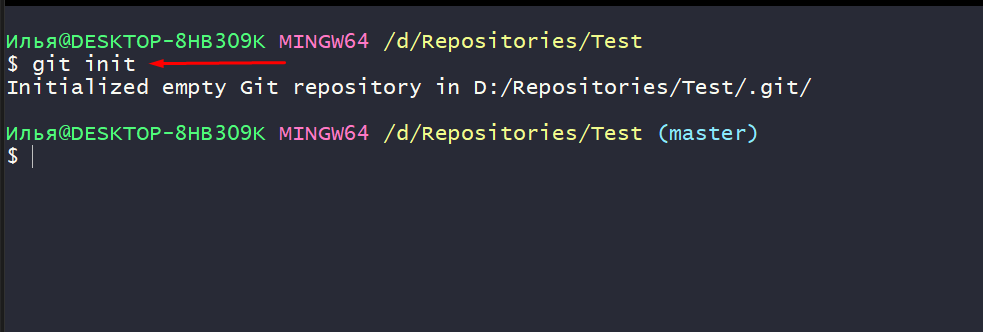
**Крок 1й – ініціалізація папки**

У новій папці клацнути ПКМ та обрати **Git Bash Here**.

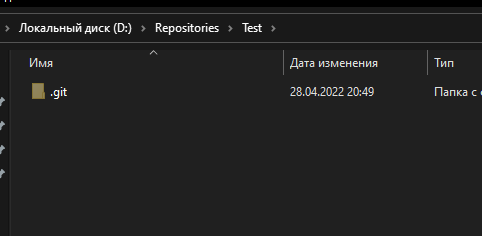


Щоб ініціалізувати папку, у консолі прописуємо наступну команду:

**git init** , натискаємо Enter.

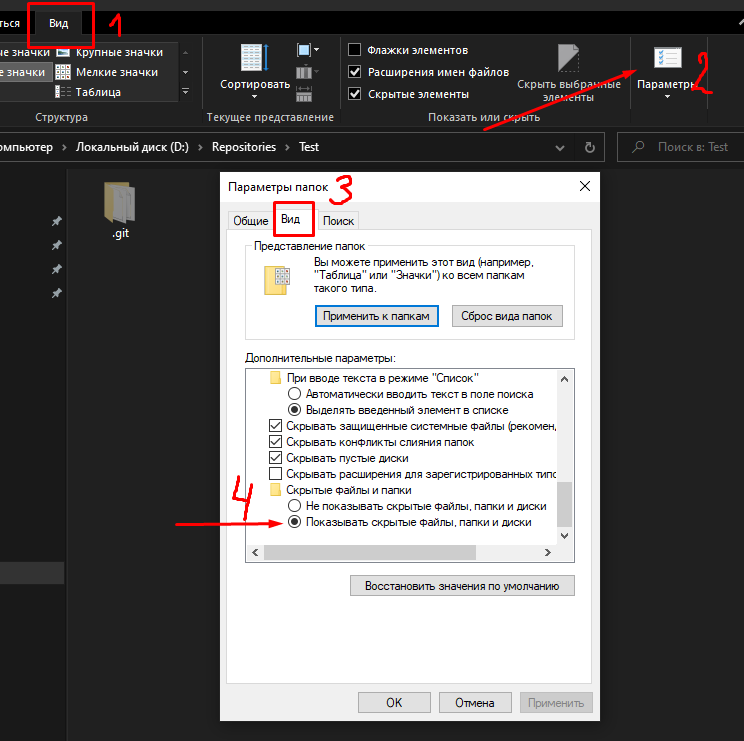


Після ініціалізації папки у вас має з’явитися наступна **прихована папка:**



**ЯКЩО ЇЇ НЕМАЄ, ПЕРЕКОНАЙТЕСЯ, ЩО У ВАС УВІМКНЕНИЙ РЕЖИМ ВІДОБРАЖЕННЯ ПРИХОВАНИХ ПАПОК**

Це можна зробити так:



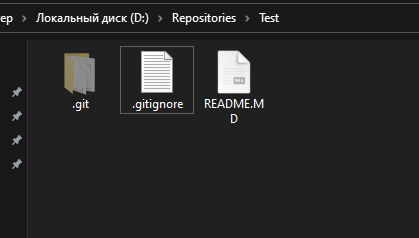
**Крок 2й – створення README.MD та .gitignore**

У цій же папці створюємо 2 файли - **README.MD** та **.gitignore**,

без яких неможлива подальша робота с GitHub.

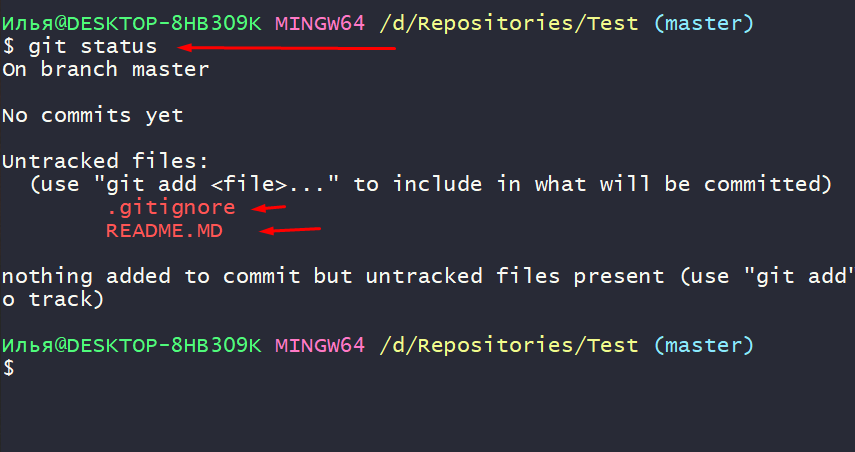
**README** з розширенням (**MD**)Markdown – зазвичай для GitHub це просто блокнот, який ведеться на мові Markdown та у якому записуються основні зміни або просто нотатки.

А ось **.gitignore** потрібен для зберігання файлів папок від системи Git. Зазвичай приховують конфігураційні файли (особливо зі словами), тимчасові файли і папки.

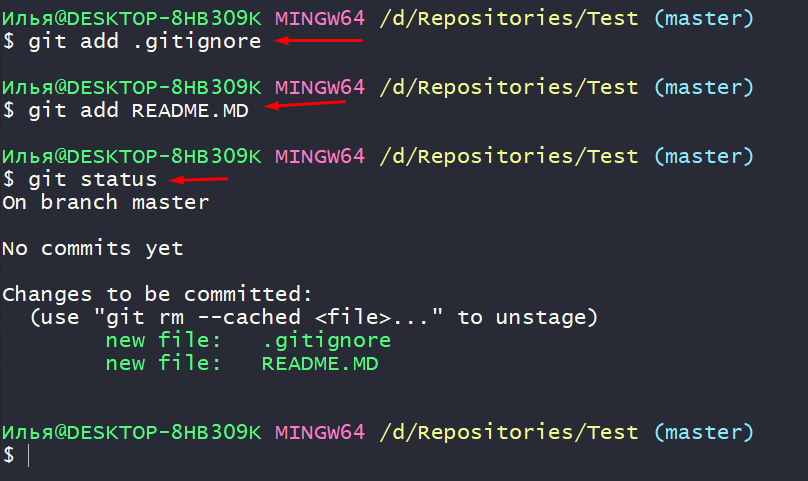


**Крок 3й – внесення файлів у головну гілку(master)**

Якщо у консолі ми використаємо команду **git status** , то побачимо нові, створені раніше нами файли(**README.MD та .gitignore**):



Щоб їх додати потрібно прописати **git add НАЗВА-ФАЙЛУ** (для одного конкретного файлу) або **git add .** (для всіх файлів одразу)

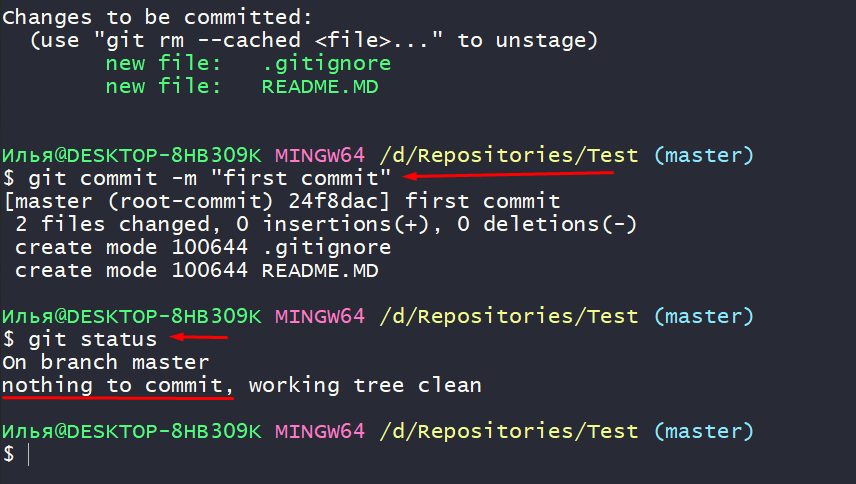


Тепер після перевірки статусу **git status** , ми побачимо додані файли (вони світяться зеленим), які чекають на комміт.

**Крок 4й – комміт (commit)**

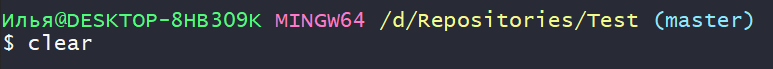
Тепер нам потрібно їх закоммітити. Комміт (**commit**) – фіксація будь-яких змін у **Git**.

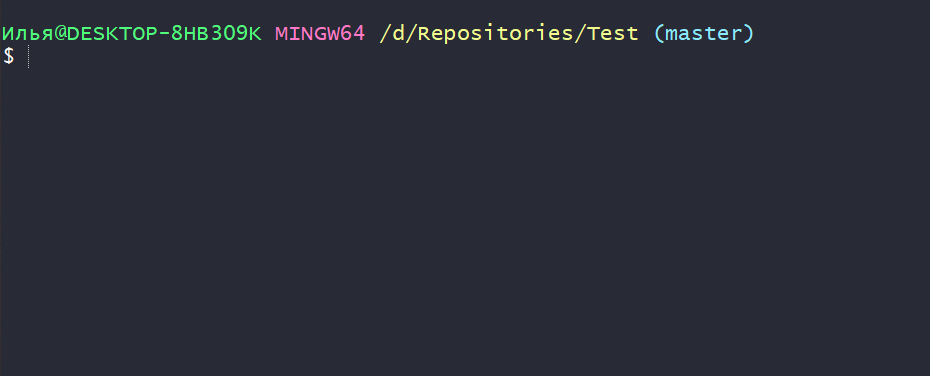
Щоб зробити комміт, у консолі ми повинні прописати наступну команду **git commit -m “***smth***”** , де **-m** (message), а у лапках назву вашого комміту. Тиснемо Enter:



Ось файли успішно закоммітились. Після перевірки статусу **git status** , ми побачимо, що немає ніяких файлів на комміт.

**ЯКЩО ВИ ХОЧЕТЕ ПОЧИСТИТИ КОНСОЛЬ, ТОДІ ПРОПИСУЄМО КОМАНДУ clear :**





**Крок 5й – гілки (branch)**

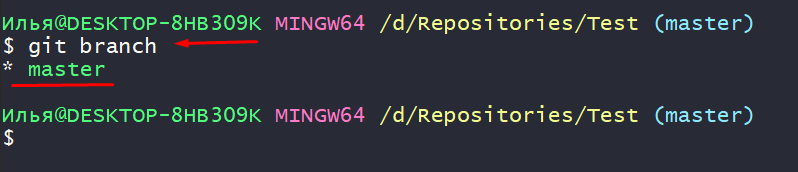
Взагалі система Git деревоподібна, тому маємо таке поняття, як гілка **(branch)** . Гілка в Git - це простий пересувний вказівник на один з таких коммітів.

Після ініціалізації ми маємо одну, проте **головну гілку** під назвою **master** .

У майбутньому Git попросить нас перейменувати гілку **master** на **main** .

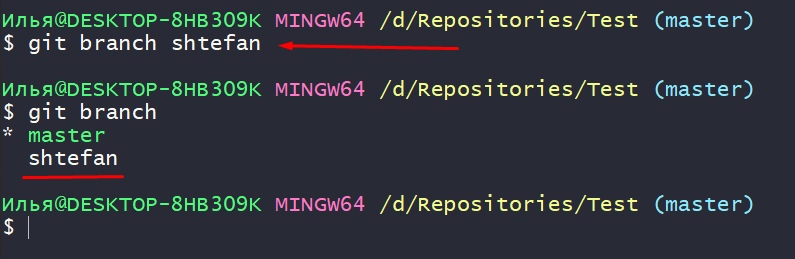
Щоб перевірити на якій саме ми гілці знаходимось, прописуємо

**git branch :**



Ось нам показало, що ми знаходимось на гілці **master** .

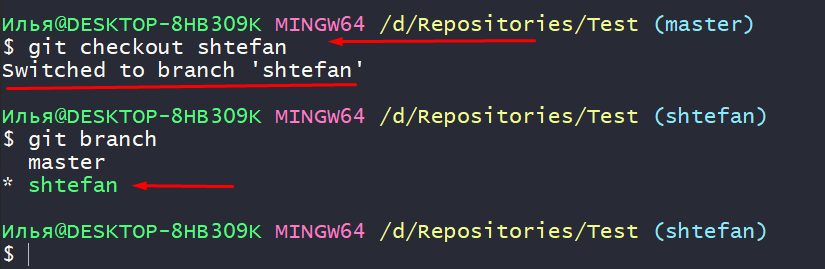
Якщо ви хочете додати нову гілку, то потрібно прописати наступне **git branch НАЗВА-ГІЛКИ** :



Після створення гілки перевіряємо її наявність командою **git branch**

Як ми бачимо, гілка створилась, проте ми все ще на гілці **master** .

Щоб переключитися на іншу гілку маємо прописати наступне **git checkout НАЗВА-ГІЛКИ** :



Робимо перевірку **git branch** . Тепер ми на гілці **shtefan** .

**Практика:**

**1. Створити новий текстовий файл у папці;**

**2. Переключитися на гілку master ;**

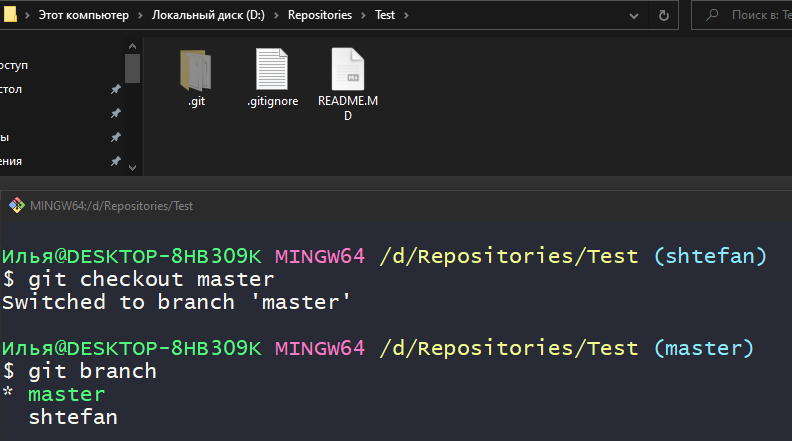
**3. Перевірити на якій саме ви гілці;**

**4. Перевірити статус;**

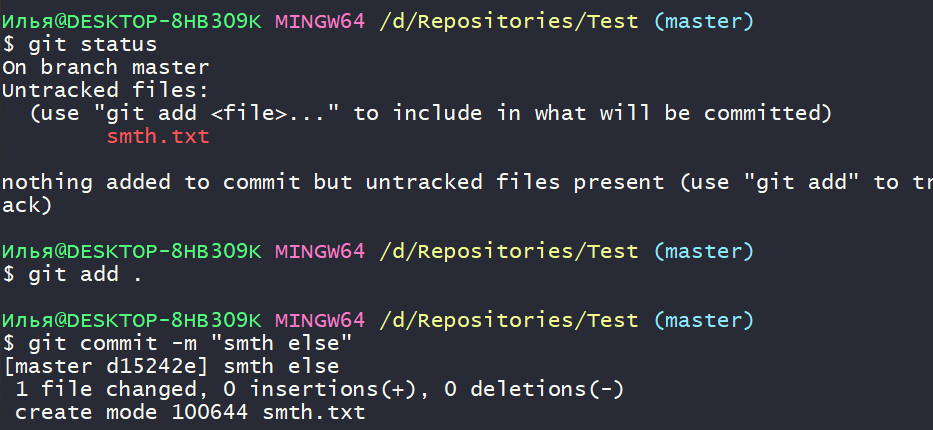
**5. Додати новий файл;**

**6. Закоммітити новий файл ;**

**Підказки:**

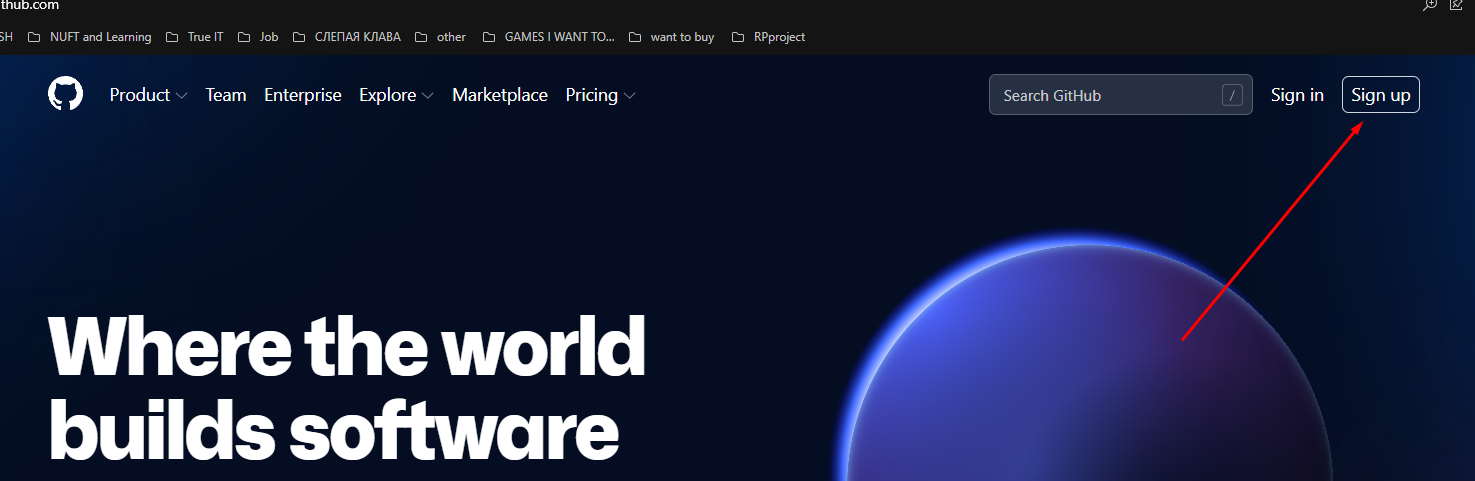
**1,2,3-** 

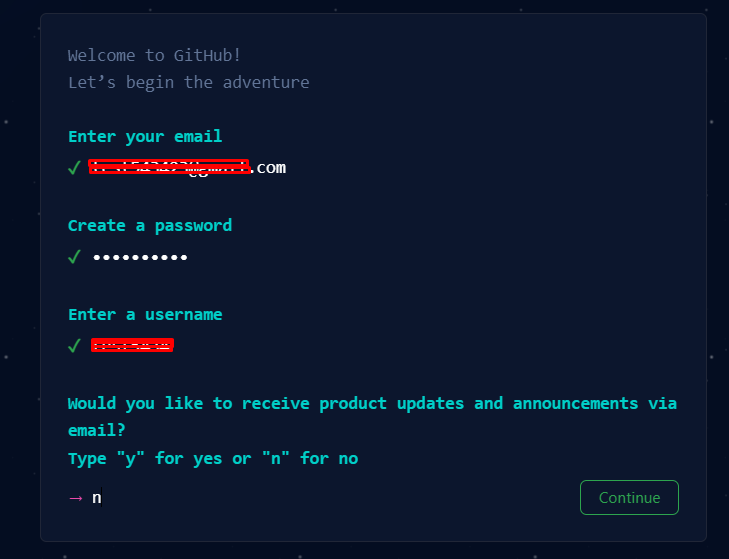
**4,5,6-**



**Крок 6й – створюємо та налаштовуємо аккаунт на Github**

Заходимо на сайт github.com та реєструємось якщо ще цього не зробили:

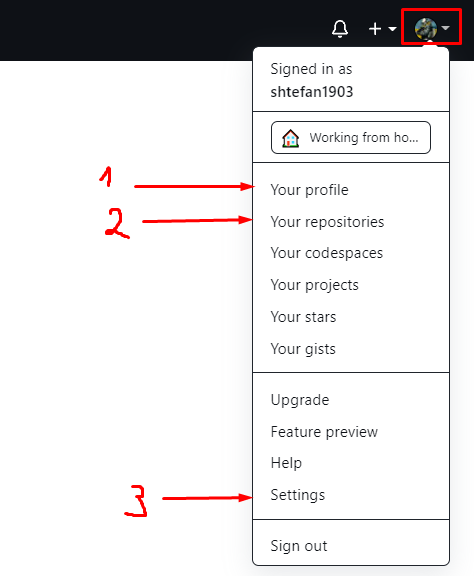




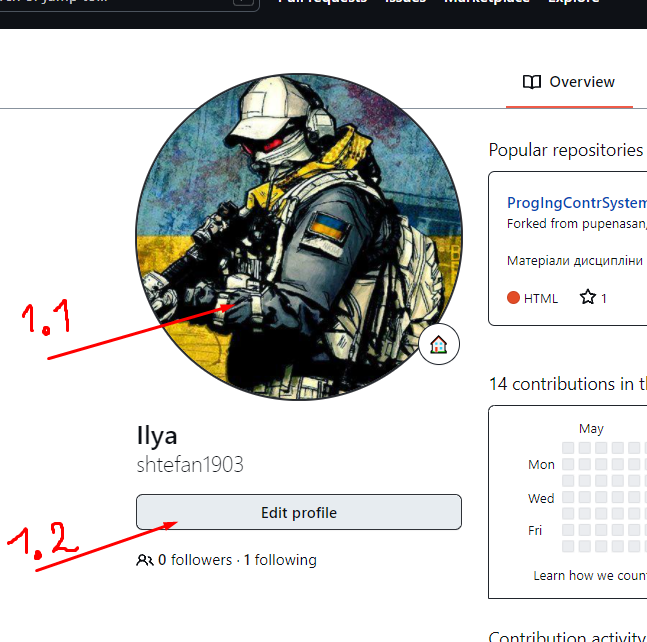
Потім ви маєте пройти спеціальний пазл на перевірку, що ви не робот)

Після того, як ви зареєструвалися і підтвердили запит на електронній пошті, вас перенесе на головну сторінку GitHub.

У вас сторінка буде з **темною** темою, щоб перейти на іншу тему потрібно зайти у налаштування (**Settings**)>**Appearance**.

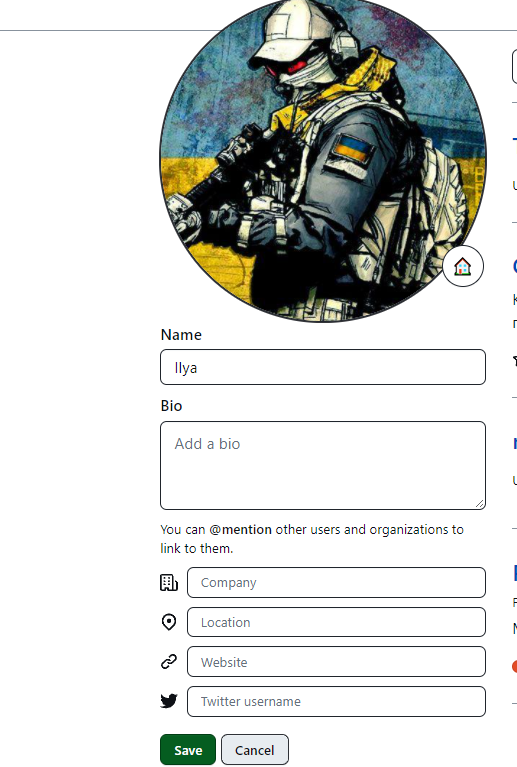


**1. Тут ваш профіль, де ви можете його редагувати;**

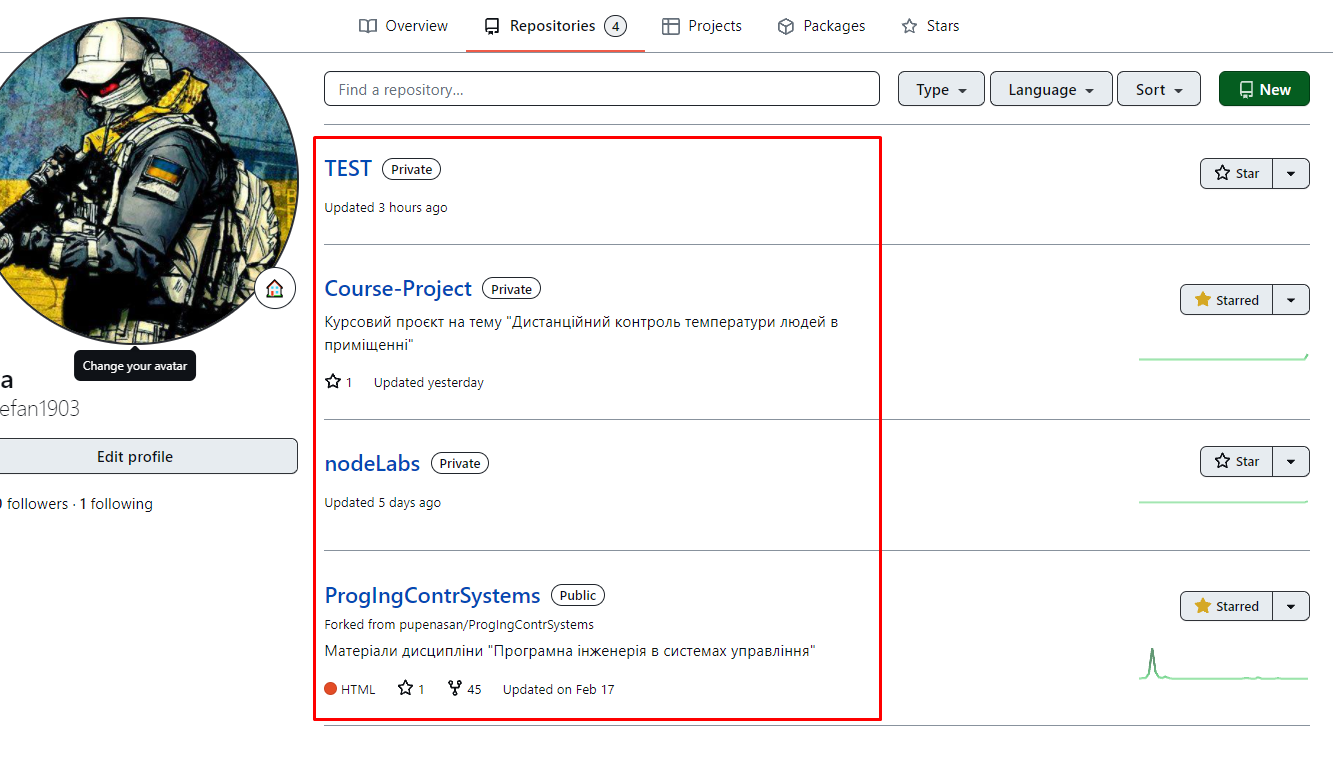


**1.1 Перехід до Settings (усі налаштування) ;**

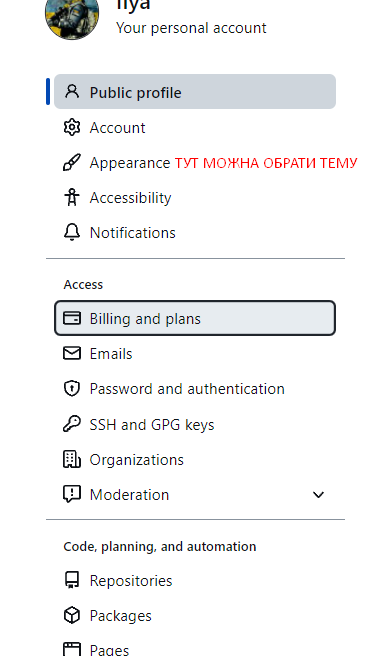
**1.2 Редагування профілю (ім’я, статус та ін) ;**



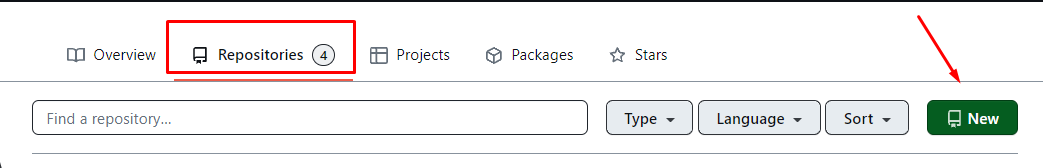
**2. Тут ваші репозиторії ;**

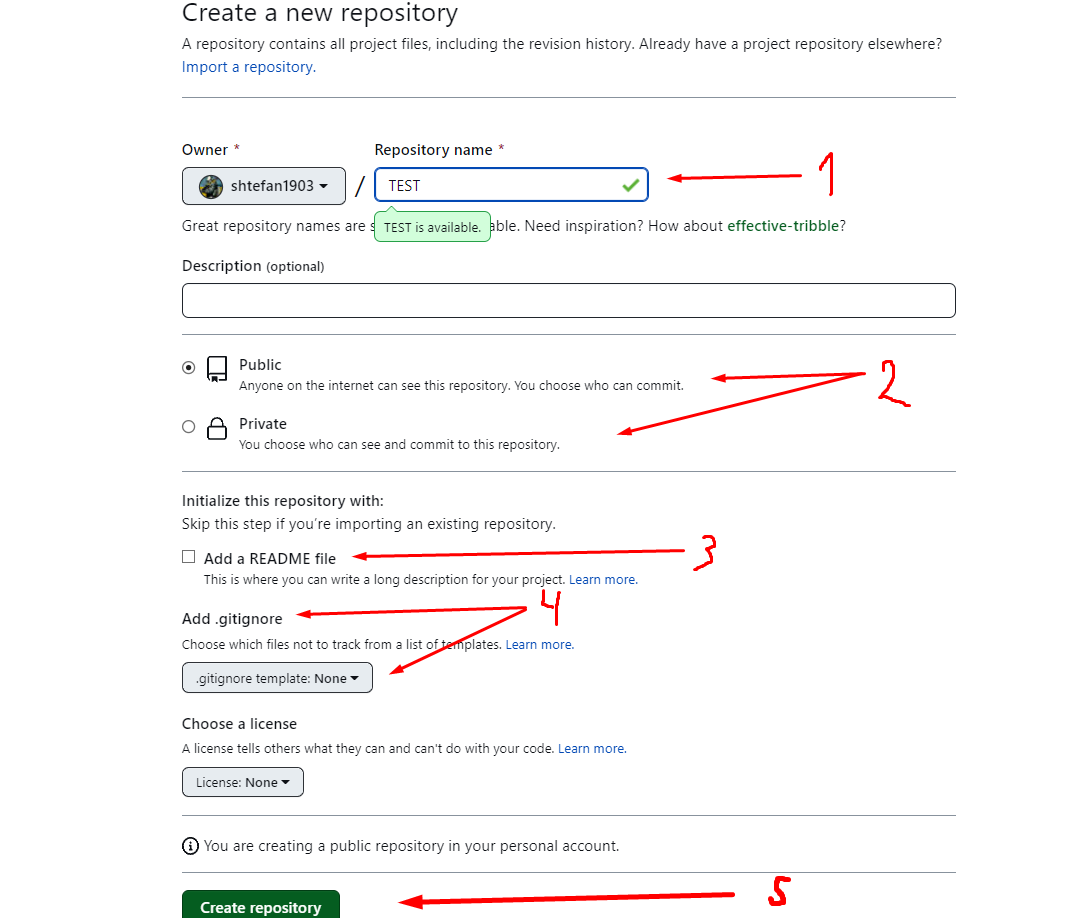


**3. Settings (усі налаштування) ;**



**Крок 7й – створюємо репозиторій**





**1 – назва вашого репозиторію;**

**2 – зробити репозиторій публічним або приватним;**

**3 – додавання файлу README.MD (У НАС ВІН ВЖЕ Є, ТОМУ НЕ СТВОРЮЄМО) ;**

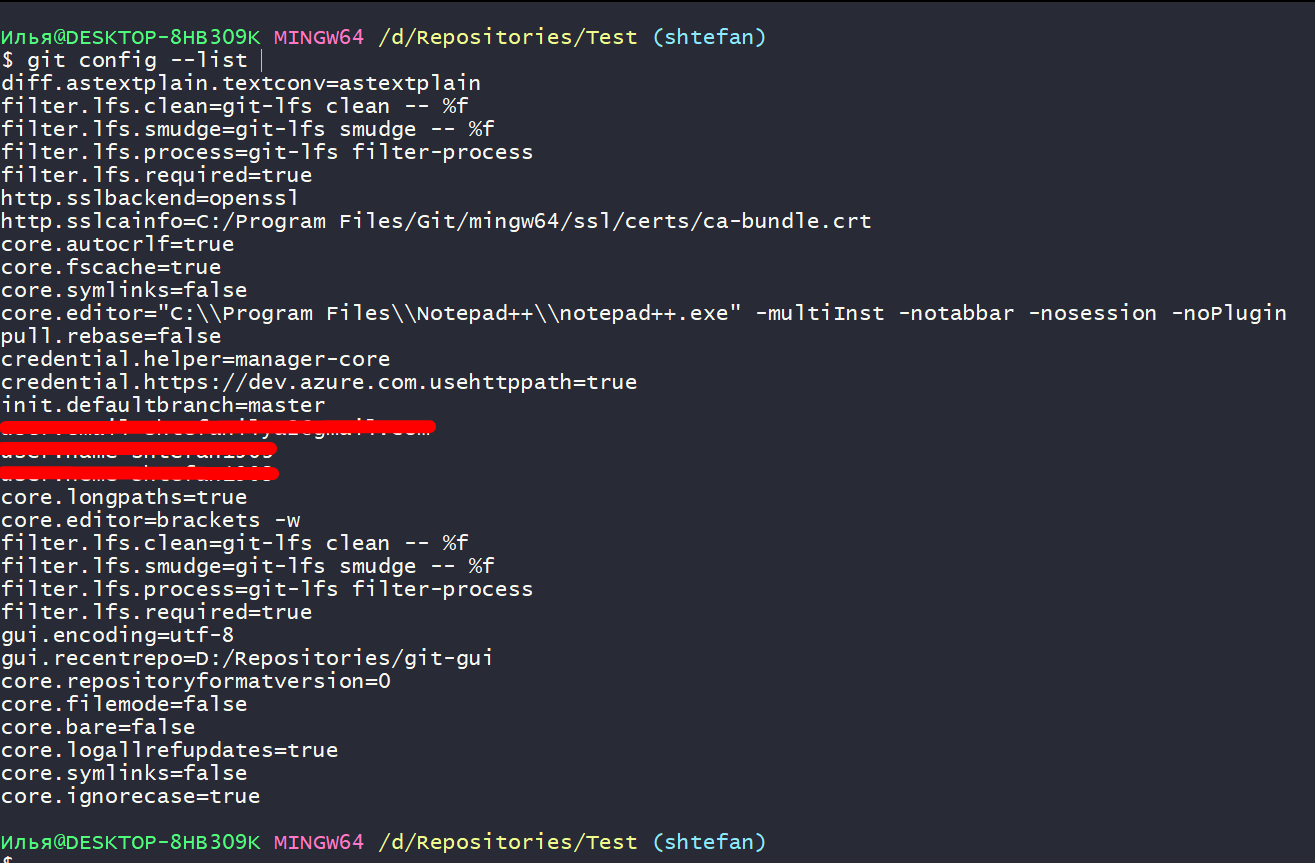
**4 – .gitignore (У НАС ВІН ВЖЕ Є, ТОМУ НЕ СТВОРЮЄМО) ;**

**5 – створюємо репозиторій;**

**Крок 8й – реєструємо себе у Git локально**

Перш ніж зареєструватися, вводимо команду **git config --list** ,

ця команда потрібна для перегляду всіх конфігурацій.



У вас буде набагато менше конфігурацій, та будуть відсутні поля з поштою та ім’ям.

Для локальної реєстрації потрібно ввести наступні команди:

Для створення імені (бажано записати ваш GitHub нікнейм ).

**git config --global user.name “ВАШЕ-ІМ’Я”** - є лапки **!**

**! ! !**Тут обов’язково email, який ви прив’язали до GitHub.

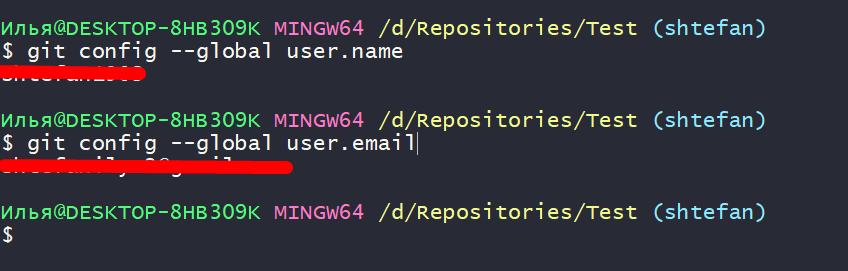
**git config --global user.email ВАШ EMAIL** - нема лапок **!**

Оскільки я вже зареєстрований, тому просто покажу як перевірити своє ім’я та пошту у Git.

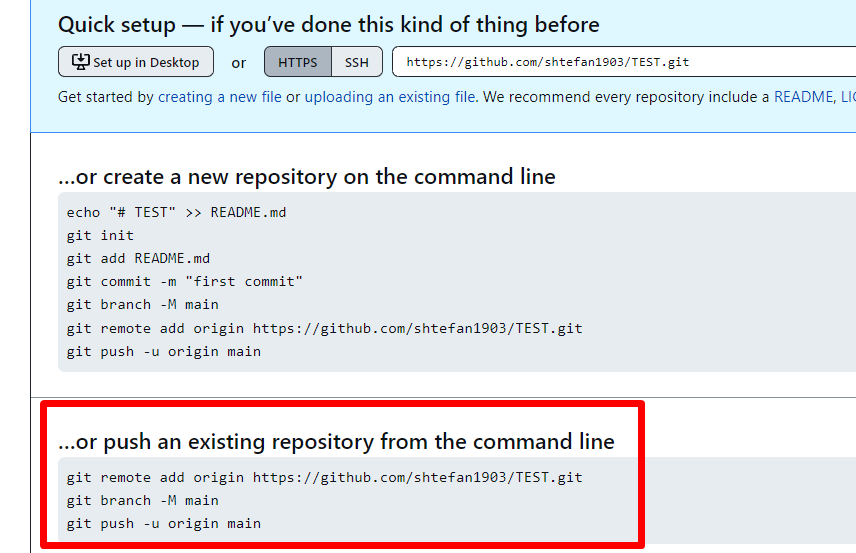
Для перевірки використовуємо наступні команди:

**git config --global user.name**

**git config --global user.email**

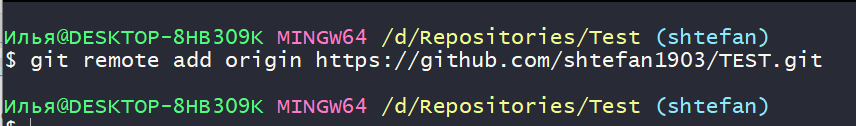


**Крок 9й –підключаємо репозиторій та пушимо(push) в нього файли**

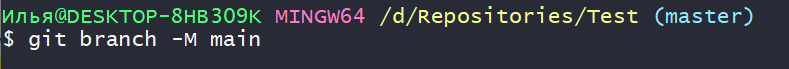


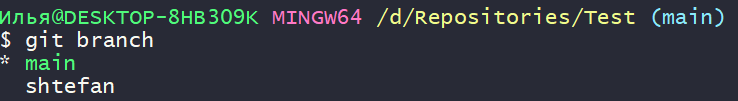
Підключаємо репозиторій:

**git remote add origin ПОСИЛАННЯ НА РЕПОЗИТОРІЙ**



Оскільки GitHub буде сваритися на те, що в нас головна гілка має ім’я **master** , тому одразу міняємо її на **main git branch -M main** :



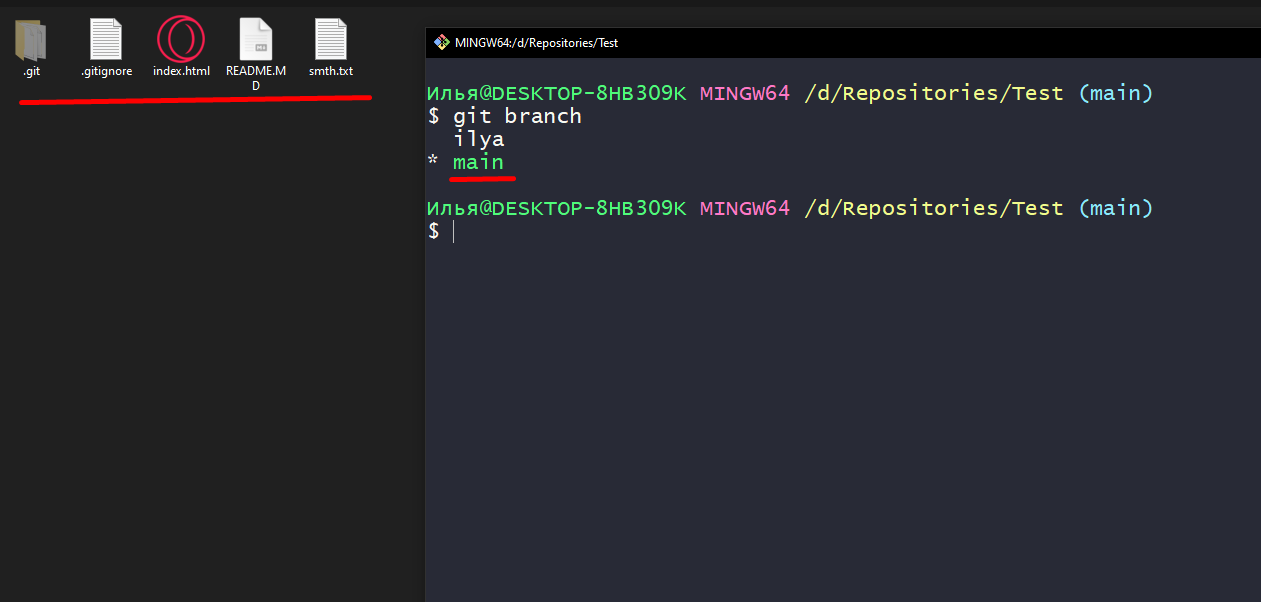


**! ! !** Якщо ви працюєте в консолі, то потрібно бути дуже уважним і відслідковувати, що ви додаєте і куди саме. **! ! !**

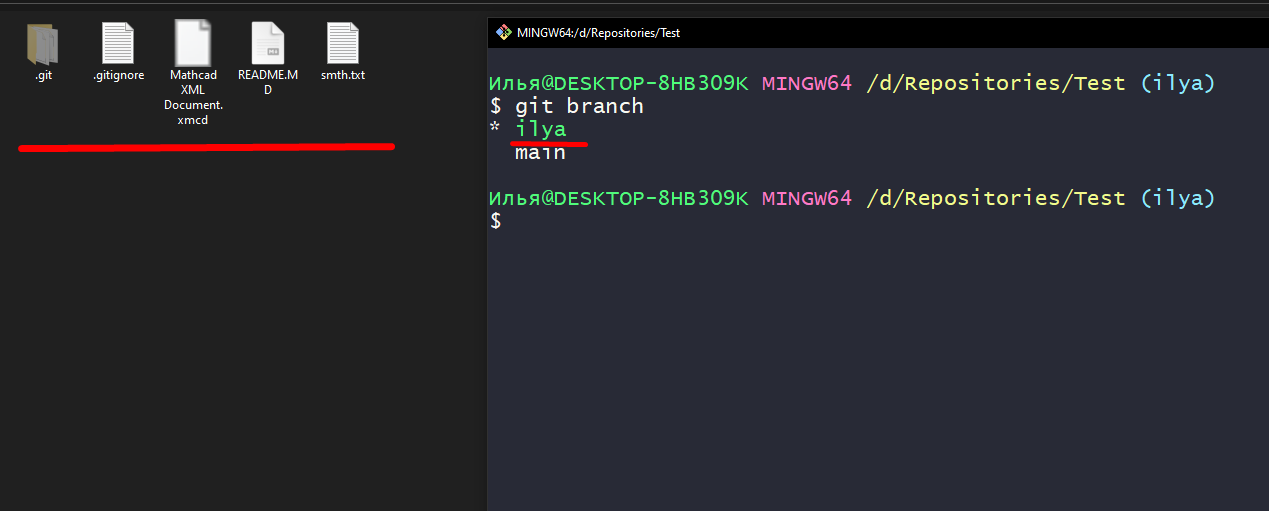
**Додано! ( у вас в папці цих файлів немає)**

**-----------------------------------------------------------------------------------------------**

**Ось такий вигляд має папка, коли ви знаходитесь на гілці main :**



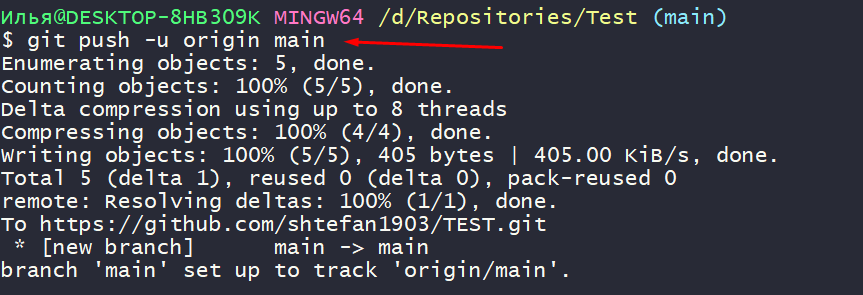
**Ось такий вигляд має папка, коли ви знаходитесь на іншій гілці :**



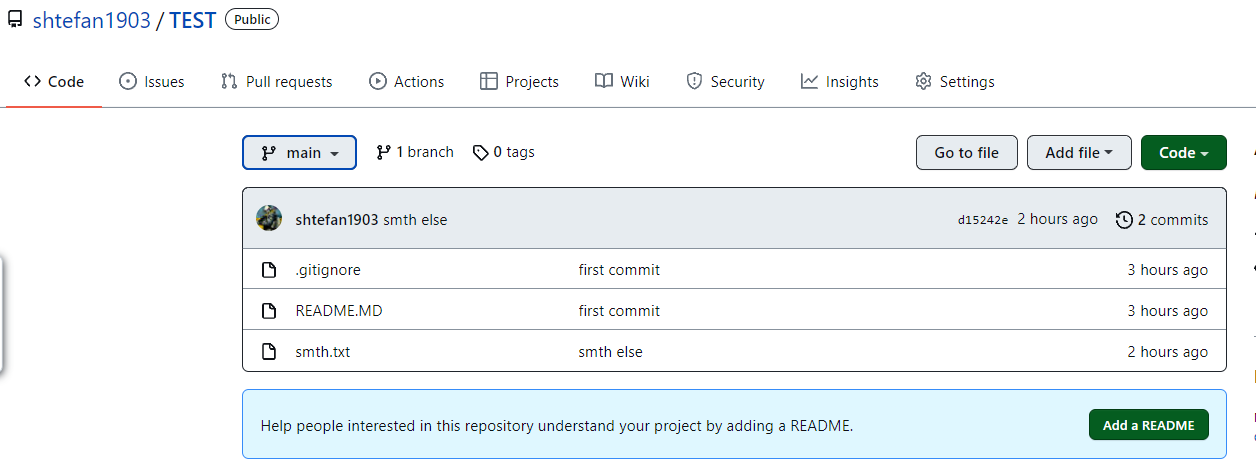
**-----------------------------------------------------------------------------------------------**

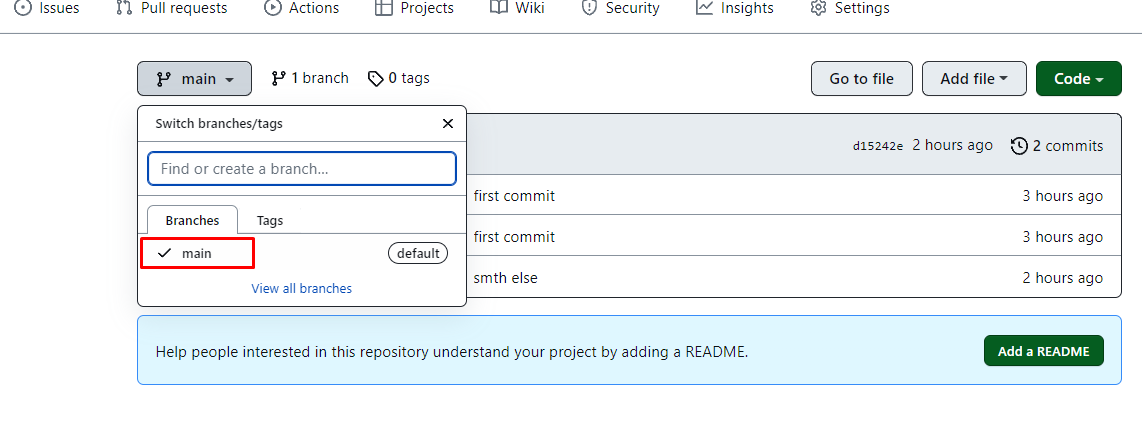
Тепер робимо **push**, тобто заштовхуємо усі наші файли до репозиторію GitHub:

**git push -u origin main** , де **main**  - гілка, в яку будуть запушені файли.



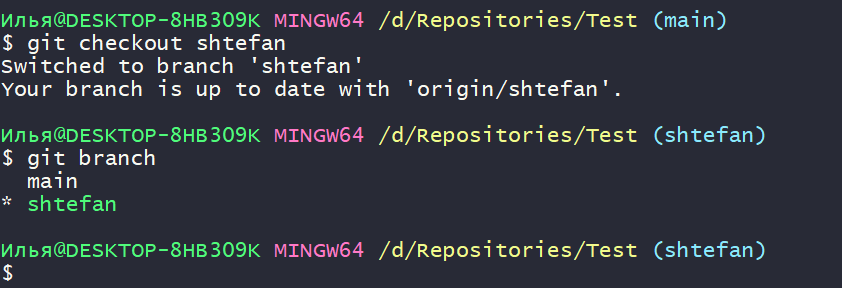
Оновіть сторінку і побачите результат.



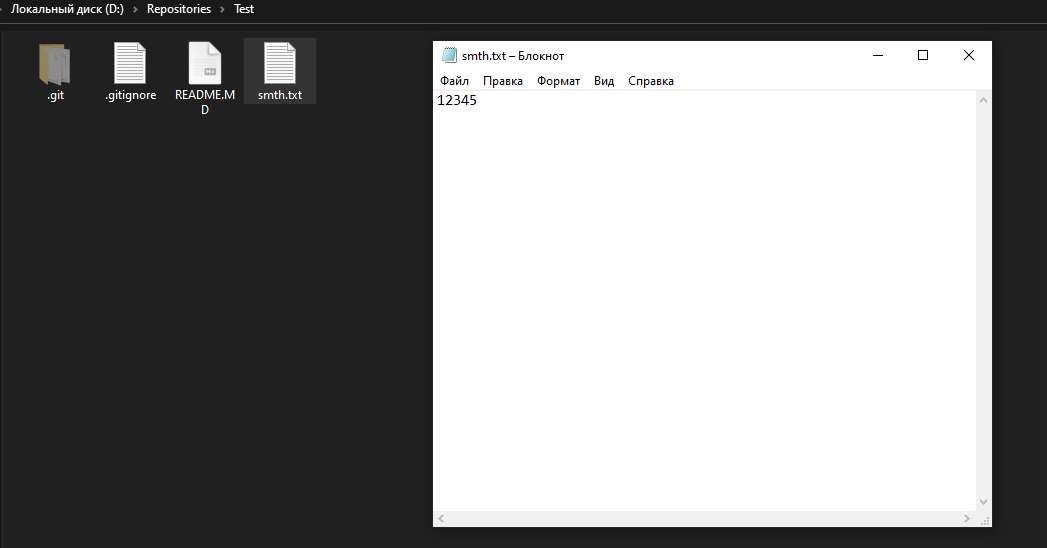


Як ми бачимо, у нас одна гілка, хоча ми створювали дві.

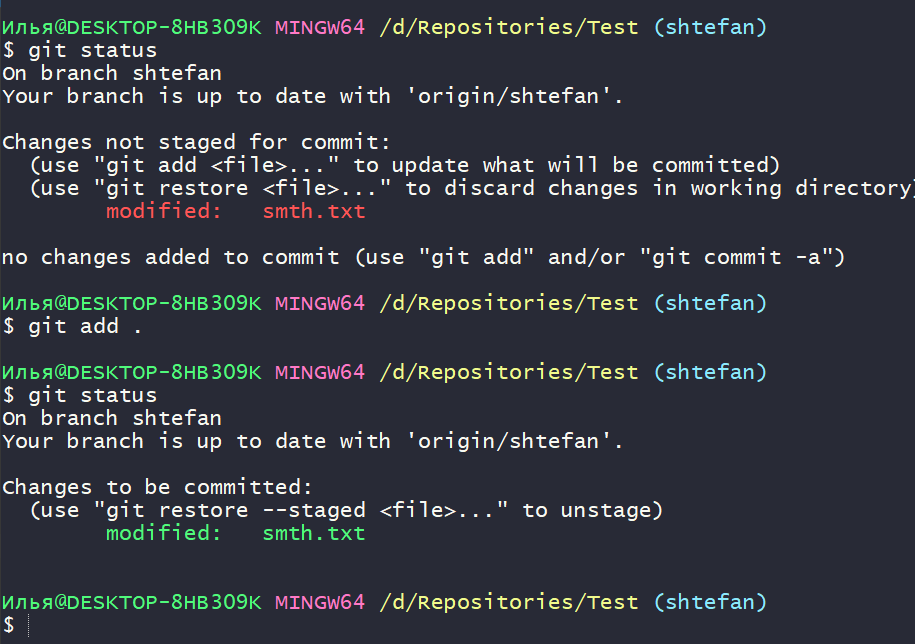
Другої гілки немає через те, що вона порожня, тому додаємо, або змінюємо **вже існуючий текстовий файл,** але спочатку перейдемо на ту гілку, в яку ми хочемо зробити push, у даному випадку на **shtefan** .

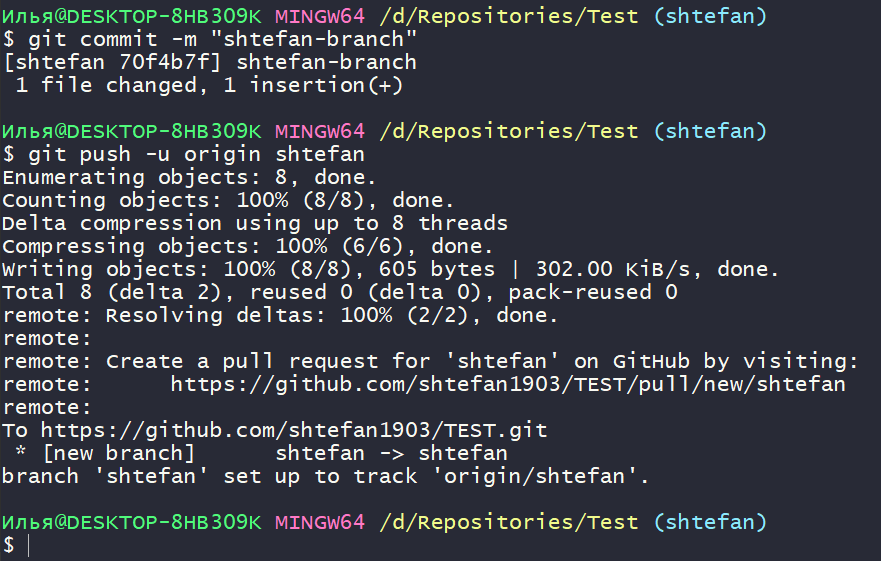


Оновлюємо **текстовий файл**.

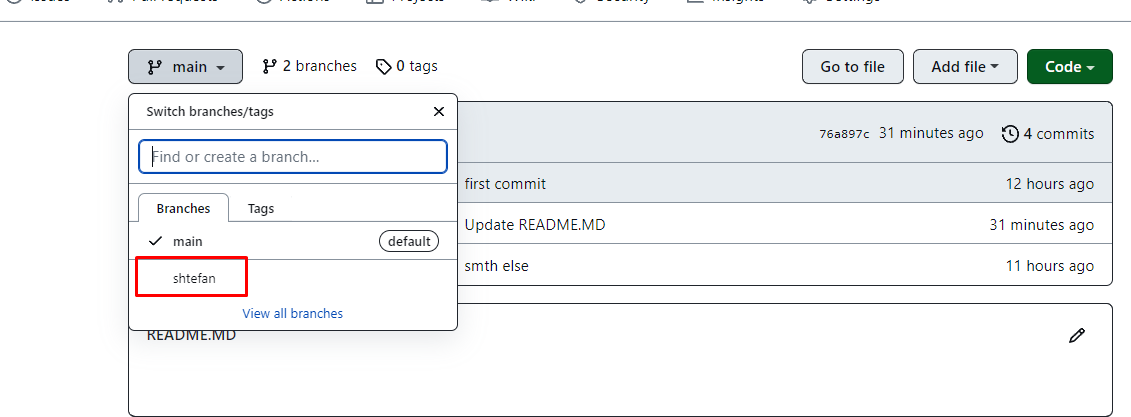


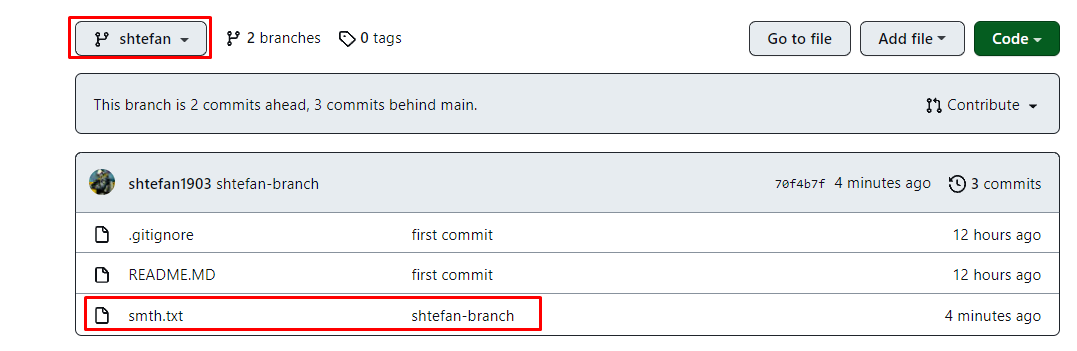
Зберігаємо зміни. Перевіряємо статус. Додаємо змінений файл. Перевіряємо статус. Комітимо файл. Пушимо на гілку **shtefan** .





Оновлюємо сторінку GitHub.



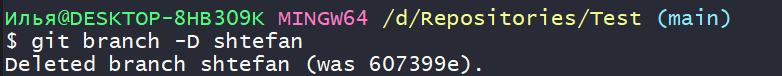


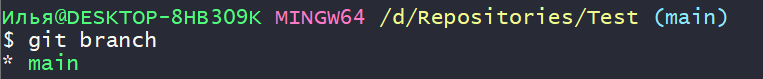
Гілка успішно додалася.

Тепер давайте видалимо гілку **shtefan** з **локального** та **віддаленого** репозиторію:

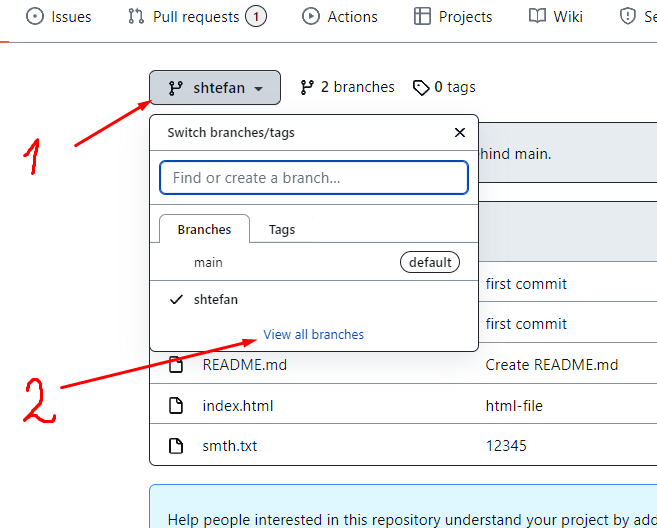
Щоб видалити гілку з **локального** репозиторію потрібно прописати наступну команду **git branch -D ІМ’Я-ГІЛКИ** :

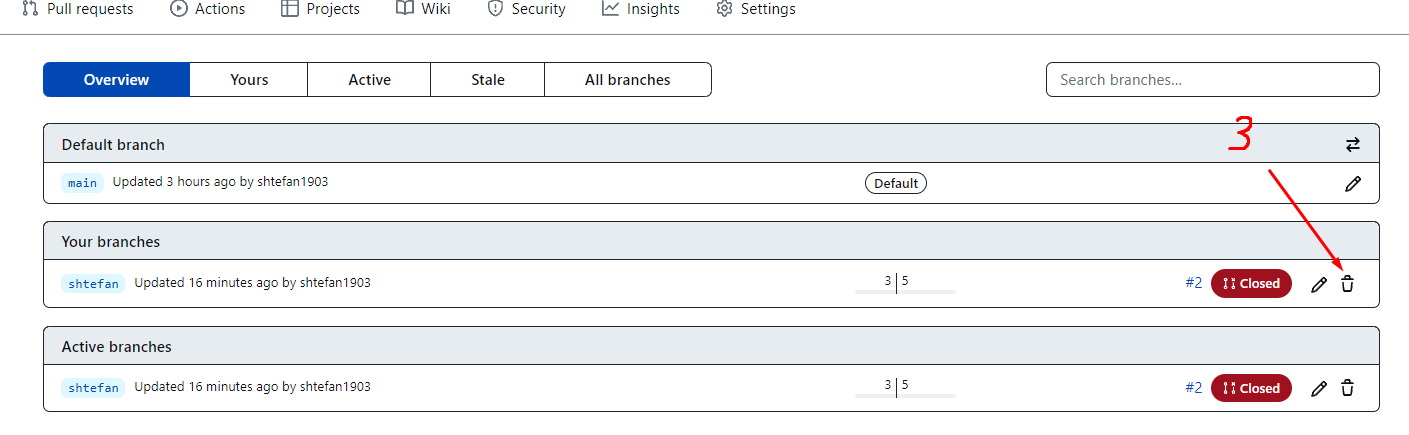
**! ! !**  Неможливо видалити ту гілку, на якій ви знаходитесь, тому переконайтеся що ви на іншій гілці **! ! !**

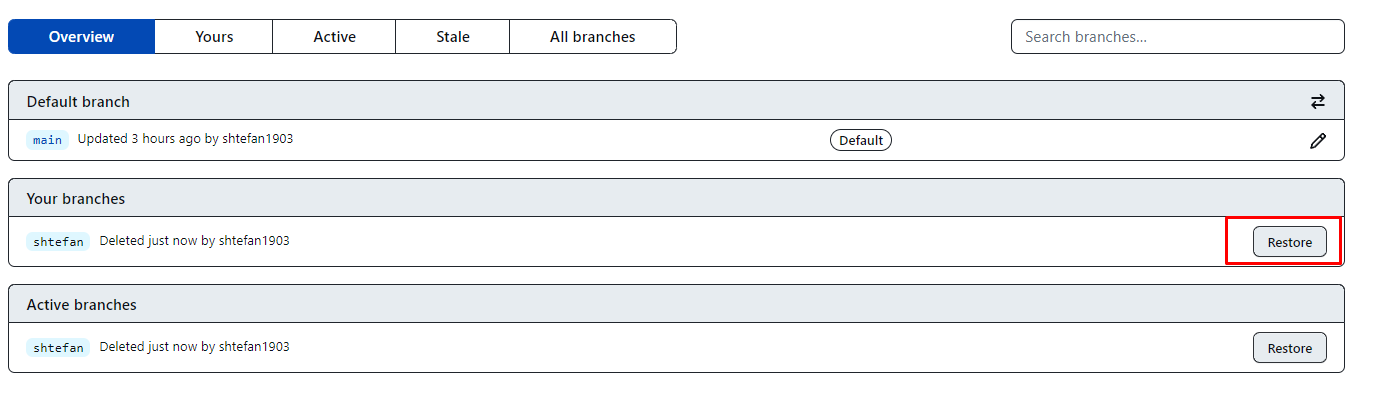




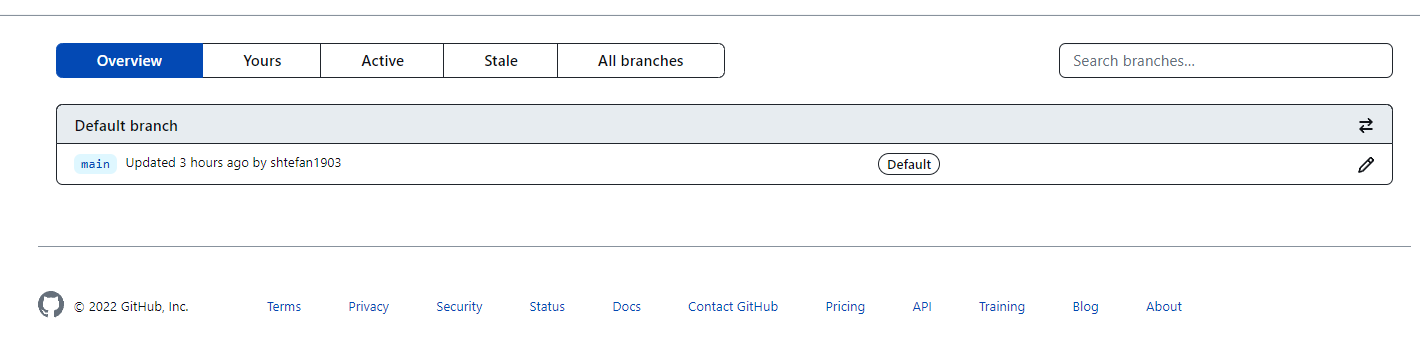
Щоб видалити гілку з **віддаленого** репозиторію потрібно зробити так:







Якщо ви випадково натиснули на “видалити” , то у вас ще є шанс повернути її назад, просто треба натиснути “**Restore**”. Але наша задача її остаточно видалити, тому просто оновлюємо сторінку:



Тепер ви знаєте базу Git.