

ДНІПРОПЕТРОВСЬКИЙ ОБЛАСНИЙ ЦЕНТР ЗАЙНЯТОСТІ
ДНІПРОПЕТРОВСЬКИЙ ЦЕНТР ПРОФЕСІЙНО-ТЕХНІЧНОЇ ОСВІТИ
ДЕРЖАВНОЇ СЛУЖБИ ЗАЙНЯТОСТІ



А. В. КУЧАЙ

РОБОТА З GIT

посібник для слухачів

курсів цільового призначення

«Основи розробки web-сайтів»



Дніпро

2018

Посібник «Робота з GIT» для слухачів курсів цільового призначення «Основи розробки web-сайтів» / укладач: Кучай А.В., – Дніпро: ДЦПТО ДСЗ. 2018. – 30 с.

Посібник містить стислу інформацію про можливості сервісу github.com, роботу з репозиторієм, описано способи використання хостингу сайтів на github. Матеріал посібника буде корисним слухачам курсів цільового призначення «Основи розробки web-сайтів».

Розглянуто та схвалено на засіданні методичної комісії інформації, комунікації та переробної промисловості Дніпропетровського центру професійно-технічної освіти державної служби зайнятості (протокол №9 від 07.09.18).

ЗМІСТ

ВСТУП	4
1. РОБОТА З САЙТОМ GITHUB.COM	5
1.1. Сутність поняття git.....	5
1.2. Реєстрація на сайті github.com.....	7
1.3. Створення репозиторію.....	8
2. РОБОТА З РЕПОЗИТОРІЄМ	9
2.1. Створення нового файлу.....	9
2.2. Завантаження файлів до репозиторію.....	11
2.3. Створення нової гілки.....	12
2.4. Злиття гілок.....	13
2.5. Завантаження репозиторію.....	15
3. РОБОТА З GIT У КОНСОЛІ	17
3.1. Створення локального git-репозиторію.....	17
3.2. Завантаження локального репозиторію до віддаленого.....	19
3.3. Додавання іншого користувача до репозиторію.....	20
3.4. Клонування репозиторіїв.....	23
4. ХОСТИНГ САЙТІВ НА GITHUB	25
4.1. Перший спосіб використання хостингу сайтів на Github.....	25
4.2. Другий спосіб використання хостингу сайтів на Github.....	27
ОПОРНА КАРТА «Можливості Github»	29
ІНФОРМАЦІЙНІ ДЖЕРЕЛА	29

ВСТУП

Сервіс Github є ефективним інструментом для спільної командної роботи над проектом розробки web-продуктів. Володіння навичками використання сервісу дозволяє web-розробнику ефективніше працювати з сайтами. Така компетенція є вимогою роботодавців до сучасних розробників web-продуктів.

Посібник «Робота з GIT» створений з метою надання допомоги слухачам курсів підвищення кваліфікації «Основи розробки web-сайтів» під час вивчення теми щодо можливостей сервісу Github. Матеріал посібника може бути цікавим та корисним web-розробникам сайтів, усім, хто займається самоосвітою.

У посібнику описано можливість використання сервісу Github як безкоштовного хостингу сайтів.

Розробка містить теоретичний матеріал в інструктивній формі, доповнена ілюстраціями, узагальненою опорною схемою «Можливості Github», містить посилання на веб-ресурси, які дозволяють більш детально розглянути запропоновані для вивчення питання. Для закріплення вивченого матеріалу в посібнику наведено запитання і практичні вправи.

1. РОБОТА З САЙТОМ GITHUB.COM

1.1. Сутність поняття git



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.

Система має ряд інтерфейсів: наприклад, gitk та git-gui.

Віддалений доступ до репозиторіїв Git забезпечується git-демоном, SSH або HTTP сервером. TCP-сервіс git-daemon входить у дистрибутив Git і є разом з SSH найпоширенішим та надійним методом доступу. Метод доступу HTTP, незважаючи на ряд обмежень, дуже популярний у контрольованих мережах, тому що дозволяє використання конфігурацій мережевих фільтрів, що існують.

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 файловою 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).

Локальні операції

На відміну від Subversion, де без підключення до мережі Інтернет можна лише редагувати файли, але зберегти зміни у вашу базу даних неможливо (оскільки вона відключена від репозиторію), у Git більшість дій можна виконувати без використання Інтернет-підключення на локальній файлової системі. Будь-який коміт спочатку робиться локально, потім вивантажується у віддалений репозиторій. Уся історія змін локально зберігається і, за необхідності, вивантажується у віддалений репозиторій.

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.

Галуження (гілки)



Галуження — це розмежування від основної лінії розробки. Git дозволяє створити декілька гілок і перемикається 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`.

Зливання та перебазування 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.

Заповіть форму (рис. 1) та натисніть на кнопку *Sign up for GitHub*.



Рис.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. 2).

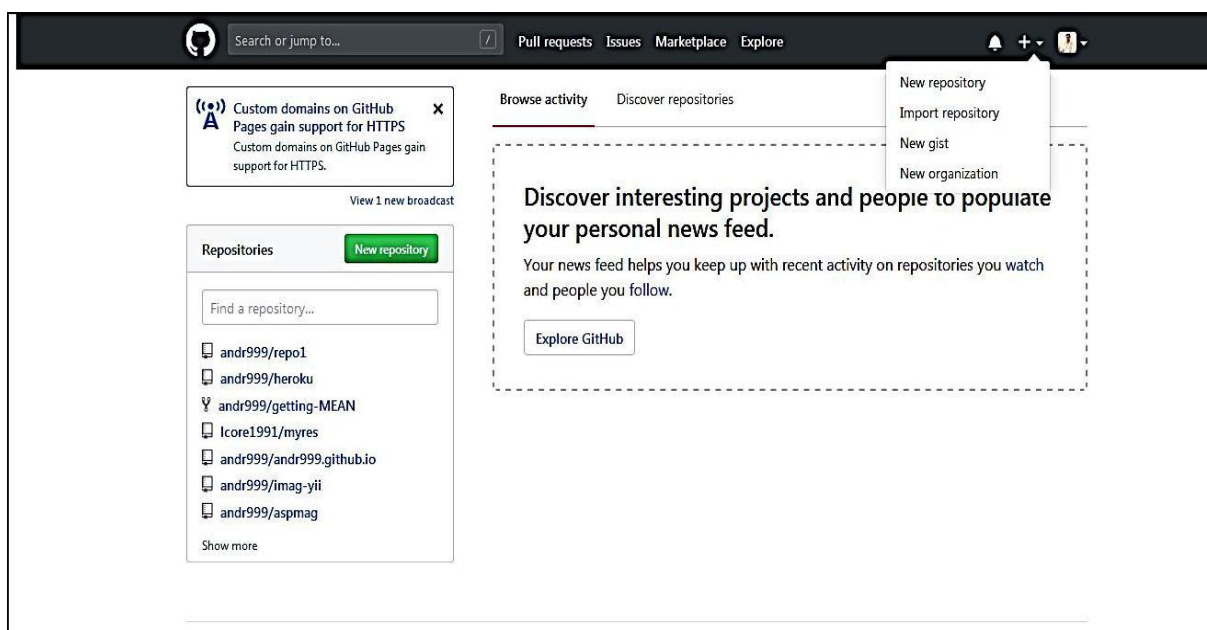


Рис. 2. Створення нового репозиторію

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

Після створення репозиторію ви потрапите на сторінку репозиторію. Поки що тут є один файл `Redme.md`, котрий було створено разом з репозиторієм (рис. 3).

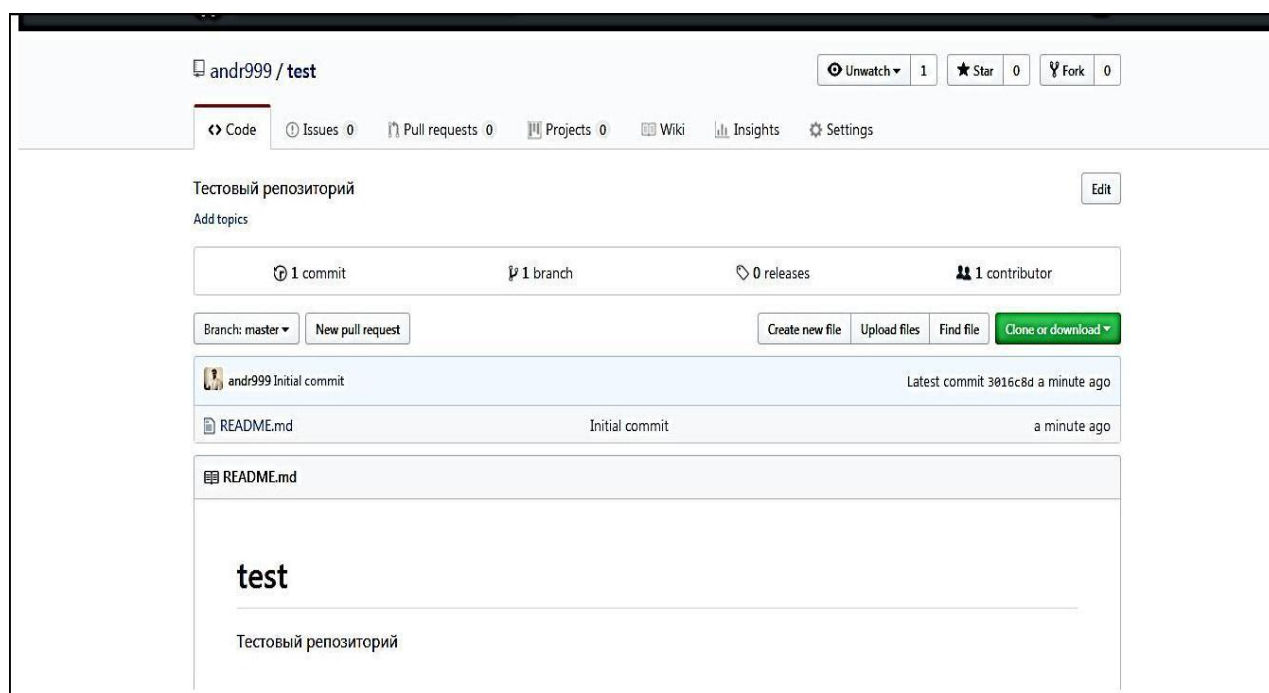


Рис. 3. Сторінка репозиторію

Тема 2. РОБОТА З РЕПОЗИТОРІЄМ

2.1. Створення нового файлу

Для додавання нового файлу у репозиторій натисніть кнопку *Create new file* (рис. 4).

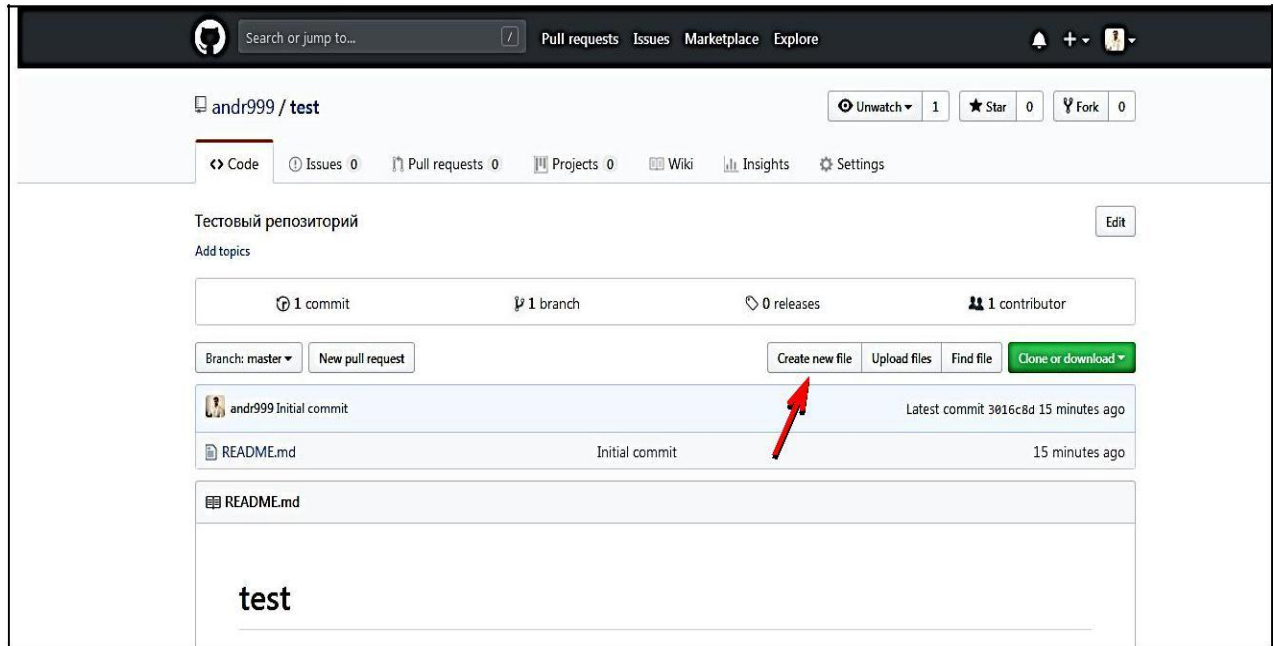


Рис. 4. Створення нового файла у репозиторії

На сторінці створення файла введіть назву файла та відредагуйте його зміст (рис. 5).

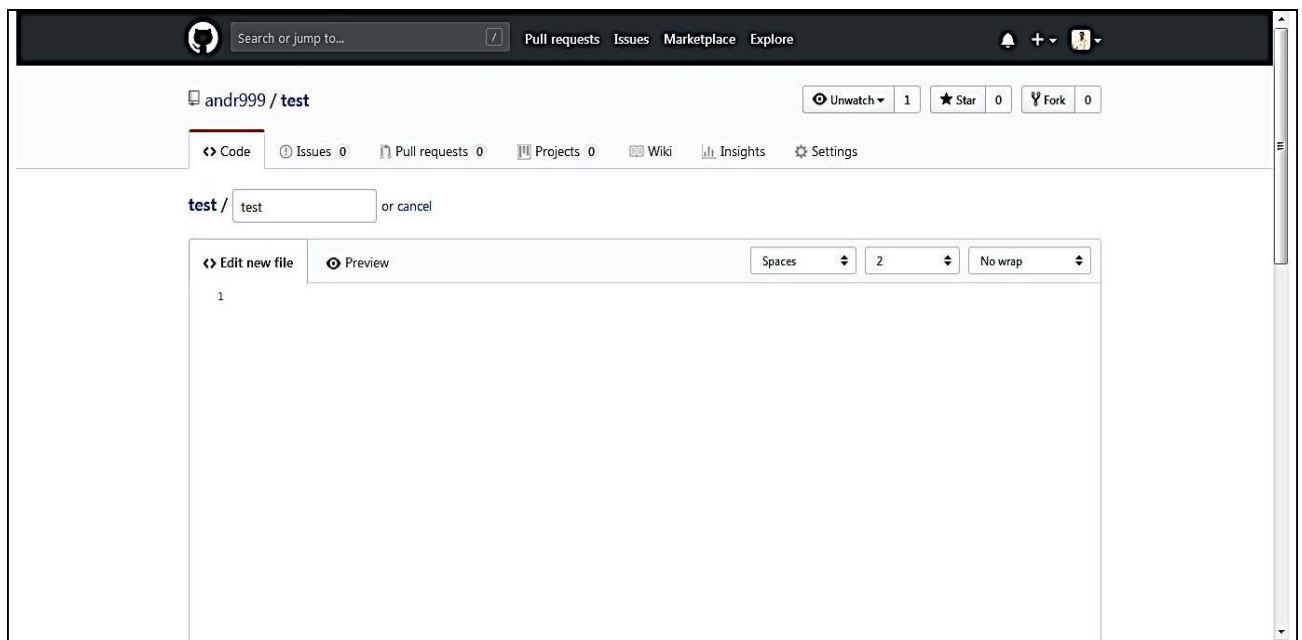


Рис. 5. Редагування файлу

Після чого натисніть на кнопку *Commit new file*, при цьому новий файл буде створено (рис. 6).

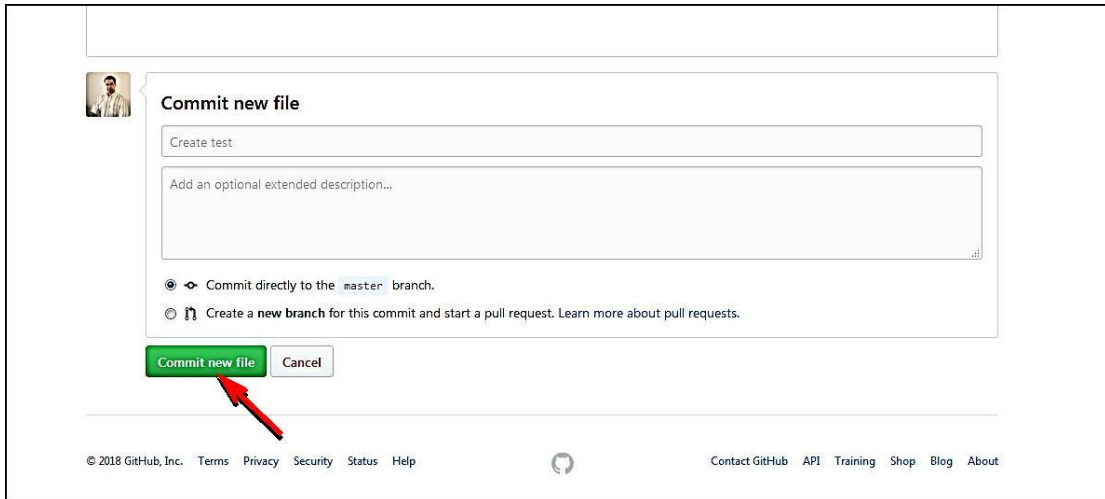


Рис. 6. Підтвердження змін у файлі

2.2. Завантаження файлів до репозиторію

Для завантаження файлів до репозиторію потрібно натиснути кнопку *Upload files* (рис. 7).

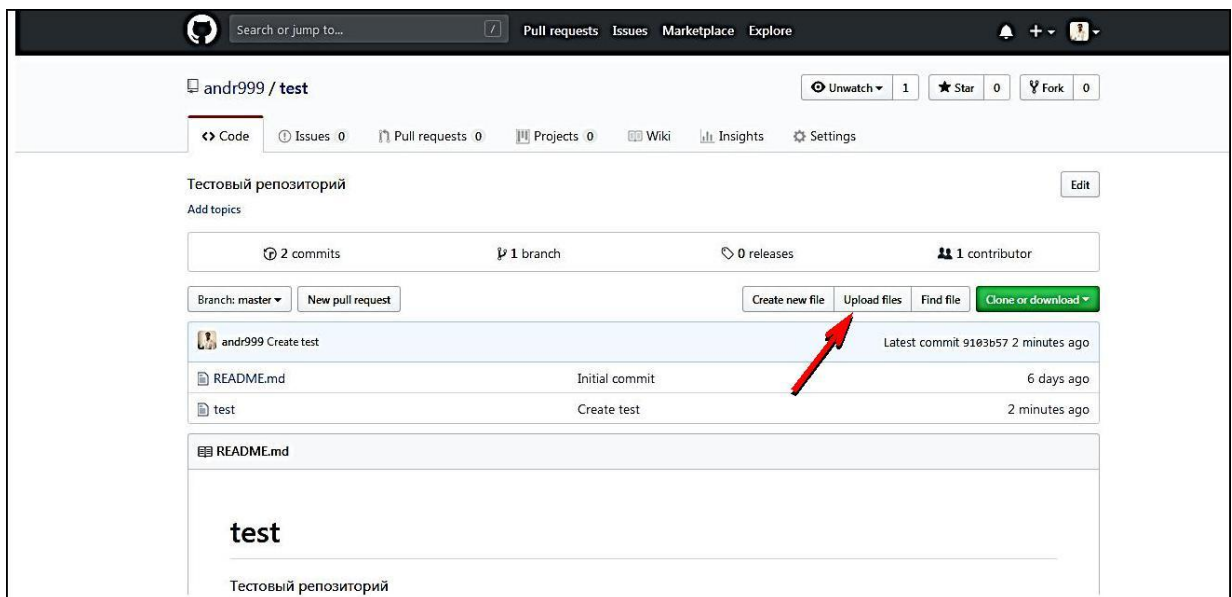


Рис. 7. Завантаження файлів до репозиторію

На наступній сторінці натисніть на посилання *Choose your files* (рис. 8).

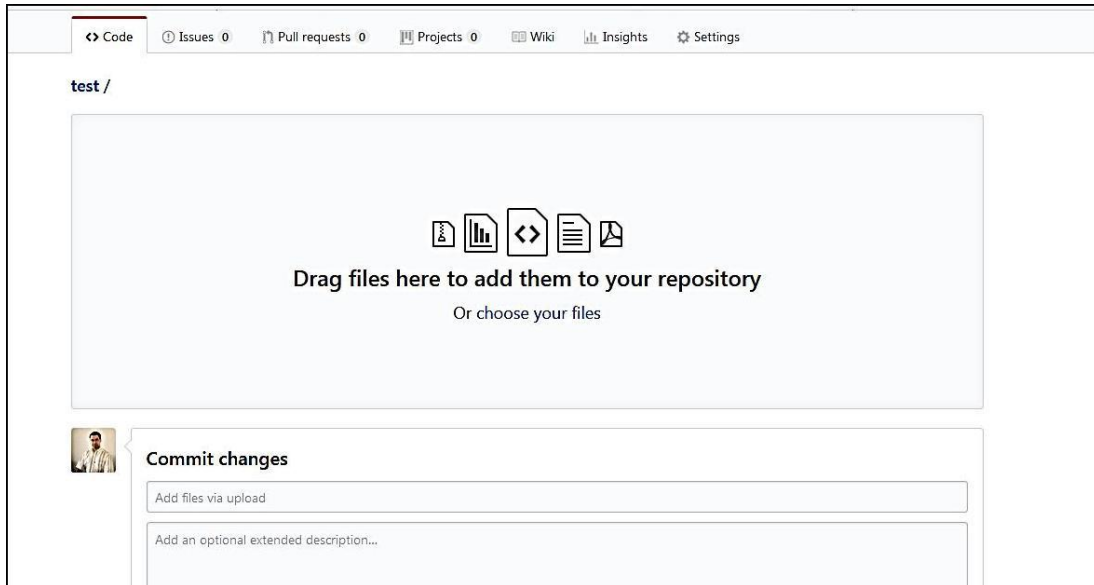


Рис. 8. Обрання файлів для завантаження

У вікні, що відкриється, оберіть файли для завантаження на Github і натисніть на кнопку *Відкрити*, при цьому файли завантажуться до репозиторію.

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.

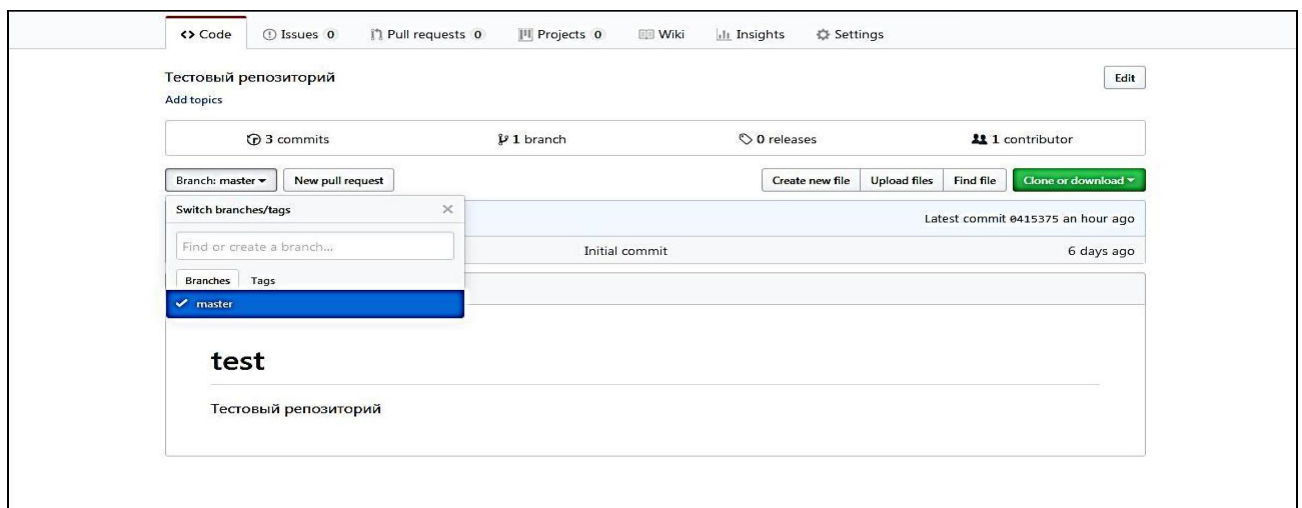


Рис. 9. Створення нової гілки

Для створення гілки потрібно натиснути на кнопку *Branch master* та у списку, що випадає, ввести ім'я нової гілки (рис. 9).

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. 10).

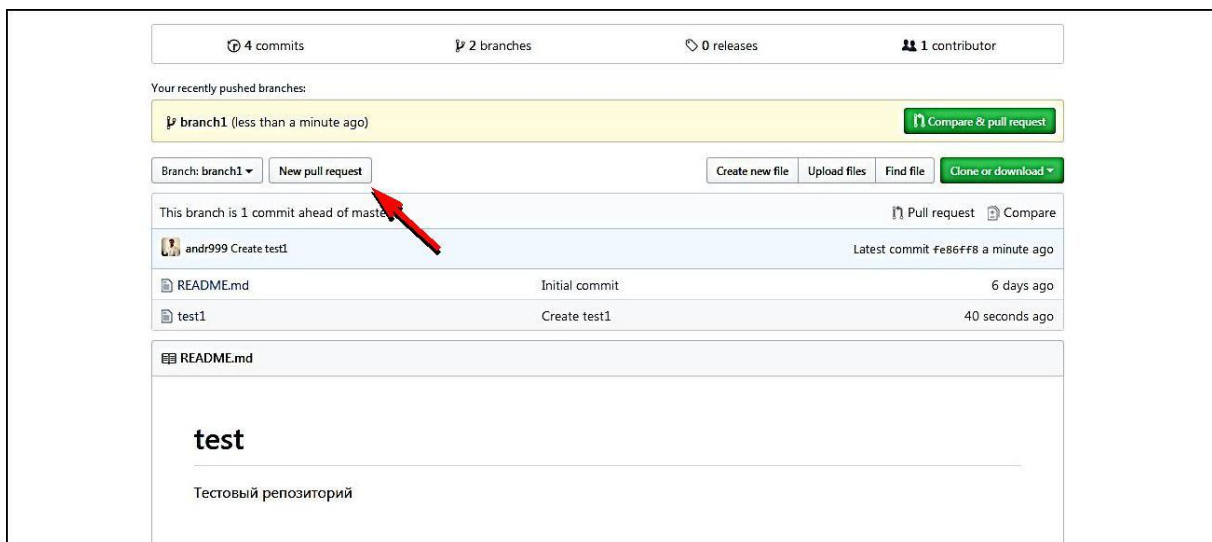


Рис. 10. Створення запиту на злиття гілок

На сторінці запиту на злиття буде спочатку перевірено можливість злиття гілок. Після перевірки та можливості злиття натисніть на кнопку *Create pull request* (рис. 11).

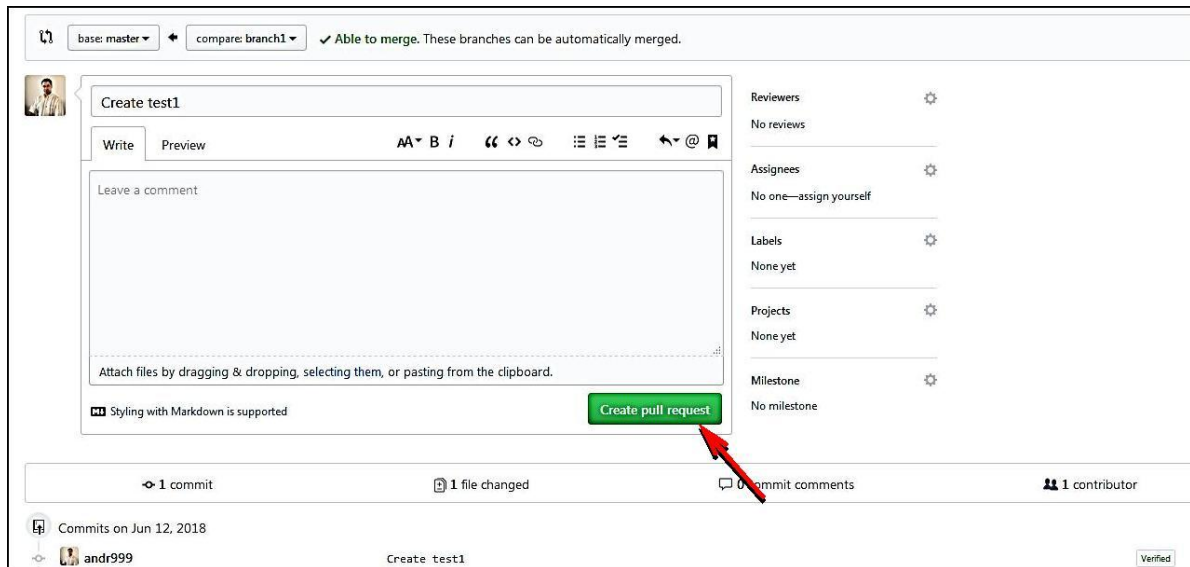


Рис. 11. Злиття гілок

На сторінці підтвердження злиття гілок буде також задіяно перевірку можливості злиття гілок і, якщо перевірка успішна, натисніть на кнопку *Merge pull request* (рис. 12).

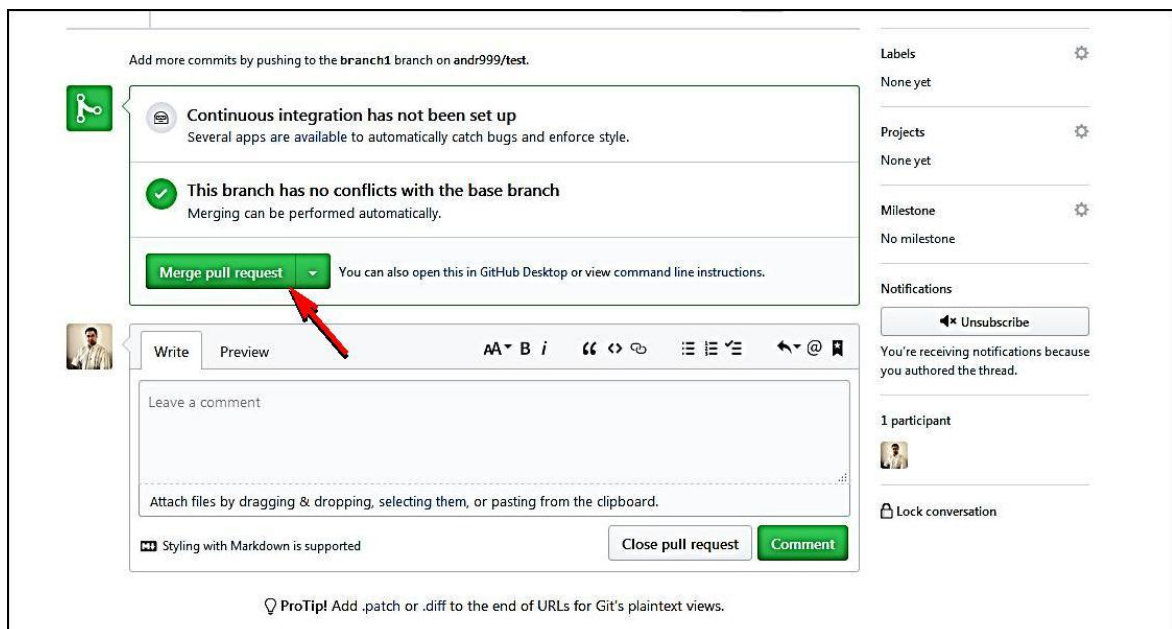


Рис. 12. Підтвердження злиття гілок

На наступній сторінці натисніть на кнопку *Confirm merge*. Після цього розпочнеться механізм злиття гілок. Після закінчення злиття необхідно видалити або залишити гілку, якщо вона ще потрібна (рис. 13).

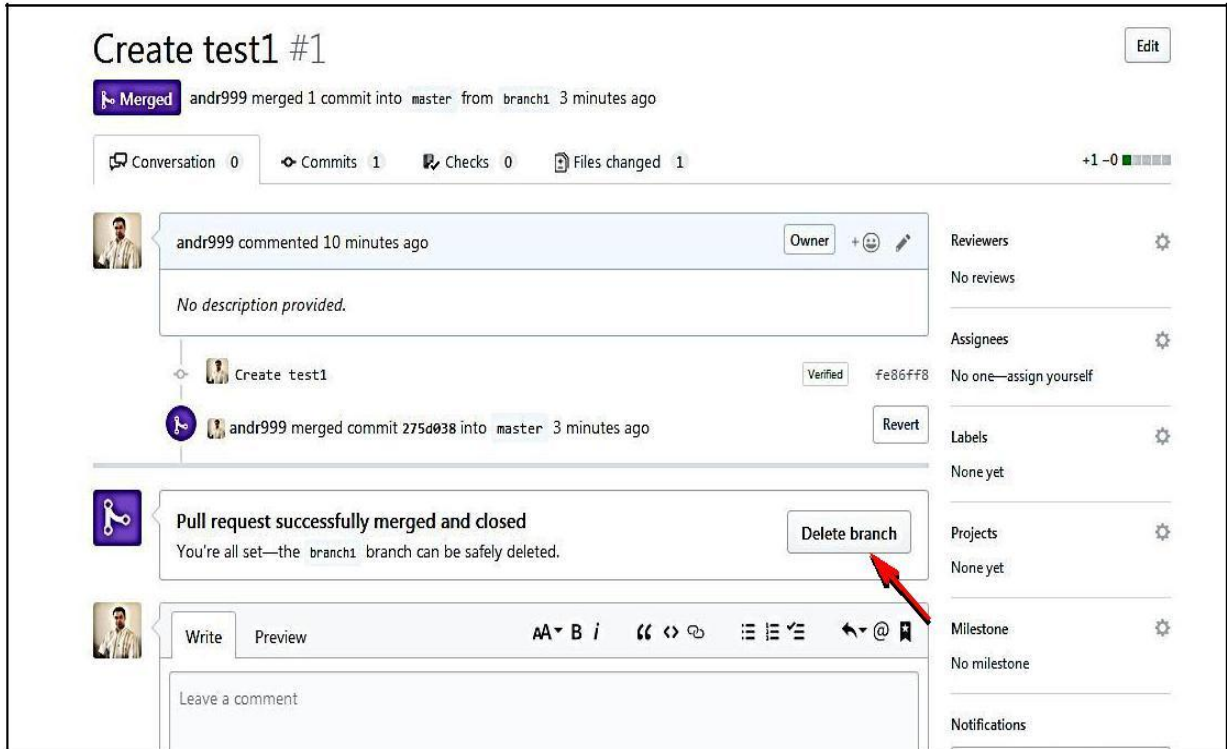


Рис. 13. Видалення гілки

2.5. Завантаження репозиторію

Для завантаження репозиторію на комп'ютер необхідно натиснути на кнопку *Clone or Download*, потім *Download ZIP* (рис. 14).

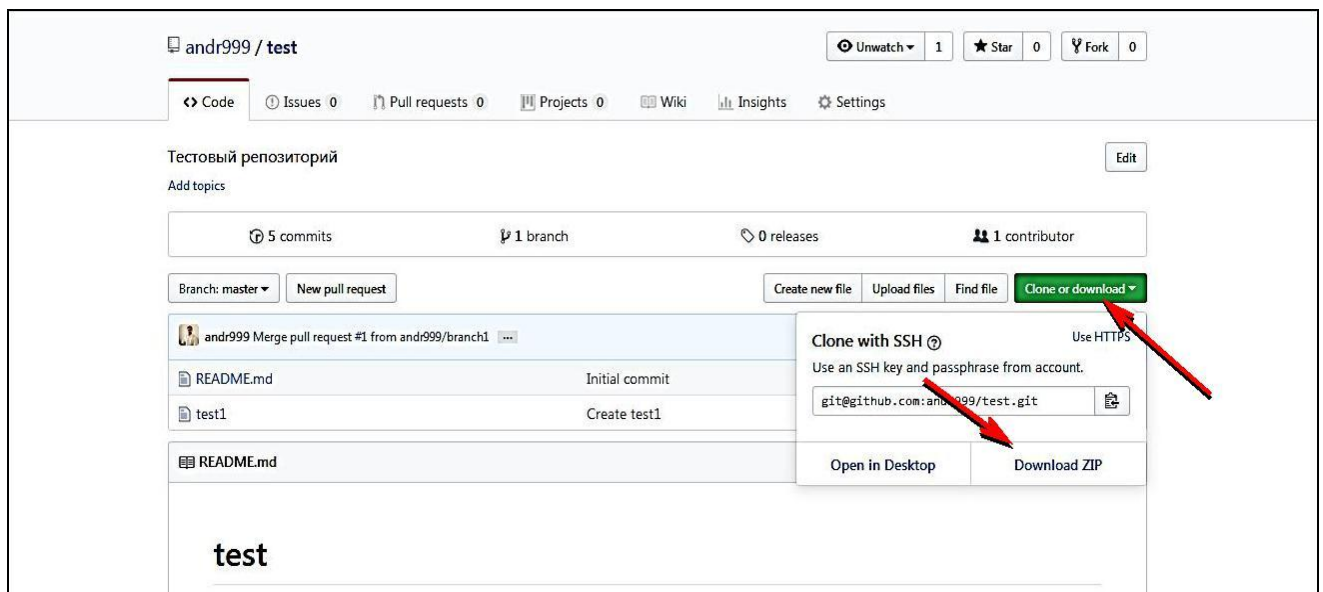


Рис. 14. Завантаження репозиторію

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



Питання для закріплення

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?



Вправи для самостійної роботи

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 GIT У КОНСОЛІ

3.1. Створення локального git-репозиторію

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

Оскільки вбудована консоль у систему Windows не підтримує утиліту Git, необхідно встановити програму Cygwin або іншу програму, котра підтримує цю утиліту. Розберемо на прикладі консолі, яка є у програмі OpenServer.

Запустіть програму OpenServer, потім у правому нижньому куті екрана натисніть на зелений прапорець та з меню, що випадає, оберіть Додатково – Консоль (рис. 15).

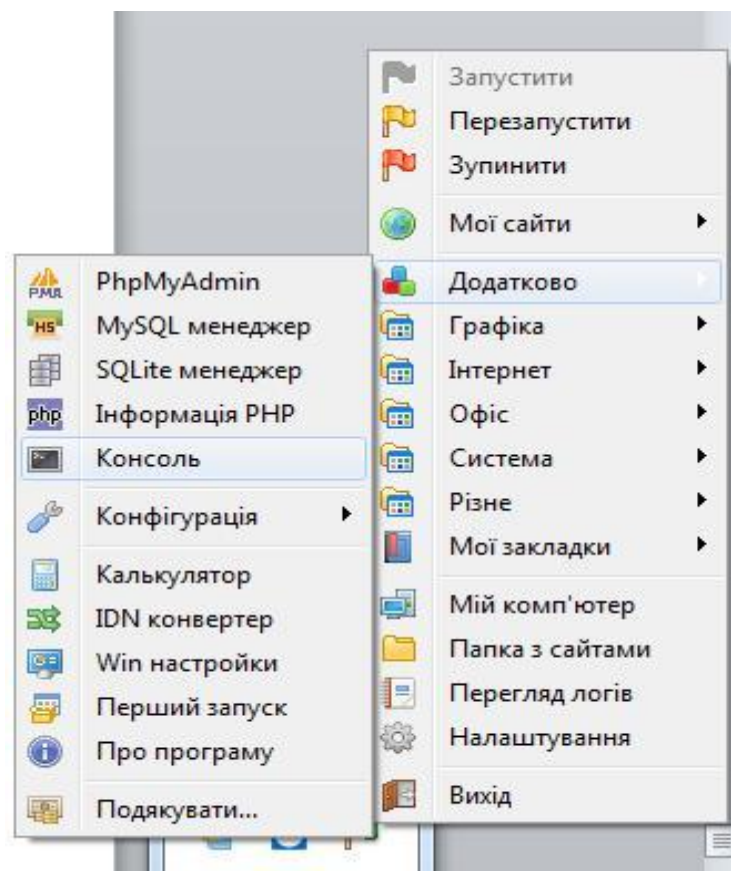


Рис. 15. Запуск консолі OpenServer

При цьому запуститься консоль, у якій потрібно вводити команди.

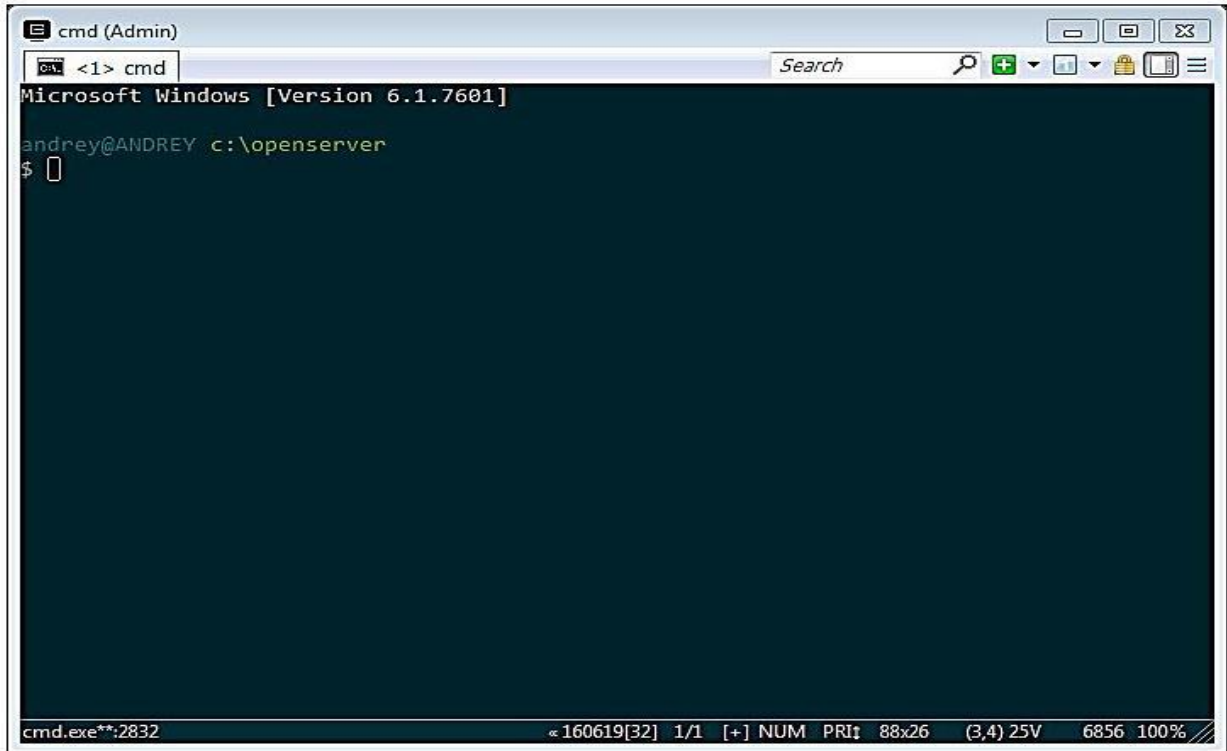


Рис. 16. Консоль OpenServer

Спочатку потрібно ввести дані для входу на сайт github.com. Для цього у консолі необхідно ввести команди (рис. 16):

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 ./domains/site.loc

Знаходячись у папці з проектом, введіть команду ***git init***, таким чином у папці з проектом буде створено приховану папку `.git`.

Додайте усі файли проекту до репозиторію. Для цього введіть команду

git add.

Підтвердіть зміни командою ***git commit -m "Перший коміт"***.

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/andr999/myrep3.git,

де замість адреси *https://github.com/andr999/myrep3.git*, потрібно ввести свою, подивитись її можна на сторінці репозиторію (рис. 17).

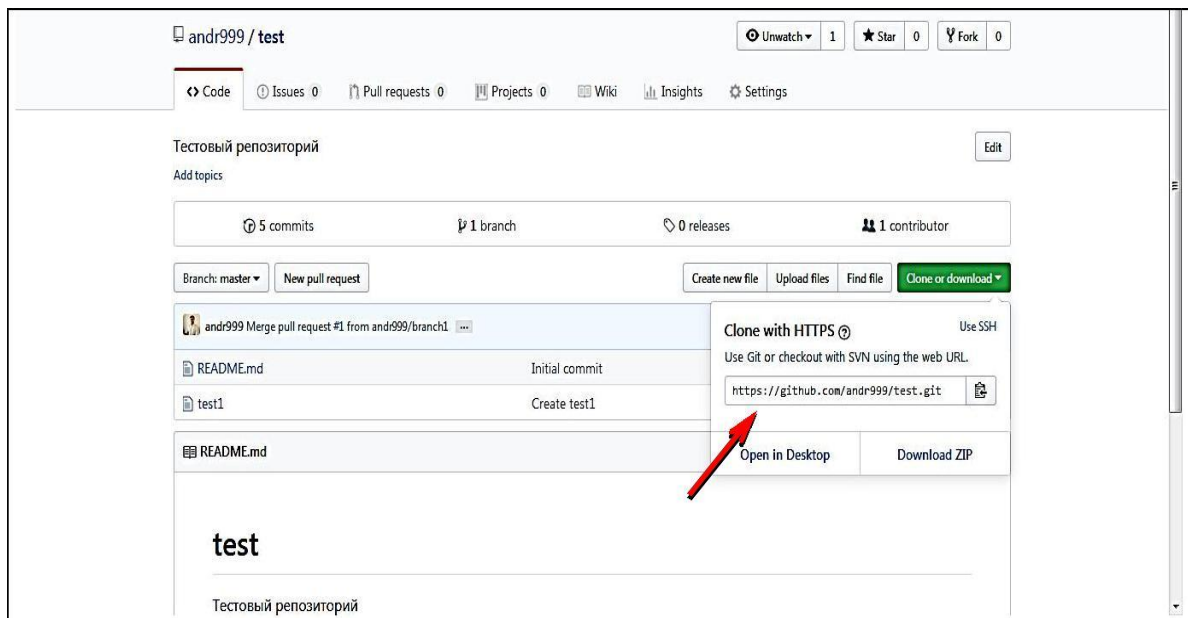


Рис. 17. Клонування репозиторію

Перевірити усі віддаленні репозиторії можна командою

git remote -v

Далі завантажте усі зміни з віддаленого репозиторію командою

git pull rep master --rebase

Після цієї команди можна завантажити усі зміни до віддаленого репозиторію командою

git push rep master

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 *Collaborators* (рис. 18).

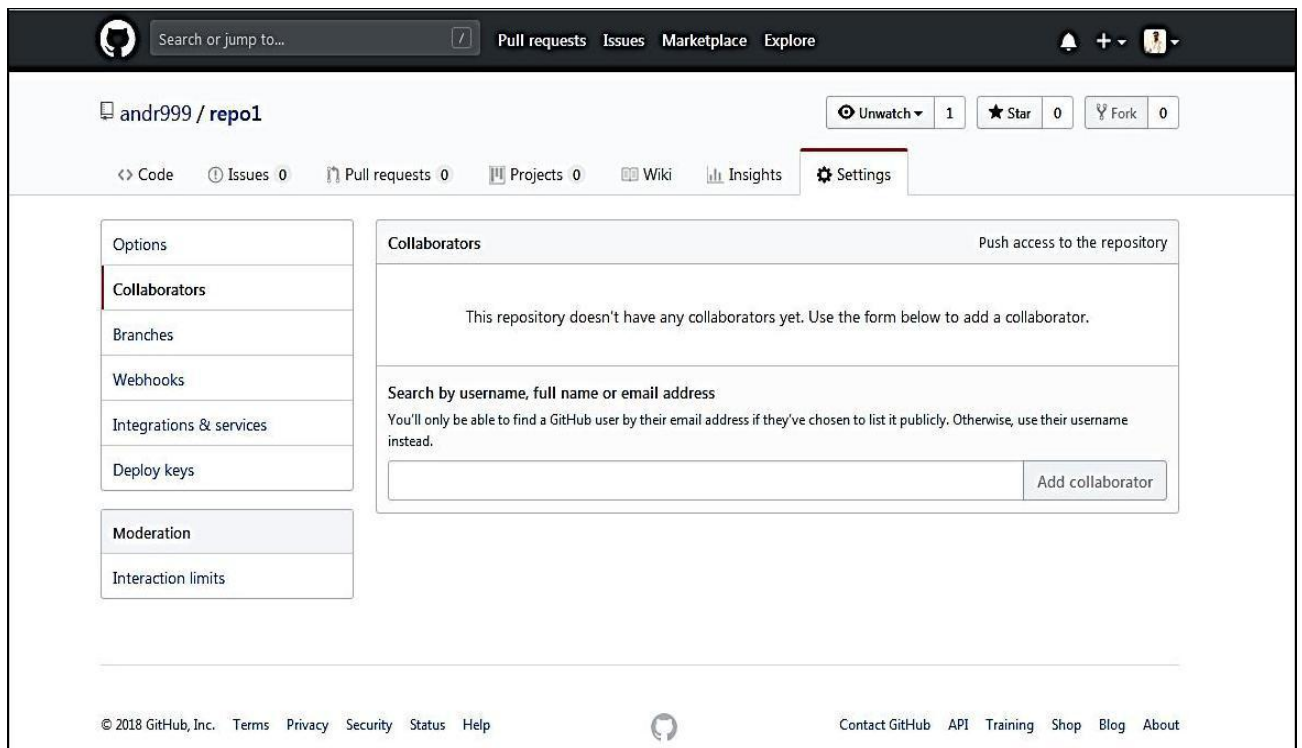


Рис. 18. Додавання іншого користувача до репозиторію

У текстовому полі введіть логін або e-mail користувача, який залучається до участі у проекті, та натисніть на кнопку *Add collaborator* (рис. 19).

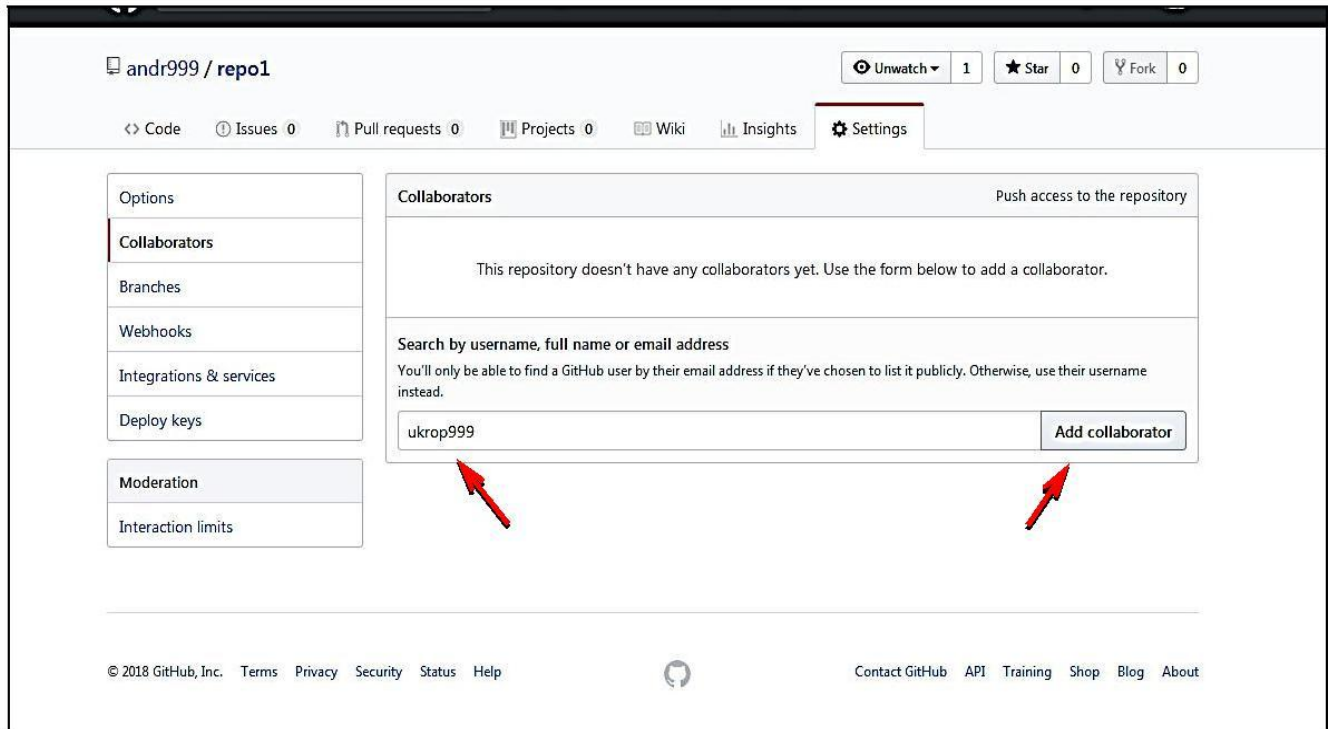


Рис. 19. Введення логіну користувача, який додається до репозиторію

Користувачеві, якого ви запросили, на електронну пошту буде надіслано лист із запрошенням (рис. 20).

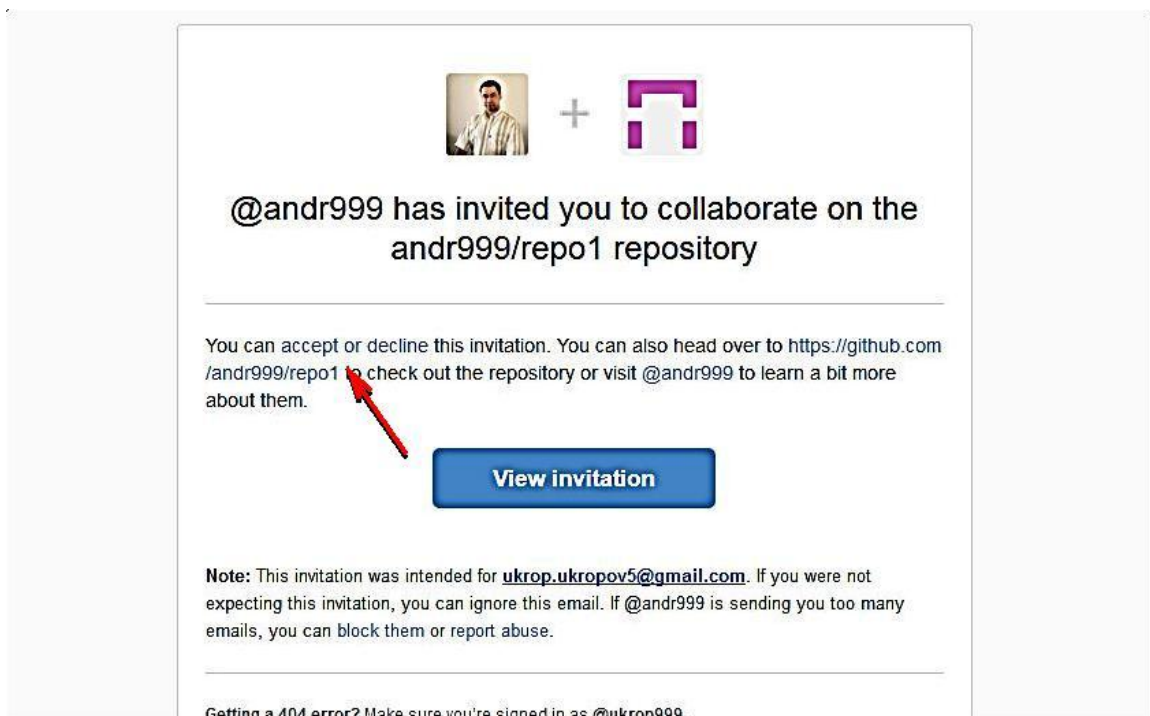


Рис. 20. Лист із посиланням на запрошення

У листі необхідно перейти за посиланням *accept or decline* та прийняти або відхилити запрошення (рис. 21).

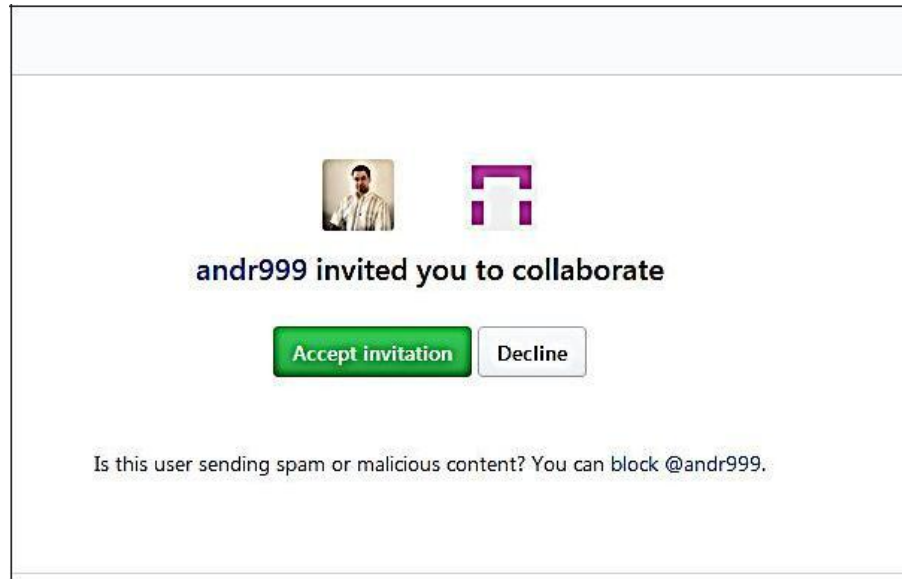


Рис. 21. Запрошення підтверджено

Після натискання кнопки *Accept invitation* відбудеться перенаправлення на сторінку з репозиторієм і з'явиться можливість редагувати файли та папки цього репозиторію (рис. 22).

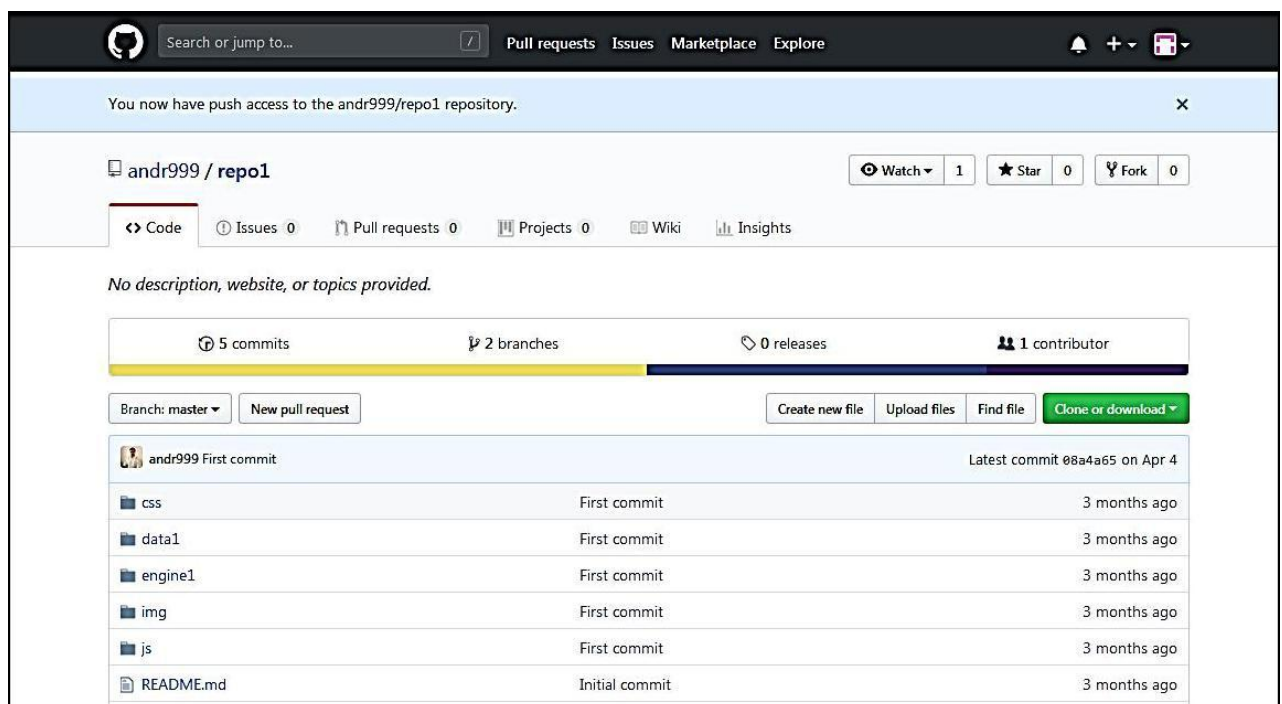


Рис. 22. Сторінка з репозиторієм

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.

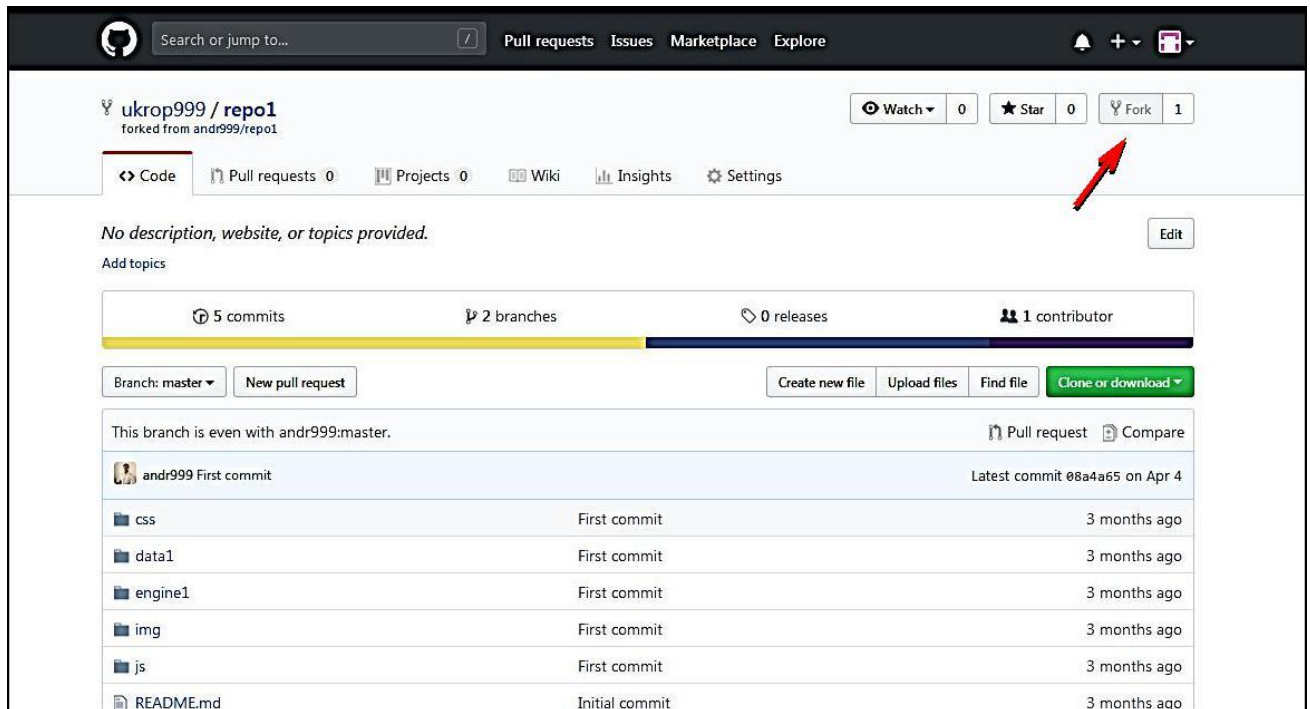


Рис. 23. Копія або форк репозиторію

3.4. Клонування репозиторіїв

Для клонування репозиторію необхідно створити теку, куди його буде клоновано. Наприклад, створюємо теку *project2* у папці *domains* і туди клонуємо репозиторій.

Запустіть консоль. Перейдіть у папку *project2*.

Для цього у консолі введіть команду

```
cd c:/OsPanel/domains/project2
```

Знаходячись у теці *project2*, введіть команду

```
git clone https://github.com/andr999/repo1.git.
```

Адресу репозиторію можна дізнатися, якщо натиснути на кнопку *Clone or Download* (рис. 24).

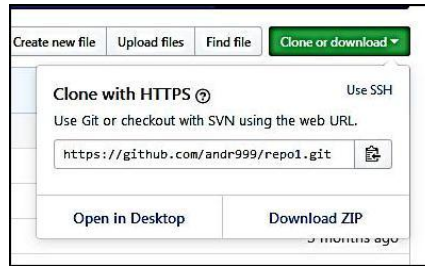


Рис. 24. Клонування репозиторію

При цьому у теку *project2* буде клоновано репозиторій.



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?



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 видаленого репозитрія.
7. Обміняйтесь своїми логіном та паролем з іншим користувачем, додайте його до вашого репозиторію.
8. Клонуйте репозиторій, до якого вас додав інший користувач.

4. ХОСТИНГ САЙТІВ НА GITHUB

Сайт `github.com` можна використовувати як хостинг для сайтів. Підтримуються тільки сайти `html`, `css`, `js`. PHP скрипти та бази даних MySQL не підтримуються.

4.1. Перший спосіб використання хостингу сайтів на Github

Необхідно створити новий репозиторій таким чином, щоб він був однаковим з вашим логіном та мав приставку `github.io` (рис. 25).

Наприклад, якщо ваш логін `andr999`, то й ім'я нового репозиторію повинно бути `andr999.github.io`.

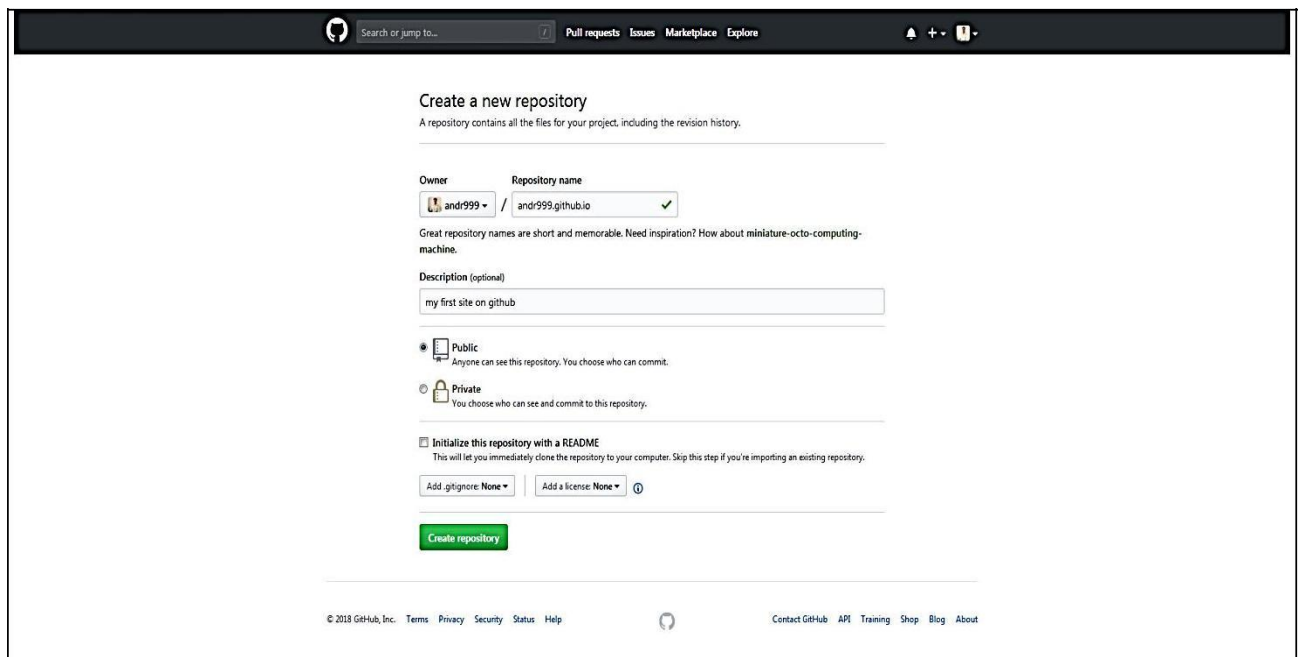


Рис. 25. Створення нового репозиторію

Після цього потрібно завантажити файли вашого репозиторію, натиснувши кнопку *Upload files*.

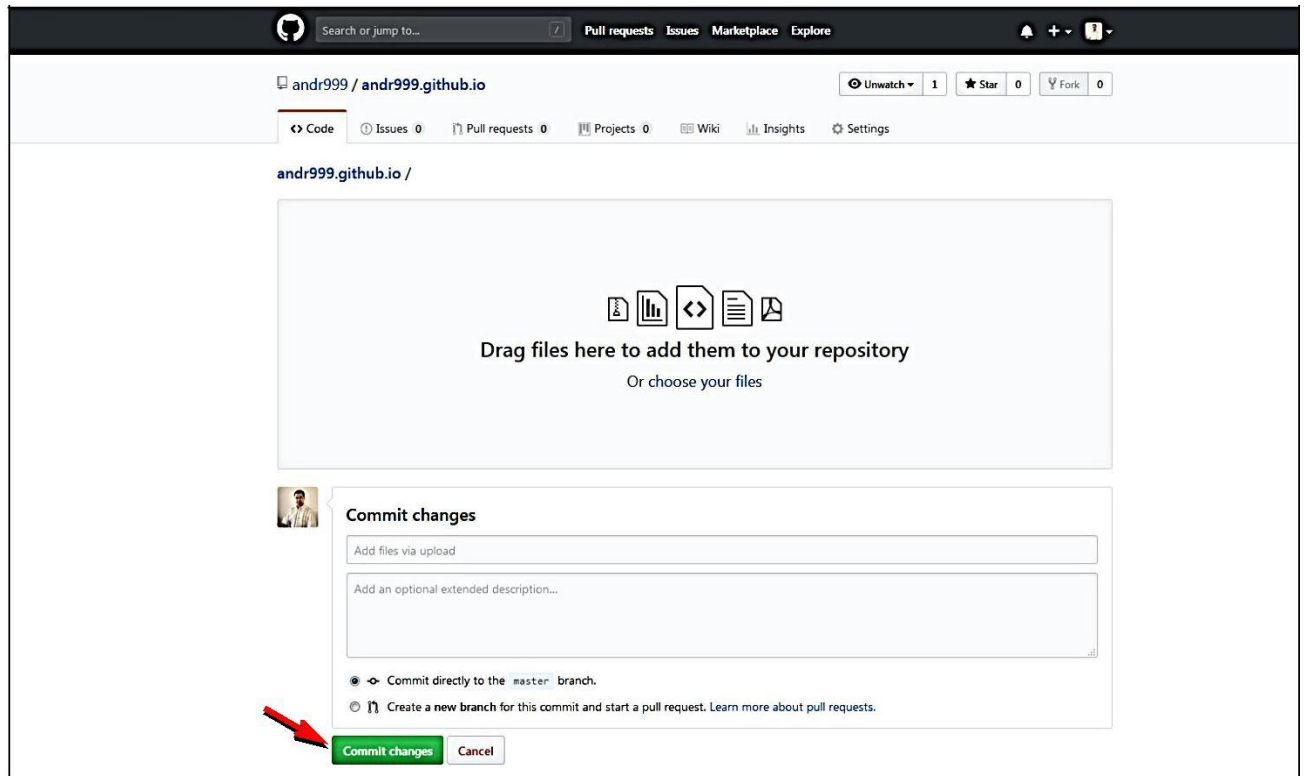


Рис. 26. Завантаження файлів до репозиторію

У цьому вікні натисніть на посилання *Choose your files* та оберіть файли. Після цього натисніть на кнопку *Commit changes* для того, щоб прийняти зміни (рис. 26).

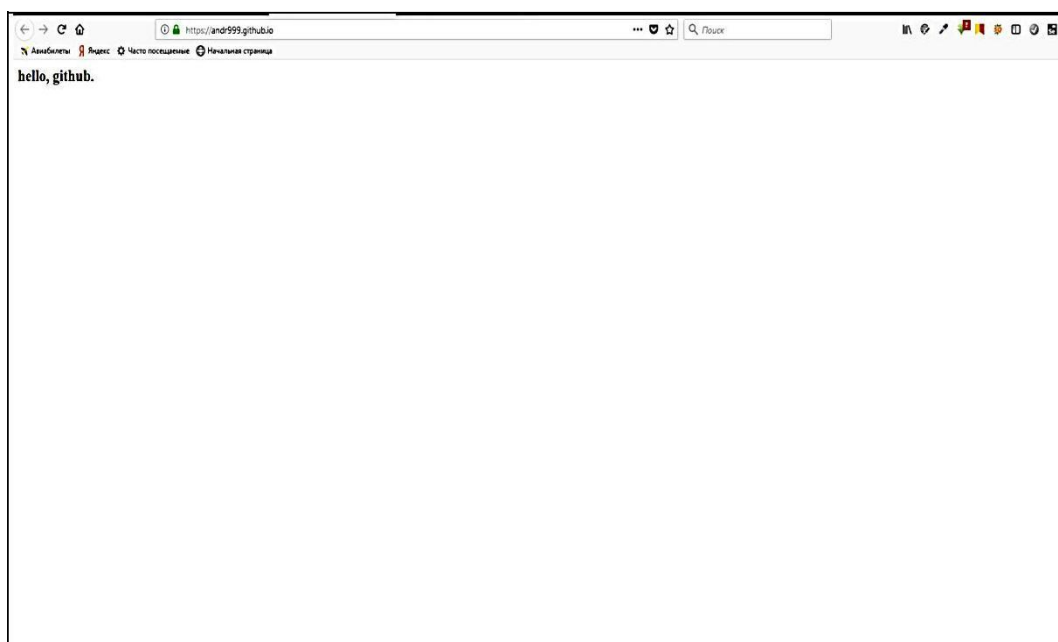


Рис. 27. Сайт в Інтернеті

Ваш сайт буде доступний за посиланням <https://andr999.github.io>. (рис. 27).

4.2. Другий спосіб використання хостингу сайтів на Github

Якщо ви бажаєте, щоб не тільки репозиторій з ім'ям вашого логіну був доступний як сайт у мережі Інтернет, а взагалі будь-який репозиторій, необхідно зробити наступне. Як і в першому способі необхідно завантажити файли сайту до репозиторію. Після цього увійти у *Налаштування* репозиторію та натиснути кнопку *Settings* (рис. 28).

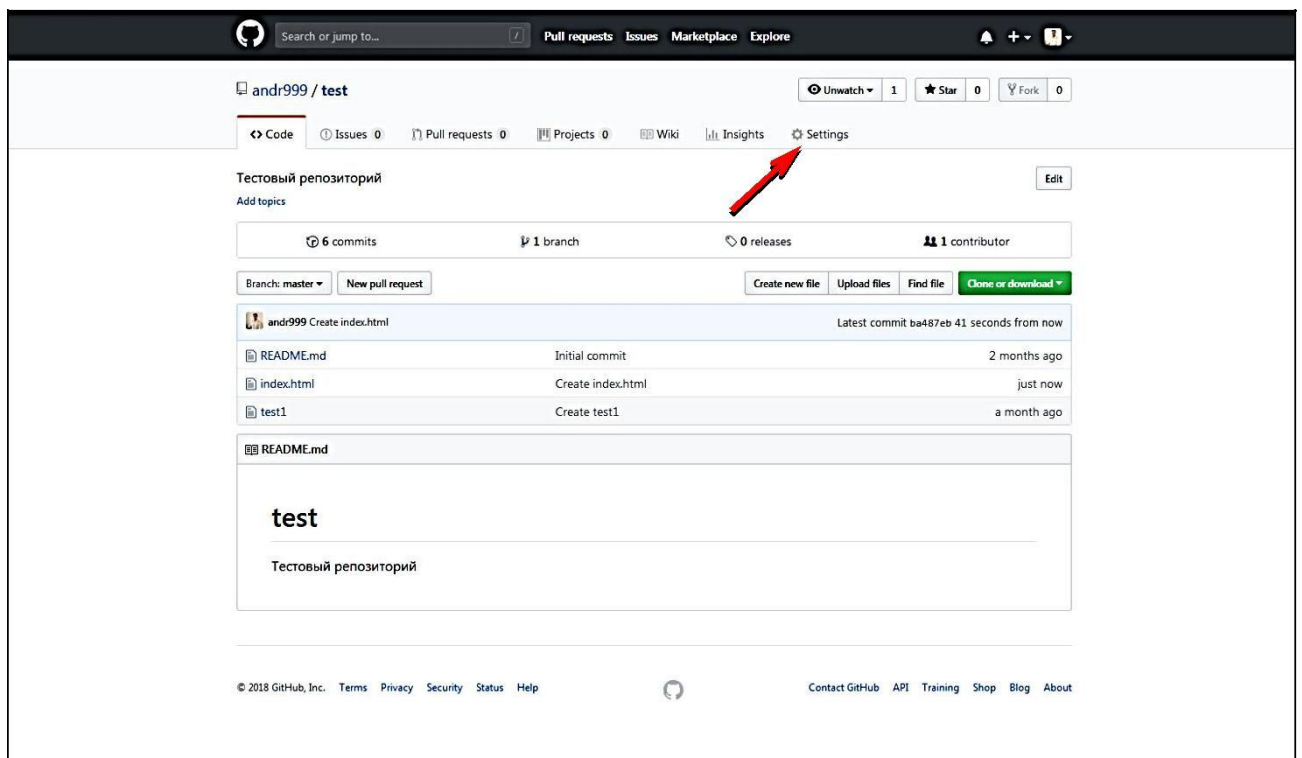


Рис. 28. Сторінка репозиторію

На сторінці налаштувань потрібно обрати *GitHub Pages*->*Source* та зі списку, що випадає, вибрати гілку *master*, натиснути на кнопку *Save* (рис. 29).

Після цього ваш репозиторій стане доступний за адресою <https://andr999.github.io/test/>.

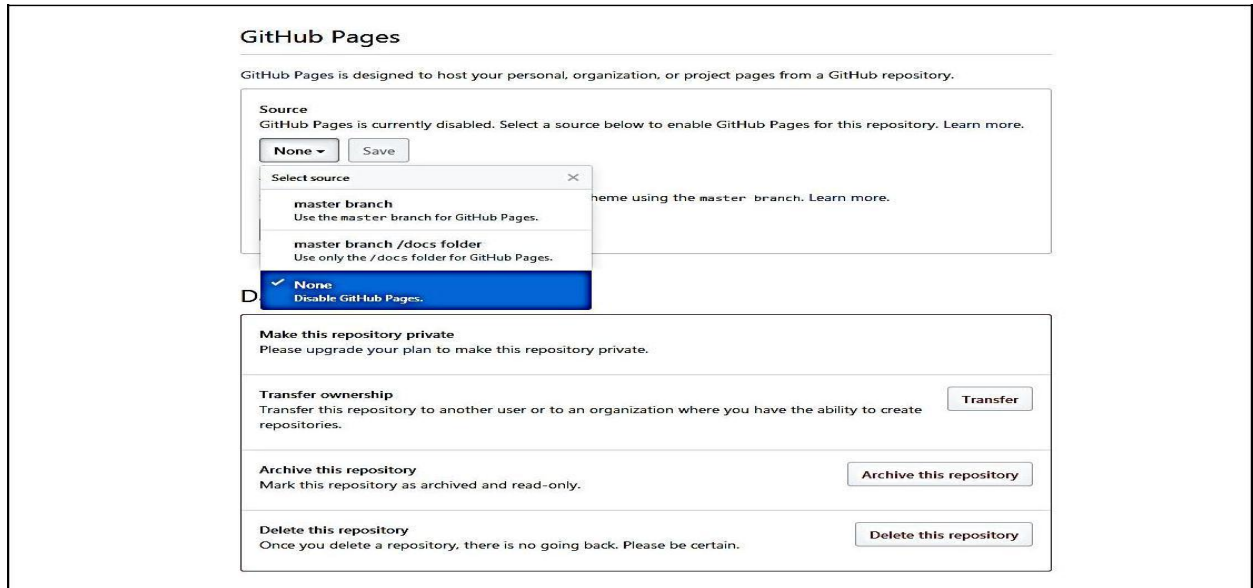


Рис. 29. Увімкнення нового сайту



Питання для закріплення

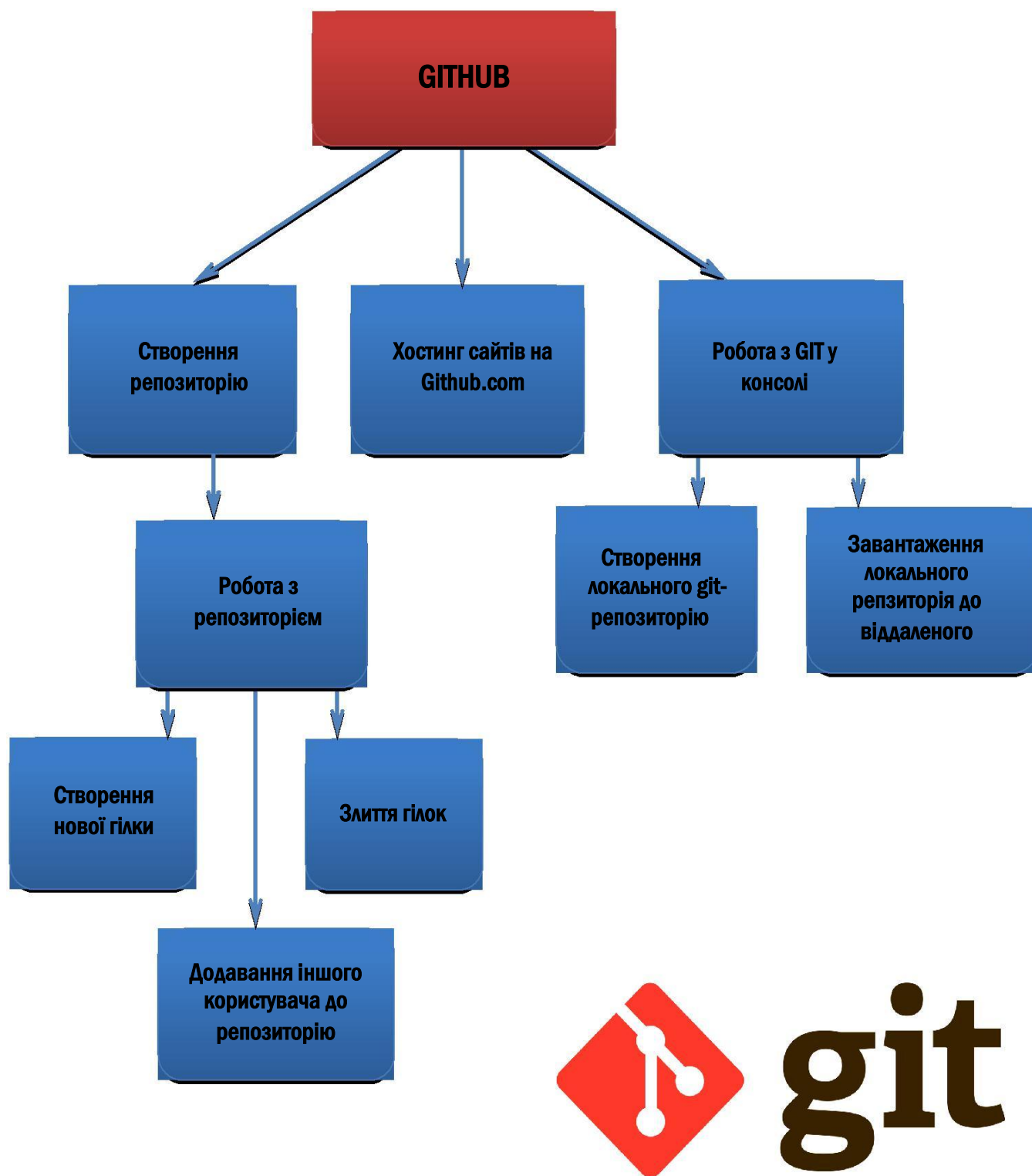
1. Як налаштувати репозиторій, щоб він мав вигляд як сайт у мережі Інтернет?
2. Як завантажити файли до репозиторію?



Вправи для самостійної роботи

1. Завантажте ваш проект до репозиторію.
2. У налаштуваннях репозиторію оберіть *git pages* та задайте адресу сайту.
3. Відкрийте наступну вкладку у браузері та введіть ім'я вашого сайту.
4. Переконайтеся, що сайт працює.

ОПОРНА СХЕМА «МОЖЛИВОСТІ GITHUB»



ІНФОРМАЦІЙНІ ДЖЕРЕЛА

1. Чакон С., Штрауб Б. Pro Git / С. Чакон, Б. Штрауб // Библиотека программиста. – М.: Питер, 2017. – 496 с.
2. Інтернет посилання: <https://proglib.io/p/git-for-half-an-hour/>
<https://habr.com/company/playrix/blog/348864/>
<http://alexprivalov.org/setup-blog-on-github/>
<https://git-scm.com/doc>