

Лабораторна робота №7

Початок роботи з GITHUB

I. Підготовка до лабораторної роботи

Для підготовки до лабораторної роботи слід проробити відповідний теоретичний матеріал, що вказаний в цьому розділі.

Також для виконання цієї лабораторної роботи необхідною умовою є виконана лабораторна робота №1.

II. Теоретичні відомості

GitHub – один з найбільших веб-сервісів для спільної розробки програмного забезпечення. Існують безкоштовні та платні тарифні плани користування сайтом. Базується на системі керування версіями Git і розроблений на Ruby on Rails і Erlang компанією GitHub, Inc (раніше Logical Awesome).

Сервіс безкоштовний для проектів з відкритим вихідним кодом, з наданням користувачам усіх своїх можливостей, а для окремих індивідуальних проектів пропонуються різні платні тарифні плани. На платних тарифних планах можна створювати приватні репозиторії, доступні обмеженому колу користувачів.

Розробники сайту називають GitHub «соціальною мережею для розробників». Окрім розміщення коду, учасники можуть спілкуватись, коментувати редагування один одного, а також слідкувати за новинами знайомих. За допомогою широких можливостей Git програмісти можуть поєднувати свої репозиторії – GitHub дає зручний інтерфейс для цього і може показувати вклад кожного учасника у вигляді дерева.

Є можливість прямого додавання нових файлів у свій репозиторій через веб-інтерфейс сервісу.

Код проектів можна не лише скопіювати через Git, але й завантажити у вигляді архіву. Окрім Git, сервіс підтримує отримання і редагування коду через SVN та Mercurial.

III. Завдання на лабораторну роботу

1. Створення репозиторію.
2. Створення гілки.
3. Створення і додавання змін до репозиторію (commit).
4. Відкриття вкладки злиття гілок Pull Request.
5. Злиття (Merge) Pull Request.

1. Створення репозиторію

Репозиторій – це сховище, яке використовується для організації проекту. Репозиторій може містити в собі папки і файли (зображення, відео, таблиці з даними).

Сервіс GitHub рекомендує додавати файл README або інформацію про робочий проект.

Для створення репозиторію (рис. 2.2) необхідно виконати такі дії:

- в правому верхньому кутку натиснути New repository;
- написати назву репозиторія (наприклад, my-first-project);
- написати короткий опис;
- вибрати (поставити галочку) Initialize this repository with a README;
- натиснути Create repository.

Create a new repository
 A repository contains all the files for your project, including the revision history.

Owner: / Repository name:

Great repository names are short and memorable. Need inspiration? How about shiny-eureka.

Description (optional):

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

Private
 You choose who can see and commit to this repository.

Initialize this repository with a README
 This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: | Add a license:

Рисунок 2.2 – Створення репозиторію

2. Створення гілки

Розгалуження проекту – це спосіб працювати на різних версіях репозиторію в один час.

За замовчуванням сховище має одну гілку *master*, яка вважається кі-нцевою гілкою. При створенні нової гілки робиться копія гілки *master*.

Діаграма, наведена нижче, ілюструє роботу розгалуження(рис. 2.3).

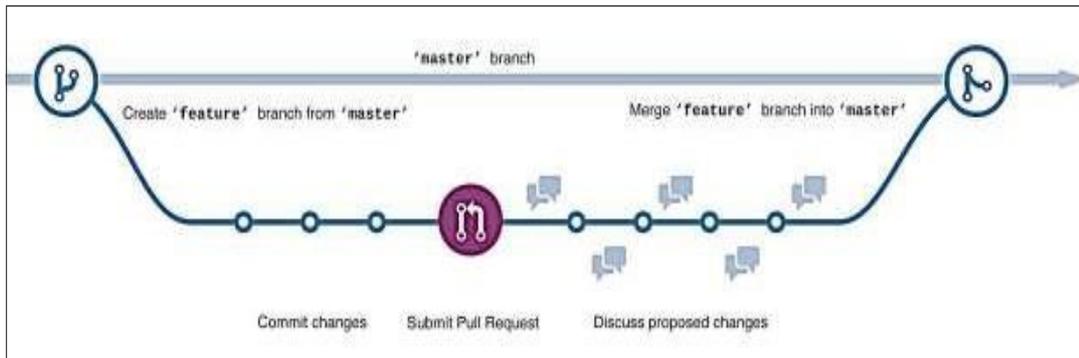


Рисунок 2.3 – Діаграма розгалуження

Для створення нової гілки (рис. 2.4) необхідно виконати такі дії:

- перейти в новий репозиторій (my-first-project);
- натиснути верхній список, що випадає, в якому буде написано branch: master;
- у текстовому полі ввести назву нової гілки (edit-readme-file);
- вибрати пункт Create branch.

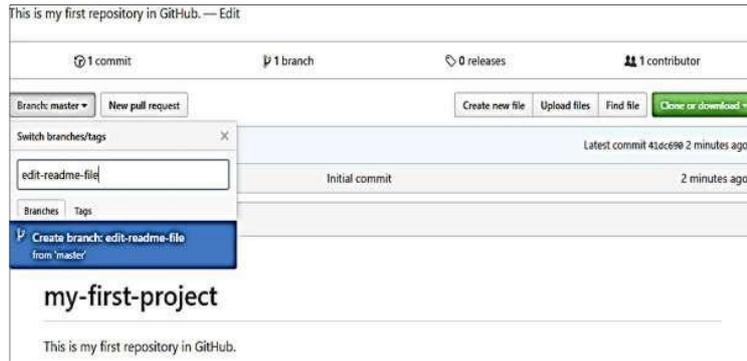


Рисунок 2.4 – Створення нової гілки

Після виконання перерахованих дій в репозиторії знаходяться дві гілки, які поки що нічим не відрізняються одна від одної. При створенні нової гілки вона автоматично стає активною.

3. Створення і додавання змін до репозиторію (commit)

Для того щоб побачити роботу гілок, необхідно додати зміни в файл.

Commit – це опис, який пояснює, чому певна зміна було зроблена. Історія commit'ів являє собою послідовність змін. Таким чином, інші учасники проекту можуть зрозуміти, що було зроблено іншими учасниками і чому.

Для створення і додавання змін до репозиторію (рис. 2.5) необхідно виконати такі дії:

- натиснути на файл README.md;
- натиснути на іконку олівця в правому верхньому куті, для того, щоб зробити зміни у файлі;
- у редакторі додати будь-яку інформацію;
- в полі Commit changes написати назву поточного commit'а. Потім написати опис змін цього commit'а;
- натиснути кнопку Commit changes.

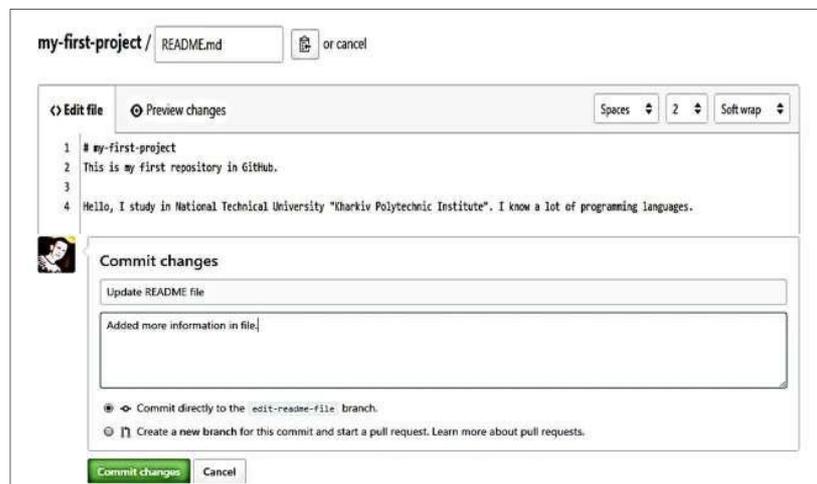


Рисунок 2.5 – Створення та commit змін

Ці зміни файлу будуть додані тільки в гілці edit-readme-file. Видно, що вміст файлу README.md відрізняється залежно від того, яка з гілок активна.

4. Відкриття вкладки злиття гілок Pull Request

У створених гілках є відмінності, тому можна злити всі зміни в основну гілку master.

Вміст вкладки Pull Request показує відмінність контенту в обох гілках. Зміни, додавання і видалення підсвічені зеленим і червоним кольорами відповідно.

Для роботи з різними гілками необхідно виконати такі дії:

- перейти на вкладку *Pull Request* і натиснути кнопку *New pull request* (рис. 2.6);



Рисунок 2.6 – Вкладка Pull Request

- вибрати дві гілки для порівняння змін (рис. 2.7);



Рисунок 2.7 – Вибір гілок

- подивитися на зміни (рис. 2.8);

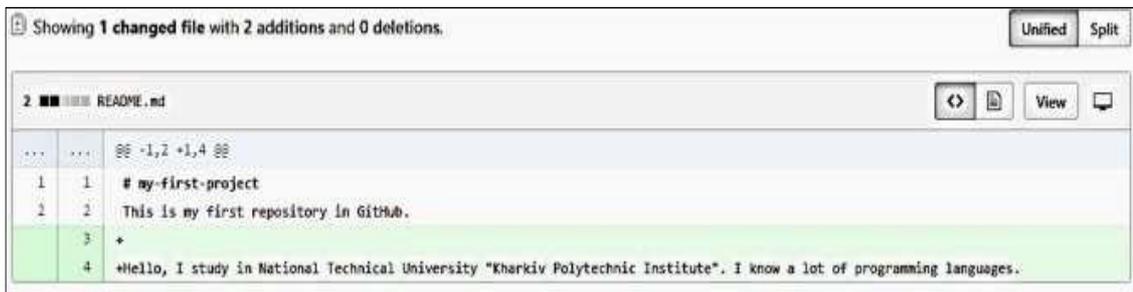


Рисунок 2.8 – Візуалізація змін

- якщо існуючі зміни коректні, натиснути кнопку *Create Pull Request* (рис. 2.9);



Рисунок 2.9 – Підтвердження змін

- для завершення роботи дати злиттю Pull Request назву, написати короткий опис поточних змін і натиснути кнопку *Create Pull Request* (рис. 2.10).

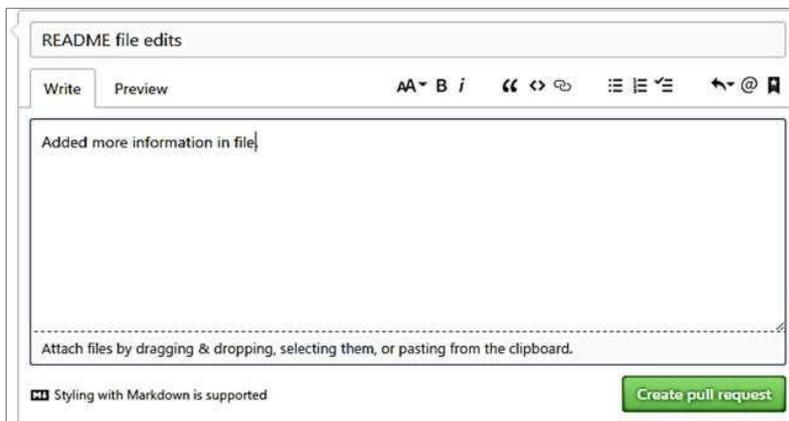


Рисунок 2.10 – Опис змін

5. Злиття (Merge) Pull Request

На останньому етапі необхідно зробити злиття всіх змін в основну гілку master.

Для цього необхідно виконати такі дії:

- натиснути кнопку Merge pull request (рис. 2.11);

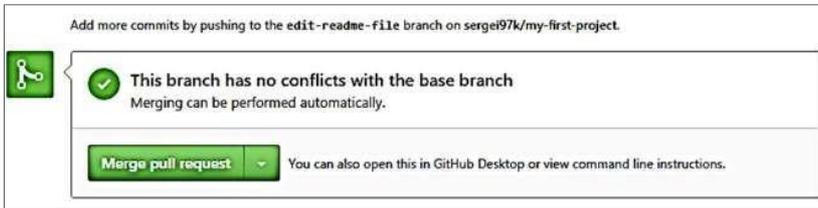


Рисунок 2.11 – Початок роботи зі злиття змін

- натиснути кнопку Confirm merge;
- видалити гілку, оскільки її зміни вже включені в основну гілку. Для цього натиснути кнопку Delete branch (рис. 2.12).



Рисунок 2.12 – Завершення роботи зі злиття змін

IV. Контрольні питання

1. Що таке GitHub?
2. Які системи контролю версій підтримуються в GitHub?
3. Яким чином створюється нова гілка в репозиторії на GitHub?
4. Як додати зміни до репозиторію?
5. Як називається основна гілка репозиторію?
6. Яким чином можна перевірити відмінність контенту в гілках?
7. Як називається команда злиття гілок?