

Завдання №1. Система керування версіями файлів та спільної роботи Git.

Мета: Ознайомитися з основними можливостями та принципами роботи системи керування версіями Git, навчитися створювати та налаштовувати локальний і віддалений репозиторії, виконувати базові операції з файлами (створення коммітів, перегляд історії змін, відновлення попередніх версій), працювати з гілками та здійснювати їх злиття, а також засвоїти основи організації спільної роботи над проєктом за допомогою віддаленого репозиторію GitHub.

Теоретичні відомості та методичні рекомендації

У сучасній веброзробці програмного забезпечення однією з ключових потреб є контроль версій. Він дозволяє:

- зберігати історію змін у проєкті;
- працювати спільно над одним кодом;
- відновлювати попередні версії файлів;
- експериментувати без ризику втратити основну робочу версію.

Найпоширенішою системою контролю версій є Git – розподілена система, створена Лінусом Торвальдсом у 2005 році для розробки ядра Linux.

Основні поняття:

- репозиторій (repository) – місце, де зберігається історія змін проєкту;
- комміт (commit) – зафіксований стан файлів із коментарем, що описує зміни;
- гілка (branch) – незалежна лінія розробки. За замовчуванням створюється гілка main (раніше master);
- злиття (merge) – поєднання змін з різних гілок;
- віддалений репозиторій (remote) – копія репозиторію на сервері (наприклад, GitHub, GitLab, Bitbucket).

Життєвий цикл файлу. У Git файл може бути у стані:

- untracked – файл новий, ще не відслідковується Git;
- staged – файл доданий у список для комміту;
- committed – зміни збережені в історії.

Базові команди:

```
git init                # створити новий репозиторій
git clone <url>         # клонувати віддалений репозиторій
git status              # перевірити стан файлів
git add <file>          # додати файл у staged
git commit -m "msg"     # зробити комміт із повідомленням
git log                 # переглянути історію коммітів
```

Отже, можна зробити висновок, що Git – це потужний інструмент, без якого важко уявити сучасну веброзробку. Він дозволяє:

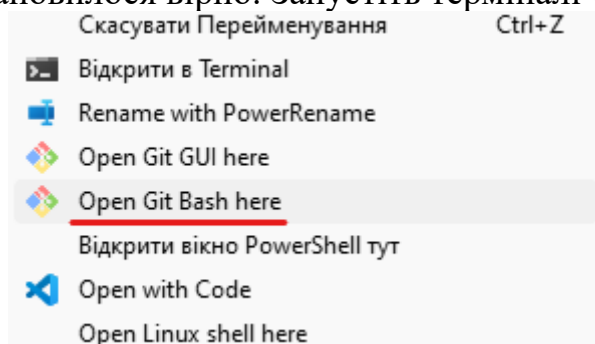
- ефективно керувати змінами у коді;
- організувати командну роботу;
- мінімізувати ризик втрати даних;
- впроваджувати кращі практики розробки програмного забезпечення.

Завдання до роботи work1

1. Встановіть систему контролю версій Git з офіційного сайту <https://git-scm.com/> (for Windows, for macOS або for Linux/Unix):



Перевірте чи все встановилося вірно. Запустіть терміналі Git Bash:



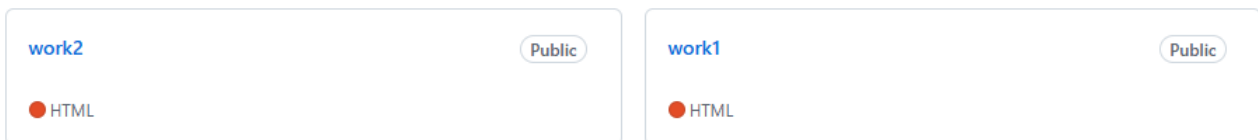
і виконайте команду `git --version` перевірки версії Git у терміналі Git Bash:

```
$ git --version  
git version 2.51.0.windows.1
```

2. Зареєструйтеся на сайті GitHub за посиланням <https://github.com/>:

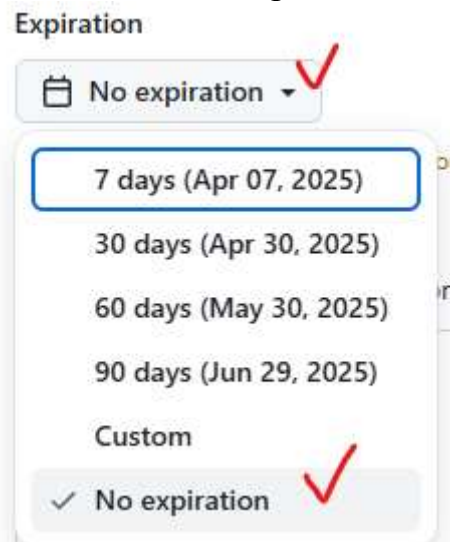


Далі створіть новий публічний репозиторій, назвавши його, наприклад, work1.



3. Для зв'язку з локальним репозиторієм отримайте Personal Access Token. Можна скористатися інструкцією за посиланням: <https://techglimpse.com/git-push-github-token-based-passwordless/>

У процесі створення Token треба буде придумати йому ім'я, налаштувати скільки він буде діяти і надати дозволи, наприклад:



Токен треба скопіювати і зберегти.

4. Створити локальний репозиторій. Задайте системні налаштування імені та e-mail за допомогою команд:

```
git config --global user.name "Your Name"  
git config --global user.email "you@example.com"
```

Далі треба зробити з папки репозиторій. Для цього існує команда, яка робить з папки на комп'ютері репозиторій (ініціює його):

```
git init
```

У локальному репозиторії створіть файл `index.html` такого вмісту:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Привіт!</h1>
  <p>Мене звати Прізвище ім'я по батькові</p>
  <p>Я навчаюсь на факультеті ... за спеціальністю ..., у
групі ...</p>
</body>
</html>
```

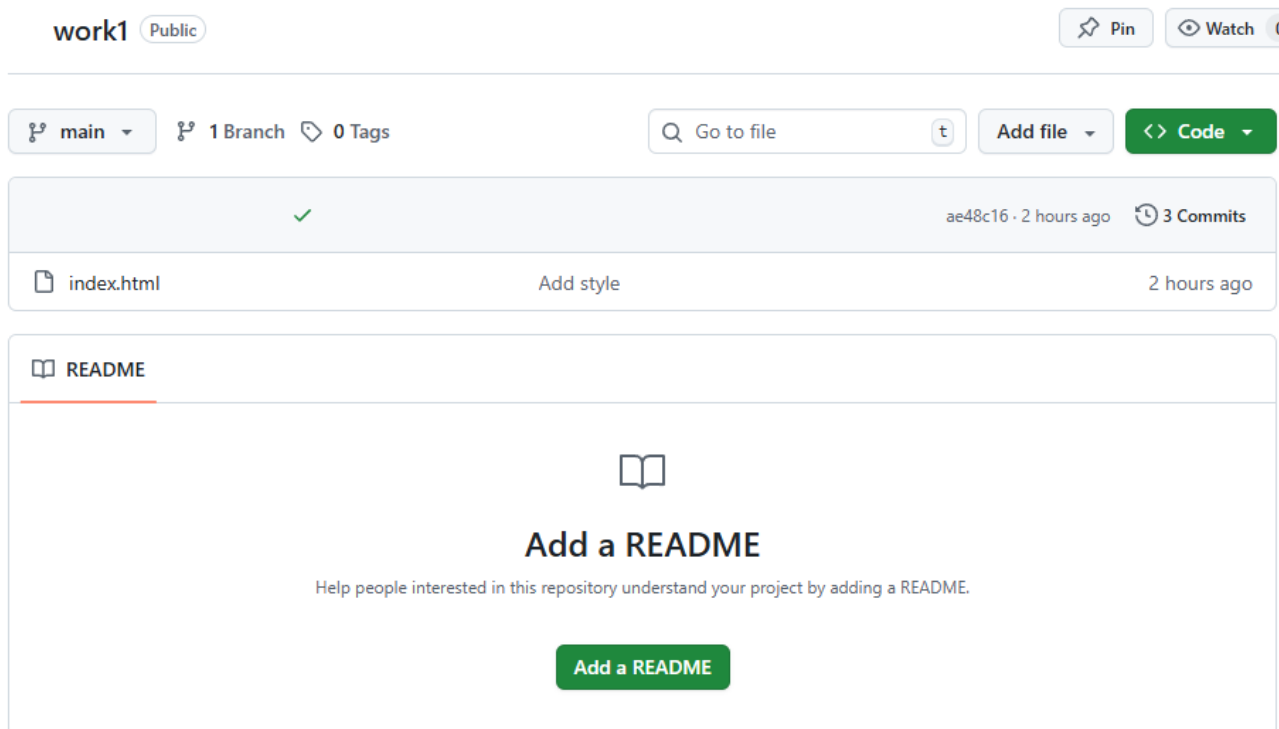
5. Далі відстежте у Git Bash всі зміни, зробіть перший коміт, перевірте статус і запусьте все на сайт <https://github.com/> у свій публічний репозиторій `work1`. Щоб запусити локальний репозиторій на сайт GitHub, використовуйте команди:

```
git remote add origin
https://github.com/<GITHUB_USERNAME>/work1.git
```

```
git push
https://<GITHUB_ACCESS_TOKEN>@github.com/<GITHUB_USERNAME>
/<REPOSITORY_NAME>.git
```

тут `<GITHUB_ACCESS_TOKEN>` – Ваш персональний Token, `<GITHUB_USERNAME>` – ім'я на GitHub, `<REPOSITORY_NAME>` – ім'я Вашого репозиторію.

У результаті повинно вийти так:



6. Дайте відповіді на питання для самоконтролю.

7. Зробіть звіт з скріншотами всіх пророблених етапів. Додайте у звіт посилання на свій публічний репозиторій GitHub. Завантажте звіт у Moodle.

Питання для самоконтролю

1. Що таке система керування версіями? Які її переваги?
2. Чим відрізняється локальний репозиторій від віддаленого?
3. Для чого використовуються команди: `git init`; `git add`; `git commit`; `git status`; `git log`?
4. Що таке гілка у Git? Для чого вона використовується?
5. Як виконати злиття гілок? Що таке конфлікт злиття та як його вирішити?
6. Які кроки потрібно зробити, щоб підключити локальний репозиторій до GitHub чи GitLab?
7. Чим відрізняються команди `git push` і `git pull`?
8. Що таке `fork` і `pull request`? Де вони застосовуються?
9. Які правила варто дотримуватися при написанні повідомлень до комітів?
10. Чому Git є розподіленою системою контролю версій?