**Лабораторна робота №5**

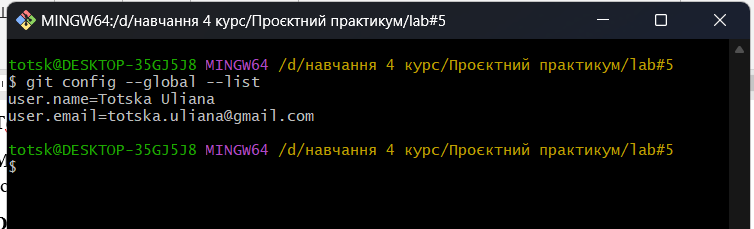
**Тема:** Внесення змін до репозиторію

**Мета роботи:** Отримати навички внесення змін до локальної робочої копії та репозиторію на Github, навчитися працювати з файлом .gitignore

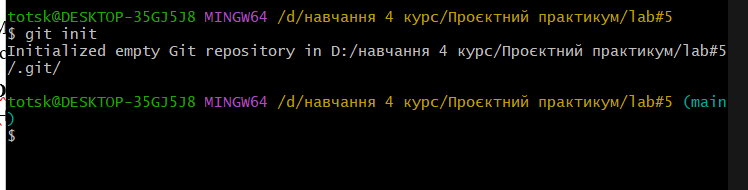
**Обладнання:**Персональний комп'ютер. Текстовий редактор Sublime Text 3 або Notepad++. Web-браузер Chrome, Firefox, Opera, Internet Explorer, Git for Windows

**Хід роботи**

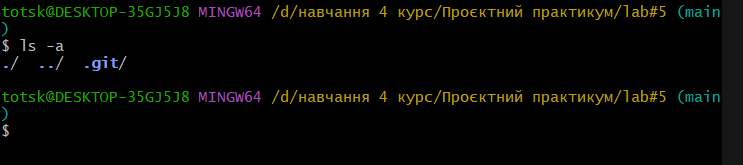
1. Перевірити глобальні налаштування Git for Windows



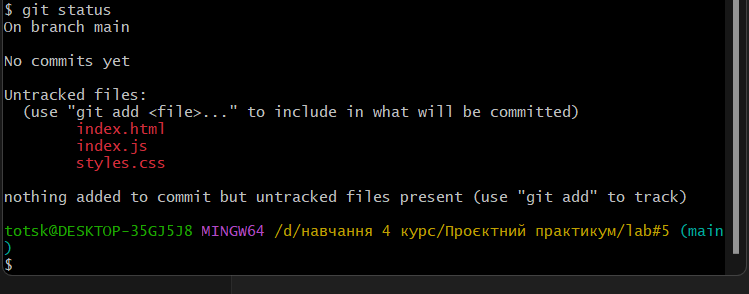
1. Створити порожній каталог для нового репозиторію
2. За допомогою команди git init ініціалізувати репозиторій



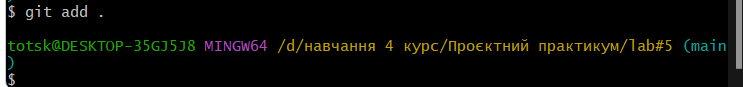
1. Якщо репозиторій проініціалізовано правильно, всередені каталогу з'явиться підкаталог .git



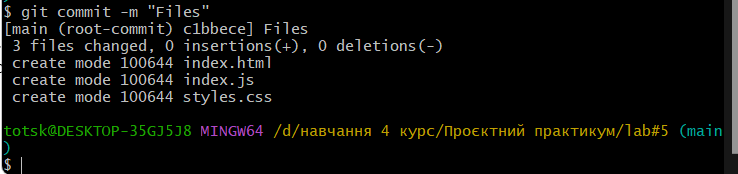
1. Помістити в каталог файли, та за допомогою консолі перевірити статус репозиторію за допомогою команди git status



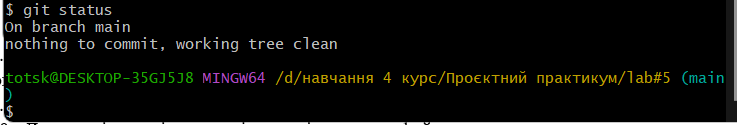
1. Додати до індексу нові файли



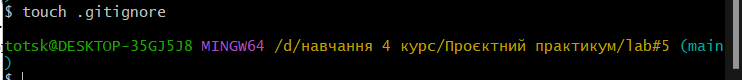
1. Зафіксувати зміни в репозиторії



1. Помістити в каталог файли через інтерфейс операційної системи або за допомогою файлового менеджера, та за допомогою консолі перевірити статус репозиторію

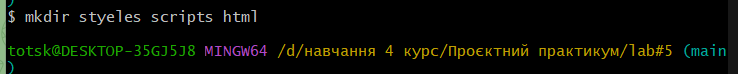


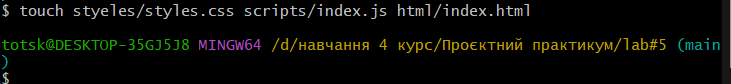
1. Додати нові файли до переліку ігнорування



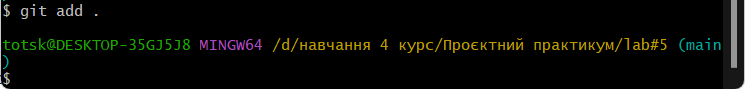


1. Додати кілька підкаталогів, розмістити там файли з розширенням \*.css, \*.js та \*.html

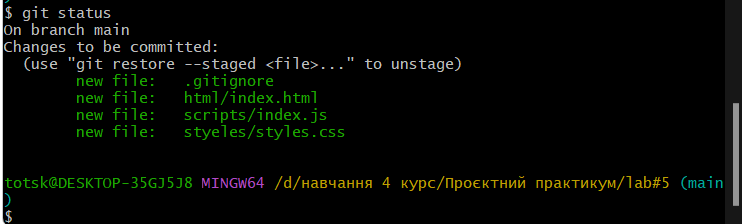




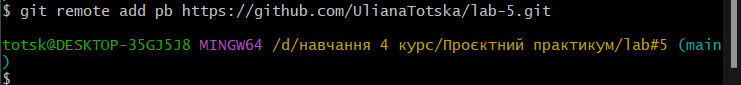
1. Додати файли \*.html до переліку ігнорування
2. Додати до індексу нові файли



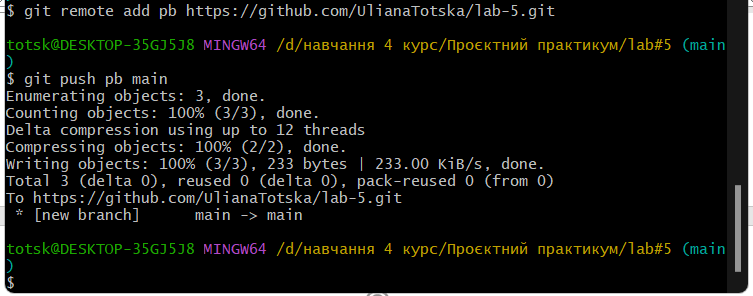
1. Перевірити статус репозиторію



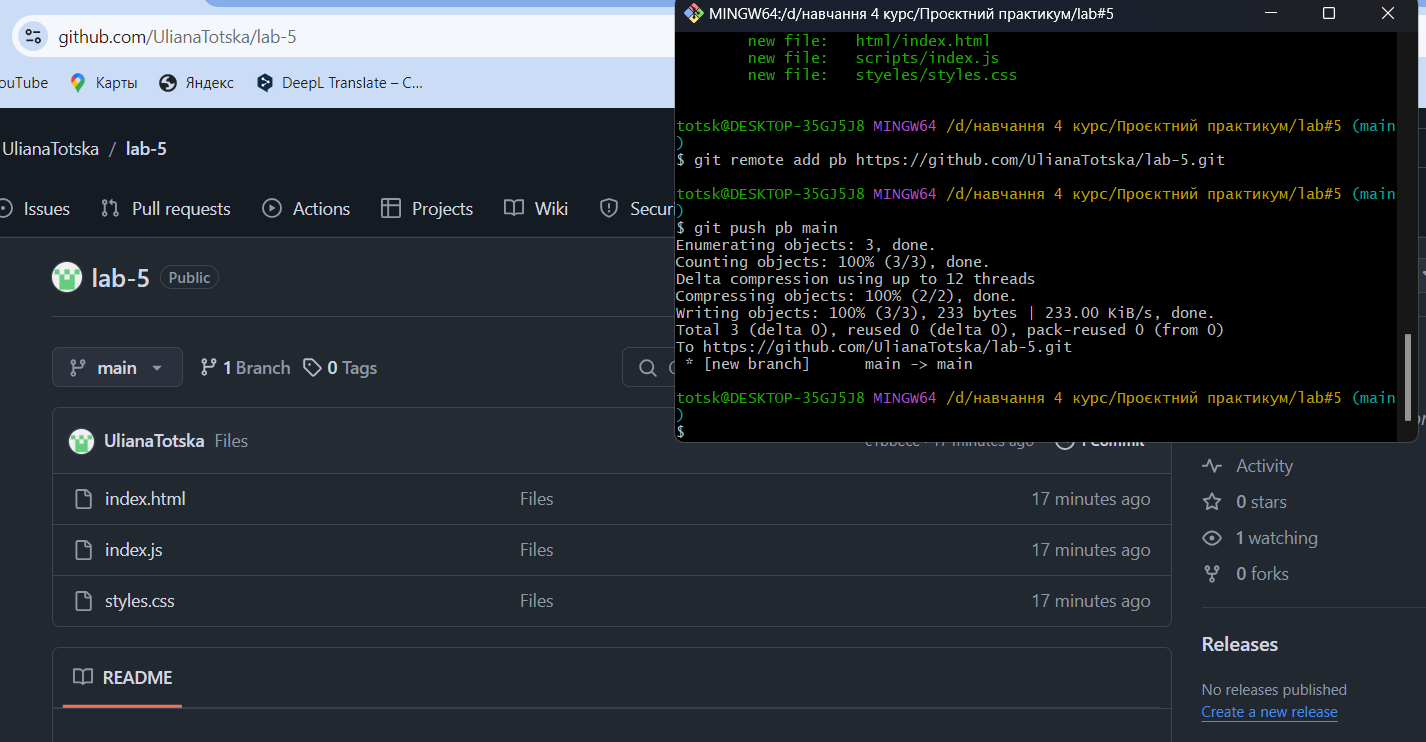
1. За допомогою команди git remote під'єднати віддалений репозиторій заздалегіть створений на GitHub



1. Підєднати ще один віддалений репозиторій на Bitbucket
2. Вивантажити файли локального репозиторію на Github та Bitbucket



1. Впевнитися за допомогою web-інтерфейсу, що завантаження пройшло корректно



1. Впевнитися за допомогою web-інтерфейсу, що Файли з переліку ігнорування не вивантажилися у віддалений репозиторій
2. Для кожного етапу роботи зробити знімки екрану або скопіювати текст консолі та додати їх у звіт з описом кожного скіншота
3. Дати відповіді на контрольні запитання
4. Зберегти звіт у форматі PDF

1. Як створити файл .gitignore в своєму проекті?

a) Відкрийте термінал у кореневому каталозі вашого проекту.

b) Введіть команду: touch .gitignore

c) Відкрийте файл у текстовому редакторі і додайте необхідні правила.

2. Чому корисно використовувати .gitignore при роботі з репозиторієм Git?

Файл .gitignore дозволяє виключати з контролю версій файли, які не мають сенсу у репозиторії, наприклад:

a) Конфігураційні файли, специфічні для локального середовища.

b) Скомпільовані файли та артефакти.

c) Тимчасові файли, які не повинні потрапляти в репозиторій.

3. Де може бути розміщений файл .gitignore?

Файл .gitignore може бути розміщений:

a) У кореневому каталозі репозиторію (глобальні правила).

b) У підкаталогах (локальні правила для конкретної папки).

4. Які типи файлів можна вказати в файлі .gitignore для їх ігнорування Git?

У файлі .gitignore можна вказувати:

a) Файли за розширенням: \*.log, \*.tmp

b) Цілі каталоги: /build/, /dist/

c) Конкретні файли: secret.txt, config.yaml

d) Шаблони: \*~, \*.bak

e) Ігнорування папок у підкаталогах: /node\_modules/

5. Яким чином ігнорувати всі файли \*.php окрім index.php?

У файлі .gitignore напишіть:

\*.php

!index.php

6. Які дії виконує шаблон ?

Шаблон означає "будь-який каталог". Наприклад, /\*.log ігнорує всі .log файли в будь-якій папці на всіх рівнях.

7. Які дії виконує шаблон !?

Шаблон ! використовується для скасування ігнорування. Наприклад, \*.php і !index.php ігноруватиме всі .php файли, крім index.php.

8. Що таке файл .gitignore і яка роль його відіграє в системі контролю версій Git?

Файл .gitignore - це текстовий файл, що містить список шаблонів, які вказують Git, які файли або каталоги потрібно ігнорувати. Це допомагає підтримувати чистоту репозиторію, уникаючи включення непотрібних файлів.

9. Як додати коментарі до файлу .gitignore для пояснення включених правил?

Коментарі в .gitignore починаються з символу #

10. Якщо виникає конфлікт між локальним та глобальним файлами .gitignore, як його вирішити?

Конфлікт між локальним і глобальним .gitignore можна вирішити, звертаючи увагу на порядок перевірки. Локальний .gitignore має пріоритет над глобальним. Якщо потрібні специфічні правила для конкретного репозиторію, розмістіть їх у локальному файлі, а загальні правила — у глобальному.