

## **Попередні кроки, які потрібно виконати перед початком виконання ДЗ:**

1. Встановіть GIT на робочу машину;
2. Створіть на вашій робочій машині окрему папку, в яку ви будете клонувати репозиторій;
3. Відкрийте звичайний термінал (Linux, MacOS) або GIT BASH (Windows);
4. Перейдіть у створену папку (яку створили у п.2) за допомогою команди `cd`;
5. Перед виконанням наступних дій виконайте команду очищення історії `history -c && rm ~/.bash_history`
6. Налаштування GIT: внесіть зміни в глобальну конфігурацію (додайте `user.name` та `user.email`);
7. Створіть новий репозиторій на GitHub (<https://github.com/>). Репозиторій має бути з публічним доступом і мати назву HW\_10. Файли README.md та .gitignore створюються разом з репозиторієм (потрібно обрати чекбокс "Add a README file"; для файла .gitignore обрати темплейт "AL").

### **Task 1. Робота з .gitignore**

1. Клонуйте репозиторій (створений у п.7) на свою робочу машину;
2. Перейдіть у сконований репозиторій на робочій машині за допомогою команди `cd`;
3. Створіть через термінал файл `text_ignore.txt` (`touch`);
4. Відредагуйте файл `.gitignore`, додавши туди назву файлу `text_ignore.txt` (потрібно просто записати рядок "`text_ignore.txt`" у файл `.gitignore`);
5. Переконайтеся, що `text_ignore.txt` файл не є видимим для GIT через виконання команди `git status`;
6. Проіндексуйте всі зміни за допомогою команди "`git add .`";
7. Зробіть `commit` для фіксації ваших змін.

### **Task 2. Створення комітів**

1. Створіть порожній файл `main_test.txt`;
2. Проіндексуйте всі зміни за допомогою команди "`git add .`";
3. Зробіть `commit` для фіксації ваших змін.

### **Task 3. Робота в іншій гілці**

1. Створіть гілку `develop` і перейдіть на неї за допомогою команди `checkout`;
2. Створіть порожній файл `develop_test.txt`;
3. Проіндексуйте всі зміни за допомогою команди "`git add .`";
4. Зробіть `commit` для фіксації ваших змін;
5. Внесіть будь-які зміни у файл `develop_test.txt` file, наприклад: додайте рядок "`just a small change`";
6. Перевірте зміни через виконання команди `git status`;
7. Проіндексуйте всі зміни за допомогою команди "`git add .`";

8. Зробіть commit для фіксації ваших змін.

#### **Task 4. Об'єднання гілок**

1. Перейдіть в гілку main за допомогою команди checkout;
2. Об'єднайте (merge) гілку develop з main (або master, якщо немає main);
3. Перевірте коміти за допомогою команди git log.

#### **Task 5. Пушимо всі зміни в репозиторій**

1. Вивантажте інформацію з git BASH командою:  
для Windows history > command\_list.csv  
для MacOS history 0 > command\_list.csv  
(Цей файл з'явиться в папці, де ви виконували домашнє завдання);
2. Проіндексуйте всі зміни за допомогою команди "git add .";
3. Зробіть commit для фіксації ваших змін;
4. Виконати команду git push для того, щоб всі зроблені зміни потрапили до віддаленого репозиторію.

#### **Формат виконання завдань з використанням Git**

Посилання на віддалений репозиторій з усіма змінами завантажити в LMS. Обов'язково перевірити, що репозиторій публічний та містить command\_list.csv.