

## Лабораторна робота №3

### Наука про дані: обмін результатами та початковий аналіз

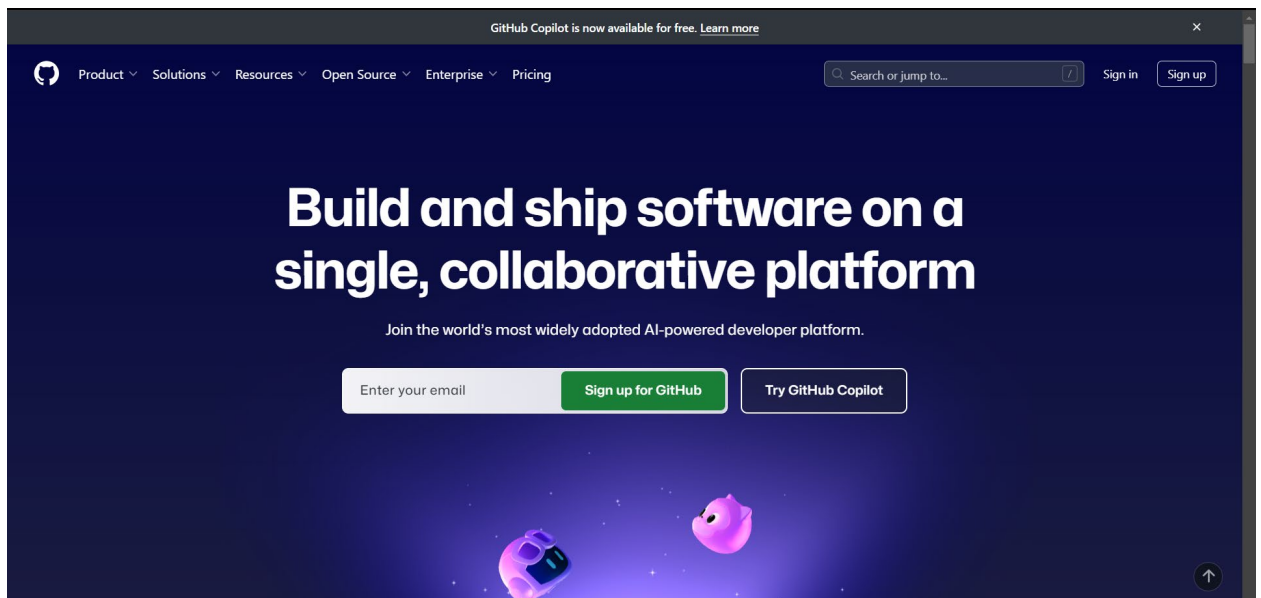
**Мета роботи:** ознайомитися з системою контролю версій GitHub, навчитися створювати прості веб-додатки для обміну результатами досліджень із використанням модуля *Streamlit*.

**Основні поняття:** система контролю версій, репозиторій, інтерактивний веб-додаток.

**GitHub** – це веб-платформа для хостингу проектів, пов'язаних із розробкою програмного забезпечення, яка працює на основі системи контролю версій Git. Сервіс пропонує користувачам широкий спектр можливостей, серед яких:

- обмін файлами між локальним репозиторієм та версією, що зберігається на GitHub;
- спільне використання репозиторіїв з іншими користувачами;
- спостереження за діяльністю інших розробників;
- використання стандартних функцій системи контролю версій.

Щоб почати користуватися GitHub, необхідно створити обліковий запис. Зареєструватися можна за посиланням: <https://github.com/>.



#### Відмінності між Git і GitHub:

**Git** – це розподілена система контролю версій, яка дозволяє відстежувати зміни у файлах, зберігати історію проекту, відновлювати попередні версії та працювати з гілками для ізольованої розробки. Git працює незалежно від інтернету, і всі дані зберігаються локально на комп'ютері користувача.

**GitHub** – це веб-сервіс, побудований на основі Git, який використовується для хостингу репозиторіїв у хмарі.

### Створення репозиторію на GitHub

Існує два основні способи створення репозиторію:

*Через веб-інтерфейс GitHub*

1. Увійдіть у свій обліковий запис GitHub.
2. Перейдіть на сторінку репозиторіїв і натисніть New repository.
3. Введіть назву репозиторію (repository name).
4. За бажанням, додайте опис (description).
5. Виберіть рівень доступу.
6. За необхідності додайте файл README, який описує ваш проєкт.

*Дублювання існуючого репозиторію (fork)*

1. Знайдіть репозиторій іншого користувача, який хочете скопіювати.
2. Натисніть кнопку Fork для створення копії цього репозиторію у своєму акаунті.

Для виконання лабораторної роботи, необхідно використати перший спосіб.

### Робота з локальним репозиторієм

Щоб працювати з репозиторієм локально, потрібен клієнт Git (Наприклад: Для Windows: [GitHub Desktop](#), що включає графічний інтерфейс і Git Shell). Для додавання файлів в репозиторій GitHub можна клонувати існуючий репозиторій або ініціалізувати новий.

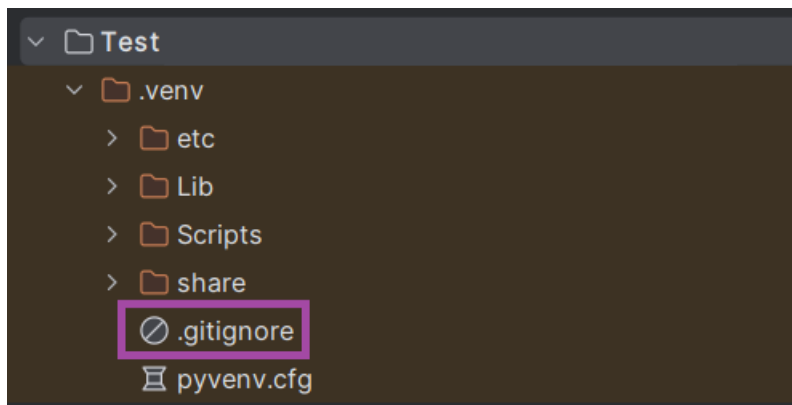
*Клонування існуючого репозиторію:*

```
git clone https://github.com/yourUserName/repoName.git
git add "filename"
git commit -m "added data"
git push origin main
```

*Ініціалізація нового локального репозиторію:*

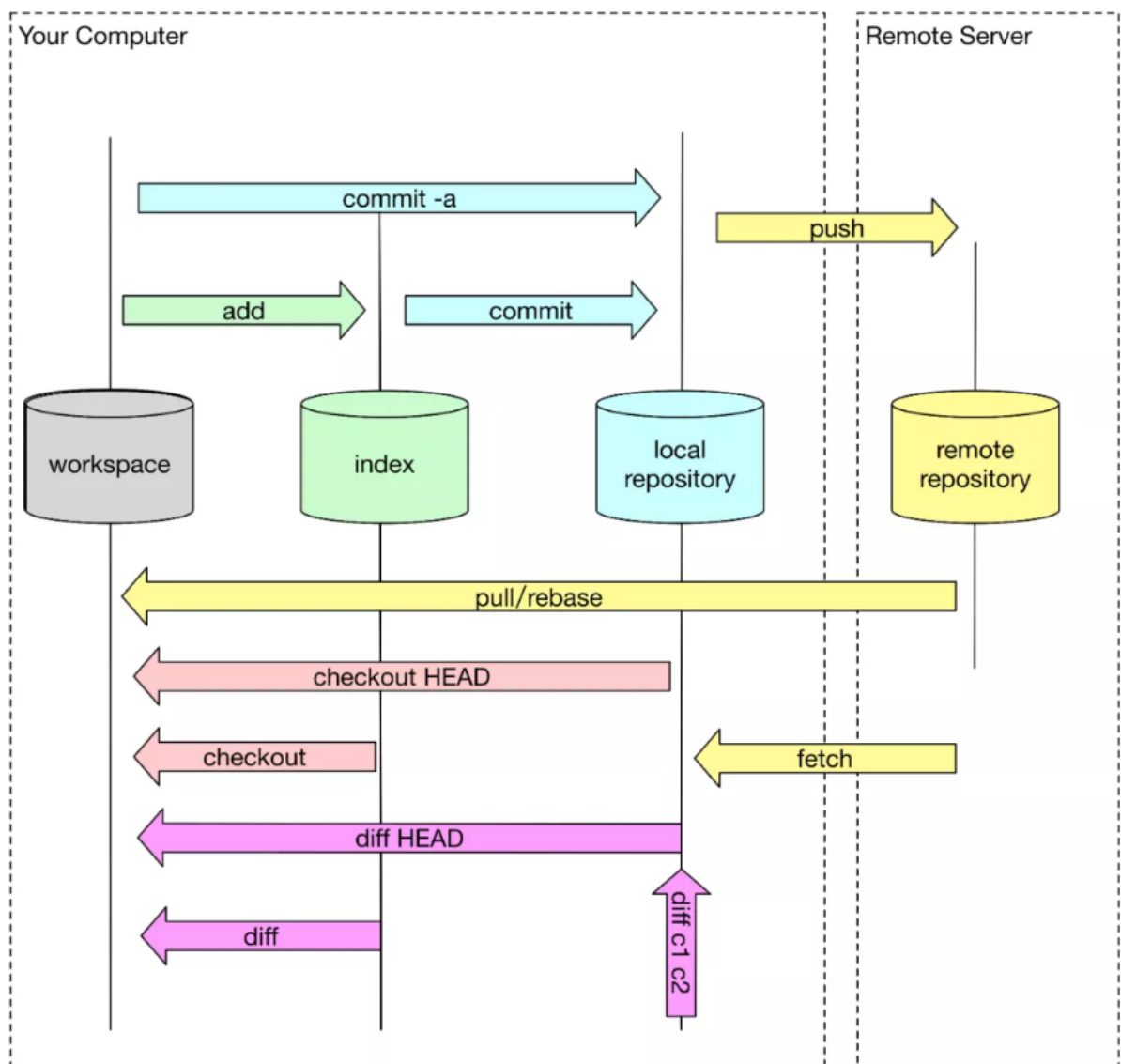
```
cd path/to/your/project
git init
git add "filename"
git commit -m "added data"
git remote add origin https://github.com/yourUserName/repoName.git
git push -u origin main
```

**Рекомендовано:** замість `git add "filename"` використовуйте `git add .`, ця команда додає всі файли (або зміни у них), які знаходяться в поточній директорії та її піддиректоріях у `commit`. Для цього необхідно заповнити `.gitignore`, вказавши файли/директорії, які не повинні бути скопійованими.



Після виконання команди `git push` зміни будуть передано на сервер GitHub, що дозволить отримати доступ до вашого коду та даних навіть з іншого комп'ютера. Ви можете клонувати репозиторій або завантажити свій проект разом із даними у вигляді архіву.

*Наглядна схема роботи з системою контролю версій:*



## Візуалізація результатів аналізу у Web

Окрім вирішення основного завдання дослідження, важливу роль відіграє обмін його результатами. Корисно надати зовнішнім користувачам можливість працювати з отриманими даними та візуалізувати їх у зручний спосіб.

Одним із ефективних інструментів для цього в Python є бібліотека *Streamlit*, яка дозволяє легко створювати веб-додатки з базовим та зручним функціоналом для роботи з даними та їх візуалізації.

Всю необхідну інформацію про роботу з *Streamlit* можна знайти за посиланням:

<https://docs.streamlit.io/get-started>

### Хід виконання роботи

Створення веб-додатку із використанням модуля *Streamlit*:

1. Створіть dropdown список, який дозволить обрати часовий ряд VCI, TCI, VHI для набору даних із лабораторної роботи 2;
2. Створіть dropdown список, який дозволить вибрати область, для якої буде виконуватись аналіз;
3. Створіть slider, який дозволить зазначити інтервал тижнів, за які відбираються дані;
4. Створіть slider, який дозволить зазначити інтервал років, за які відбираються дані;
5. Створіть button для скидання всіх фільтрів і повернення до початкового стану даних (відповідно інтерактивні елементи повинні мати початкові значення);
6. Створіть три вкладки для відображення таблиці з відфільтрованими даними, відповідного до неї графіка та графіка порівняння даних по областях.
7. Перший графік повинен відображати відфільтровані дані (часові ряди за діапазон років, що обмежені інтервалом тижнів). Другий графік має відображати порівняння значень VCI, TCI або VHI (залежно від обраної опції у списку dropdown) для обраної області з усіма іншими областями за вказаний часовий інтервал. Продумайте вигляд цих графіків.
8. Створіть два checkbox для сортування даних за зростанням та спаданням значень VCI, TCI або VHI (залежно від обраної опції у списку dropdown). Продумайте реакцію програми, якщо увімкнені обидва чекбокси.
9. Інтерактивні елементи мають бути розміщені в одній колонці, а графіки з таблицею — в іншій.

Робота з GitHub:

1. Зареєструйтесь на GitHub;
2. Створіть public репозиторій;
3. Додайте другу та третю лабораторні роботи в цей репозиторій. Кожна лабораторна повинна бути в окремій директорії;
4. Репозиторій повинен мати заповнений файл [README.MD](#) та [.gitignore](#), в якому прописані винятки для Git: таблиці та інші дані, що використовуються.

### **Контрольні запитання**

1. Для кого призначена система git?
2. Назвати основні можливості, які надає система контролю версій Git та платформа GitHub.
3. Які ще способи створення інтерактивних додатків в Python Ви знаєте? Які існують аналоги в інших мовах, що застосовуються для аналізу даних (R, Matlab).

### **Література**

1. <http://git-scm.com/doc>
2. <https://docs.streamlit.io/get-started>
3. <https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-readmes>
4. <https://docs.github.com/en/get-started/getting-started-with-git/ignoring-files>