

**Міністерство освіти і науки України**  
Національний університет “Львівська політехніка”  
Кафедра систем автоматизованого проектування  
Інститут комп’ютерних наук та інформаційних технологій



**Звіт**  
про виконання лабораторної роботи № 1  
з дисципліни «Управління ІТ-проектами»  
на тему:  
«Системи контролю версій»

Виконала студентка групи КН-408:  
Ольга КИЧУК

Прийняв:  
Іван ШПАКОВИЧ

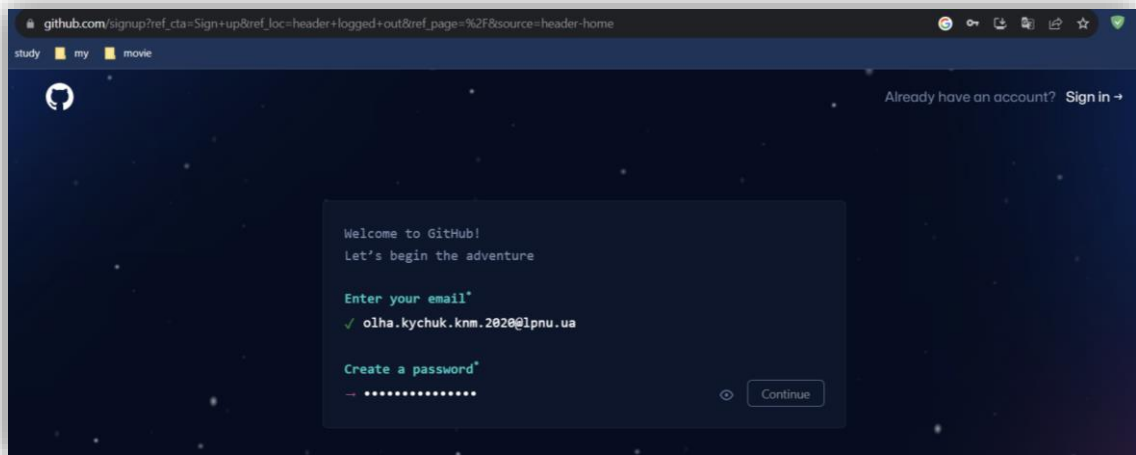
Львів 2023

**Мета.** Ознайомитись з системами контролю версій. Зрозуміти принципи синхронізації робочих груп у гнучких командах. Набути навичок використання технічних засобів та протоколів для роботи з системами контролю версій.

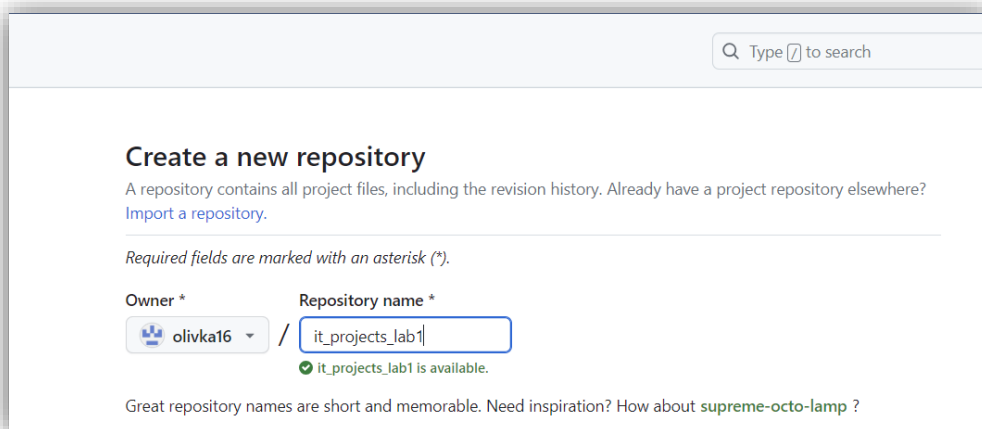
### Лабораторне завдання:

1. Ознайомитись з принципами роботи систем контролю версій.
2. Отримати індивідуальне завдання у викладача.
3. Написати програму згідно з індивідуальним завданням.
4. Створити репозиторій та завантажити туди свій програмний код.
5. Внести зміни в програму, при необхідності провести вирішення конфліктів.

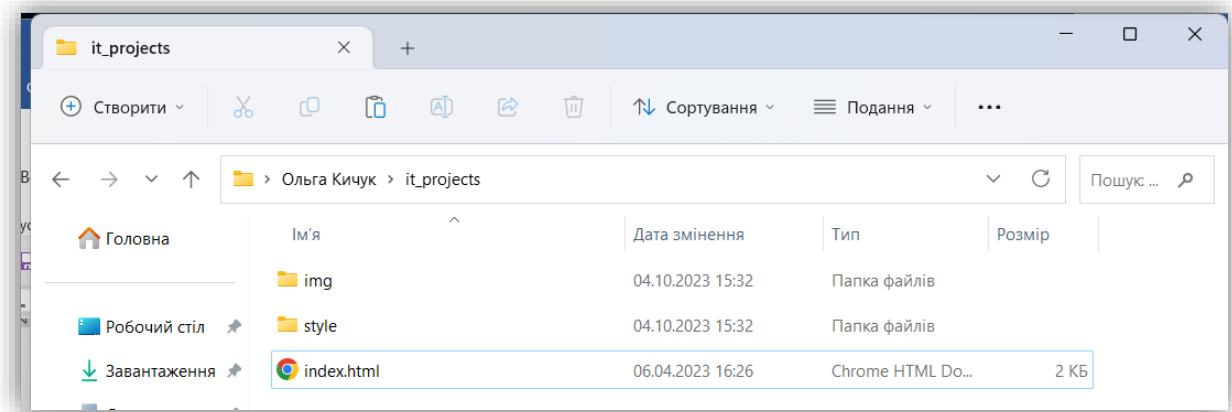
Створюю акаунт на GitHub:



Створення репозиторію:



Мій локальний проект:



Перевіряю чи встановлений Git:

```
olhak@olia MINGW64 ~/it_projects
$ git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
          [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
          [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
          [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
          [--super-prefix=<path>] [--config-env=<name>=<envvar>]
          <command> [<args>]
```

Ініціалізую порожній git репозиторій:

```
olhak@olia MINGW64 ~/it_projects
$ git init -b main
Initialized empty Git repository in C:/Users/olhak/it_projects/.git/
```

Додаю локальний репозиторій на GitHub і перевіряю з'єднання:

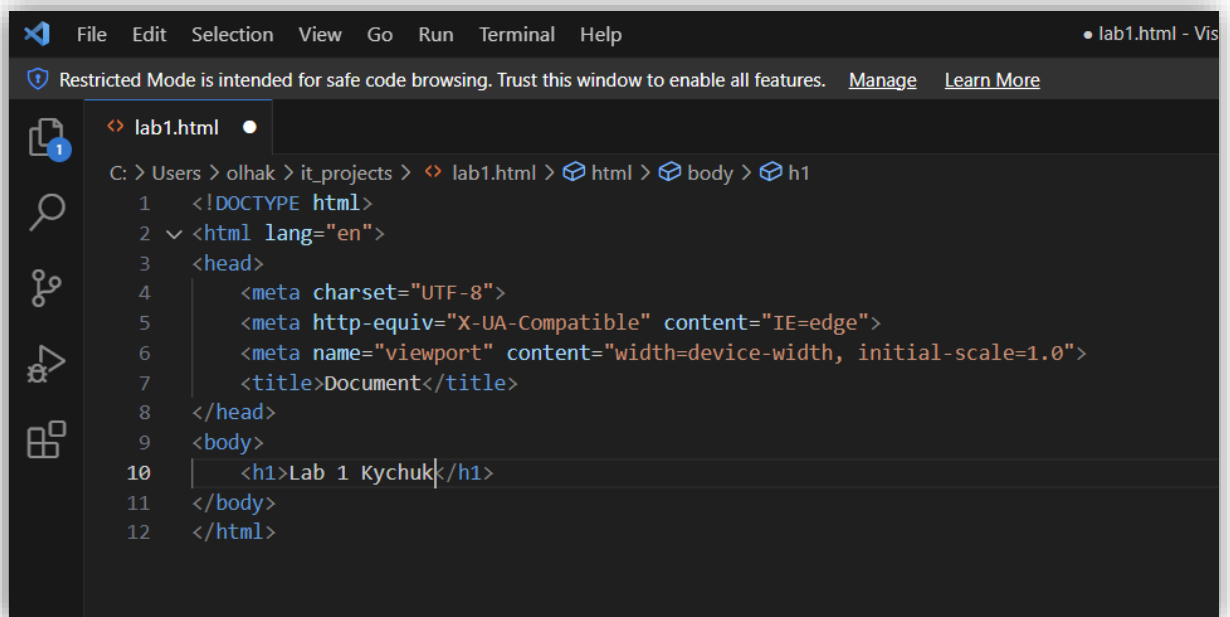
```
olhak@olia MINGW64 ~/it_projects (main)
$ git remote add origin https://github.com/olivka16/it_projects_lab1.git

olhak@olia MINGW64 ~/it_projects (main)
$ git remote -v
origin https://github.com/olivka16/it_projects_lab1.git (fetch)
origin https://github.com/olivka16/it_projects_lab1.git (push)
```

Створюю новий файл:

```
olhak@olia MINGW64 ~/it_projects (main)
$ touch lab1.html
```

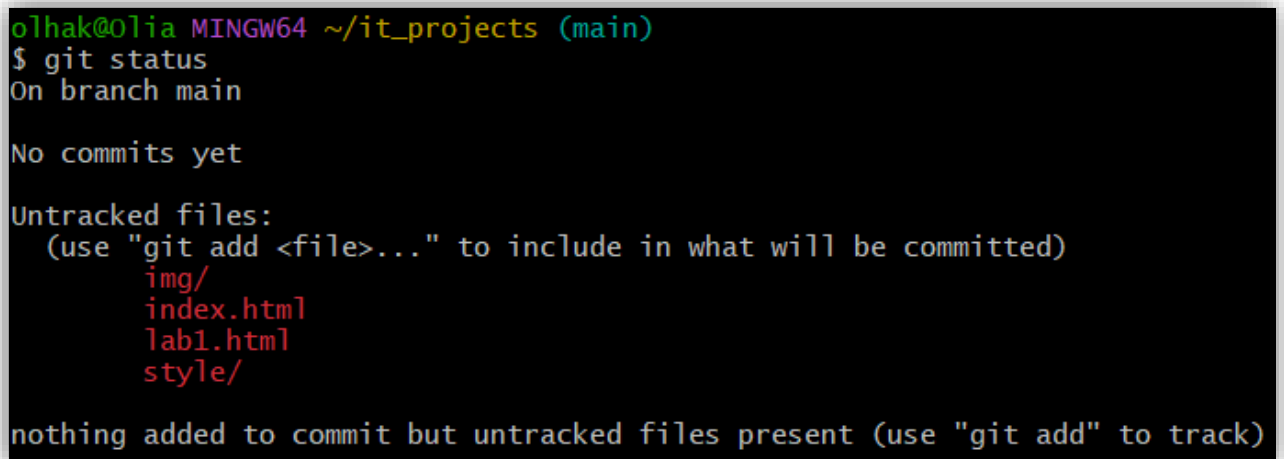
Записую мінімальний код у новостворений файл:



The screenshot shows the Visual Studio Code editor with a file named `lab1.html` open. The file path is `C:\Users\olhak\it_projects\lab1.html`. The code is a minimal HTML document with the following structure:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Document</title>
8   </head>
9   <body>
10    <h1>Lab 1 Kychuk</h1>
11  </body>
12 </html>
```

Перевіряю чи є змінені файли:



The screenshot shows a terminal window with the following output:

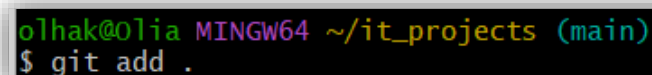
```
olhak@olia MINGW64 ~/it_projects (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    img/
    index.html
    lab1.html
    style/

nothing added to commit but untracked files present (use "git add" to track)
```

Додаю усі файли:



The screenshot shows a terminal window with the following command and output:

```
olhak@olia MINGW64 ~/it_projects (main)
$ git add .
```

Перевіряю чи усі файли додалися:

```
olhak@Olia MINGW64 ~/it_projects (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   img/img.png
    new file:   index.html
    new file:   lab1.html
    new file:   style/style.css
```

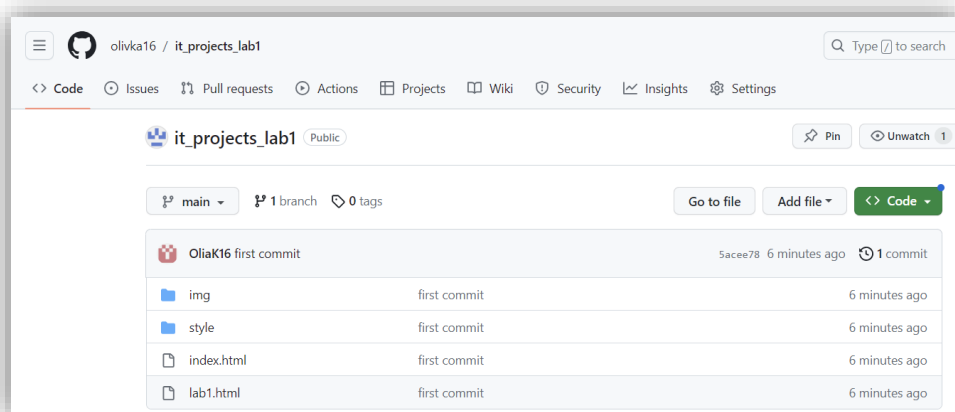
Зробила коміт:

```
olhak@Olia MINGW64 ~/it_projects (main)
$ git commit -m "first commit"
[main (root-commit) 5acee78] first commit
4 files changed, 107 insertions(+)
create mode 100644 img/img.png
create mode 100644 index.html
create mode 100644 lab1.html
create mode 100644 style/style.css
```

Відправила на Git:

```
olhak@Olia MINGW64 ~/it_projects (main)
$ git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (8/8), 3.64 MiB | 806.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/olivka16/it_projects_lab1.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Перевірила чи є на Git:



Тепер вношу ще зміни у файл:

```
C: > Users > olhak > it_projects > <> lab1.html > html > body > h2
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  |   <title>Document</title>
8  </head>
9  <body>
10 |   <h1>Lab 1 Kychuk</h1>
11 |   <h2>second commit</h2>
12 </body>
13 </html>
```

Перевіряю статус, додаю зміни і коммічу їх:

```
olhak@Olia MINGW64 ~/it_projects (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   lab1.html

no changes added to commit (use "git add" and/or "git commit -a")

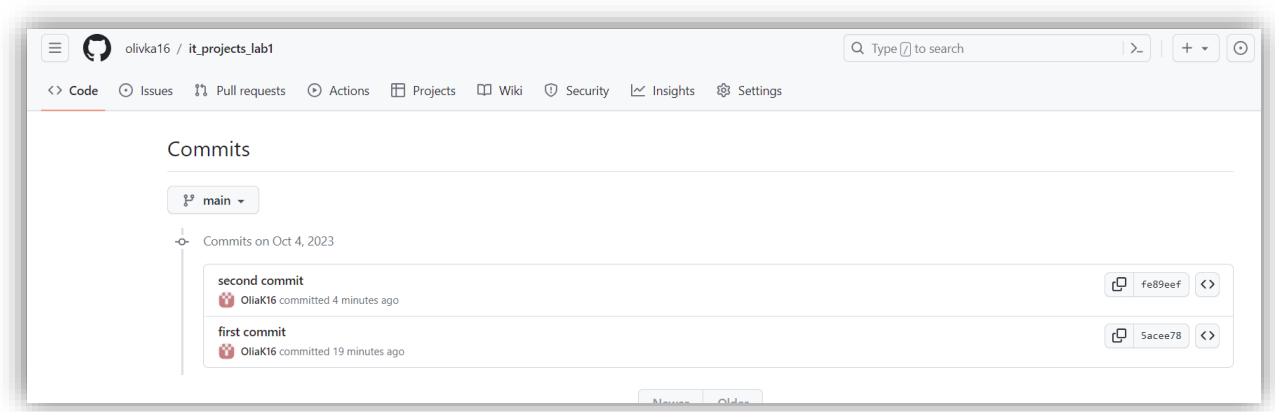
olhak@Olia MINGW64 ~/it_projects (main)
$ git add .

olhak@Olia MINGW64 ~/it_projects (main)
$ git commit -m "second commit"
[main fe89eef] second commit
1 file changed, 13 insertions(+)
```

Запустила зміни:

```
olhak@Olia MINGW64 ~/it_projects (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 540 bytes | 540.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/olivka16/it_projects_lab1.git
5acee78..fe89eef  main -> main
```

Перевірила зміни:



**Висновок:** завдяки цій лабораторній роботі я ознайомилася з системами контролю версій. Зрозуміла принципи синхронізації робочих груп у гнучких командах. Набула навичок використання технічних засобів та протоколів для роботи з системами контролю версій.