

Міністерство освіти і науки, молоді і спорту України  
Західноукраїнський національний університет  
Факультет комп'ютерних інформаційних технологій

Кафедра ІОСУ

Лабораторна робота №6.1  
З дисципліни “Основи комп'ютерних наук”  
На тему: “Основи Роботи з GIT та GitHub”

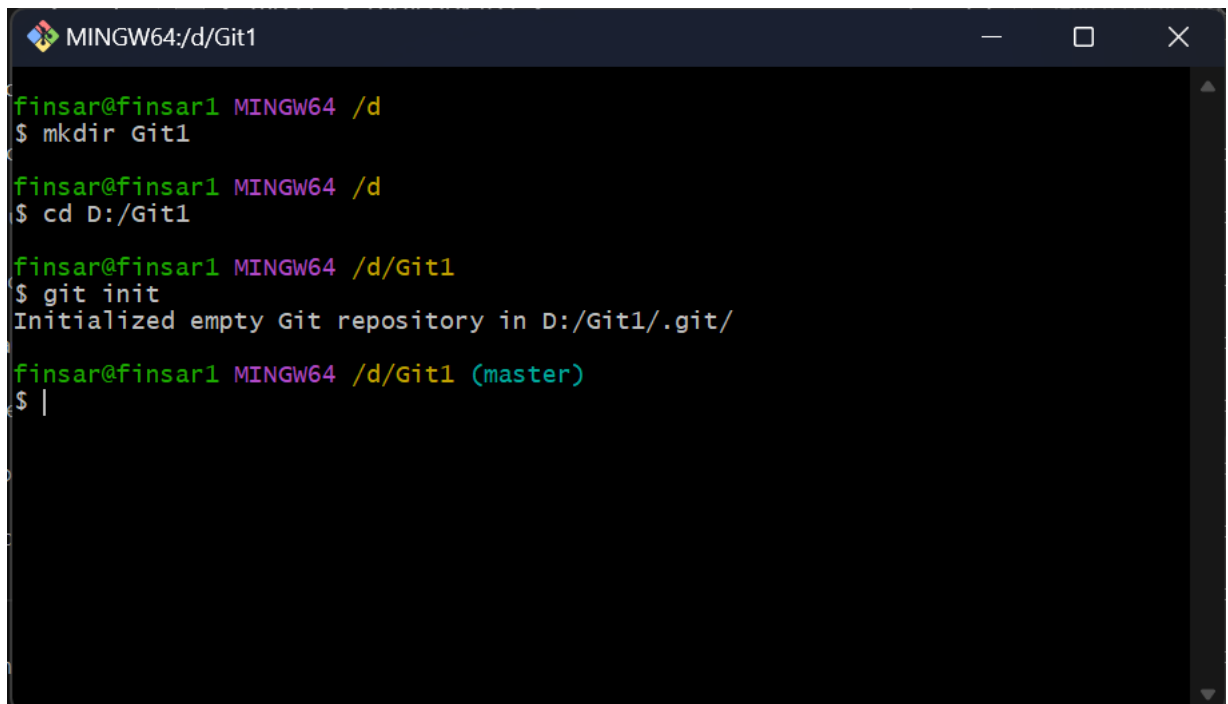
Виконав  
студент групи КН-11  
Третьяк Максим

Тернопіль 2023

**Мета:** Ознайомлення студентів з основами системи контролю версій GIT та платформи GitHub. Розробка навичок управління версіями коду та організації спільної роботи над проектом.

### Хід роботи

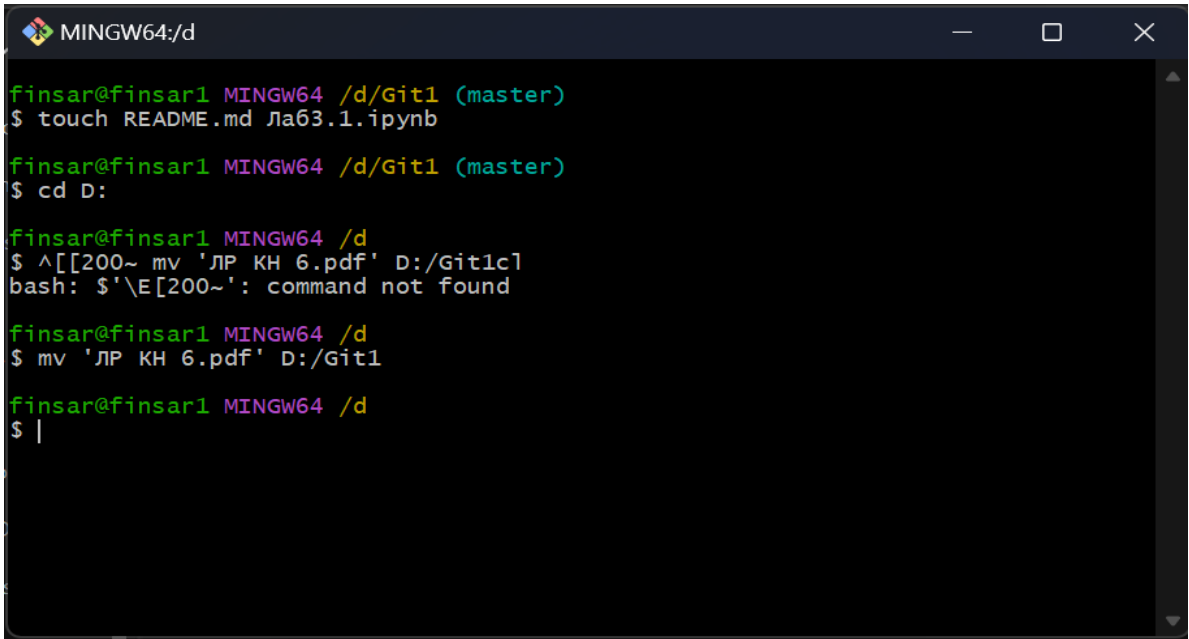
1. Створюю нову папку для репозиторію та ініціалізую її як GIT репозиторій (рисунок 1).



```
MINGW64:/d/Git1
finsar@finsar1 MINGW64 /d
$ mkdir Git1
finsar@finsar1 MINGW64 /d
$ cd D:/Git1
finsar@finsar1 MINGW64 /d/Git1
$ git init
Initialized empty Git repository in D:/Git1/.git/
finsar@finsar1 MINGW64 /d/Git1 (master)
$ |
```

Рисунок 1 – Вікно консолі Git Bash з введеними командами.

2. Додаю кілька файлів до репозиторію (README.md, Лаб3.1 ірпnb та звіт по ЛБ6) (рисунок 2).



```
MINGW64:/d

finsar@finsar1 MINGW64 /d/Git1 (master)
$ touch README.md Лаб3.1.ірпnb

finsar@finsar1 MINGW64 /d/Git1 (master)
$ cd D:

finsar@finsar1 MINGW64 /d
$ ^[[200~ mv 'ЛР КН 6.pdf' D:/Git1c1
bash: $'\E[200~': command not found

finsar@finsar1 MINGW64 /d
$ mv 'ЛР КН 6.pdf' D:/Git1

finsar@finsar1 MINGW64 /d
$ |
```

Рисунок 2 – Вікно Git Bash з введеними командами для створення файлів.

3. Роблю зміни в файлі README.md (рисунок 3).

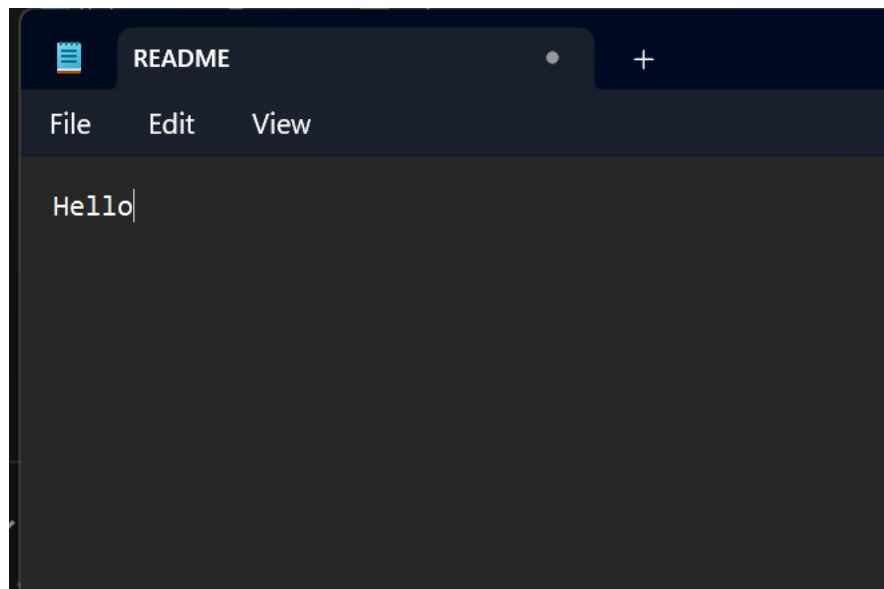
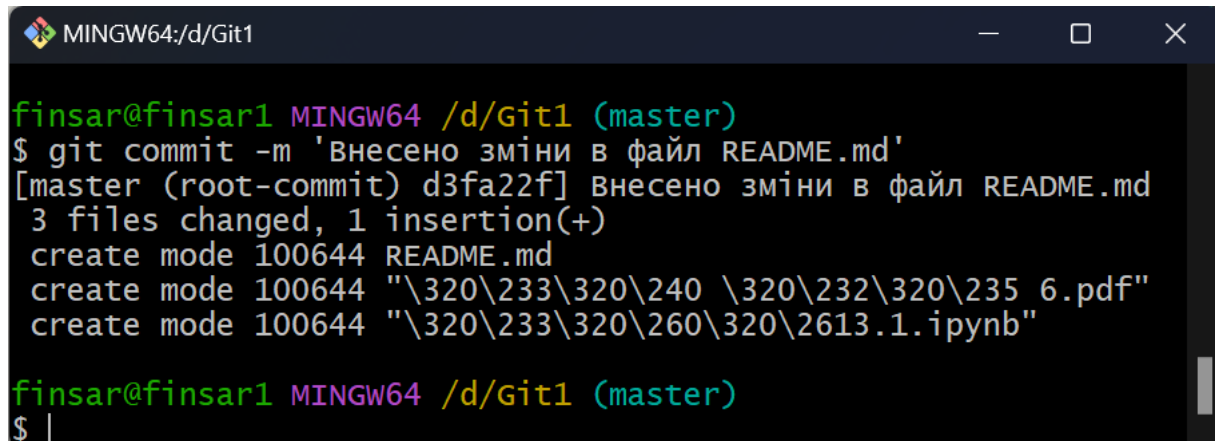


Рисунок 3 – Вікно файла README.md .

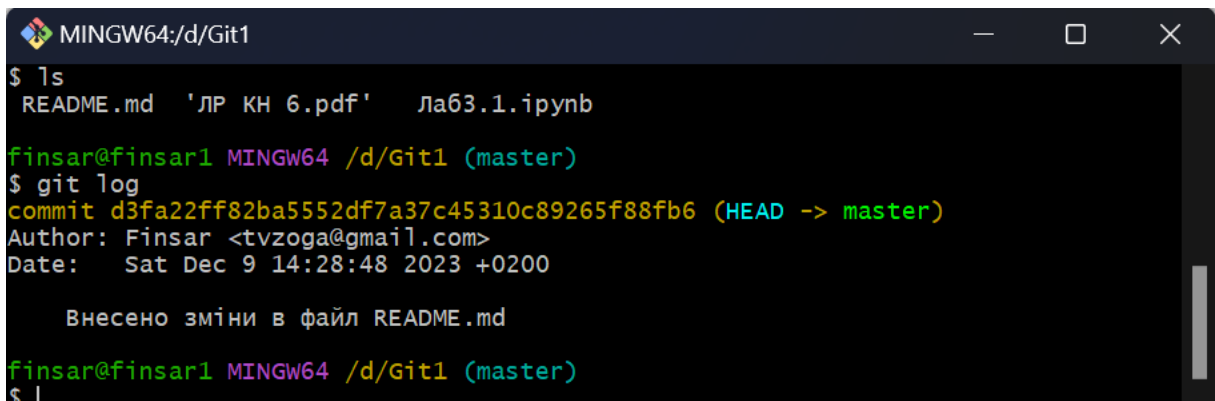
4. Використовую команди `git add`, `git commit` для збереження змін (рисунок 4).



```
MINGW64:/d/Git1
finsar@finsar1 MINGW64 /d/Git1 (master)
$ git commit -m 'Внесено зміни в файл README.md'
[master (root-commit) d3fa22f] Внесено зміни в файл README.md
3 files changed, 1 insertion(+)
create mode 100644 README.md
create mode 100644 "\\320\\233\\320\\240 \\320\\232\\320\\235 6.pdf"
create mode 100644 "\\320\\233\\320\\260\\320\\2613.1.ipynb"
finsar@finsar1 MINGW64 /d/Git1 (master)
$ |
```

Рисунок 4 – Вікно Git Bash з введеними командами для збереження змін.

5. Переглядаю історію комітів за допомогою `git log` (рисунок 5).



```
MINGW64:/d/Git1
$ ls
README.md 'ЛР КН 6.pdf' лаб3.1.ipynb
finsar@finsar1 MINGW64 /d/Git1 (master)
$ git log
commit d3fa22ff82ba5552df7a37c45310c89265f88fb6 (HEAD -> master)
Author: Finsar <tvzoga@gmail.com>
Date: Sat Dec 9 14:28:48 2023 +0200

    Внесено зміни в файл README.md
finsar@finsar1 MINGW64 /d/Git1 (master)
$ |
```

Рисунок 5 – Вікно Git Bash з введеною командою `git log`.

6. Створюю новий репозиторій на GitHub (рисунок 6).

**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

*Required fields are marked with an asterisk (\*).*

**Owner \*** **Repository name \***

Finsar / Git

✔ Git is available.

Great repository names are short and memorable. Need inspiration? How about [super-duper-doodle](#) ?

**Description** (optional)

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**

.gitignore template: **None**

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

**Choose a license**

License: **None**

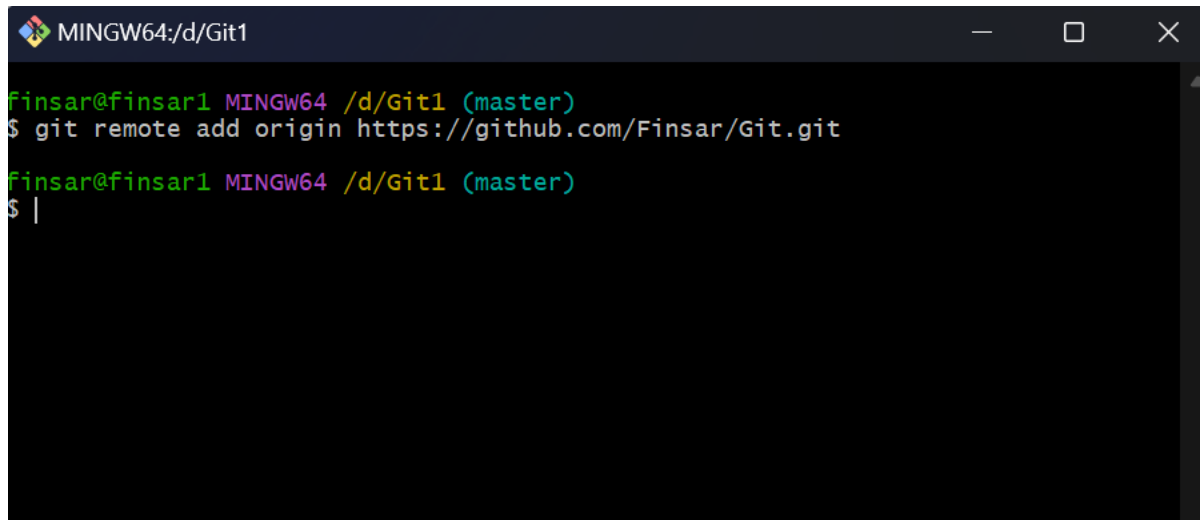
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

ⓘ You are creating a public repository in your personal account.

**Create repository**

Рисунок 6 – Вікно створення репозиторію на сайті GitHub

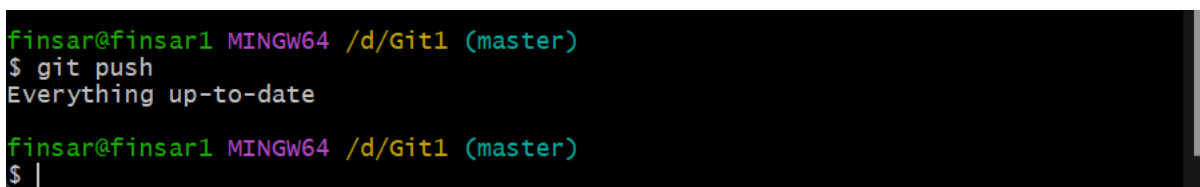
7. Зв'язую локальний репозиторій з віддаленим на GitHub за допомогою `git remote add origin [https://github.com/Finsar/Git.git]` (рисунок 7).

A screenshot of a Git Bash terminal window. The title bar at the top reads "MINGW64:/d/Git1". The terminal shows the prompt "finsar@finsar1 MINGW64 /d/Git1 (master)" followed by the command "\$ git remote add origin https://github.com/Finsar/Git.git". The prompt returns, and the next line shows the prompt "\$ |" with a vertical cursor.

```
MINGW64:/d/Git1
finsar@finsar1 MINGW64 /d/Git1 (master)
$ git remote add origin https://github.com/Finsar/Git.git
finsar@finsar1 MINGW64 /d/Git1 (master)
$ |
```

Рисунок 7 – Вікно Git Bash з введенною командою `git remote add origin [URL]`.

8. Ввожу `git push` для відправлення локальних змін на GitHub (рисунок 8).

A screenshot of a Git Bash terminal window. The title bar at the top reads "MINGW64:/d/Git1". The terminal shows the prompt "finsar@finsar1 MINGW64 /d/Git1 (master)" followed by the command "\$ git push". The output "Everything up-to-date" is displayed. The prompt returns, and the next line shows the prompt "\$ |" with a vertical cursor.

```
MINGW64:/d/Git1
finsar@finsar1 MINGW64 /d/Git1 (master)
$ git push
Everything up-to-date
finsar@finsar1 MINGW64 /d/Git1 (master)
$ |
```

Рисунок 8 – Вікно Git Bash з введенною командою `git push`.

9. Вношу зміни в файл на GitHub і виконую `git pull`, щоб синхронізувати зміни з локальним репозиторієм (рисунок 9-11).

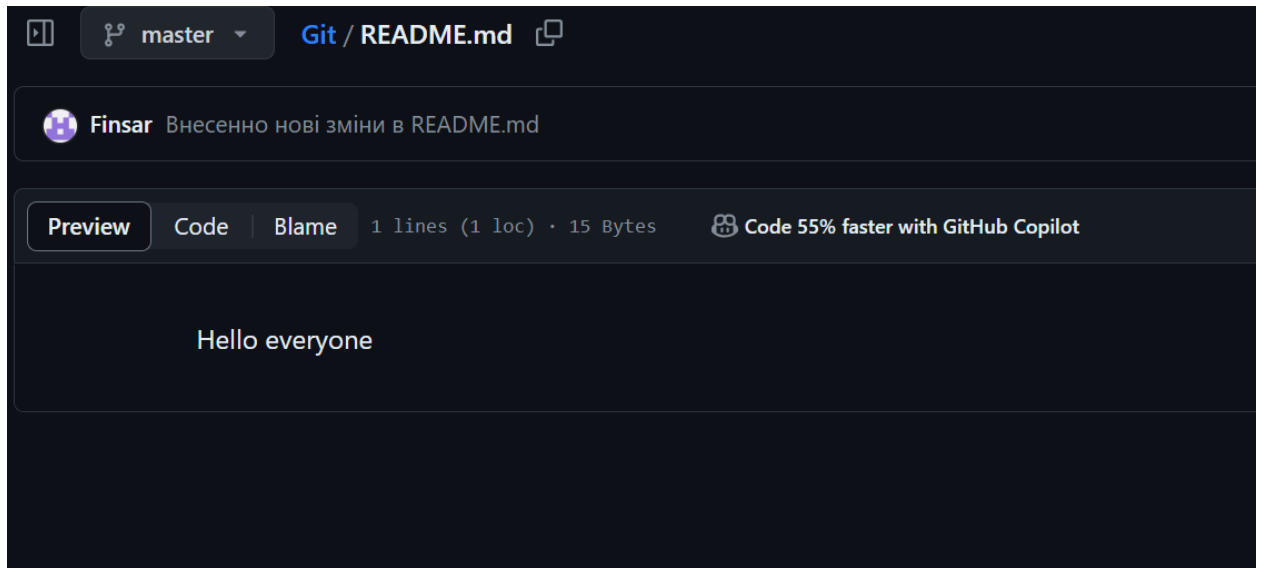


Рисунок 9 – Вікно GitHub з внесеними змінами в файлі.

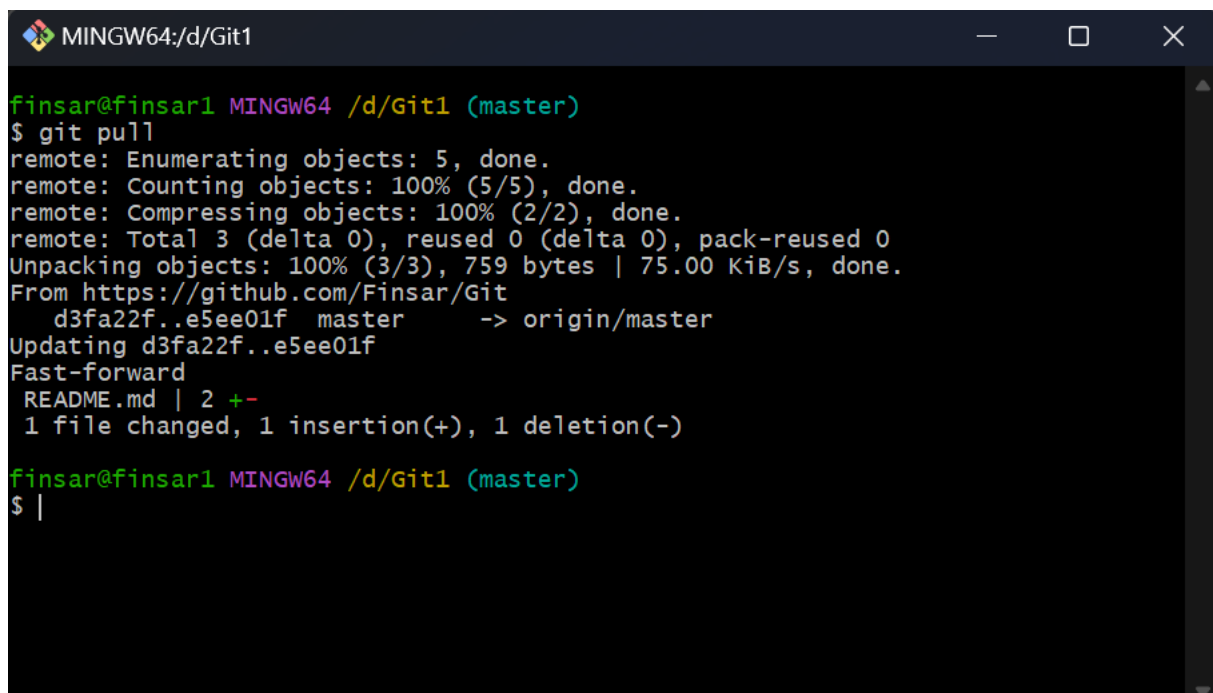


Рисунок 10 – Вікно Git Bash з введенною командою git pull.

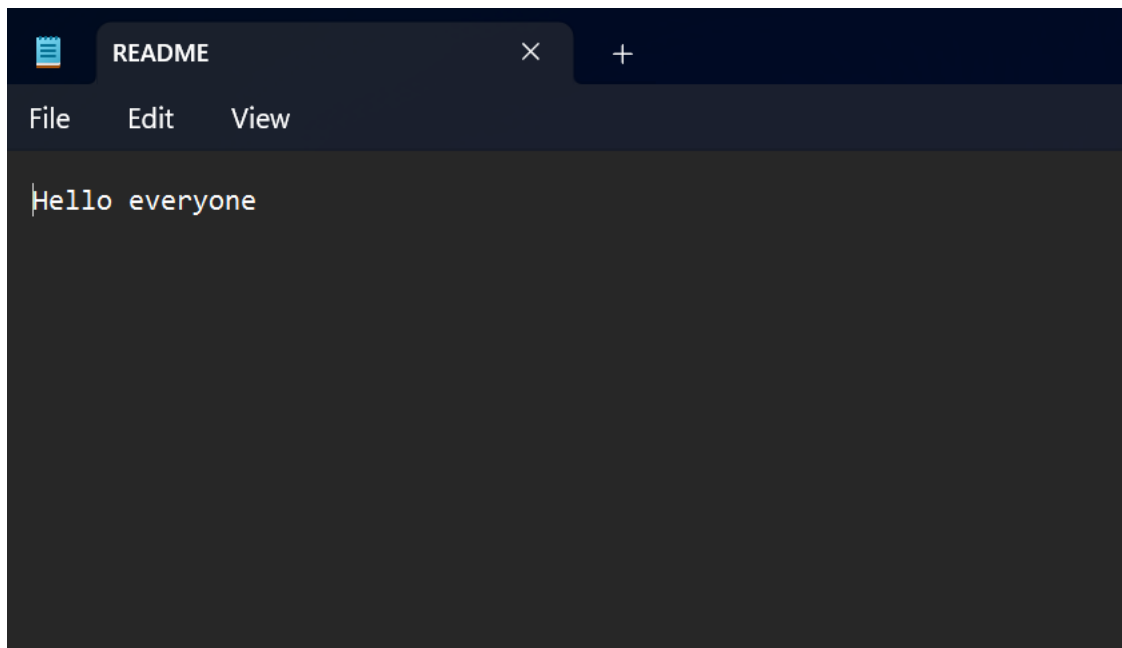


Рисунок 11 – Вікно файла який був змінений.

10. Створюю нову гілку за допомогою `git branch [LR6.1]` та переключаюсь на неї за допомогою `git checkout [LR6.1]` (рисунок 12).

A screenshot of a Git Bash terminal window. The title bar shows the path 'MINGW64:/d/Git1' and standard window controls. The terminal output is as follows:

```
finsar@finsar1 MINGW64 /d/Git1 (master)
$ git branch LR6.1

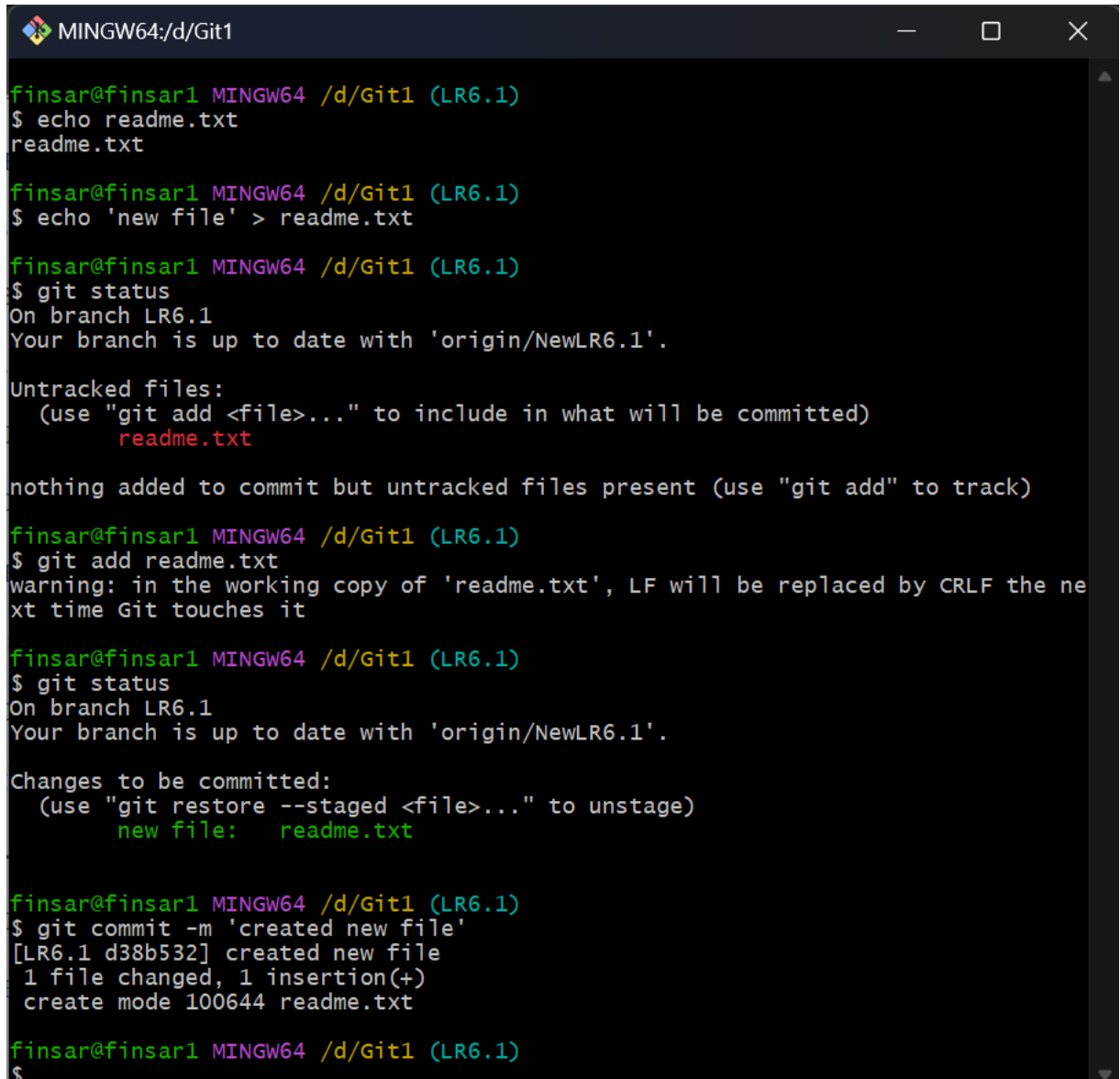
finsar@finsar1 MINGW64 /d/Git1 (master)
$ git checkout LR6.1
Switched to branch 'LR6.1'

finsar@finsar1 MINGW64 /d/Git1 (LR6.1)
$
```

Рисунок 12 – Вікно Git Bash з введеними командами `git branch` і `git checkout`.



11. Вношу зміни в гілці та роблю коміт (рисунок 13).



```
MINGW64:/d/Git1

finsar@finsar1 MINGW64 /d/Git1 (LR6.1)
$ echo readme.txt
readme.txt

finsar@finsar1 MINGW64 /d/Git1 (LR6.1)
$ echo 'new file' > readme.txt

finsar@finsar1 MINGW64 /d/Git1 (LR6.1)
$ git status
On branch LR6.1
Your branch is up to date with 'origin/NewLR6.1'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  readme.txt

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

finsar@finsar1 MINGW64 /d/Git1 (LR6.1)
$ git add readme.txt
warning: in the working copy of 'readme.txt', LF will be replaced by CRLF the next time Git touches it

finsar@finsar1 MINGW64 /d/Git1 (LR6.1)
$ git status
On branch LR6.1
Your branch is up to date with 'origin/NewLR6.1'.

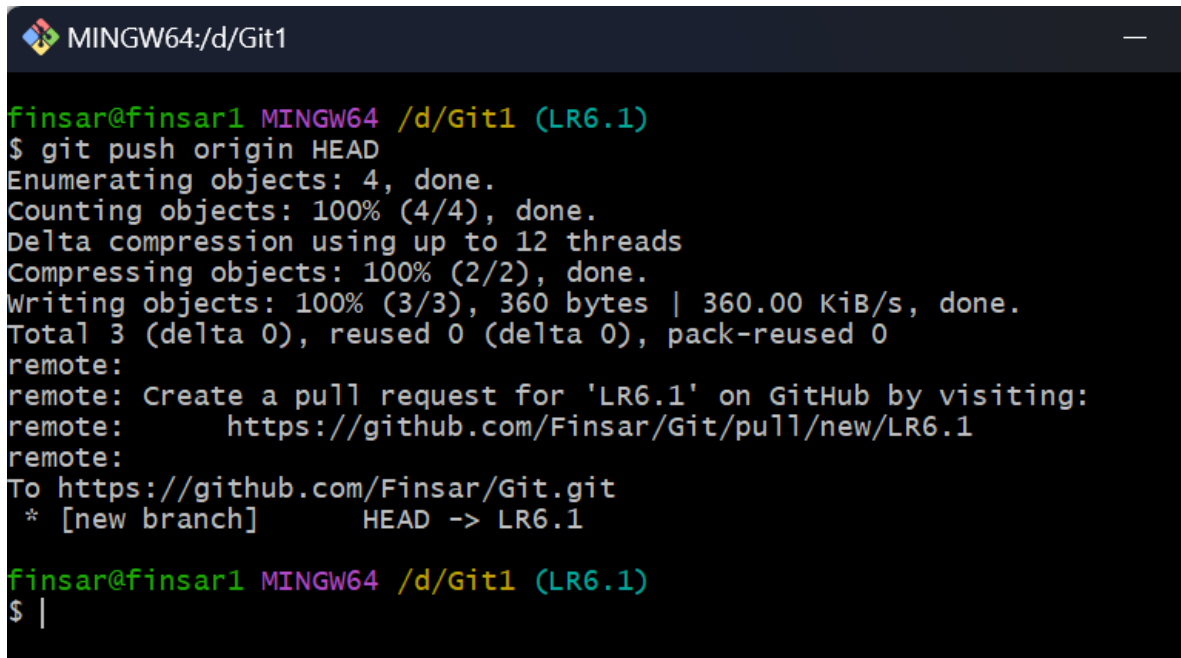
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  new file:   readme.txt

finsar@finsar1 MINGW64 /d/Git1 (LR6.1)
$ git commit -m 'created new file'
[LR6.1 d38b532] created new file
1 file changed, 1 insertion(+)
create mode 100644 readme.txt

finsar@finsar1 MINGW64 /d/Git1 (LR6.1)
$
```

Рисунок 13 – Вікно з внесеними змінами в гілці та добавленими комітом.

12. Відправляю зміни з гілки на GitHub за допомогою git push (рисунок 14).



```
MINGW64:/d/Git1

finsar@finsar1 MINGW64 /d/Git1 (LR6.1)
$ git push origin HEAD
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 360 bytes | 360.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'LR6.1' on GitHub by visiting:
remote:      https://github.com/Finsar/Git/pull/new/LR6.1
remote:
To https://github.com/Finsar/Git.git
 * [new branch]      HEAD -> LR6.1

finsar@finsar1 MINGW64 /d/Git1 (LR6.1)
$ |
```

Рисунок 14 – Вікно Git Bash з введеною командою git push.

13. На GitHub створюю Pull Request для об'єднання гілки з основною гілкою. Опишіть зміни та виконайте Merge Pull Request. (рисунок 15-17).

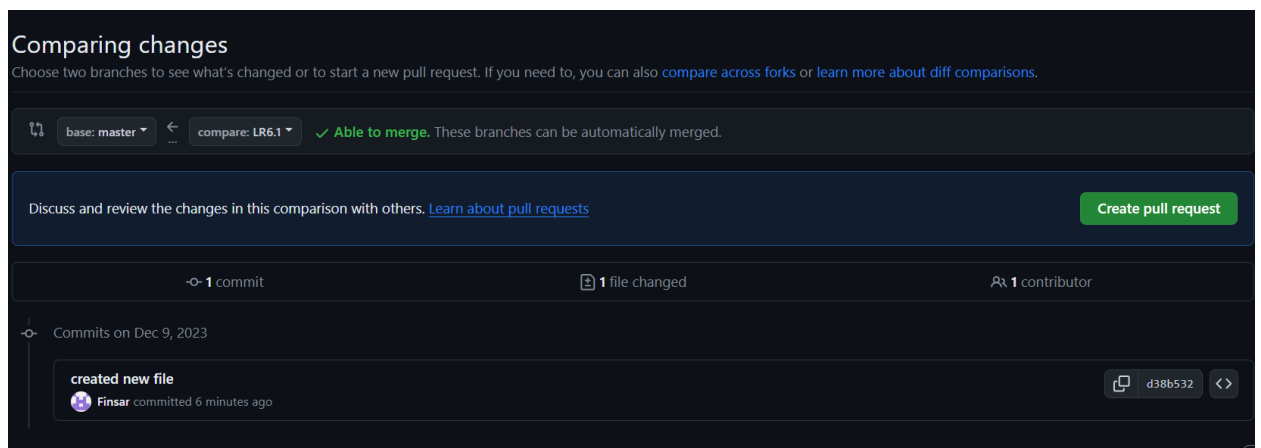


Рисунок 15 – Вікно створення Pull Request на GitHub.

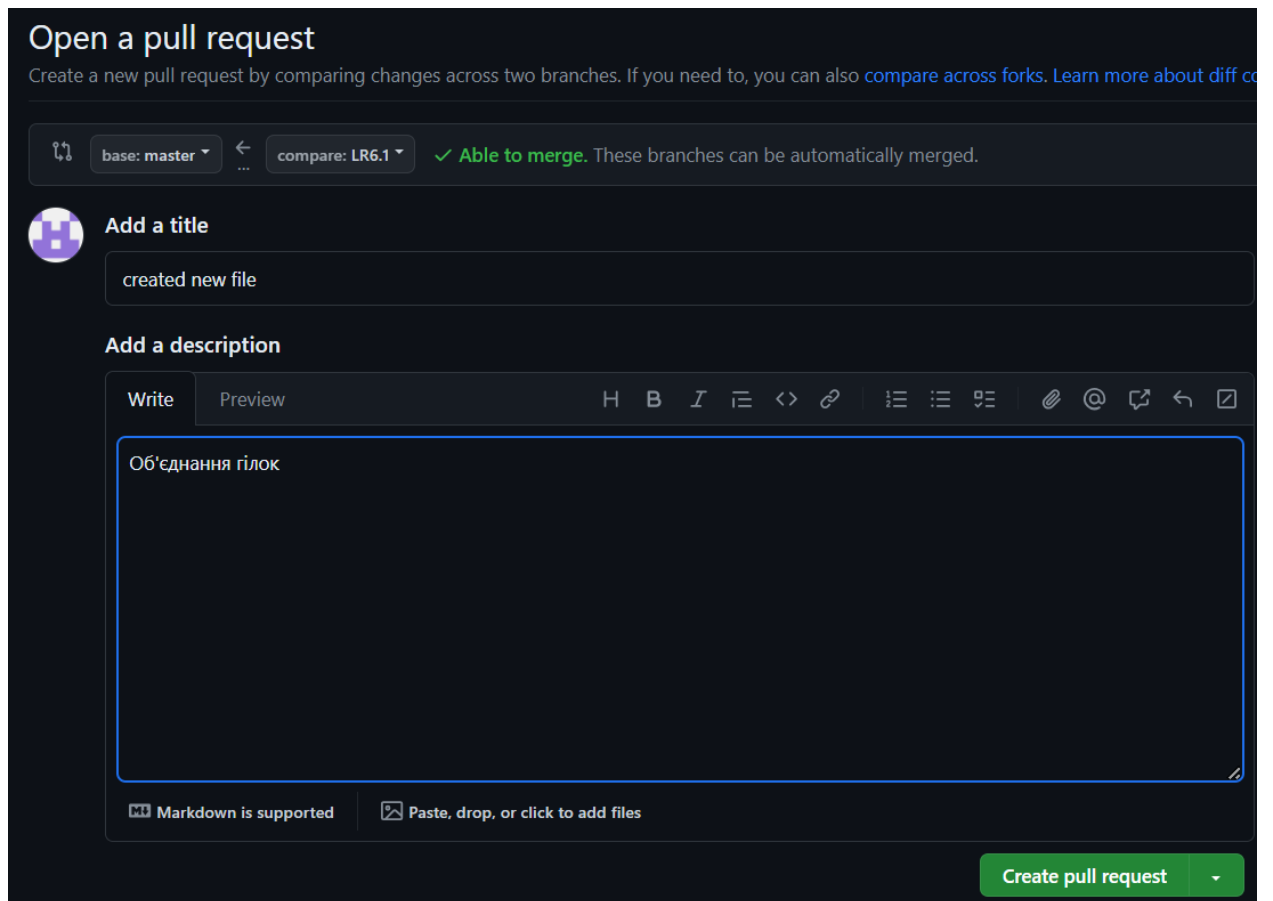


Рисунок 16 – Вікно опису змін GitHub.

```
MINGW64:/d/Git1

finsar@finsar1 MINGW64 /d/Git1 (LR6.1)
$ git merge master
Already up to date.

finsar@finsar1 MINGW64 /d/Git1 (LR6.1)
$ |
```

Рисунок 17 – Вікно Git Bash з введеною командою git merge.

14. Прохожу тренажер «Вступ», «Їдемо далі» та «Переміщуємо роботу туди-сюди» (рисунок 18).

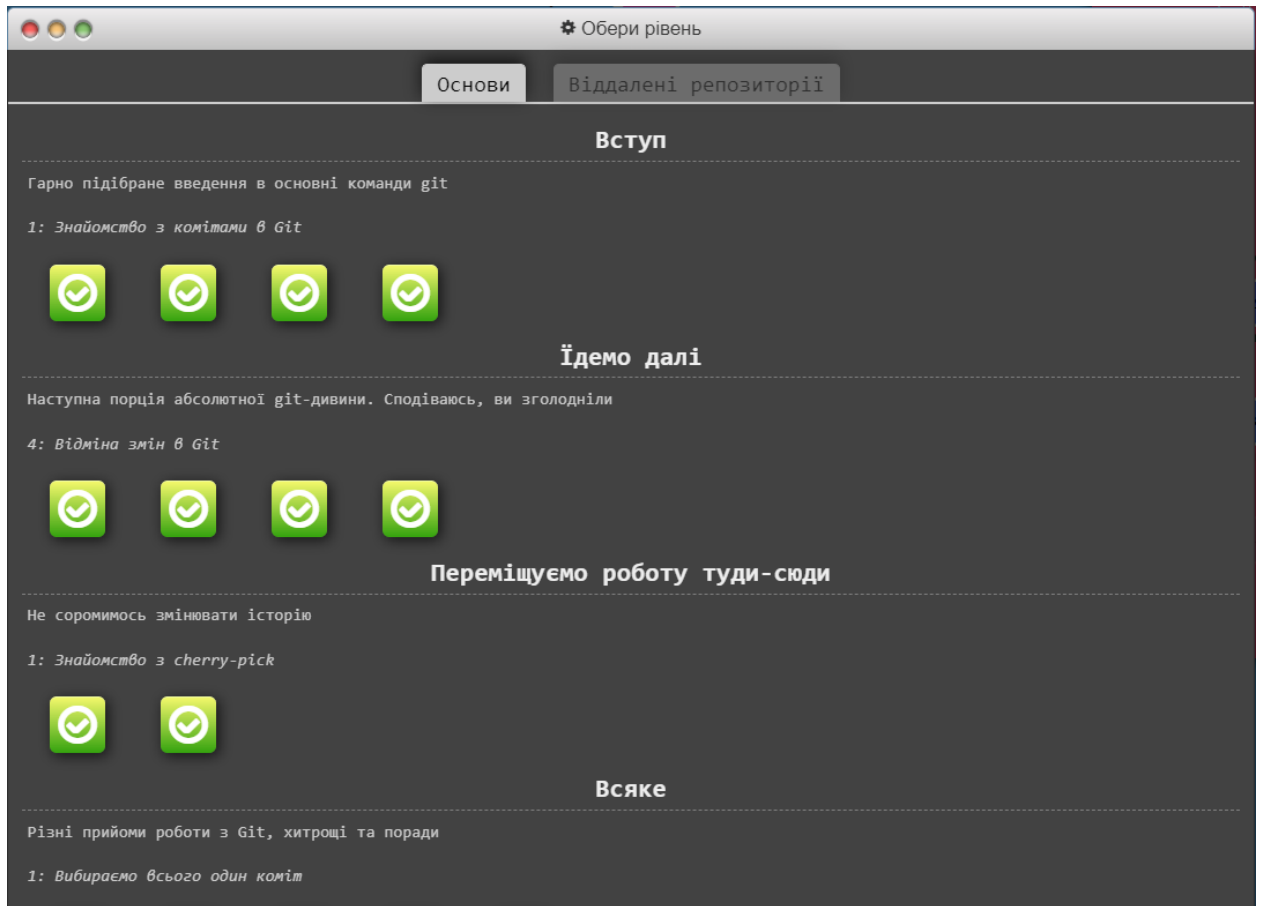


Рисунок 18 – Вікно пройденого тренажеру.

### Висновок:

Я навчився основам роботи з GIT та GitHub, включаючи управління версіями, роботу з гілками, створення Pull Request та співпрацю в командах та пройшов тренажер.