

Міністерство освіти і науки, молоді та 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 КНШІ-11

Гарболінський Владислав

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 ініціалізуємо її як git репозиторій (рисунок 1-3)

| Ім'я | Дата змінення | Тип | Розмір |
|----------------|------------------|--------------|--------|
| .VirtualBox | 27.10.2023 9:33 | Папка файлів | |
| 3D-об'єкти | 29.07.2023 19:31 | Папка файлів | |
| ansel | 29.07.2023 21:50 | Папка файлів | |
| AppData | 29.07.2023 19:31 | Папка файлів | |
| build | 07.10.2023 11:46 | Папка файлів | |
| GIT4 | 09.12.2023 14:32 | Папка файлів | |
| node_modules | 08.10.2023 18:33 | Папка файлів | |
| OneDrive | 29.07.2023 19:33 | Папка файлів | |
| source | 04.09.2023 15:45 | Папка файлів | |
| telegram-bot | 08.10.2023 18:30 | Папка файлів | |
| tgbot-cpp | 08.10.2023 18:03 | Папка файлів | |
| venv | 07.10.2023 12:40 | Папка файлів | |
| VirtualBox VMs | 26.10.2023 21:05 | Папка файлів | |
| Відеозаписи | 12.10.2023 19:16 | Папка файлів | |

Рисунок 1 - Вікно провідника та створення папки для репозиторію

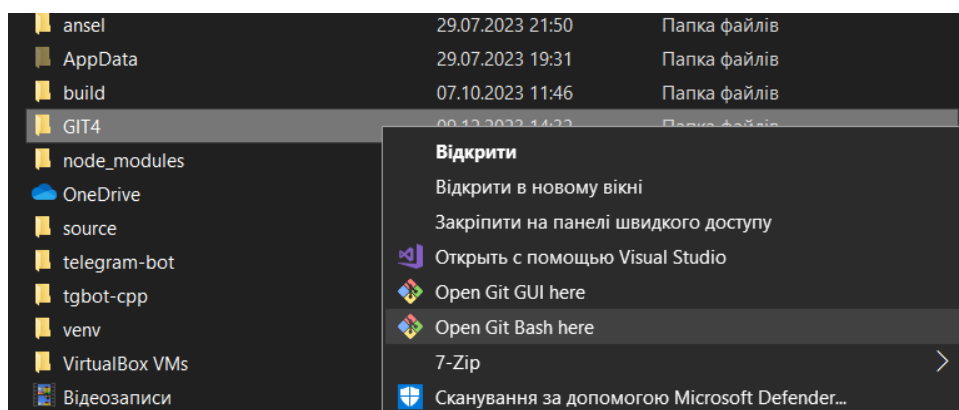


Рисунок 2 - Вікно провідника, відкриття папки через GitBash

```
MINGW64:/c/Users/Владік/GIT4
Владік@DESKTOP-R0B2PFM MINGW64 ~/GIT4
$ git init
Initialized empty Git repository in C:/Users/Владік/GIT4/.git/
```

Рисунок 3 - Вікно Git Bash з командою git init

2. Додаю до папки декілька файлів, робимо зміни в одному (рисунок 4-5)

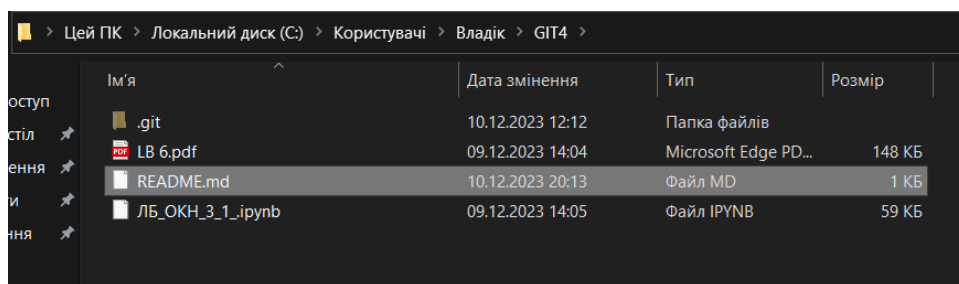


Рисунок 4 - Вікно провідника та декілька файлів в папці

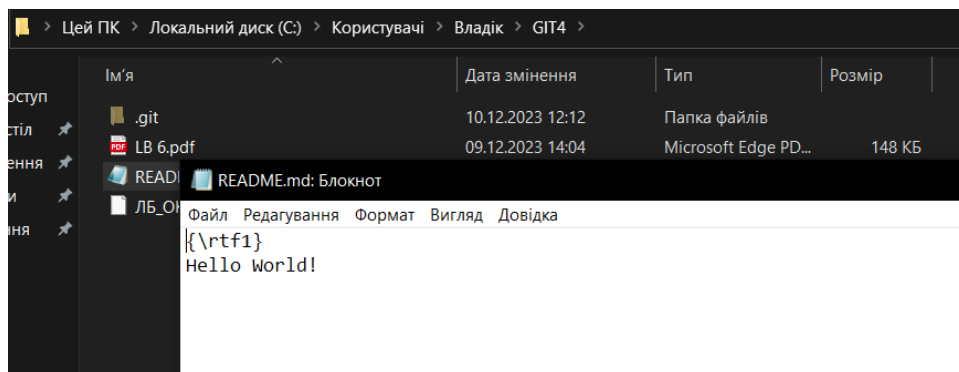
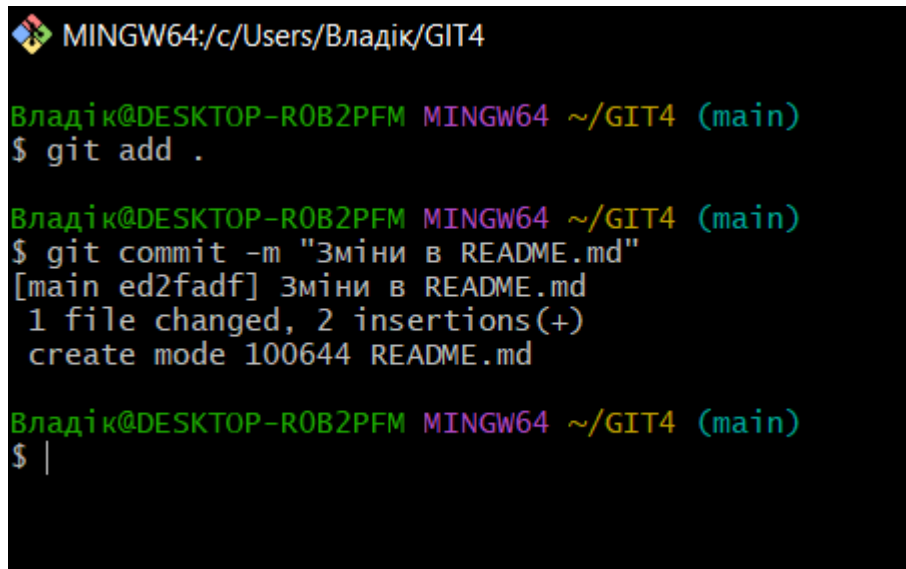


Рисунок 5 - Вікно документу README.md

3. Використовую команди `git add`, `git commit` для збереження змін та `git log` для перевірки змін (рисунок 6-7)



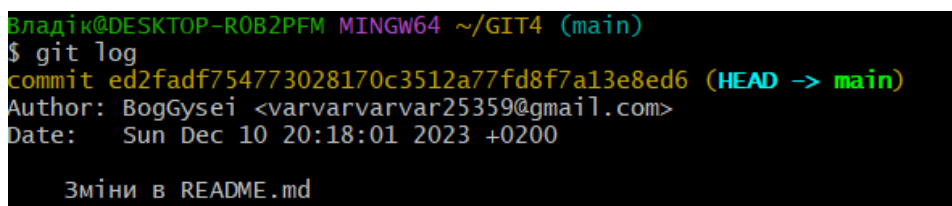
```
MINGW64:/c/Users/Владік/GIT4

Владік@DESKTOP-R0B2PFM MINGW64 ~/GIT4 (main)
$ git add .

Владік@DESKTOP-R0B2PFM MINGW64 ~/GIT4 (main)
$ git commit -m "Зміни в README.md"
[main ed2fadf] Зміни в README.md
1 file changed, 2 insertions(+)
create mode 100644 README.md

Владік@DESKTOP-R0B2PFM MINGW64 ~/GIT4 (main)
$ |
```

Рисунок 6 - збереження змін командами `git add` та `git commit`



```
Владік@DESKTOP-R0B2PFM MINGW64 ~/GIT4 (main)
$ git log
commit ed2fadf754773028170c3512a77fd8f7a13e8ed6 (HEAD -> main)
Author: BogGysei <varvarvarvar25359@gmail.com>
Date:   Sun Dec 10 20:18:01 2023 +0200

    Зміни в README.md
```

Рисунок 7 - вікно Git Bash з командою `git log`

3. 1) Створення нового репозиторій на GitHub (рисунок 8)

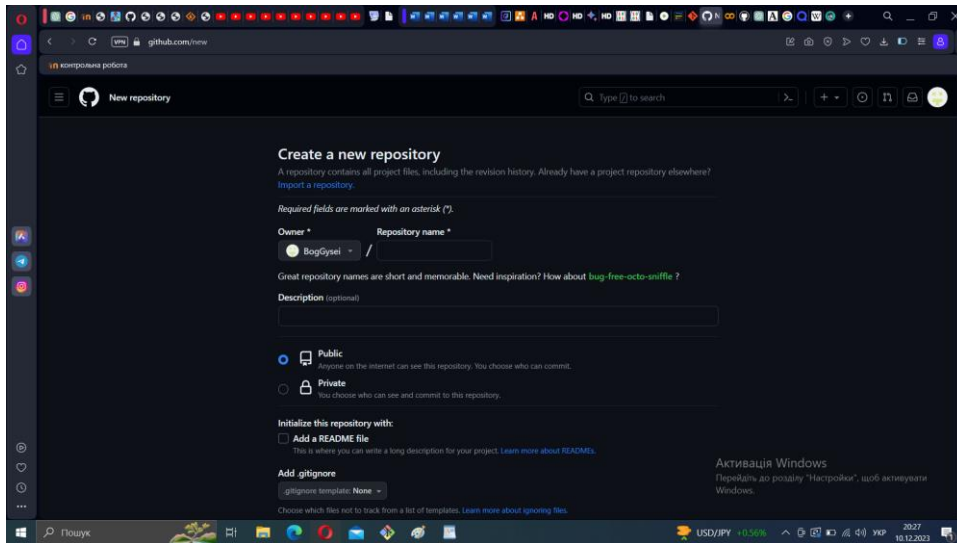


Рисунок 8 - створення нового репозиторія на сайті Github

2) Зв'язую локальний репозиторій з віддаленим на GitHub за допомогою `git remote add origin https://github.com/BogGysei/GIT4.git` (рисунок 9)



Рисунок 9 - Вікно Git Bash з командою `git remote add origin`
<https://github.com/BogGysei/GIT4.git>

4. 1) Ввожу `git push` для відправлення локальних змін на GitHub (рисунок 10)

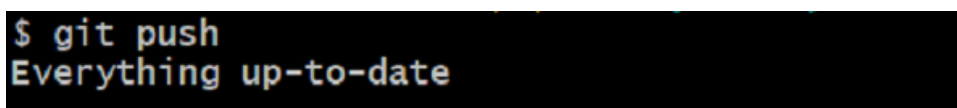


Рисунок 10 - Вікно Git Bash з введеною командою `git push`

2) Вношу зміни в файл на GitHub і виконую `git pull`, щоб синхронізувати зміни з локальним репозиторієм (рисунок 10-11).

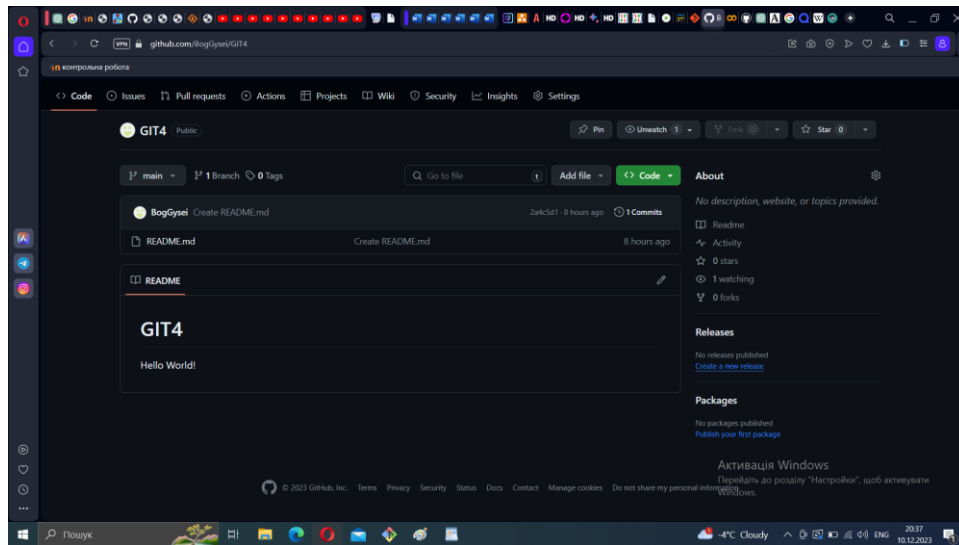


Рисунок 10 - Вікно браузера з внесеними змінами в файл README на сайті Github

```
Владік@DESKTOP-ROB2PFM MINGW64 ~/GIT4 (main)
$ git pull
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 603 bytes | 21.00 KiB/s, done.
From https://github.com/BogGyse/GIT4
* [new branch]      main       -> origin/main
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

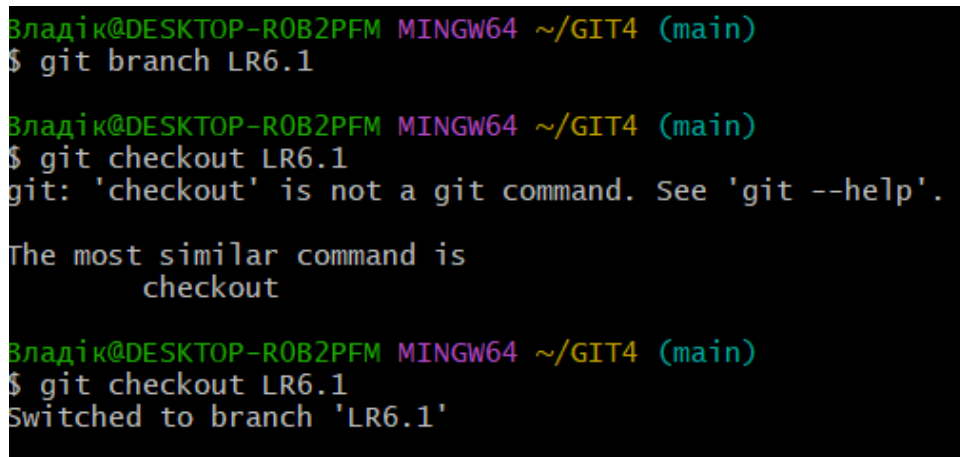
    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=origin/<branch> main
```

Рисунок 11 - Вікно Git Bash з введеною командою `git pull`

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



```
Владік@DESKTOP-R0B2PFM MINGW64 ~/GIT4 (main)
$ git branch LR6.1

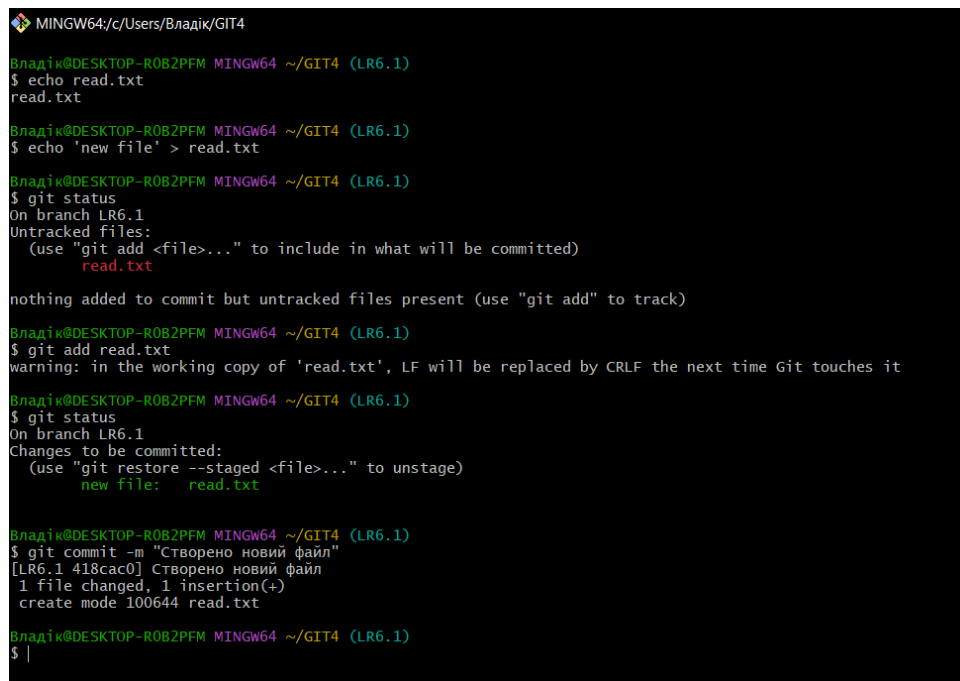
Владік@DESKTOP-R0B2PFM MINGW64 ~/GIT4 (main)
$ git checkout LR6.1
git: 'checkout' is not a git command. See 'git --help'.

The most similar command is
    checkout

Владік@DESKTOP-R0B2PFM MINGW64 ~/GIT4 (main)
$ git checkout LR6.1
Switched to branch 'LR6.1'
```

Рисунок 12 - Вікно GitBash з введеними командами `git branch LR6.1` та `git checkout LR6.1`

2) Вношу зміни та створюю новий коміт (рисунок 13)



```
MINGW64/c/Users/Владік/GIT4

Владік@DESKTOP-R0B2PFM MINGW64 ~/GIT4 (LR6.1)
$ echo read.txt
read.txt

Владік@DESKTOP-R0B2PFM MINGW64 ~/GIT4 (LR6.1)
$ echo 'new file' > read.txt

Владік@DESKTOP-R0B2PFM MINGW64 ~/GIT4 (LR6.1)
$ git status
On branch LR6.1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    read.txt

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

Владік@DESKTOP-R0B2PFM MINGW64 ~/GIT4 (LR6.1)
$ git add read.txt
warning: in the working copy of 'read.txt', LF will be replaced by CRLF the next time Git touches it

Владік@DESKTOP-R0B2PFM MINGW64 ~/GIT4 (LR6.1)
$ git status
On branch LR6.1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   read.txt

Владік@DESKTOP-R0B2PFM MINGW64 ~/GIT4 (LR6.1)
$ git commit -m "Створено новий файл"
[LR6.1 418cac0] Створено новий файл
1 file changed, 1 insertion(+)
create mode 100644 read.txt

Владік@DESKTOP-R0B2PFM MINGW64 ~/GIT4 (LR6.1)
$ |
```

Рисунок 13 - Вікно GitBash з процесом створення нового файлу та коміту

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

```
Владік@DESKTOP-ROB2PFM MINGW64 ~/GIT4 (LR6.1)
$ git push --set-upstream origin LR6.1
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), 312 bytes | 312.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object
remote:
remote: Create a pull request for 'LR6.1' on GitHub by visiting:
remote:   https://github.com/BogGysei/GIT4/pull/new/LR6.1
remote:
To https://github.com/BogGysei/GIT4.git
 * [new branch]      LR6.1 -> LR6.1
branch 'LR6.1' set up to track 'origin/LR6.1'.
```

Рисунок 14 - Вікно GitBash з командою git push

2) На GitHub створюю Pull Request для об'єднання гілки з основною гілкою (рисунок 15-17)

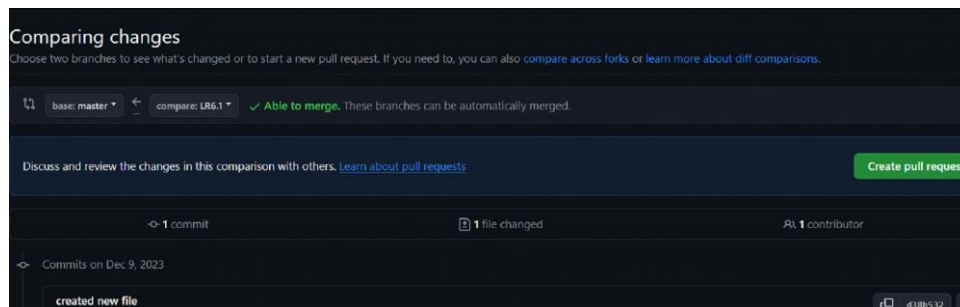


Рисунок 15 - Вікно створення Pull Request на GitHub.

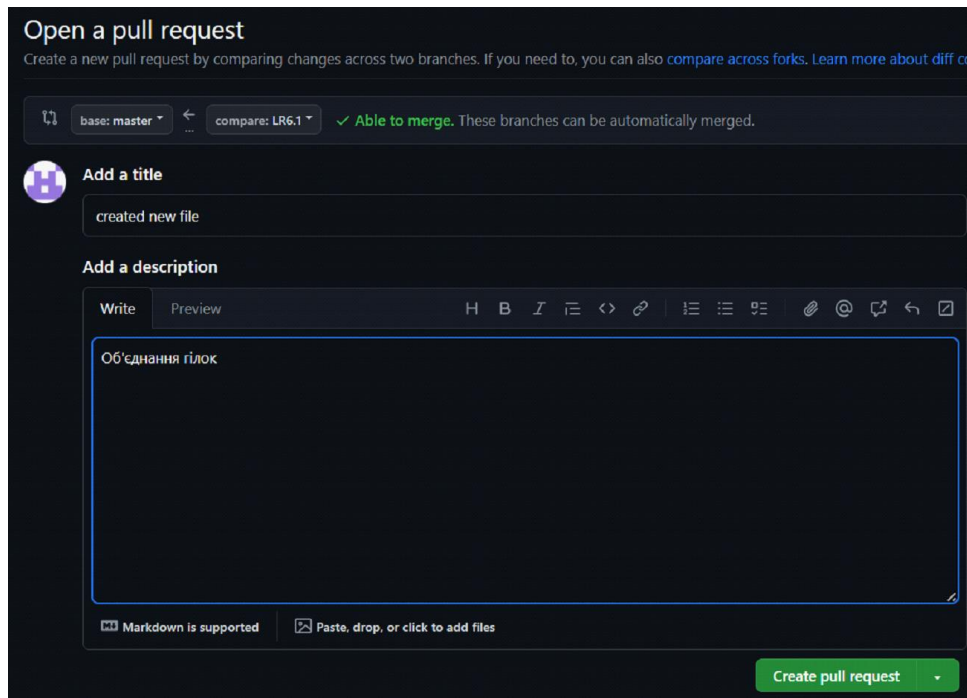


Рисунок 16 - Вікно опису змін GitHub.

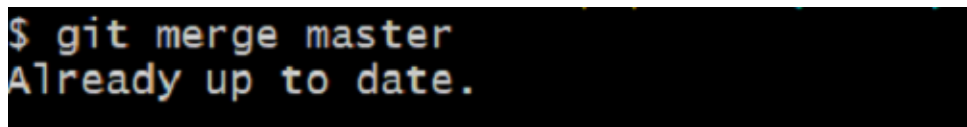


Рисунок 17 - Вікно Git Bash з введеною командою git merge.

7. Проходжу на тренажері «Вступ», «Їдемо далі» та «Переміщуємо роботу туди-сюди» (рисунок 18)

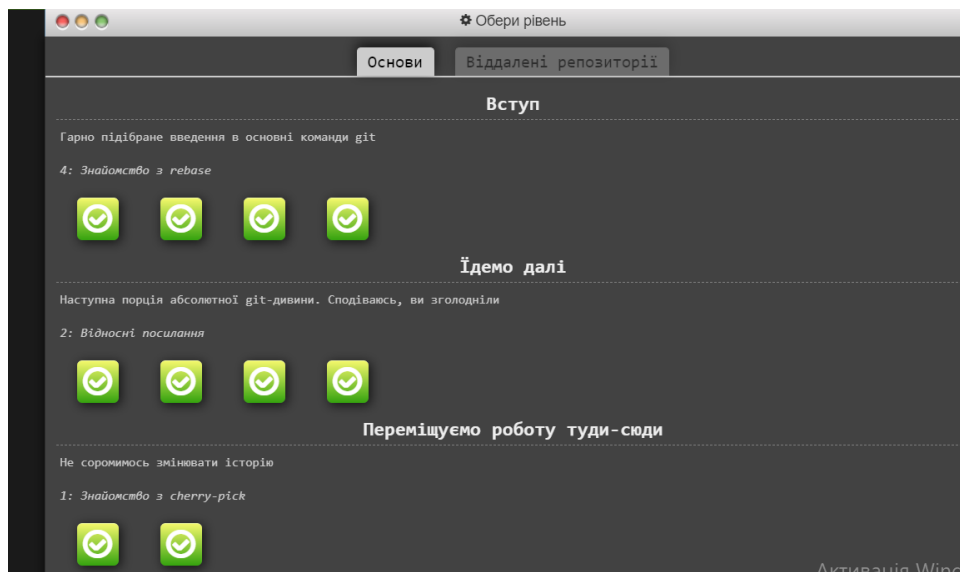


Рисунок 18 - Пройдений тренажер

Висновок: Ознайомлено з основами системи контролю версій GIT та платформи GitHub. Розроблено навички управління версіями коду та організації спільної роботи над проектом