

**Міністерство освіти і науки України
Національний університет «Львівська політехніка»**



ЗВІТ

Про виконання лабораторної роботи № 1

**На тему: “Налаштування *Git*”
з дисципліни «Моделювання та аналіз ПЗ»**

Лектор:

доцент кафедри ПЗ
Сердюк П.В.

Виконав:

студент. групи ПЗ-22
Місяйло О.О.

Прийняв:

викладач ПЗ
Микуляк А.В.

«___» _____ 2023 р.

Σ = _____

Львів – 2023

Тема: Налаштування Git.

Мета: Освоїти базові навички роботи з Git.

Теоретичні відомості

Git – це розподілена система контролю версій, яка дозволяє відстежувати історію розробки ПЗ і спільно працювати над складними проектами з будь-якої точки світу.

Переваги Git над іншими системами:

Головна перевага Git – в тому, що він дуже швидкий і прозорий. Він зручний для нелінійної розробки і ефективний як для невеликих проектів, так і для великих систем з тисячами учасників.

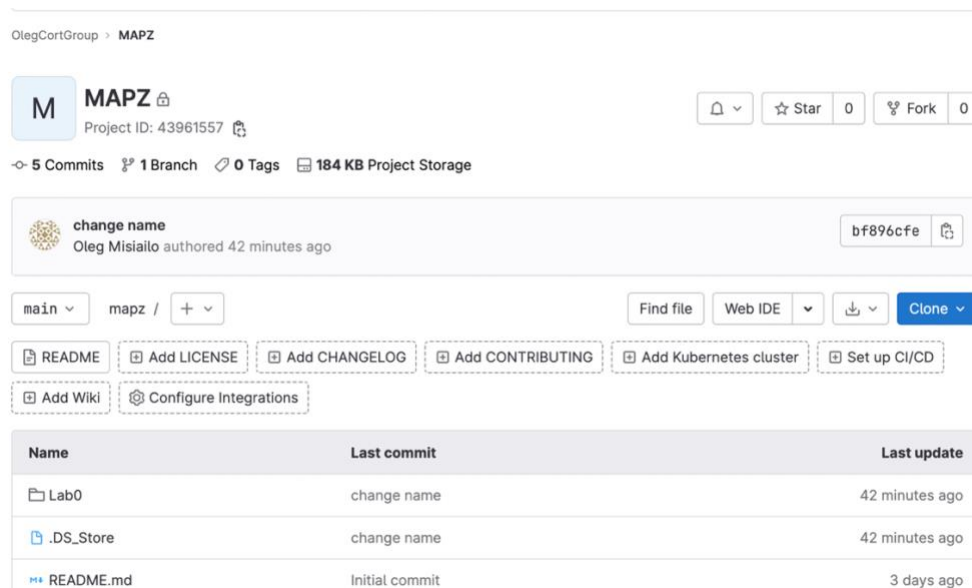
Завдання

1. Переглянути відео
<https://www.youtube.com/watch?v=IO4IGv9TLnQ&list=PLCSO5njdNs1-jYfAkuteDSm1eivMqs6Bw>
2. Створити аккаунт створювати на GitLab (тільки GitLab - не GitHub, і не жоден інший)
3. Створити репозиторій назвати MAPZ (тільки така назва)
4. Створити папку для лабораторних на компютері. Запустити там git clone для репозиторію
`git clone [url репозиторію]`
5. Створити структура папок локально на компютері- верхня MAPZ, всередині Lab0, Lab1 ,.....
6. Надати доступ до вашого репозиторію **pavlo_serdyuk**, з правами доступу до коду (низькі права не дозволяють переглядати код)
7. Додати посилання на ваш репозиторійу spreadsheet файл, доступ до кого має забезпечити староста групи
8. Проробити усі дії на сайті <https://learngitbranching.js.org/> до 5 рівня (Advanced не потрібно) і повторити їх у себе локально на комп'ютері
9. Усі дії робити у папці Lab0. Для комітів використовувати текстові файли, щоб зробити новий коміт, достатньо змінити лінійку тексту всередині файлу.
10. Для кожного завдання використовуйте новий текстовий файл.

11. Звіти повинні містити скріншоти з операціями - для цього можна використати лог, що відобразить історію, або любую візуальну надбудову Git - GitKraken, SourceTree, TortoiseGit.
12. Звіти завантажити на ВНС.

Хід виконання

1. На початку виконання я створив акаунт на GitLab та зробив там репозиторій з назвою MAPZ



До папки MAPZ я добавив папку lab0.

2. Після цього я проходжу по рівнях та пунктах з сайту learngitbranching та паралельно виконую ті ж операції на комп'ютері.

Рис 1. Завдання 1.1. Створення комітів

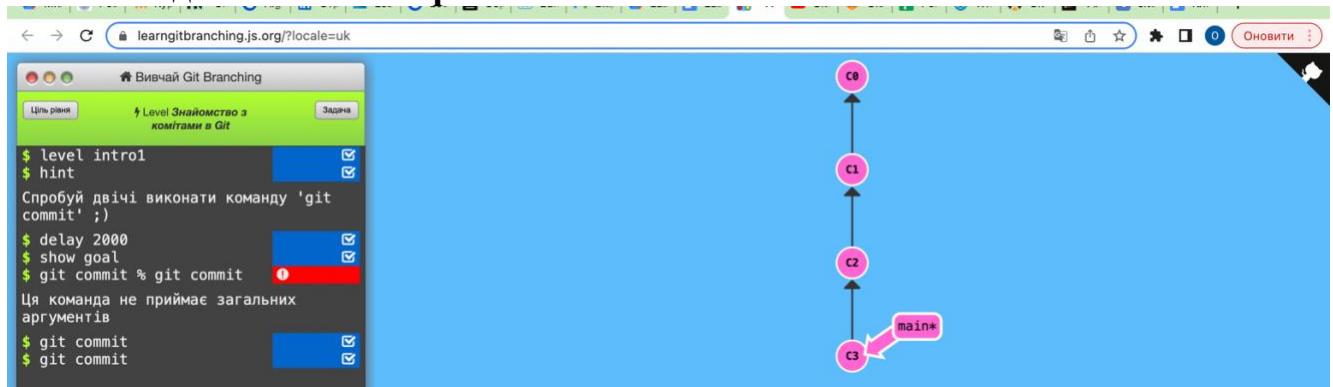


Рис 2.

```

[olegmisialo@MacBook-Air-Oleg mapz % git log
commit 520f069f5d6c1672b17c0457225055a840fef7b1 (HEAD -> main, origin/main, origin/HEAD)
Author: OlegCort <olegmisialo@gmail.com>
Date: Sun Mar 5 17:20:13 2023 +0200

    1:1 com2

commit a325a7d56cf6b0bc7cc6d5c005379236aaac0d0e
Author: OlegCort <olegmisialo@gmail.com>
Date: Sun Mar 5 17:19:41 2023 +0200

    1:1 com1

```

Рис 3. Завдання 1.2. Перенесення вказівника гілки

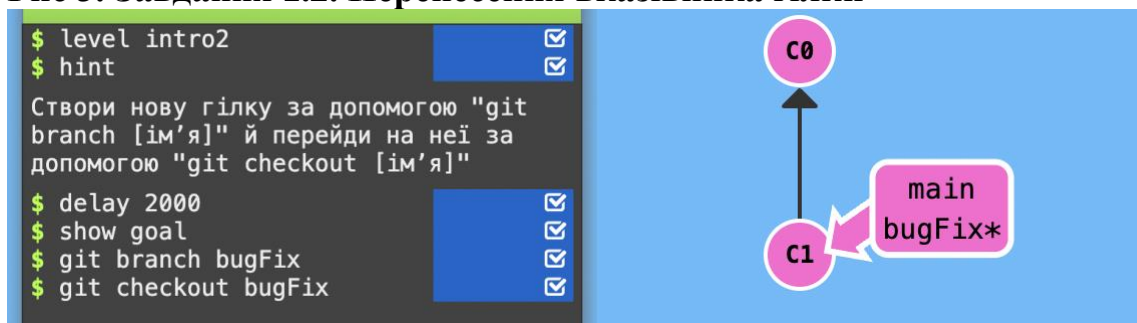


Рис. 4

```

[olegmisialo@MacBook-Air-Oleg mapz % git log
commit daf806435a62ab82dc16c13bd327f91fb5c784da (HEAD -> bugFix)
Author: OlegCort <olegmisialo@gmail.com>
Date: Sun Mar 5 17:32:30 2023 +0200

    1:2

```

Рис 5. Завдання 1.3. Застосування merge

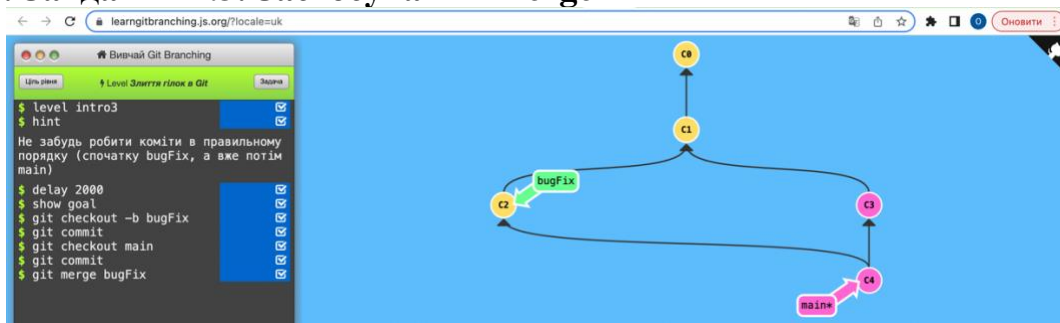


Рис 6.

```
olegmisialo@MacBook-Air-Oleg mapz % git checkout bugFix
Already on 'bugFix'
Your branch is up to date with 'origin/bugFix'.
olegmisialo@MacBook-Air-Oleg mapz % touch Lab0/song1_3.txt
olegmisialo@MacBook-Air-Oleg mapz % git add .
olegmisialo@MacBook-Air-Oleg mapz % git commit -m "added 2 lines"
[bugFix 3ae05ba] added 2 lines
1 file changed, 4 insertions(+)
create mode 100644 Lab0/song1_3.txt
olegmisialo@MacBook-Air-Oleg mapz % git main
git: 'main' is not a git command. See 'git --help'.

The most similar command is
    mailinfo
olegmisialo@MacBook-Air-Oleg mapz % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
olegmisialo@MacBook-Air-Oleg mapz % git merge bugFix
```

Рис 7. Завдання 1.4. Застосування rebase

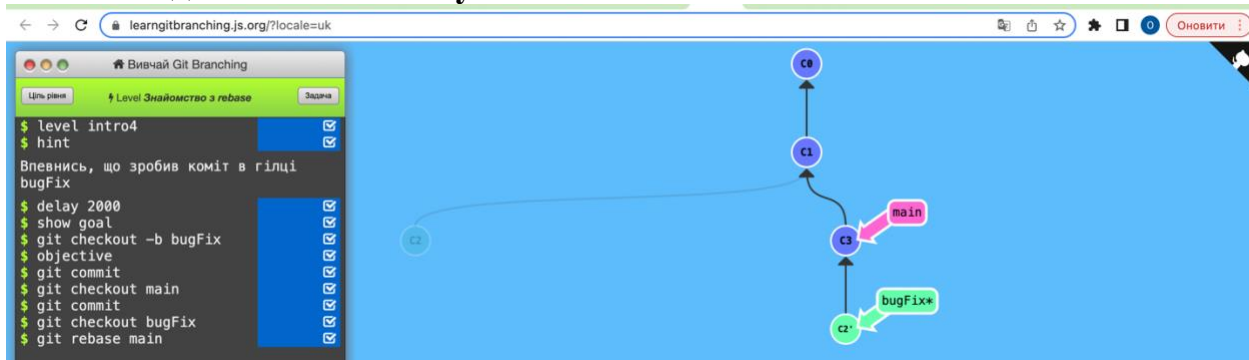


Рис 8.

```

olegmisialo@MacBook-Air-Oleg mapz % git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
olegmisialo@MacBook-Air-Oleg mapz % git commit -m "1:4 com from main"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
olegmisialo@MacBook-Air-Oleg mapz % git checkout bugFix
Switched to branch 'bugFix'
Your branch is ahead of 'origin/bugFix' by 2 commits.
(use "git push" to publish your local commits)
olegmisialo@MacBook-Air-Oleg mapz % git rebase main
Successfully rebased and updated refs/heads/bugFix.
olegmisialo@MacBook-Air-Oleg mapz % git push

```

Рис 9. Завдання 2.1. Detached HEAD

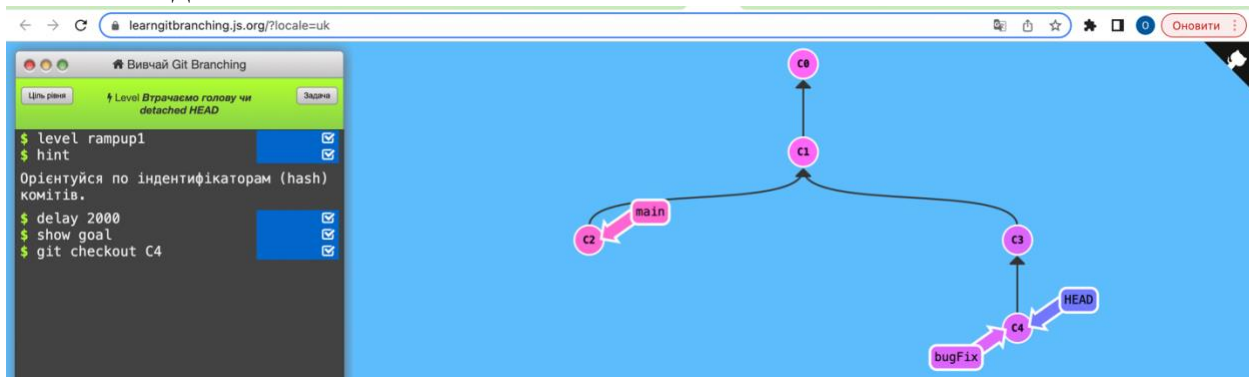


Рис. 10

```

olegmisialo@MacBook-Air-Oleg mapz % git checkout daf806435a62ab82dc16c13bd327f91fb5c784da
Note: switching to 'daf806435a62ab82dc16c13bd327f91fb5c784da'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at daf8064 1:2
olegmisialo@MacBook-Air-Oleg mapz %

```

Рис 11. Завдання 2.2. Відносні посилання

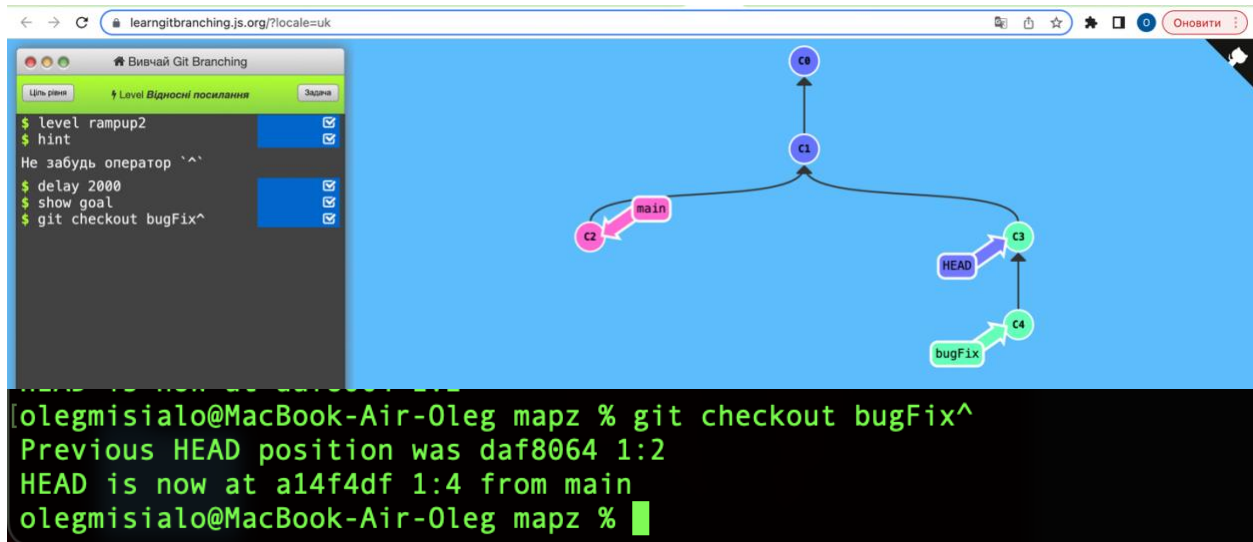


Рис 12. Завдання 2.3. Також відносні посилання

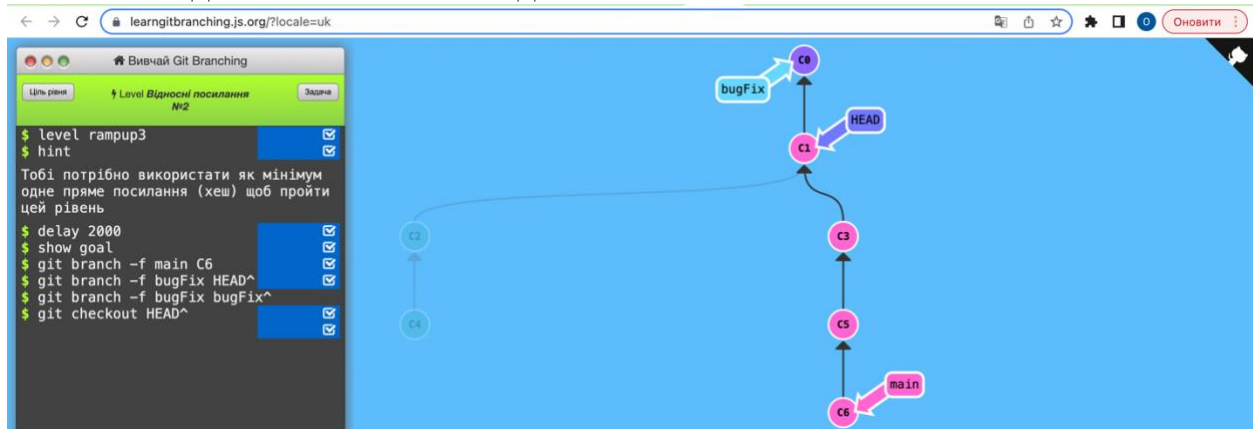


Рис 13.


```

acdcad9 Initial commit
[olegmisialo@MacBook-Air-Oleg mapz % git branch -f bugFix bugFix~3
[olegmisialo@MacBook-Air-Oleg mapz % git checkout a14f4df
Note: switching to 'a14f4df'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at a14f4df 1:4 from main
olegmisialo@MacBook-Air-Oleg mapz % git log --graph --all --oneline --decorat
e
* 896d98c (main) 2:3 main2
* c169f9e 2:3 main
| * a581822 (origin/bugFix) 1:4 from bf
|/
* a14f4df (HEAD) 1:4 from main
* 3ae05ba (origin/main, origin/HEAD, bugFix) added 2 lines
* daf8064 1:2
* 520f069 1:1 com2
* a325a7d 1:1 com1
* bf896cf change name
* ac2d3ab Create folder and file
* 59f8437 Delete .DS_Store
* 474d8eb create folder for lab01
* acdcad9 Initial commit
olegmisialo@MacBook-Air-Oleg mapz %

```

Рис 14. Завдання 2.4. Можливості відміни змін

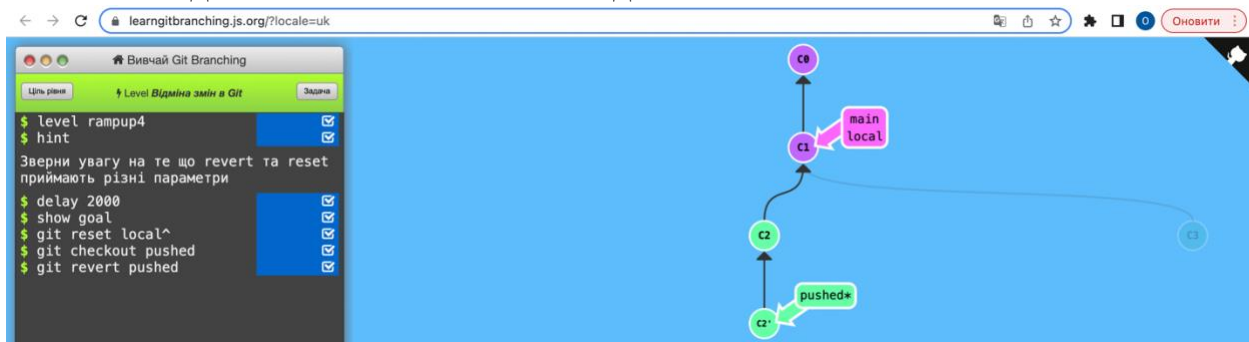


Рис 15.


```

olegmisialo@MacBook-Air-Oleg mapz % git log --graph --all --oneline --decorate
e
* 1341248 (HEAD -> bugFix) 2:4 com1
| * ef5c715 (main) 2_4 main2
| * eafee50 2_4 main1
|/
* cc40fb6 ///
* 1684c66 2:4 2
* a581822 (origin/bugFix) 1:4 from bf
* a14f4df 1:4 from main
* 3ae05ba (origin/main, origin/HEAD) added 2 lines
* daf8064 1:2
* 520f069 1:1 com2
* a325a7d 1:1 com1
* bf896cf change name
* ac2d3ab Create folder and file
* 59f8437 Delete .DS_Store
* 474d8eb create folder for lab01
* acdcad9 Initial commit
olegmisialo@MacBook-Air-Oleg mapz %

```

Рис 16.

```

olegmisialo@MacBook-Air-Oleg mapz % git log --graph --all --oneline --decorate
* 04e149a (bugFix) 2:4 bf2
* dab1a8b 2:4 bF
| * eafee50 (HEAD -> main) 2_4 main1
|/
* cc40fb6 ///
* 1684c66 2:4 2
* a581822 (origin/bugFix) 1:4 from bf
* a14f4df 1:4 from main
* 3ae05ba (origin/main, origin/HEAD) added 2 lines

```

Рис 17. Завдання 3.1. Cherry-pick

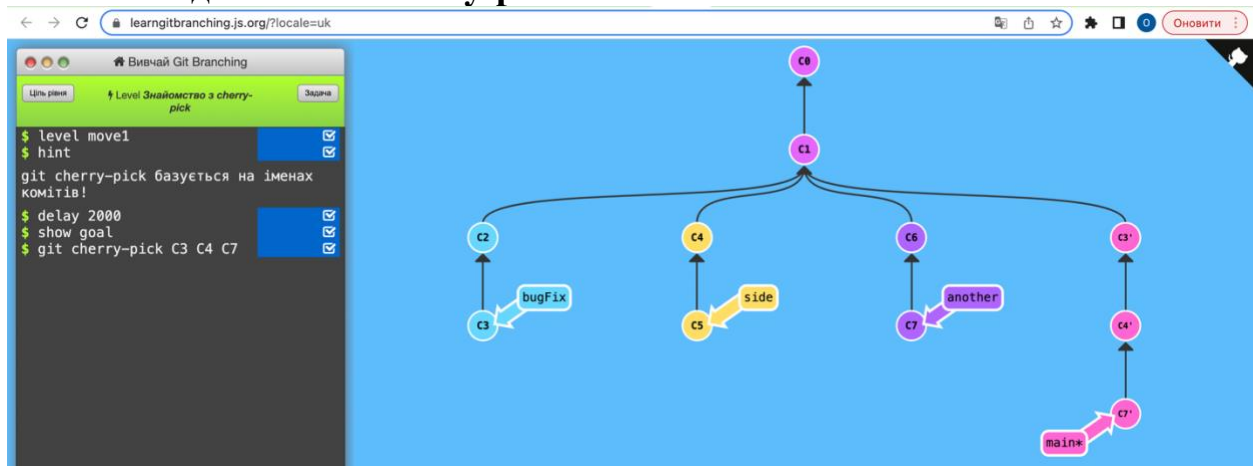


Рис 18.

```
(use "git push" to publish your local commits)
[olegmisialo@MacBook-Air-Oleg mapz % git cherry-pick db22b9a
[main d7cf11e] 3:1
Date: Sun Mar 5 21:44:28 2023 +0200
1 file changed, 1 insertion(+)
[olegmisialo@MacBook-Air-Oleg mapz % git log --graph --all --oneline --decorate
* d7cf11e (HEAD -> main) 3:1
|
| * db22b9a (bugFix) 3:1
|/
|/
```

Рис 19. Завдання 3.2. Інтерактивний rebase

```
$ level move2
$ hint
ти можеш використовувати гілки чи
відносні посилання (HEAD~) щоб
вказувати ціль для rebase

$ delay 2000
$ show goal
$ git rebase HEAD~4
Гілку вже оновлено
$ git rebase -i HEAD~4
```

```
olegmisialo@MacBook-Air-Oleg mapz % git log --graph --all --oneline --decorate
* c04067c (HEAD -> main) 3:2 3
* 6af3de2 3:2 2
* dc61a34 3:2 1
```

```
olegmisialo@MacBook-Air-Oleg mapz % git log --graph --all --oneline --decorate
* dc61a34 (HEAD -> main) 3:2 1
```

Рис 20. Завдання 4.1. Cherry-pick

```
$ level mixed1
$ hint
Не забувай, що інтерактивний rebase та
cherry-pick -- це твої друзі!

$ delay 2000
$ show goal
$ git checkout main
$ git cherry-pick C4
```

Рис 21.

```

olegmisialo@MacBook-Air-Oleg mapz % git log
* 97b40a8 (HEAD -> bugFix) 4:1
* db22b9a 3:1
|
| * dc61a34 (main) 3:2 1
| * d7cf11e (origin/main, origin/HEAD) 3:1
|/

```

Рис 22.

```

olegmisialo@MacBook-Air-Oleg mapz % git log --graph --all --oneline --decorate
* a7dec6b (HEAD -> main) 3:1 2
* dc61a34 3:2 1
* d7cf11e (origin/main, origin/HEAD) 3:1
|
| * 97b40a8 (bugFix) 4:1
| * db22b9a 3:1
|/

```

Рис 23. Завдання 4.2. Branch -f

The screenshot shows a web browser window with the URL `learn git branching.js.org/?locale=uk`. The browser displays a diagram of a branching strategy. The diagram shows a central node `C1` with three branches: `newImage` (purple), `C2` (blue), and `C3` (red). The `newImage` branch has a sub-branch `C2`. The `C3` branch has a sub-branch `C3`. The `main` branch is labeled `caption*`. The terminal window shows the execution of `git log --graph --all --oneline --decorate` twice. The first execution shows the state of the repository before the branch force push. The second execution shows the state of the repository after the branch force push, where the `main` branch is now `3511433` and the `newImage` branch is `a43c9d9`.

```

olegmisialo@MacBook-Air-Oleg mapz % git log --graph --all --oneline --decorate
* a7dec6b (HEAD -> main) 3:1 2
* dc61a34 3:2 1
* d7cf11e (origin/main, origin/HEAD) 3:1
|
| * 97b40a8 (bugFix) 4:1
| * db22b9a 3:1
|/

olegmisialo@MacBook-Air-Oleg mapz % git log --graph --all --oneline --decorate
* 3511433 (HEAD -> main) 3:2 1
* a43c9d9 3:1 2
* d7cf11e (origin/main, origin/HEAD) 3:1
|
| * 97b40a8 (bugFix) 4:1
| * db22b9a 3:1
|/

```

Рис 24. Завдання 4.3.Жанглювання комітами

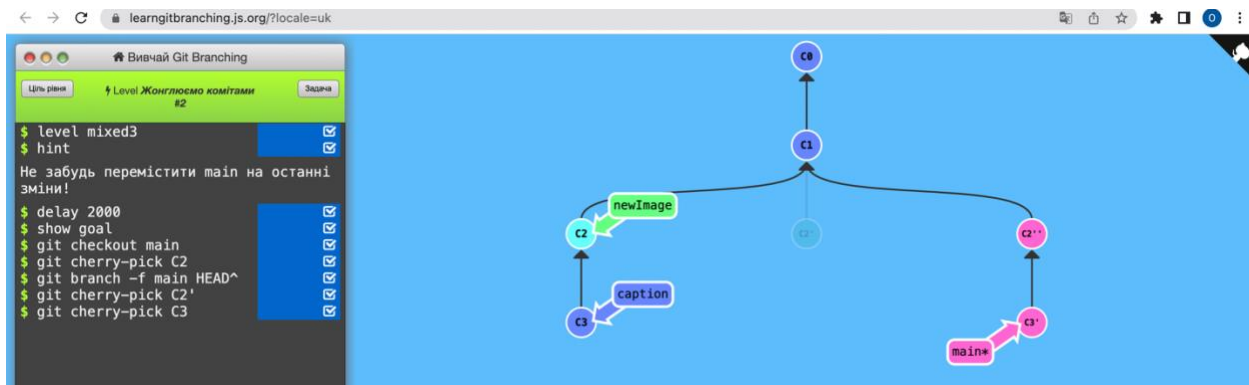


Рис 25.

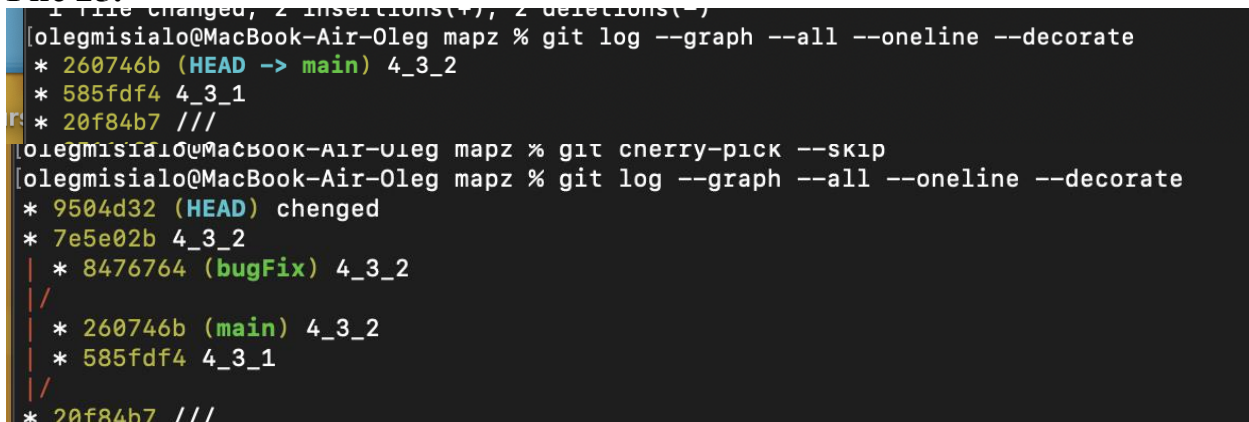


Рис 26. Завдання 4.4. Теги.

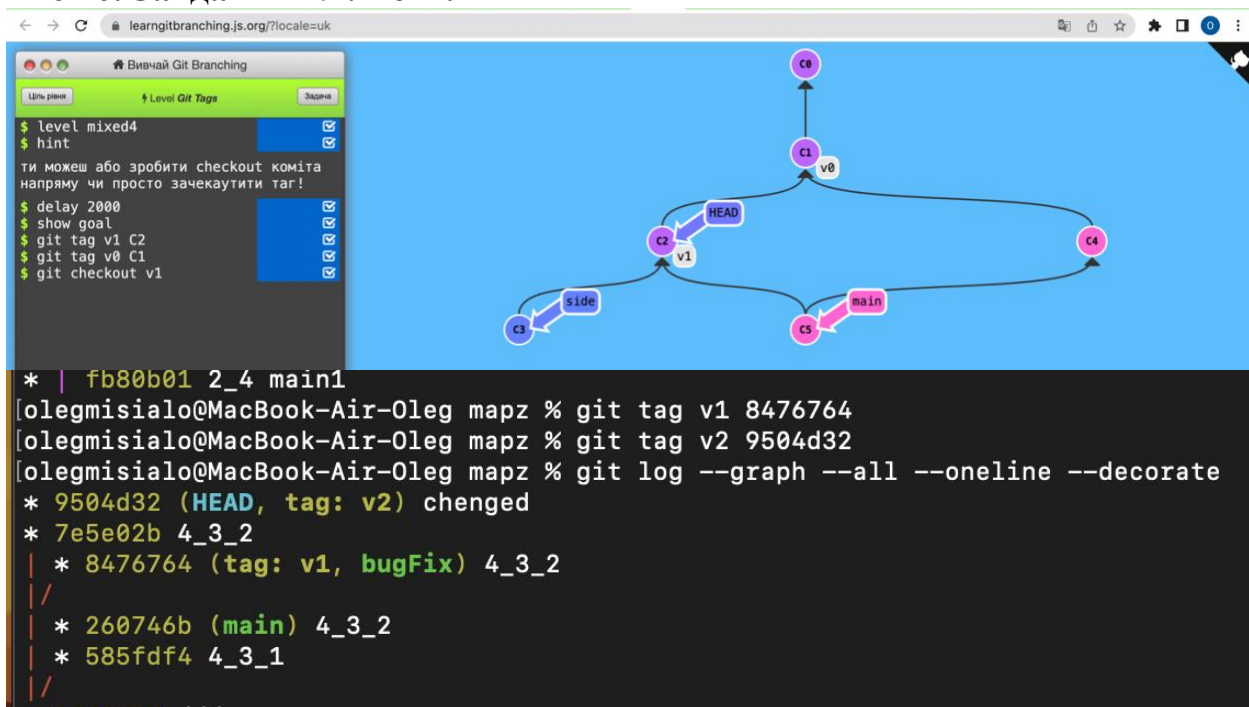


Рис 27. Завдання 4.5. Інформація про коміти

The screenshot displays the 'learnitbranching.js.org' website. On the left, a sidebar titled 'Вивчай Git Branching' contains a checklist of tasks with checkboxes. The main area features a Git branching model diagram with branches labeled 'main', 'side', 'bugFix*', 'v0', 'v1', 'v2', 'C1', 'C2', 'C3', 'C4', 'C5', and 'C6'. Below the diagram, a terminal window shows the following commands and output:

```
v2
[olegmisialo@MacBook-Air-Oleg mapz % git describe --tags HEAD
v2-1-g12e72f4
[olegmisialo@MacBook-Air-Oleg mapz % git log --graph --all --oneline --decorate
* 12e72f4 (HEAD) describe
* 9504d32 (tag: v2) changed
* 7e5e02b 4_3_2
* 8476764 (tag: v1, bugFix) 4_3_2
//
* 260746b (main) 4_3_2
* 585fdf4 4_3_1
//
```

Рис 28. Скрін меню сайту

The screenshot shows the 'learnitbranching.js.org' website with a progress menu. The menu is titled 'Вступ' and lists several topics with progress indicators (green checkmarks):

- Гарно підібране введення в основні команди git
- Знайомство з комітами в Git
- Ідемо далі
- Наступна порція абсолютної git-диалекти. Сподіваюсь, ви згодуєтесь
- Втрачаємо голову чи detached HEAD
- Переміщуємо роботу туди-сюди
- Не соромимось змінювати історію
- Знайомство з cherry-pick
- Всяке
- Різні прийоми роботи з Git, хитрощі та поради
- Вибираємо всього один коміт
- Досвідчений рівень
- Для хоробрих

Висновок

У цій лабораторній роботі було вивчено роботу з інструментом Git та хостингом репозиторіїв Gitlab. Були вивчені такі поширені команди, як commit, push, checkout, branch, tag, rebase, merge, describe тощо. Також, було виконано 4 розділи завдань на веб-сайті <https://learngitbranching.js.org>. Під час виконання цієї лабораторної роботи було характерним відчуття апатії. Помітно, що це відчуття корелювало із кількістю знімків екрану, що було зроблено в процесі виконання роботи.