Лабораторна робота №3

Юдін Станіслав 451 група

Тема: Клонування репозиторію засобами Git for Windows.

Мета: Отримати навички створення локальної робочої копії репозиторію за допомогою Git for Windows.

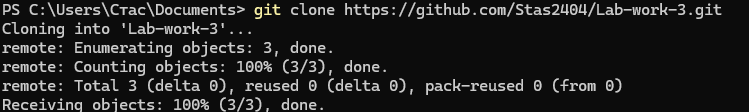
Хід роботи

Перевірити глобальні налаштування Git for Windows

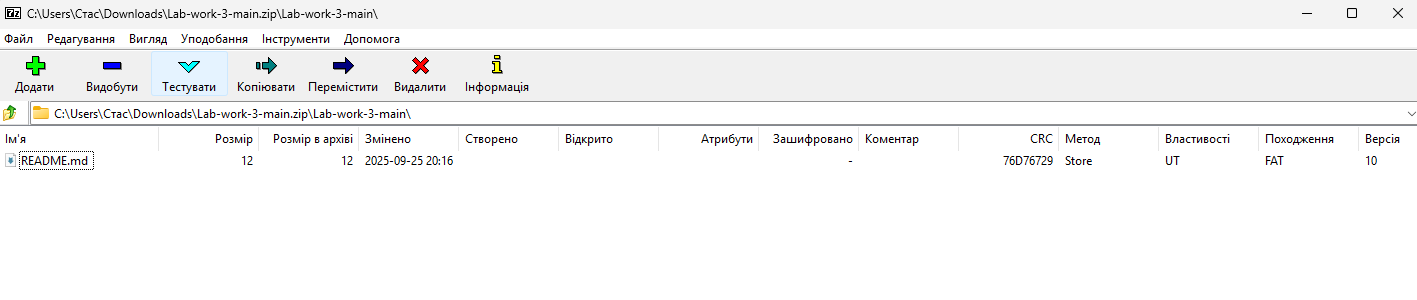


У випадку невідповідності вашим даним налаштувати глобальну конфігурацію Git for Windows

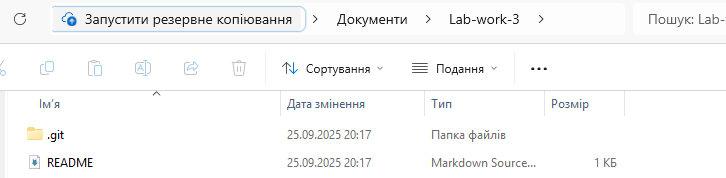
Клонувати створений раніше репозиторій як локальну робочу копію



Завантажити раніше створений репозиторій з Github в архіві. Дослідити зміст клонованих репозиторіїв, розархівованих репозиторіїв зі змістом на сервісі Github



В архіві отримаємо повну копію файлів проєкту, але, немає прихованої папки .git. Також не можемо робити git pull, git push, коміти, бо це просто файли без історії.



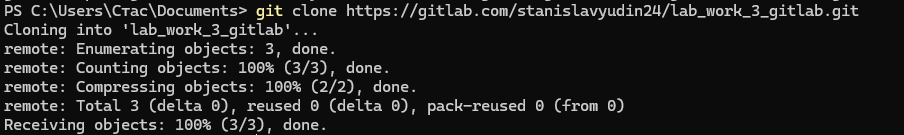
При клонуванні через Git, створюється папка з усіма файлами проєкту та папкою .git, де зберігається вся історія змін, гілки, коміти. Отримаємо повноцінну локальну копію репозиторію, з якою можна працювати: комітити, пушити, створювати гілки.

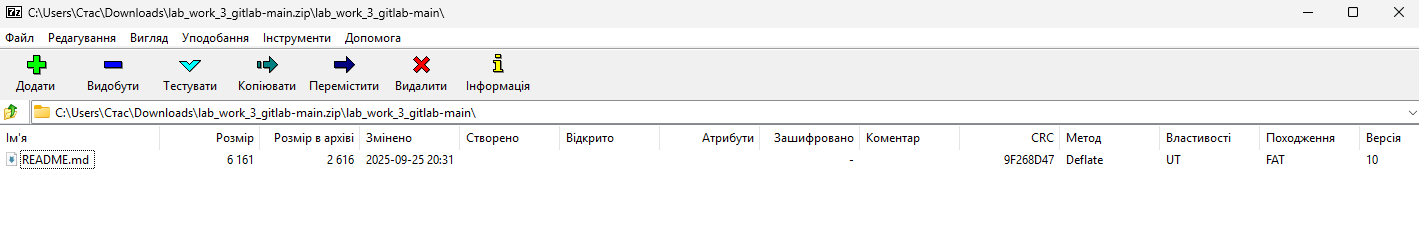
Зробити висновок про відмінності у змісті клонованого репозиторію, завантаженого та даних, що відображає веб-інтерфейс

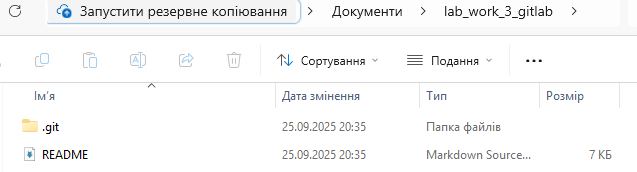
* Клонований репозиторій містить усі файли проєкту та приховану папку .git, у якій зберігається повна історія змін, коміти, гілки та можливість синхронізації з GitHub. Це повноцінна робоча копія, що дозволяє виконувати команди Git (commit, push, pull).
* Завантажений репозиторій у вигляді архіву (ZIP) містить лише знімок файлів на момент завантаження, без історії змін та без папки .git. Його можна використовувати для ознайомлення з проєктом чи запуску коду, але не для подальшої роботи з системою контролю версій.
* Веб-інтерфейс GitHub відображає як поточний стан файлів (аналог завантаженого архіву), так і додаткові дані: історію комітів, гілки, пул-реквести, issues тощо. Це дозволяє досліджувати як вміст, так і повний процес розробки.

Повторити пункти 3 - 5 пункти для сервісів Bitbucket та Gitlab з урахуванням змін глобальних налаштувань, якщо це потрібно

GitLab:

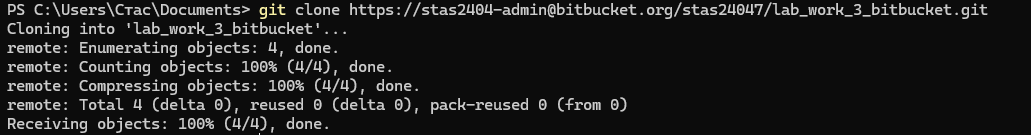


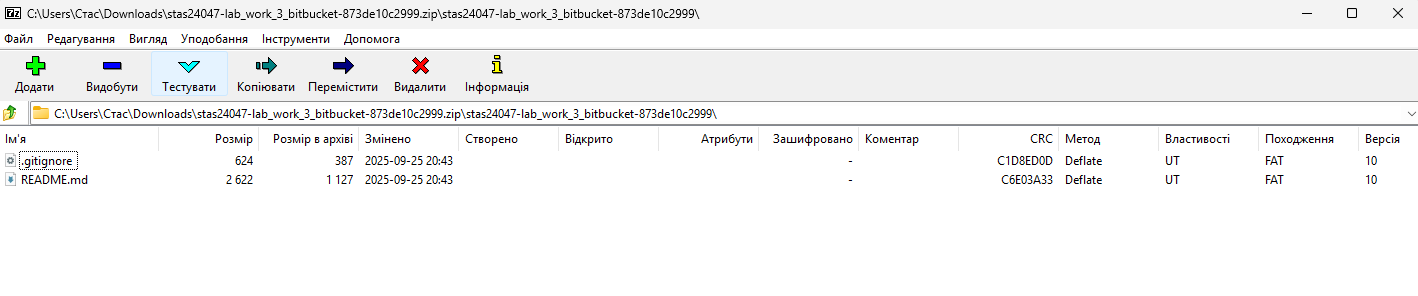


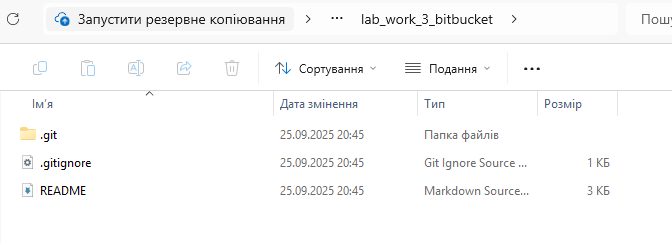


Висновок: маємо ті ж самі файли та папки, як і при клонуванні та архівуванні репозиторію у GitHub.

Bitbucket:



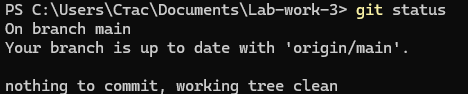


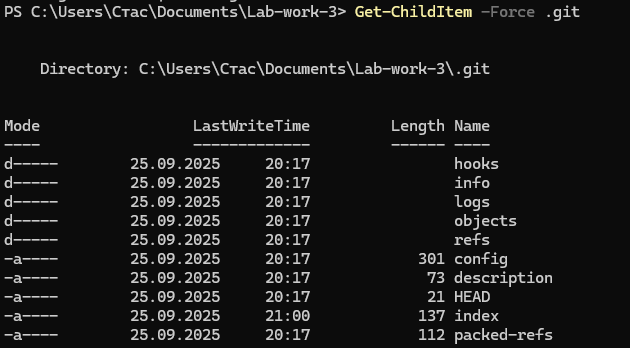


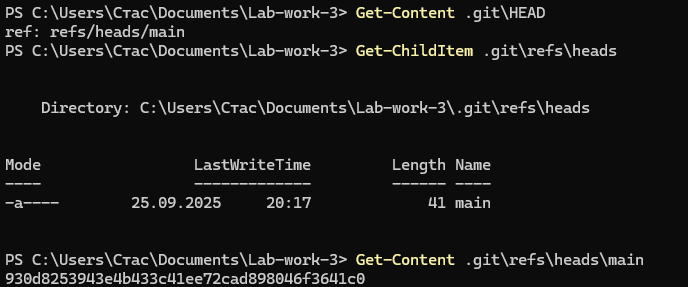
Висновок: маємо ідентичну ситуацію до GitHub та GitLab. .zip – містить тільки поточний стан коду (snapshot). git clone – містить ще й історію, гілки, коміти.

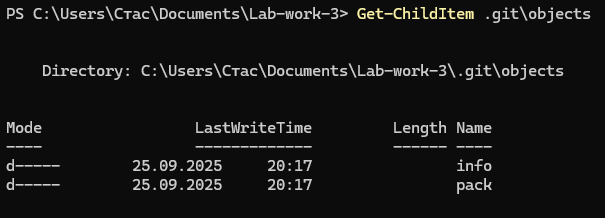
Дослідити зміст каталогу .git за методом описаним в посиланні [2]

Зробити висновок про зміст вашого каталогу .git та додати цю інформацію у звіт









Каталог .git у проекті Lab-work-3 містить всі внутрішні дані Git-репозиторію, що дозволяє відстежувати версії файлів та історію змін. Структура каталогу виглядає так:

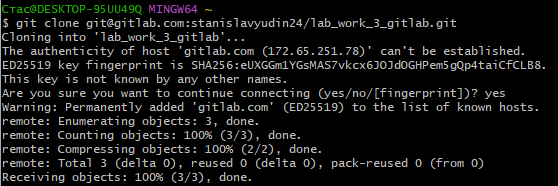
* hooks - Скрипти, які можна запускати на певні події Git (наприклад, pre-commit)
* info - Інформаційні файли репозиторію, наприклад, exclude для ігнорування файлів
* logs - Логи змін посилань на гілки (HEAD) та коміти
* objects - Основні об’єкти Git (коміти, дерева, блоби), збережені за хешами SHA-1
* refs - Посилання на гілки (heads) та теги (tags)
* config - Локальні налаштування репозиторію
* description - Короткий опис репозиторію (для bare репозиторіїв)
* HEAD - Поточне положення в репозиторії (вказує на гілку main)
* index - Список змінених файлів для коміту (staging area)
* packed-refs - Файл для оптимізованого зберігання посилань на коміти та теги

Деталі стану репозиторію:

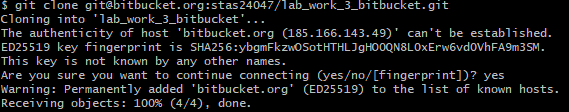
* Поточна гілка: main (.git/HEAD вказує на refs/heads/main)
* Хеш останнього коміту на гілці main: 930d8253943e4b433c41ee72cad898046f3641c0
* Папка objects містить підпапки info та pack, де зберігаються об’єкти Git у стисненому вигляді.

З урахуванням навичок, отриманих під час виконання лабораторної роботи №2, виконати клонування репозиторію з сервісів Bitbucket та Gitlab з використанням SSH-тунелю.

GitLab:



Bitbucket:



Контрольні питання

1. Яка команда відповідає за клонування віддаленого репозиторію у локальну робочу копію?

git clone <URL>

1. Що потрібно зробити, для завантаження репозиторію, якщо на комп'ютері відсутній Git для сервісу Github?

Завантажити і встановити Git із офіційного сайту. Або скористатися веб-інтерфейсом GitHub: відкрити репозиторій → кнопка Code → Download ZIP, після чого розархівувати.

1. Що потрібно зробити, для завантаження репозиторію, якщо на комп'ютері відсутній Git для сервісу Bitbucket?

Завантажити і встановити Git із офіційного сайту. Або у веб-інтерфейсі Bitbucket відкрити потрібний репозиторій → меню Download repository → вибрати ZIP-архів і завантажити його.

1. Що таке Github Desktop, яке його призначення?

GitHub Desktop — це офіційна графічна оболонка Git від GitHub.

Вона дозволяє працювати з репозиторіями без командного рядка: клонувати, створювати коміти, робити push/pull, створювати гілки та виконувати merge у зручному віконному інтерфейсі.

1. Які візуальні оболонки Git ви знаєте?

* GitHub Desktop (для GitHub).
* Sourcetree (від Atlassian, працює з GitHub, GitLab, Bitbucket).
* GitKraken (кросплатформена графічна оболонка).
* TortoiseGit (для Windows, інтегрується в провідник).

Висновок: під час виконання л.р., я отримав навички створення локальної робочої копії репозиторію за допомогою Git for Windows.