**Міністерство освіти і науки України**

**Національний університет «Запорізька Політехніка»**

Кафедра програмних засобів

**ЗВІТ**

з лабораторної роботи № 3

з дисципліни «Soft skills, групова динаміка та комунікації» на тему:

«ГРУПОВА РОБОТА З СИСТЕМАМИ КЕРУВАННЯ ВЕРСІЯМИ»

**Виконав:**

студент групи КНТ-122 О. А. Онищенко

**Прийняли:**

асистент: Д. А. Каврін

2023

1 ГРУПОВА РОБОТА З СИСТЕМАМИ КЕРУВАННЯ ВЕРСІЯМИ

1.1 Мета роботи

2.1.1 Вивчити основні можливості систем керування версіями на прикладі системи Subversion.

2.1.2 Вивчити основні можливості системи керування версіями Git та порівняти їх з можливостями Subversion.

2.1.3 Навчитися використовувати можливості систем керування версіями Subversion та Git для групової роботи.

1.2 Завдання роботи

1.2.1 Ознайомитися з теоретичними відомостями, необхідними для виконання роботи.

1.2.2 Ознайомитися з основними можливостями систем керування версіями Subversion та Git.

1.2.3 Використовуючи систему керування версіями Subversion (для пунктів 1.2.3-1.2.15), створити каталог проекту та базову структуру репозиторію.

1.2.4 Створити файли проекту у каталозі. Файли проекту можуть містити код проекту або бути текстовими файлами.

1.2.5 Відредагувати файли проекту і зберегти версії.

1.2.6 Створити нову гілку та переключити робочу копію на неї.

1.2.7 Відредагувати файли проекту.

1.2.8 Переключити робочу копію на піддиректорію trunk.

1.2.9 Злити зміни між гілками.

1.2.10 Вилучити гілку.

1.2.11 Додати до репозиторію проект на основі даних одного з проектів з відкритим вихідним кодом.

1.2.12 В якості такого проекту можна використати, наприклад, Notepad++ (svn://svn.code.sf.net/p/notepad-plus/code/trunk notepad-plus-code) або будь-який власний.

1.2.13 Внести зміни в файли проекту.

1.2.14 Відправити зміни до репозиторію та пояснити отримані результати.

1.2.15 Пункти завдань 1.2.6–1.2.14 виконати за допомогою клієнту та командного рядка.

1.2.16 Використовуючи систему керування версіями Git (для пунктів 1.2.16-1.2.25), створити репозиторій на основі одного з раніше розроблених проектів.

1.2.17 Заборонити автоматичне додання в репозиторій файлів з розширенням \*.exe.

1.2.18 Внести зміни в текст проекту та зафіксувати їх, для підпису використовуючи власні дані.

1.2.19 Переглянути різницю між новою версією проекту та початковою.

1.2.20 Створити власне віддалене сховище, використовуючи сервіс GitHub.

1.2.21 Налаштувати локальне сховище для синхронізації з віддаленим та відправити локальну версію на сервер.

1.2.22 Переглянути історію проекту та сторінку проекту через web-інтерфейс.

1.2.23 Внести зміни в текст проекту та зафіксувати їх.

1.2.24 Узгодити локальну версію репозиторію з сервером.

1.2.25 Пункти завдань 1.2.17–1.2.25 виконати за допомогою графічної оболонки та командного рядка.

1.2.26 Оформити звіт з роботи.

1.2.27 Відповісти на контрольні питання.

1.3 Короткі теоретичні відомості

Git та Subversion - дві найпопулярніші системи контролю версій, що використовуються при розробці програмного забезпечення. Ось короткий огляд того, як вони працюють:

Git: Git - це розподілена система контролю версій, яка дозволяє декільком розробникам одночасно працювати над однією і тією ж кодовою базою. Кожен розробник має власну копію сховища, в яку він може вносити зміни незалежно. Зміни потім об'єднуються шляхом порівняння та об'єднання змін, зроблених кожним розробником. Git використовує модель розгалуження, де кожна гілка представляє окрему версію кодової бази. Розробники можуть створювати та об'єднувати гілки за потреби для роботи над різними функціями або виправленнями. Git відомий своєю швидкістю, гнучкістю та потужним інтерфейсом командного рядка.

Subversion: Subversion - це централізована система контролю версій, яка використовує модель клієнт-сервер. Всі зміни до кодової бази вносяться до центрального репозиторію, який розміщується на сервері. Розробники можуть отримати копію сховища на своїй локальній машині, внести зміни, а потім зафіксувати ці зміни назад до центрального сховища. Subversion використовує структуру "стовбур-гілки-теги" (TBT), де основна лінія розробки (стовбур) відокремлена від відгалужень (гілок) і версій (тегів) релізів. Subversion відомий своєю простотою, легкістю у використанні та сумісністю з іншими інструментами.

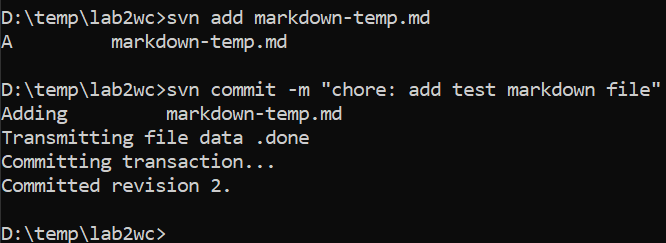
І Git, і Subversion надають потужні можливості контролю версій, які можуть допомогти командам ефективно співпрацювати над проектами з розробки програмного забезпечення. Вибір між Git та Subversion залежить від ваших конкретних потреб, уподобань та вимог проекту. Git часто обирають за швидкість, гнучкість та модель розгалуження, тоді як Subversion - за простоту, легкість у використанні та централізований підхід.

1.4 Результати роботи

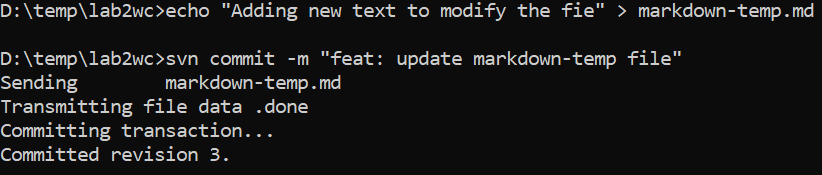
2.3.3 Використовуючи систему керування версіями Subversion (для пунктів 2.3.3-2.3.15), створити каталог проекту та базову структуру репозиторію.



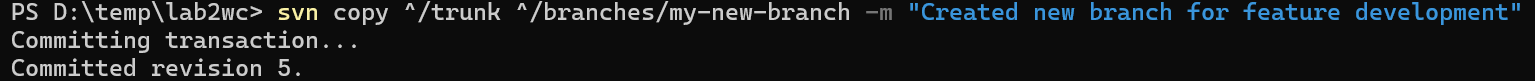
2.3.4 Створити файли проекту у каталозі. Файли проекту можуть містити код проекту або бути текстовими файлами.

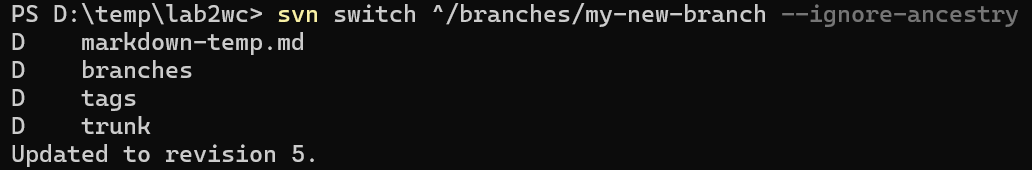


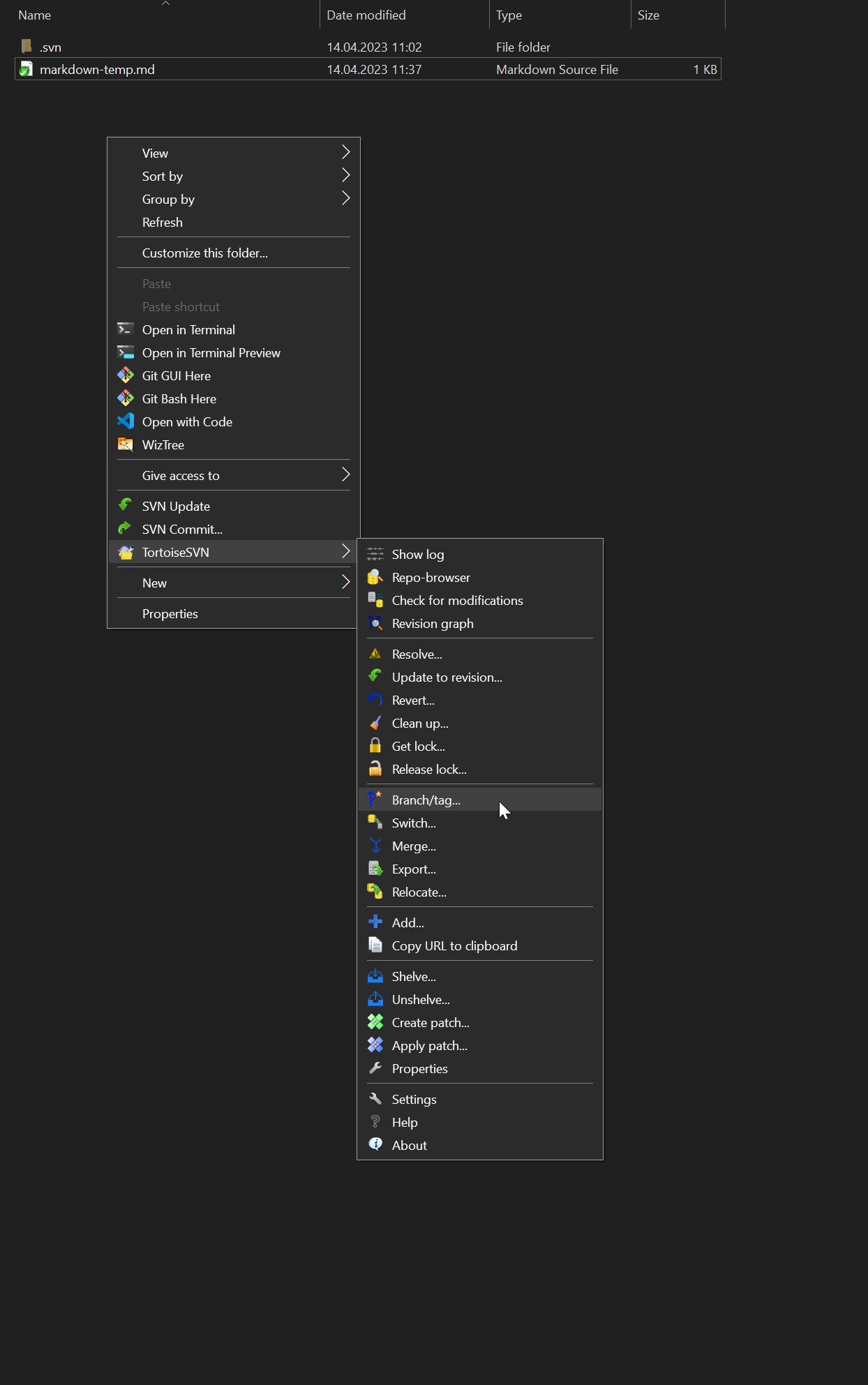
2.3.5 Відредагувати файли проекту і зберегти версії.

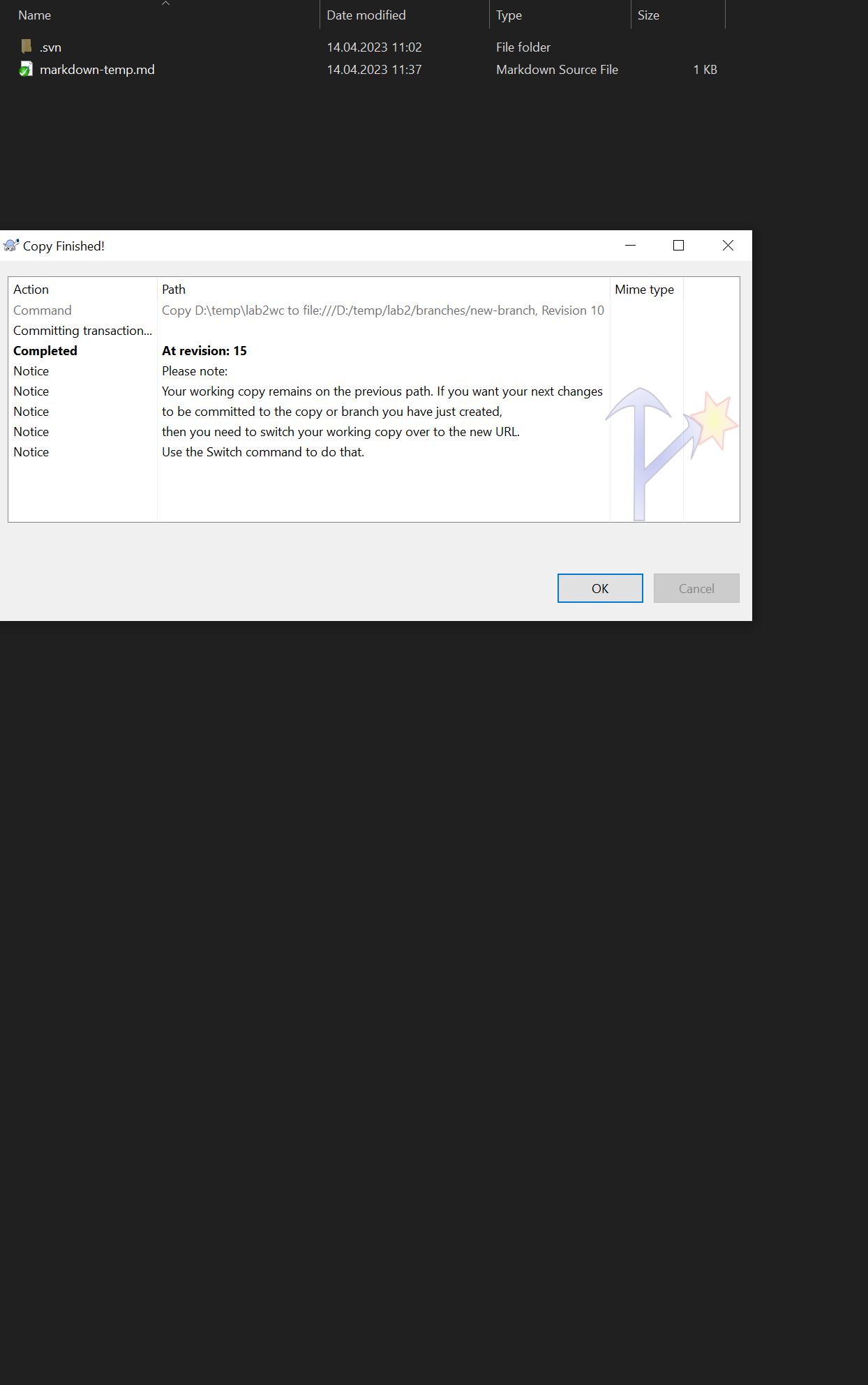


2.3.6 Створити нову гілку та переключити робочу копію на неї.

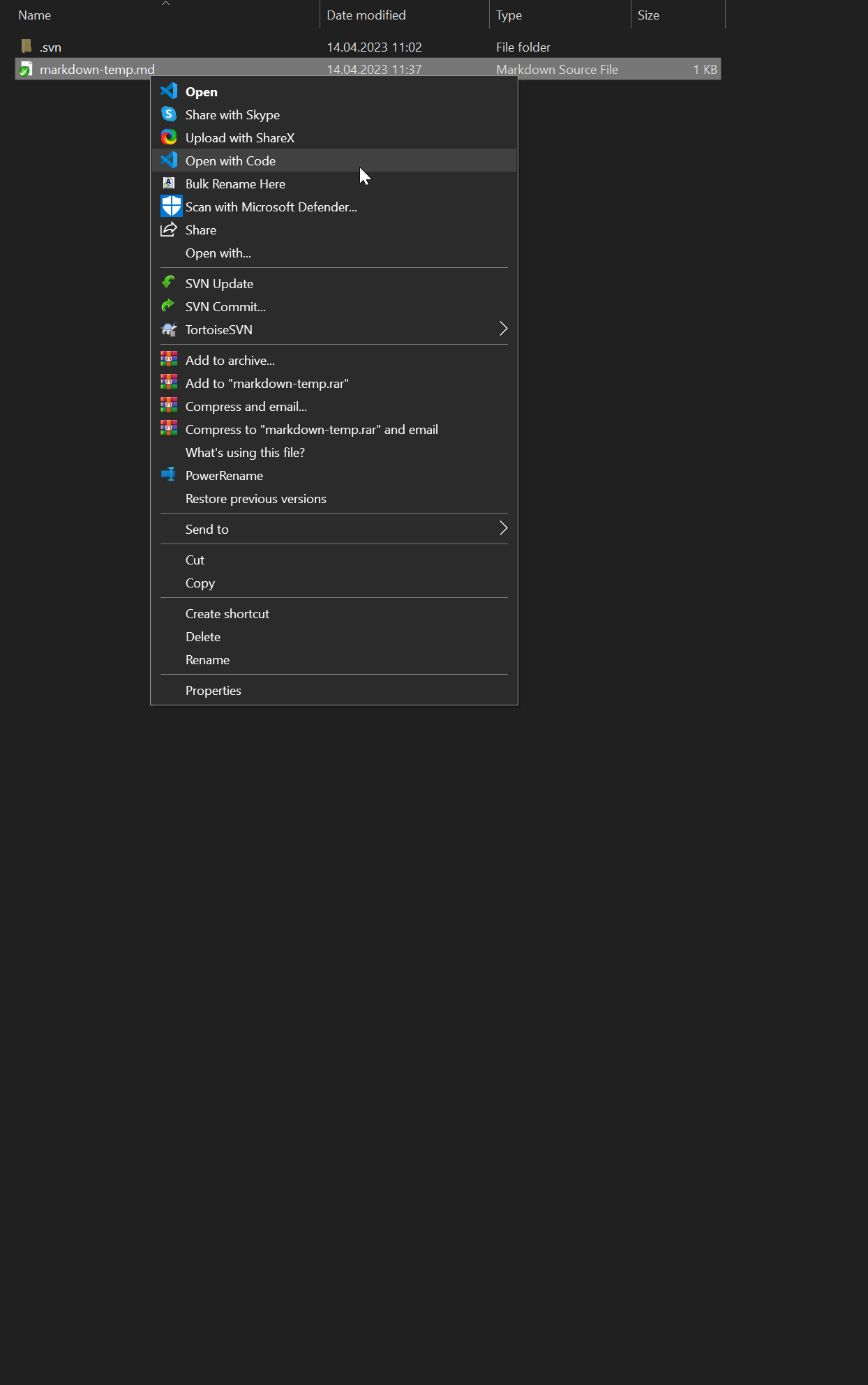


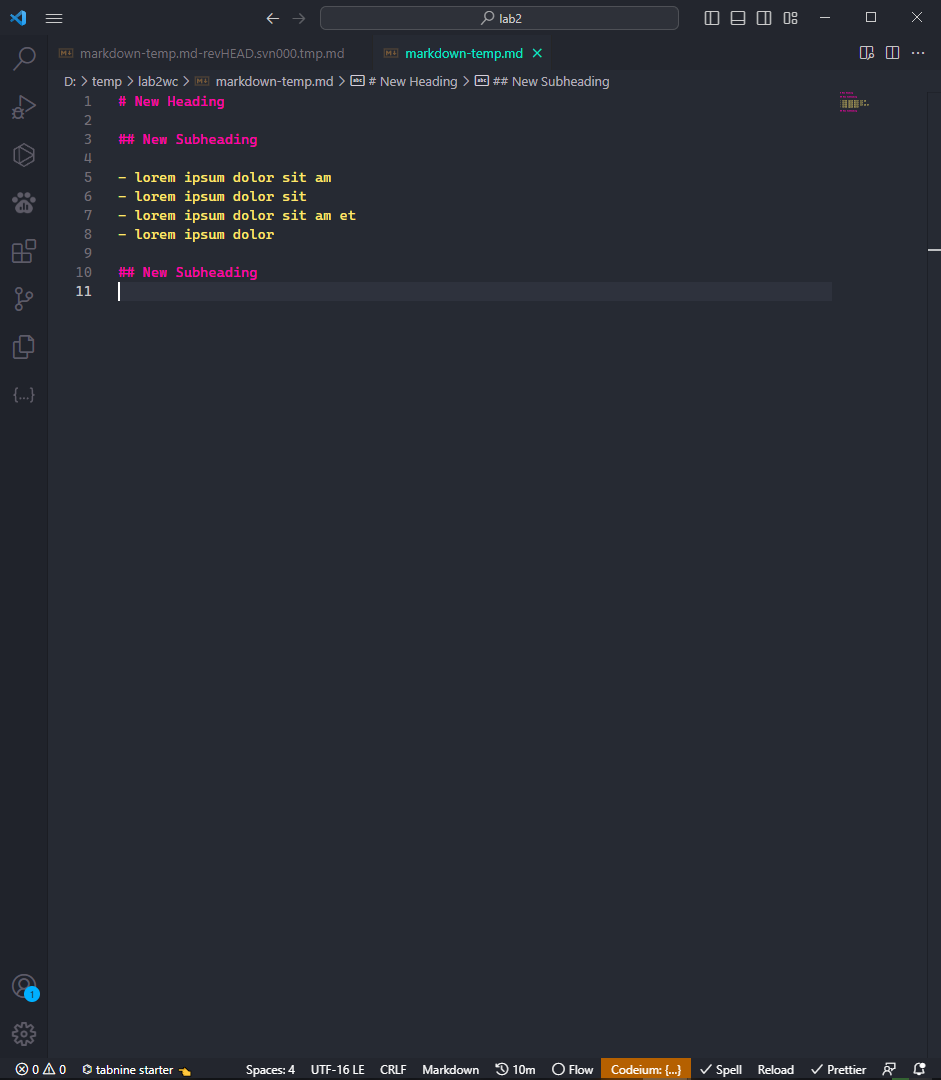


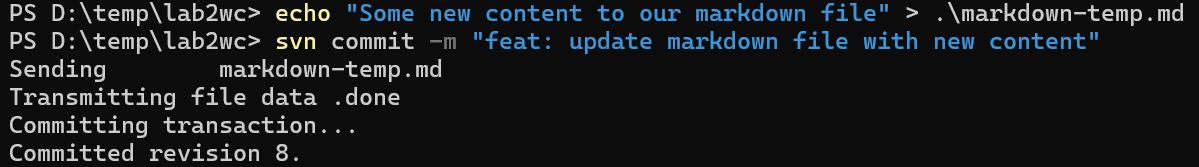




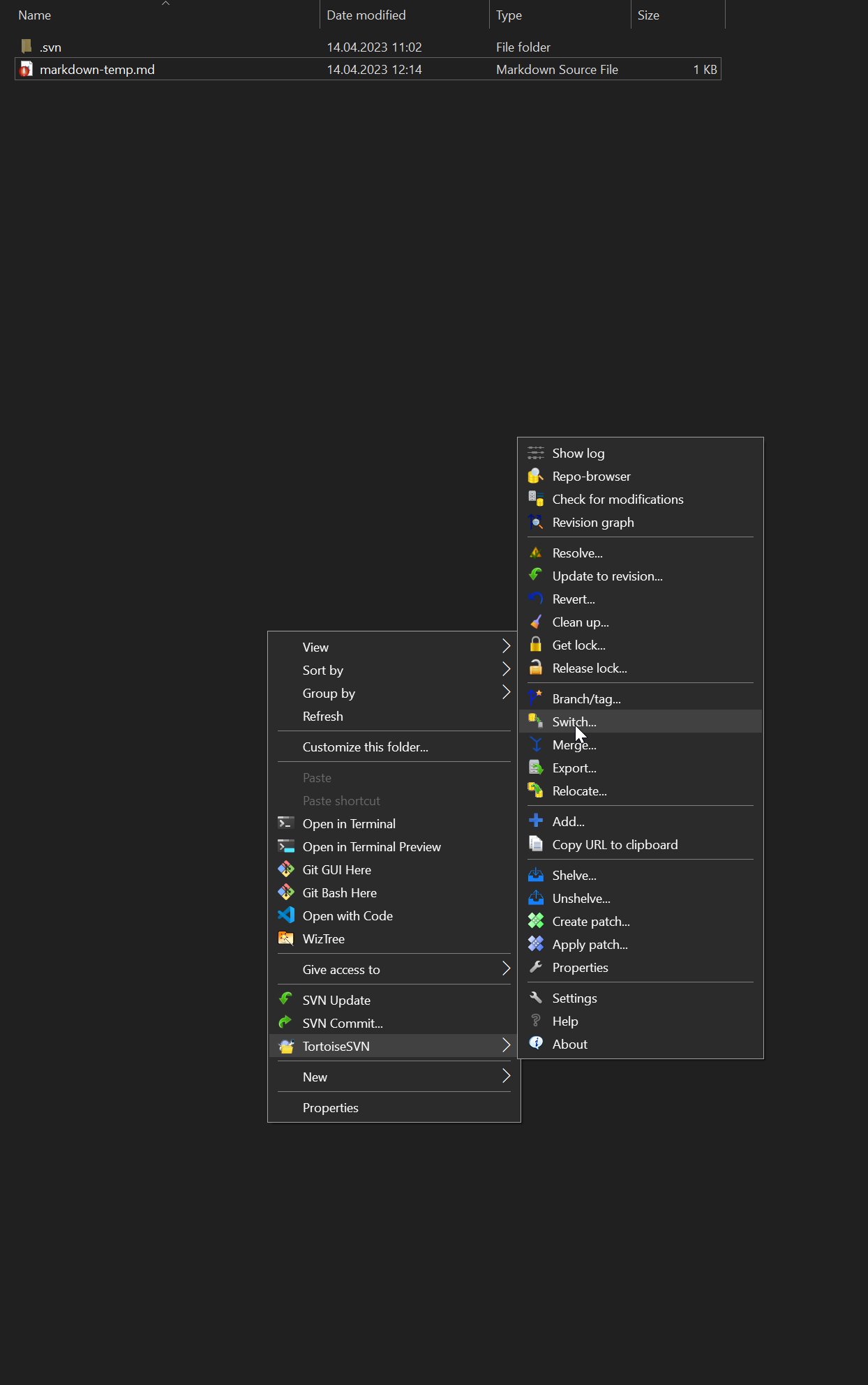
2.3.7 Відредагувати файли проекту.

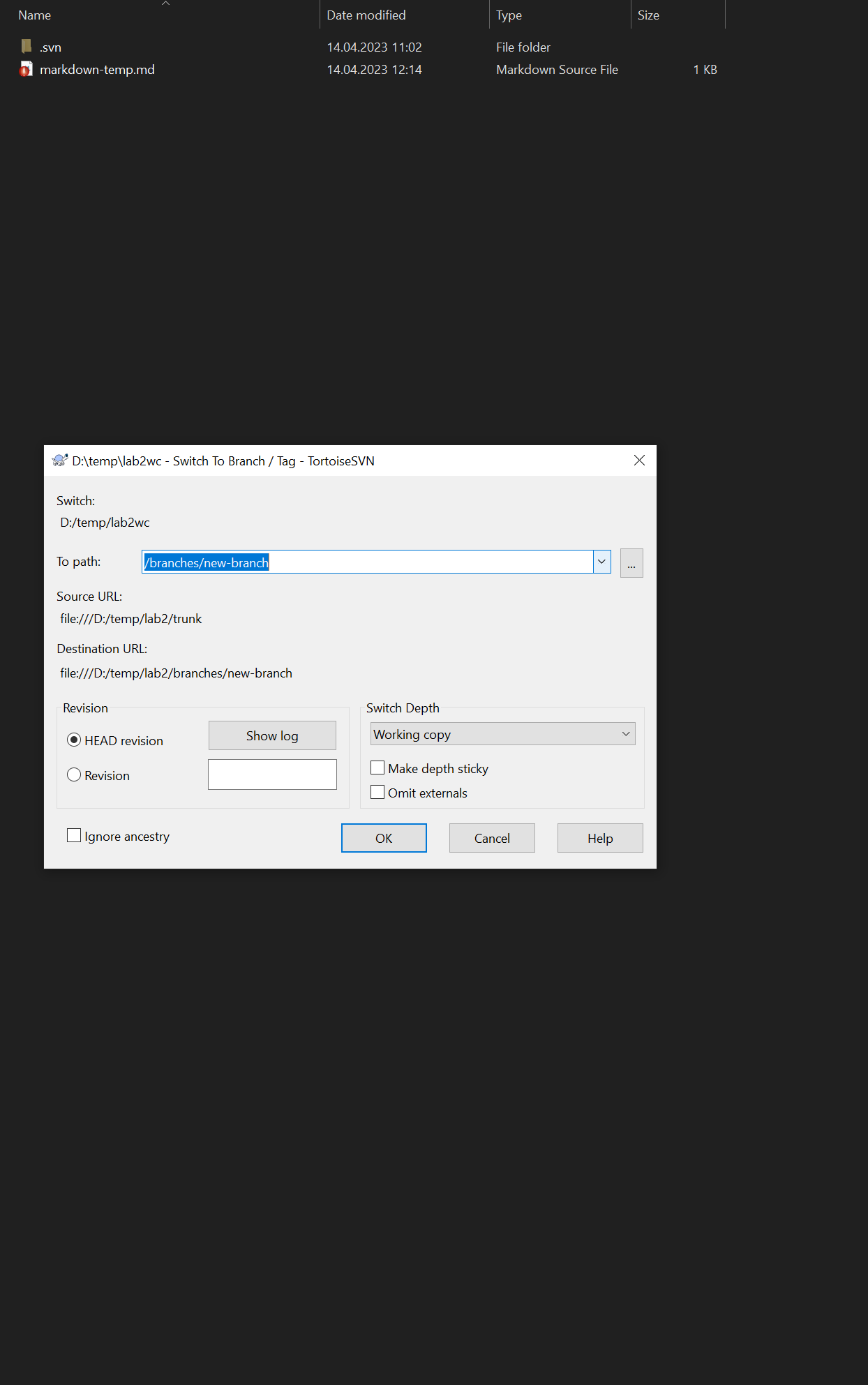


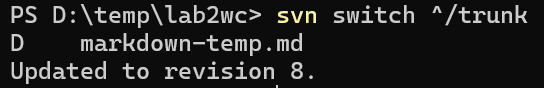




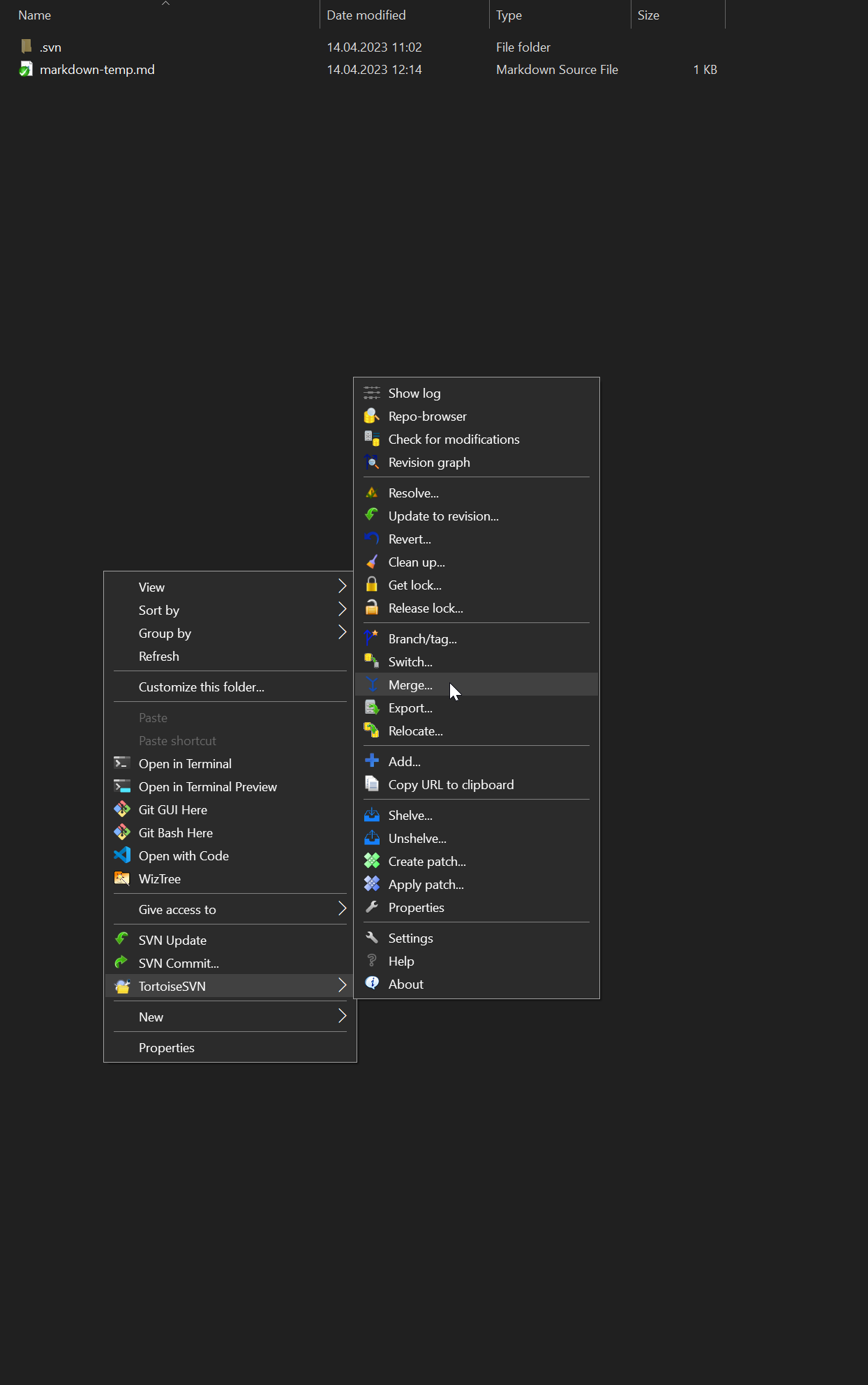
2.3.8 Переключити робочу копію на піддиректорію trunk.

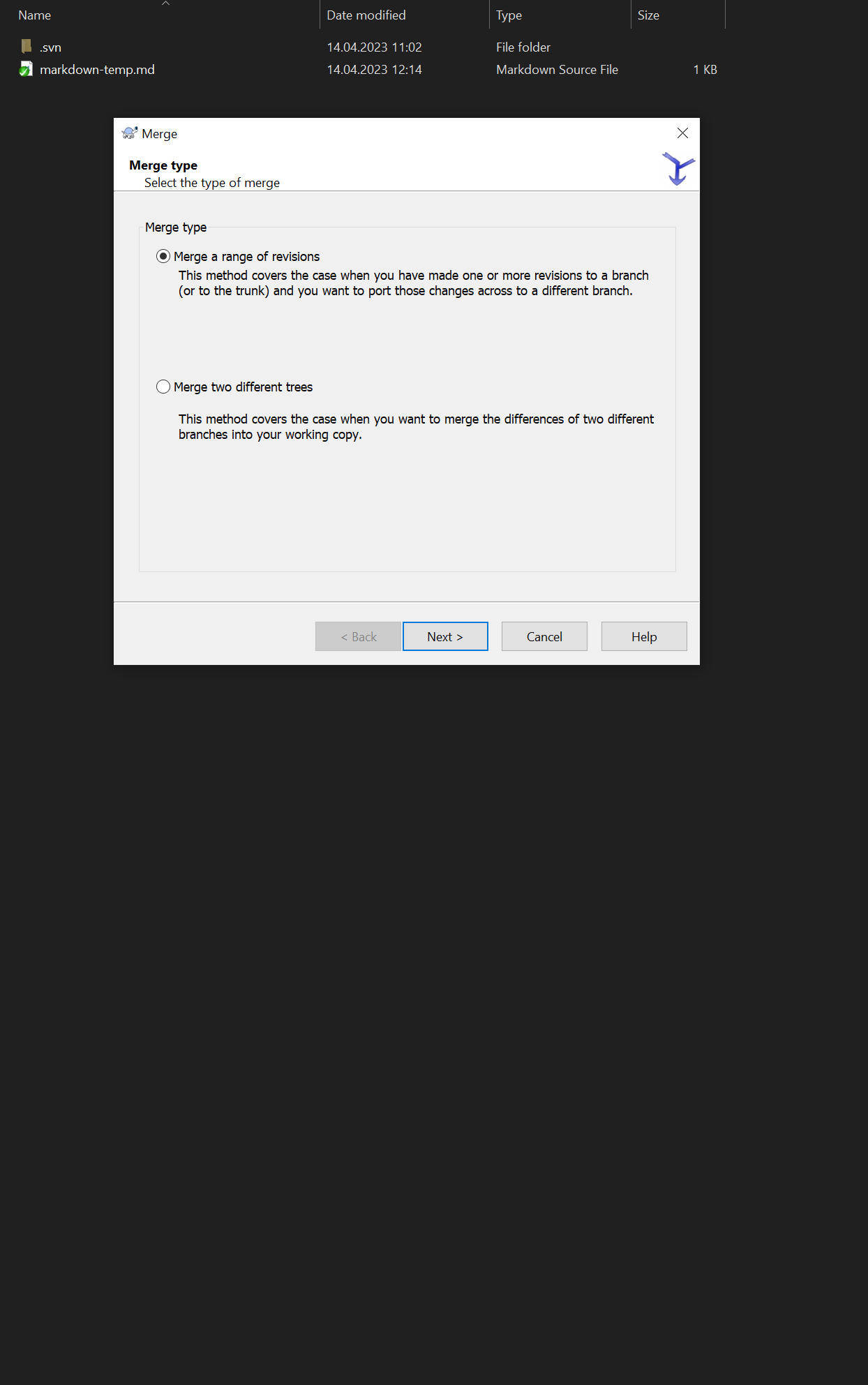


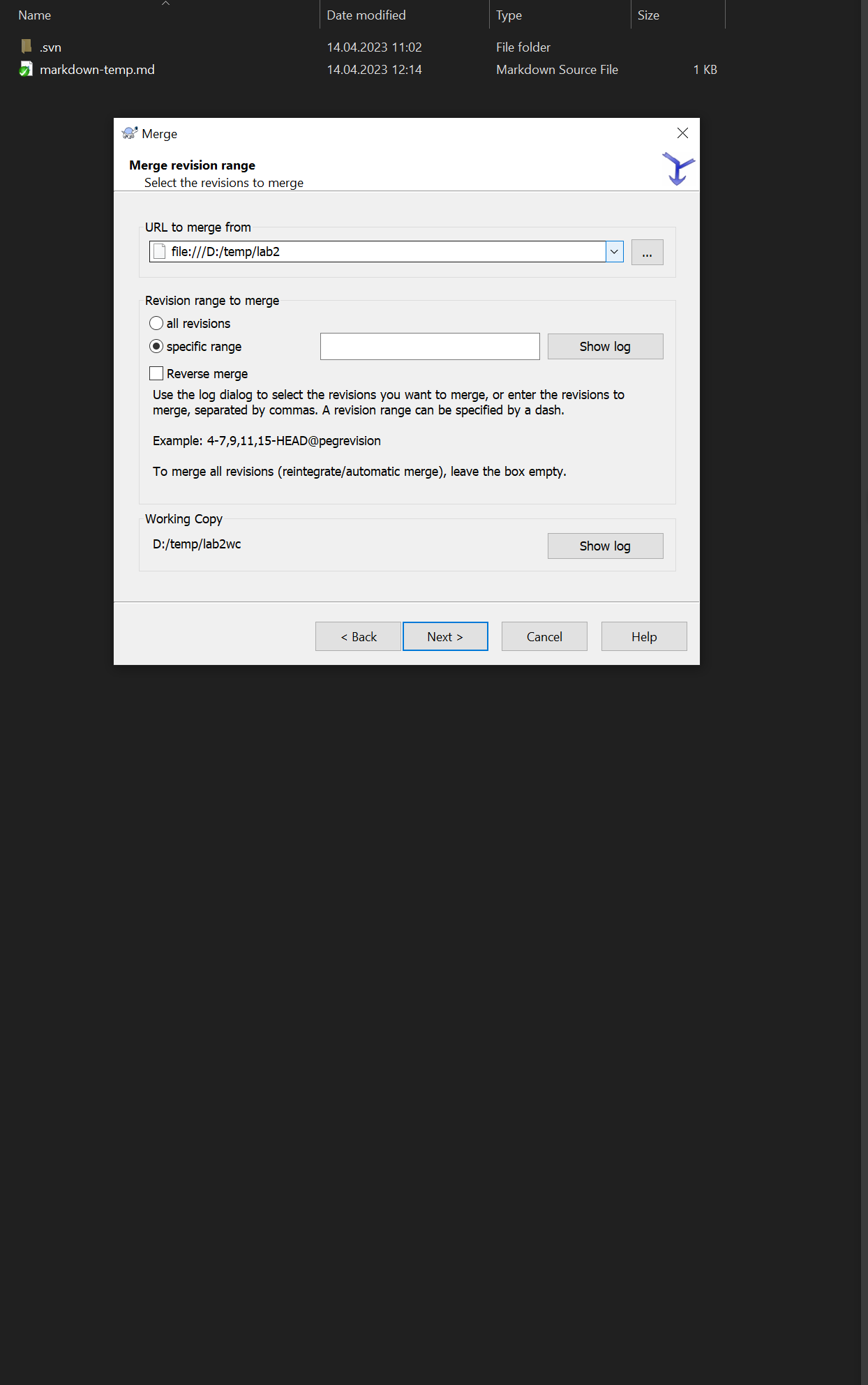


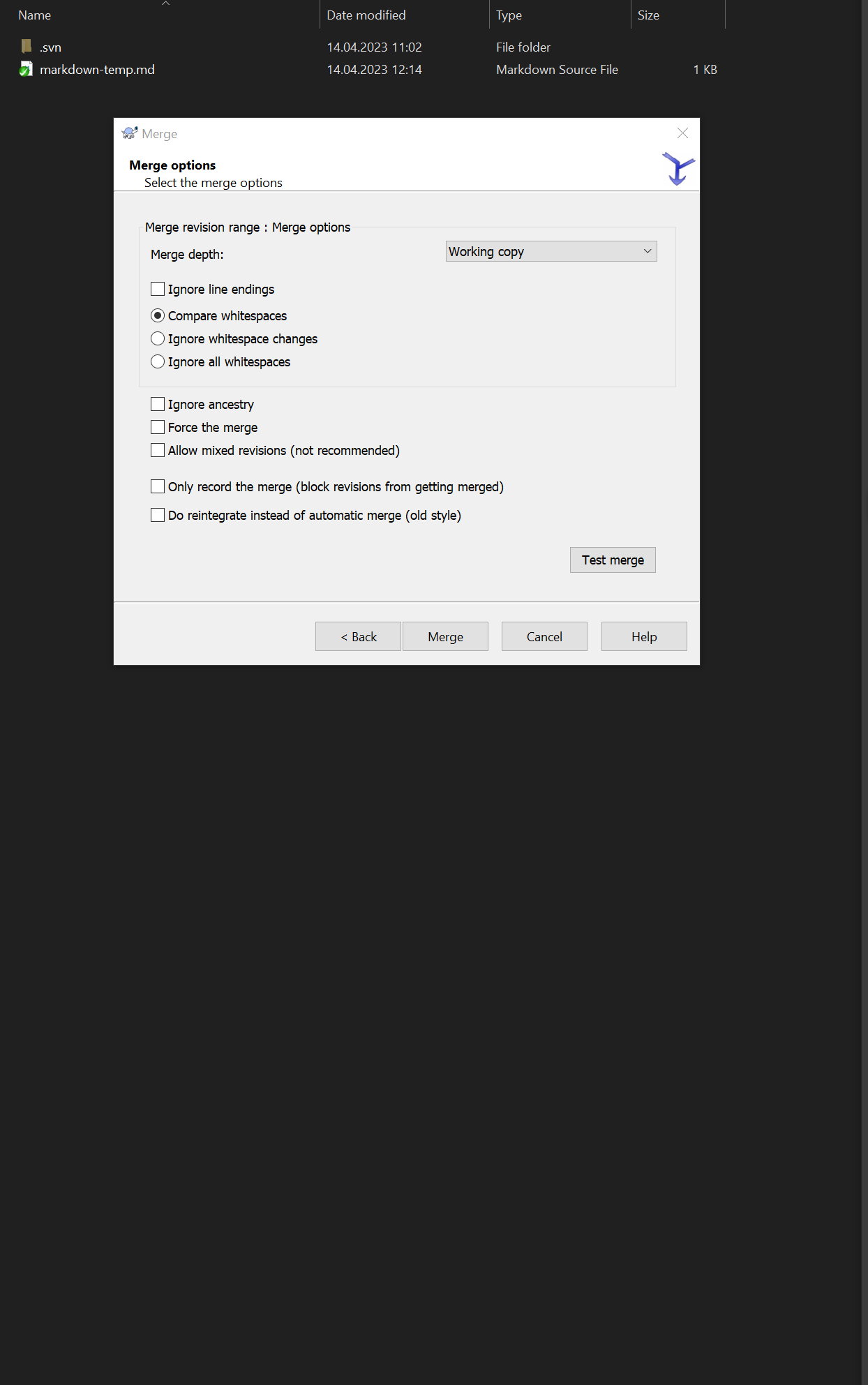


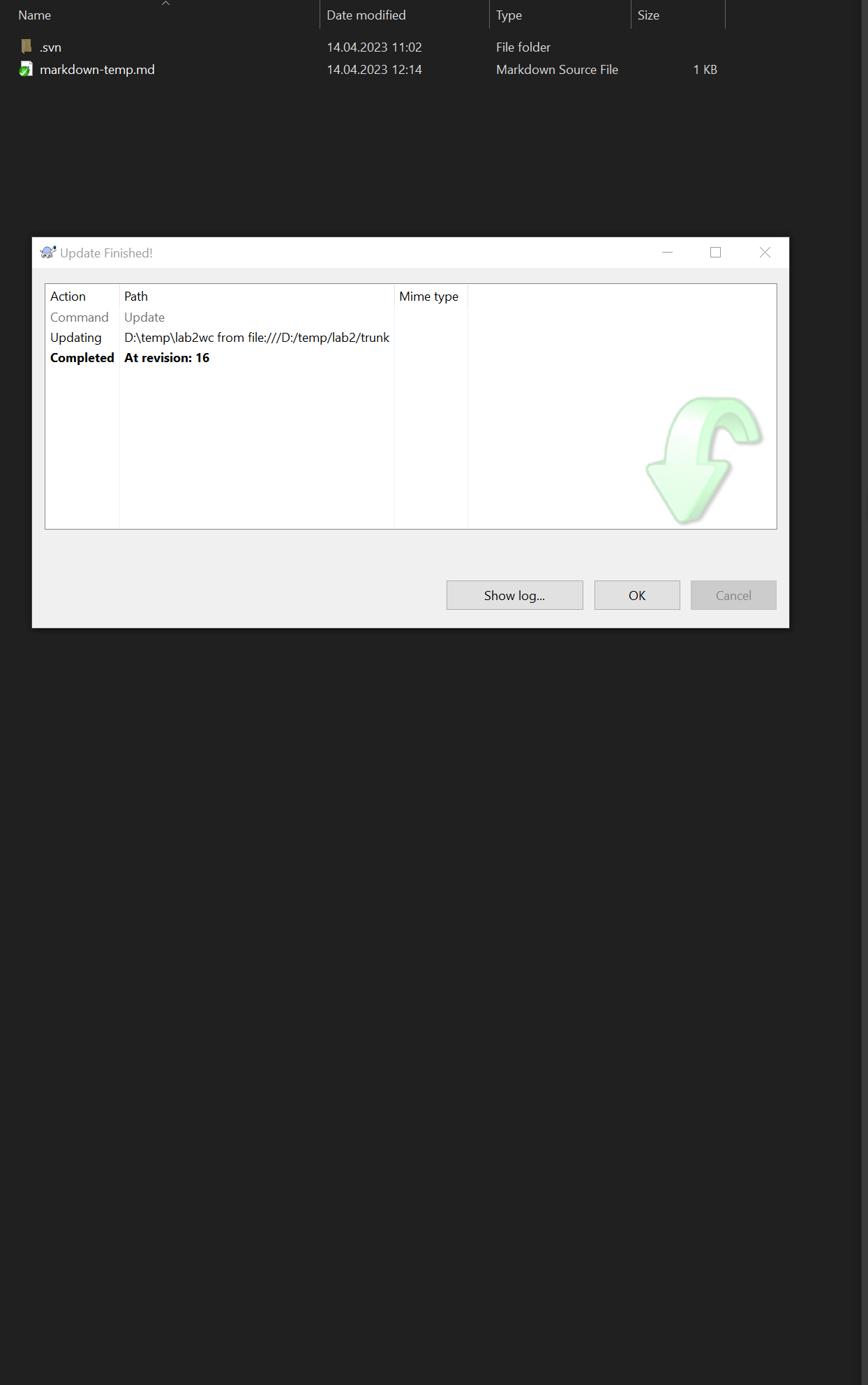
2.3.9 Злити зміни між гілками.







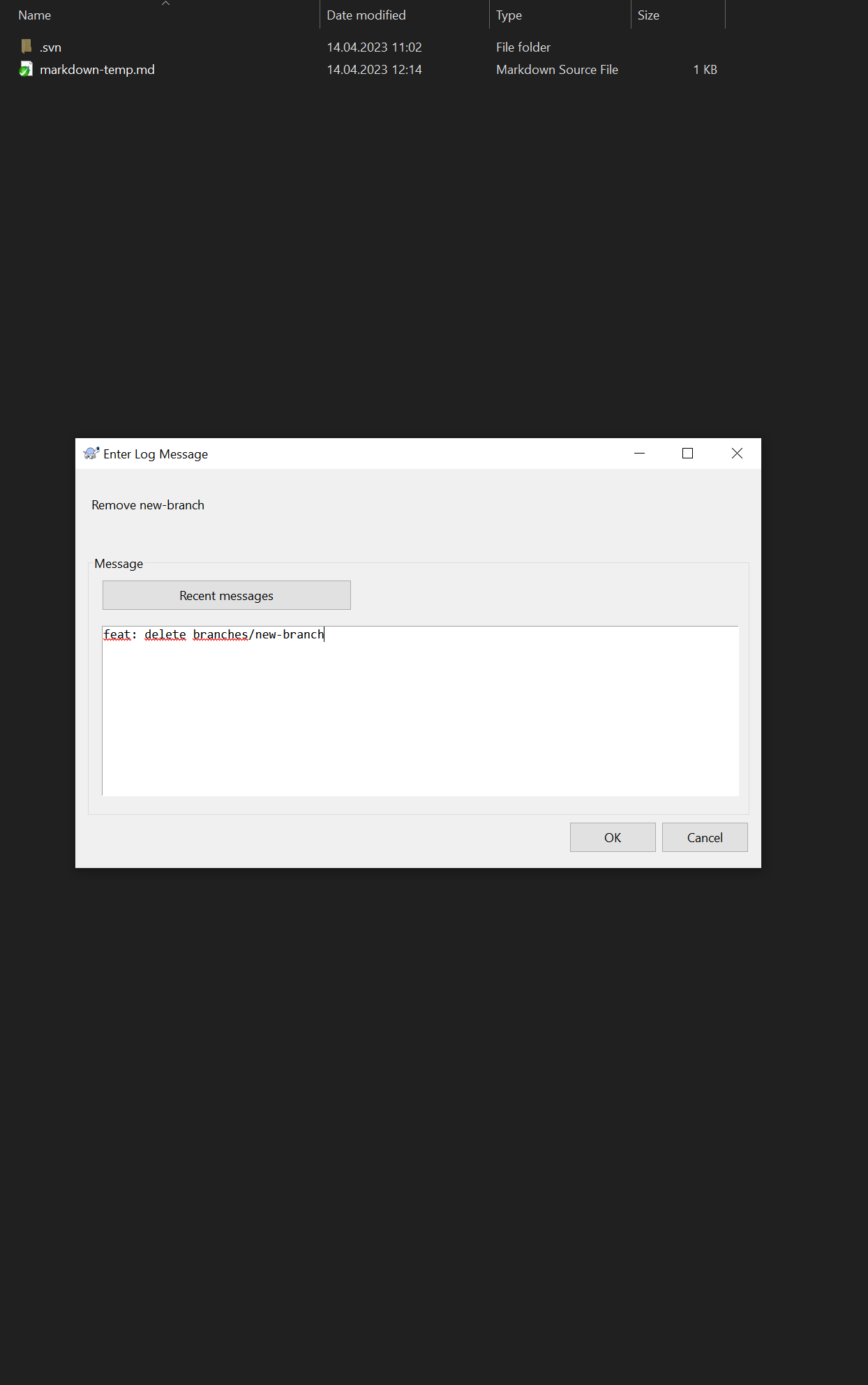


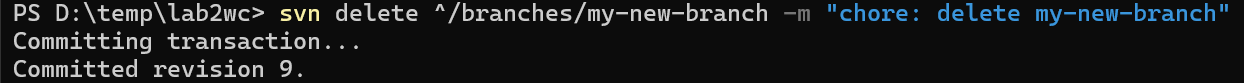




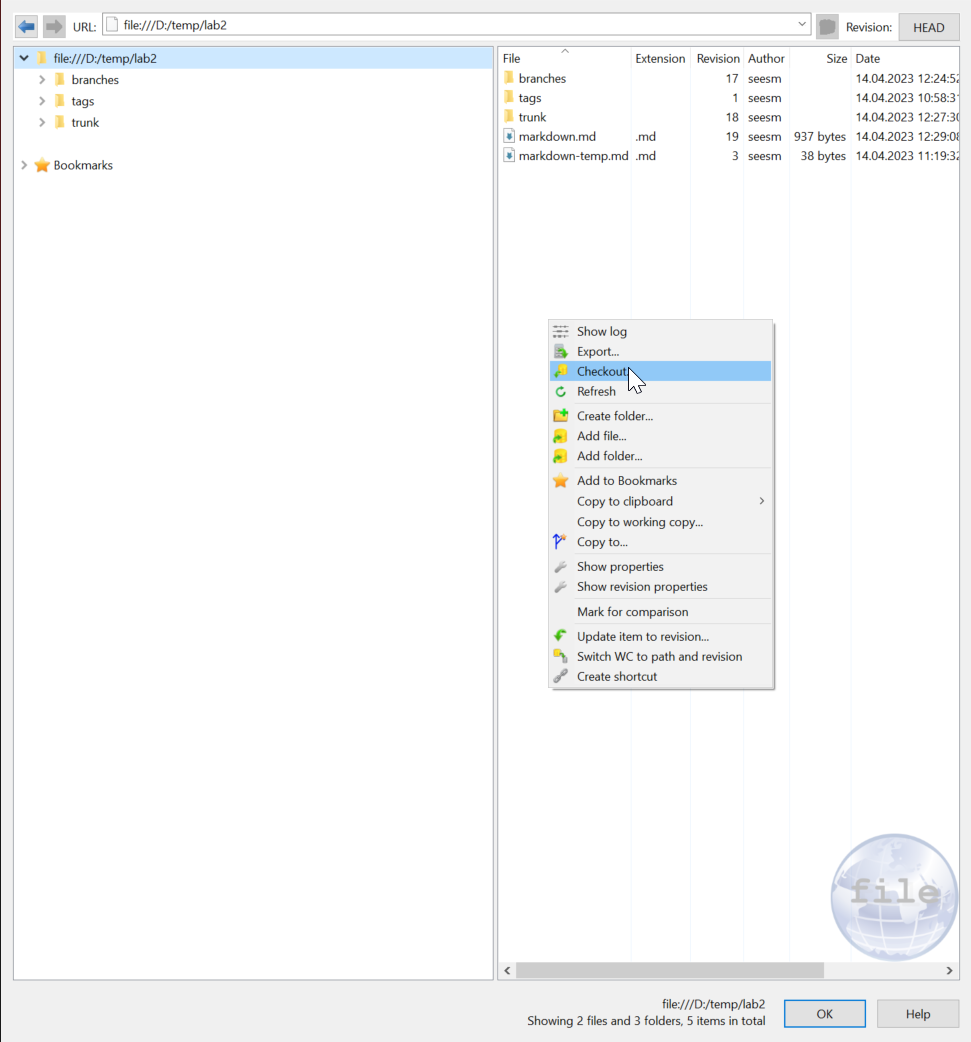
2.3.10 Вилучити гілку.

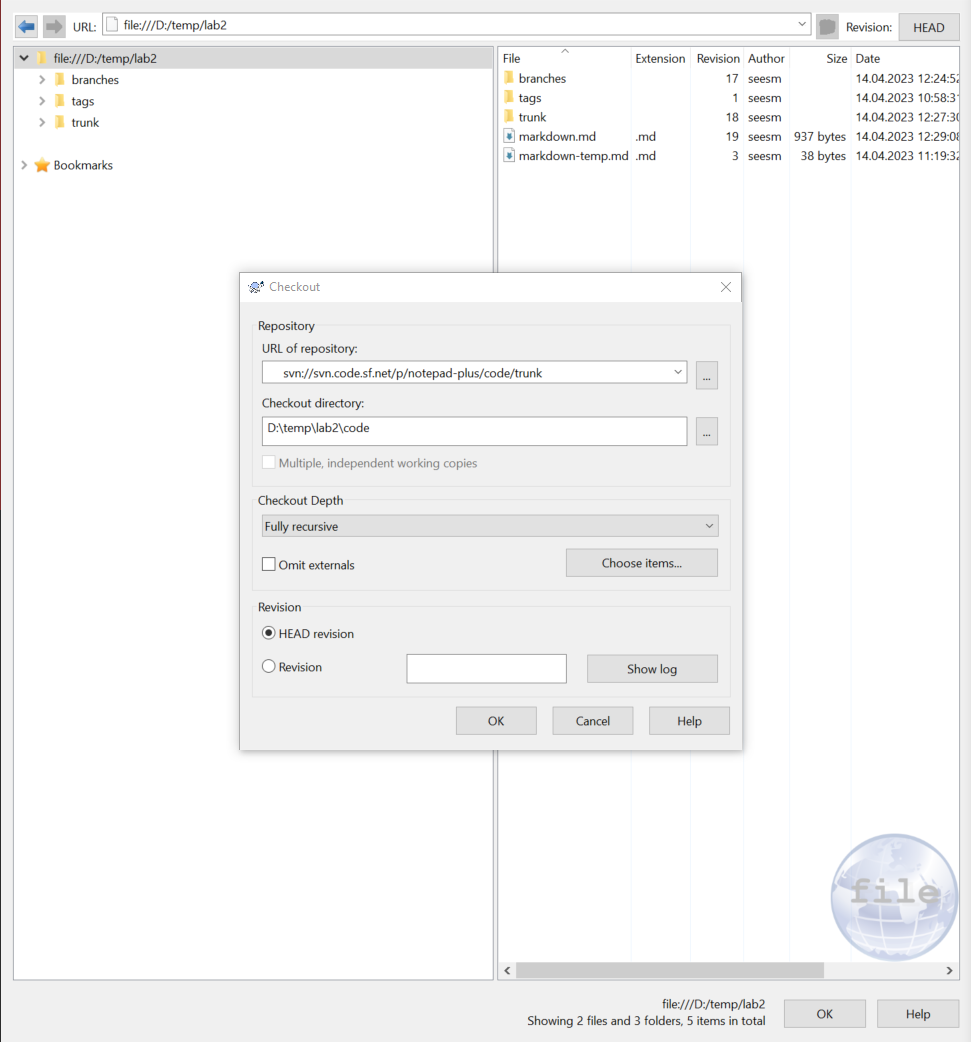


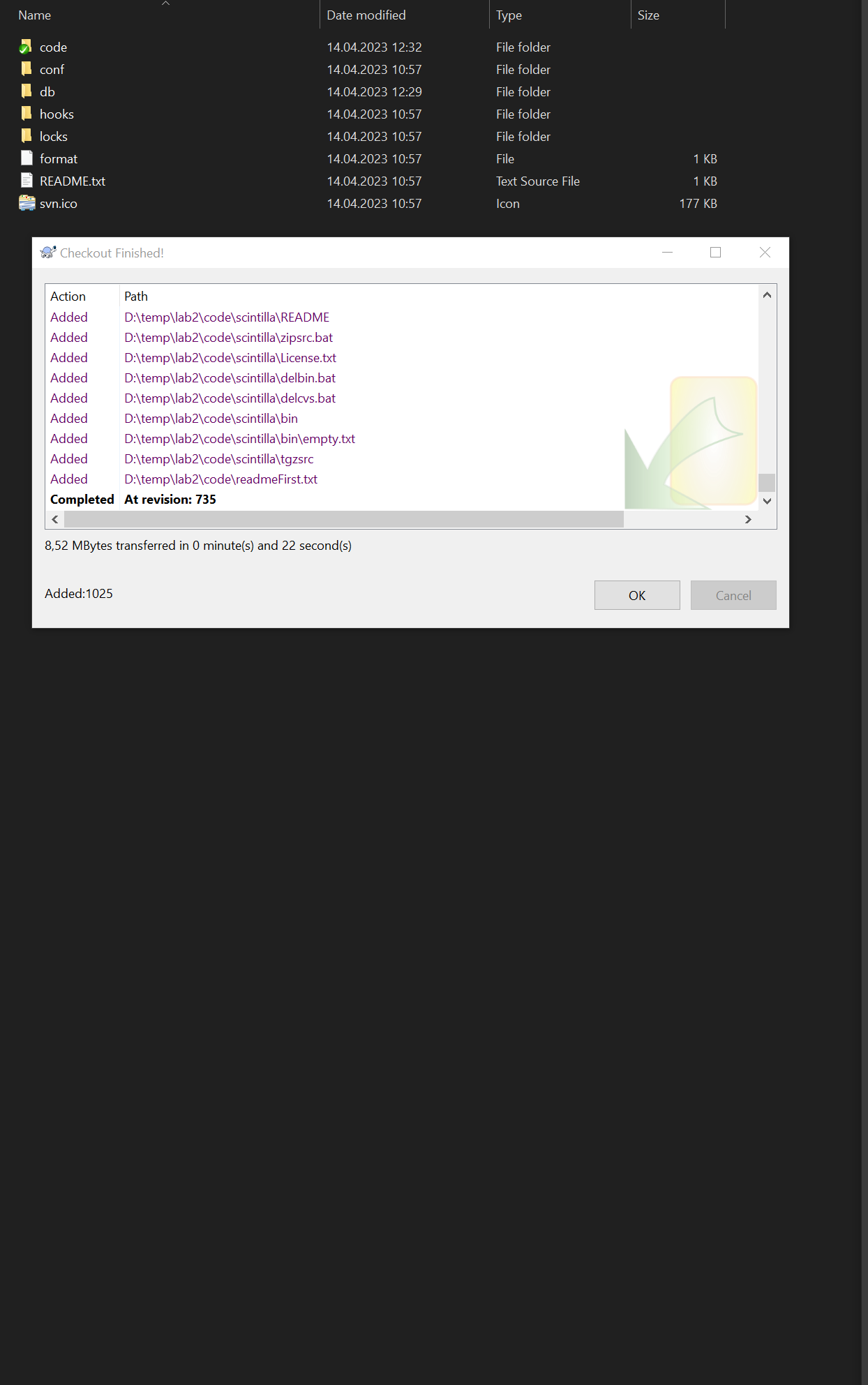


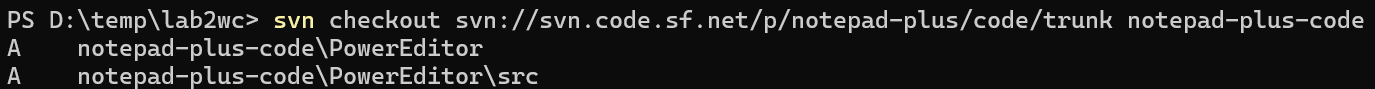


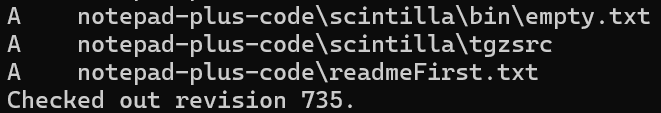
2.3.11 Додати до репозиторію проект на основі даних одного з проектів з відкритим вихідним кодом. 2.3.12 В якості такого проекту можна використати, наприклад, Notepad++ (svn://svn.code.sf.net/p/notepad-plus/code/trunk notepad-plus-code) або будь-який власний.



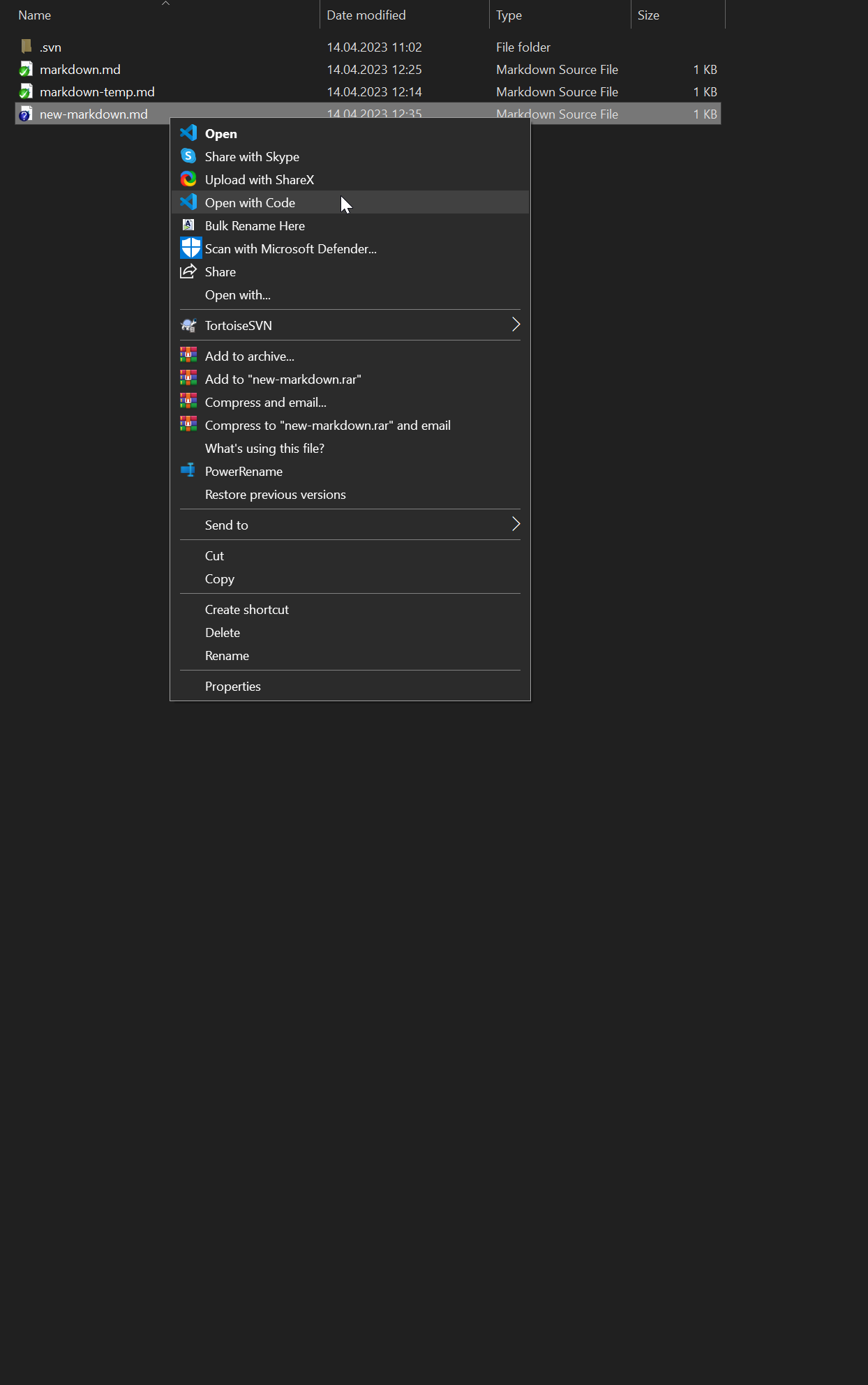






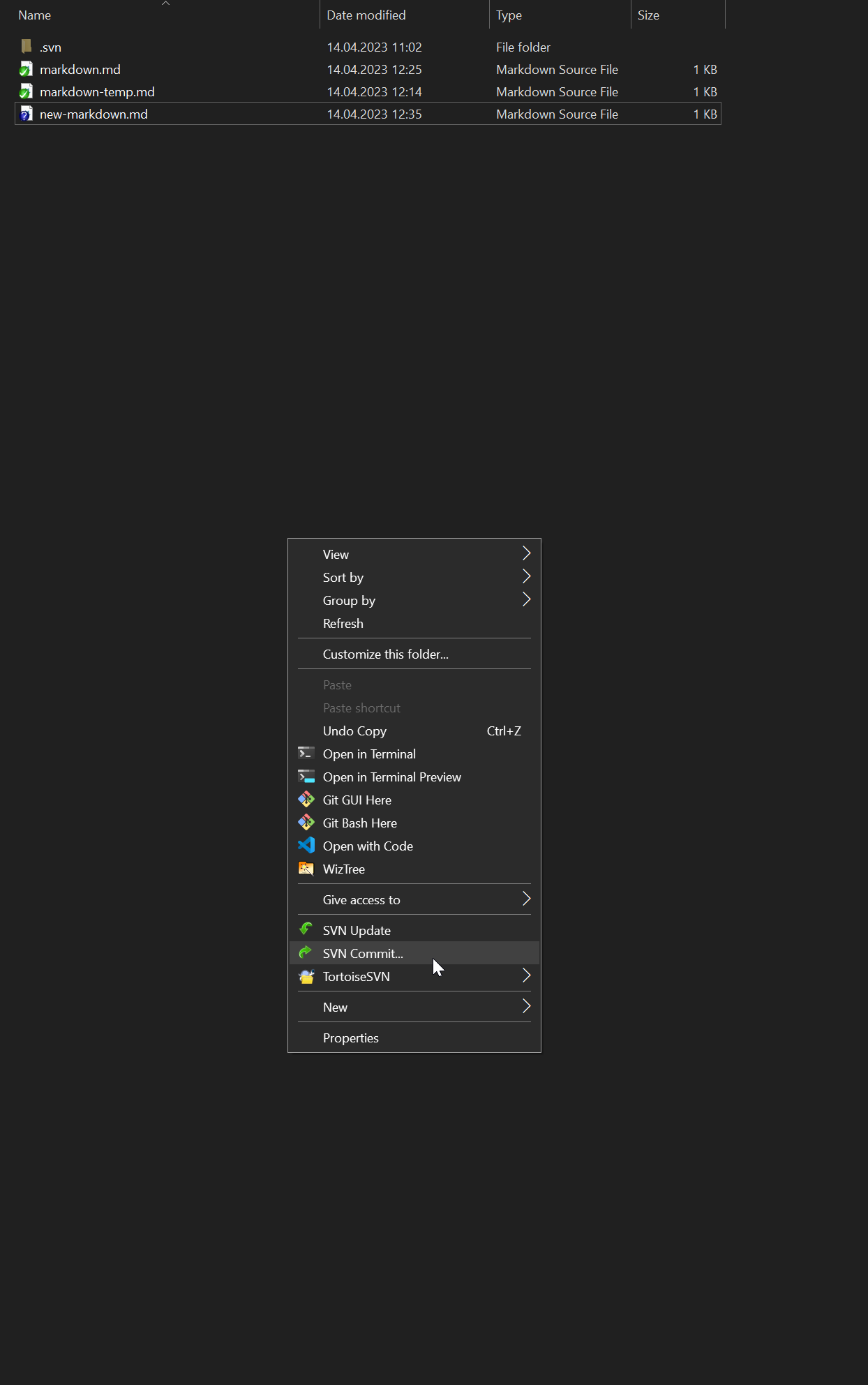


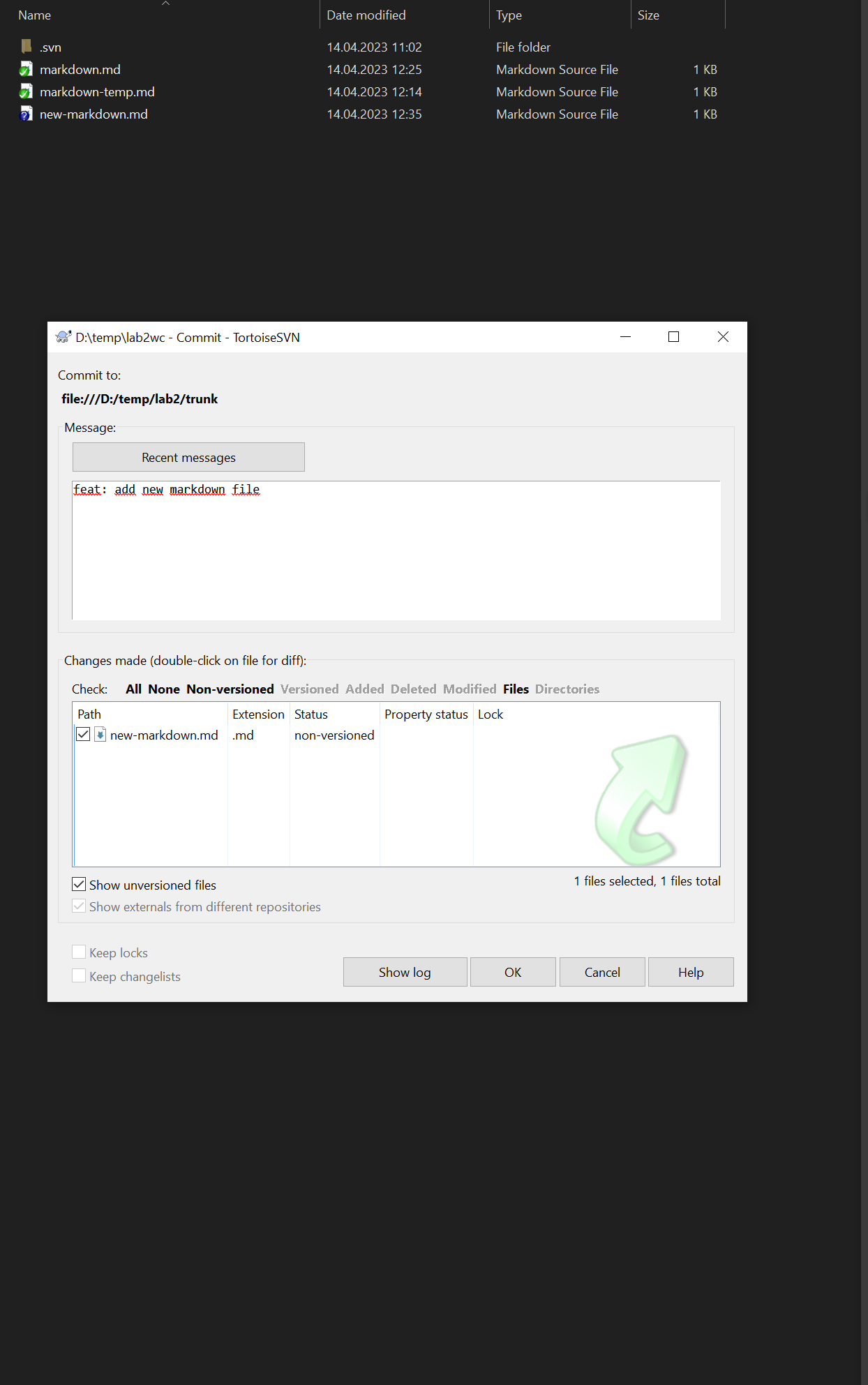
2.3.13 Внести зміни в файли проекту.

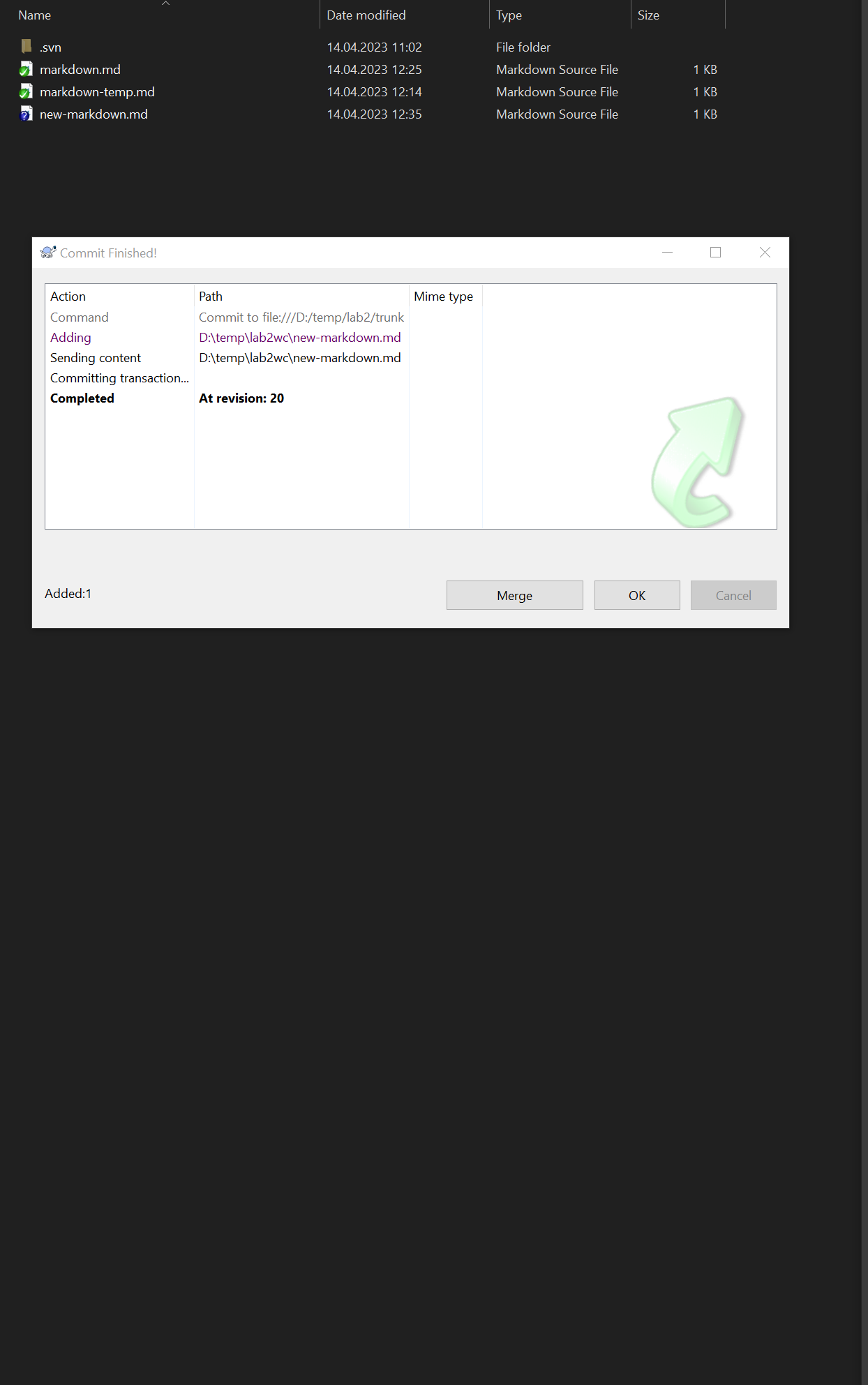


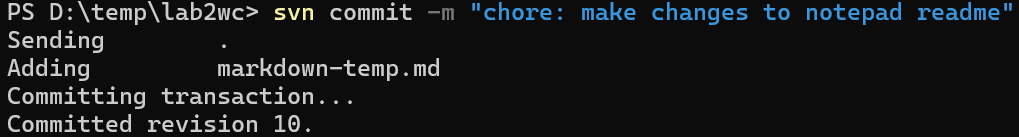


2.3.14 Відправити зміни до репозиторію та пояснити отримані результати.

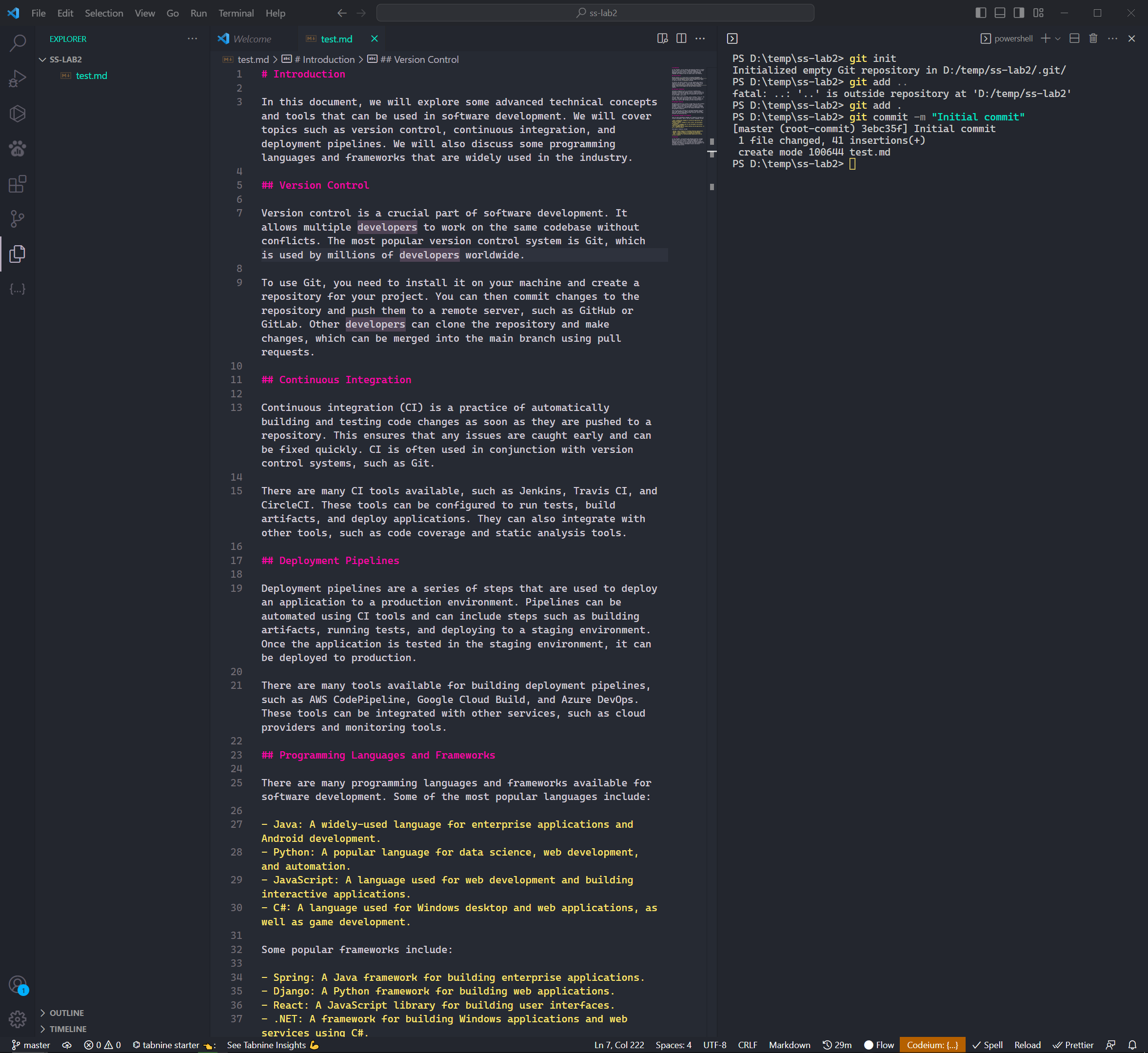




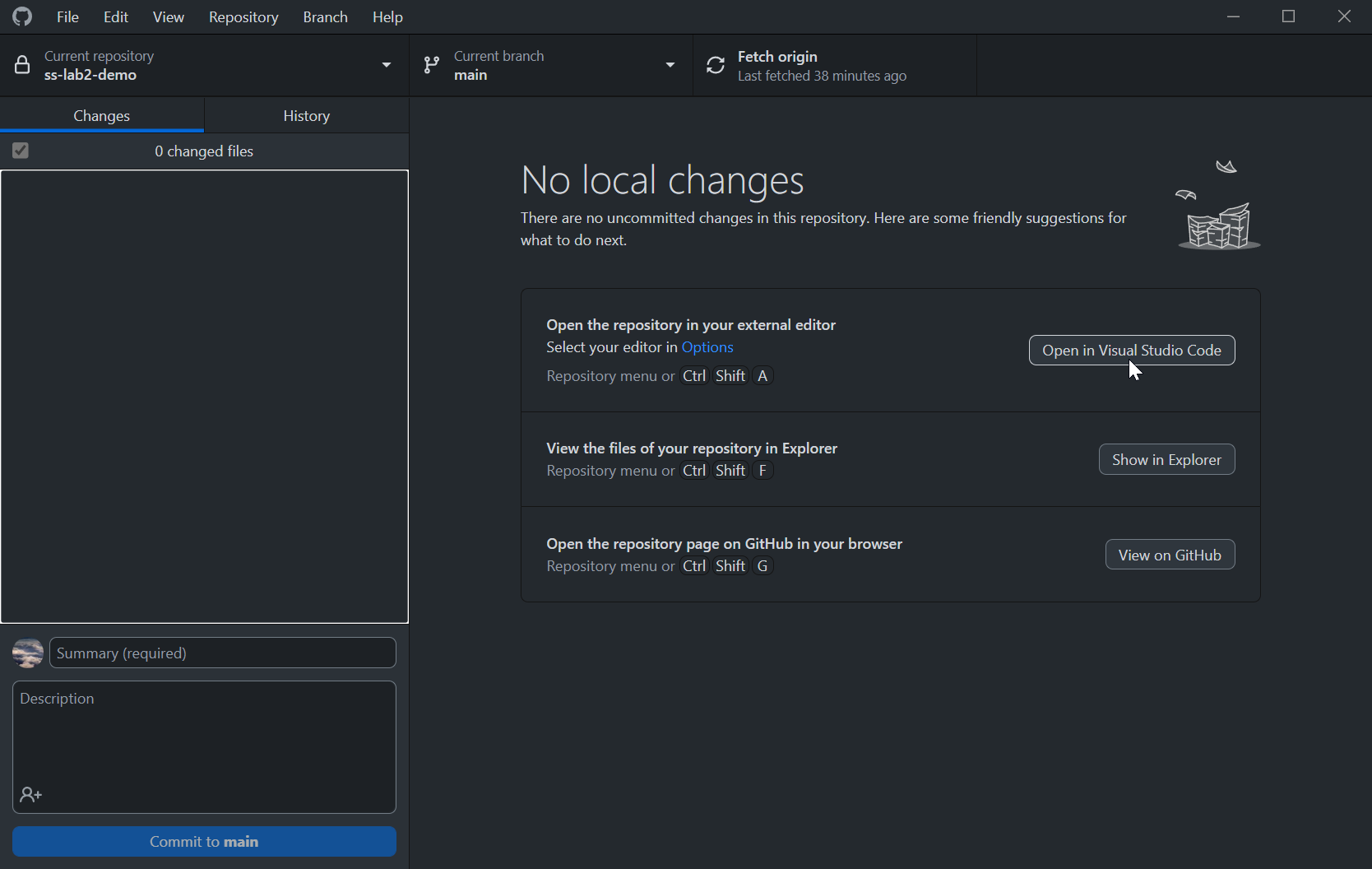


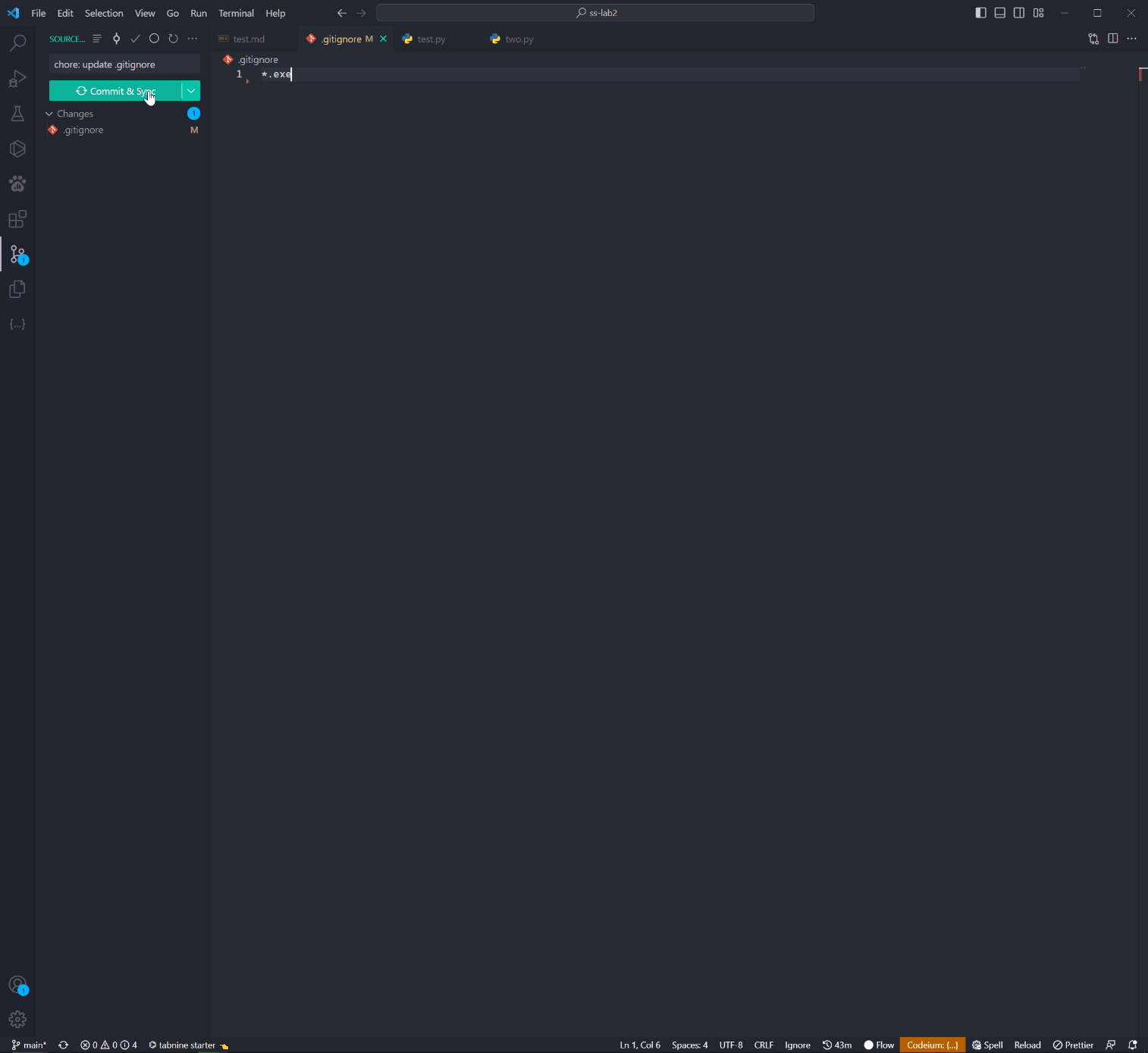


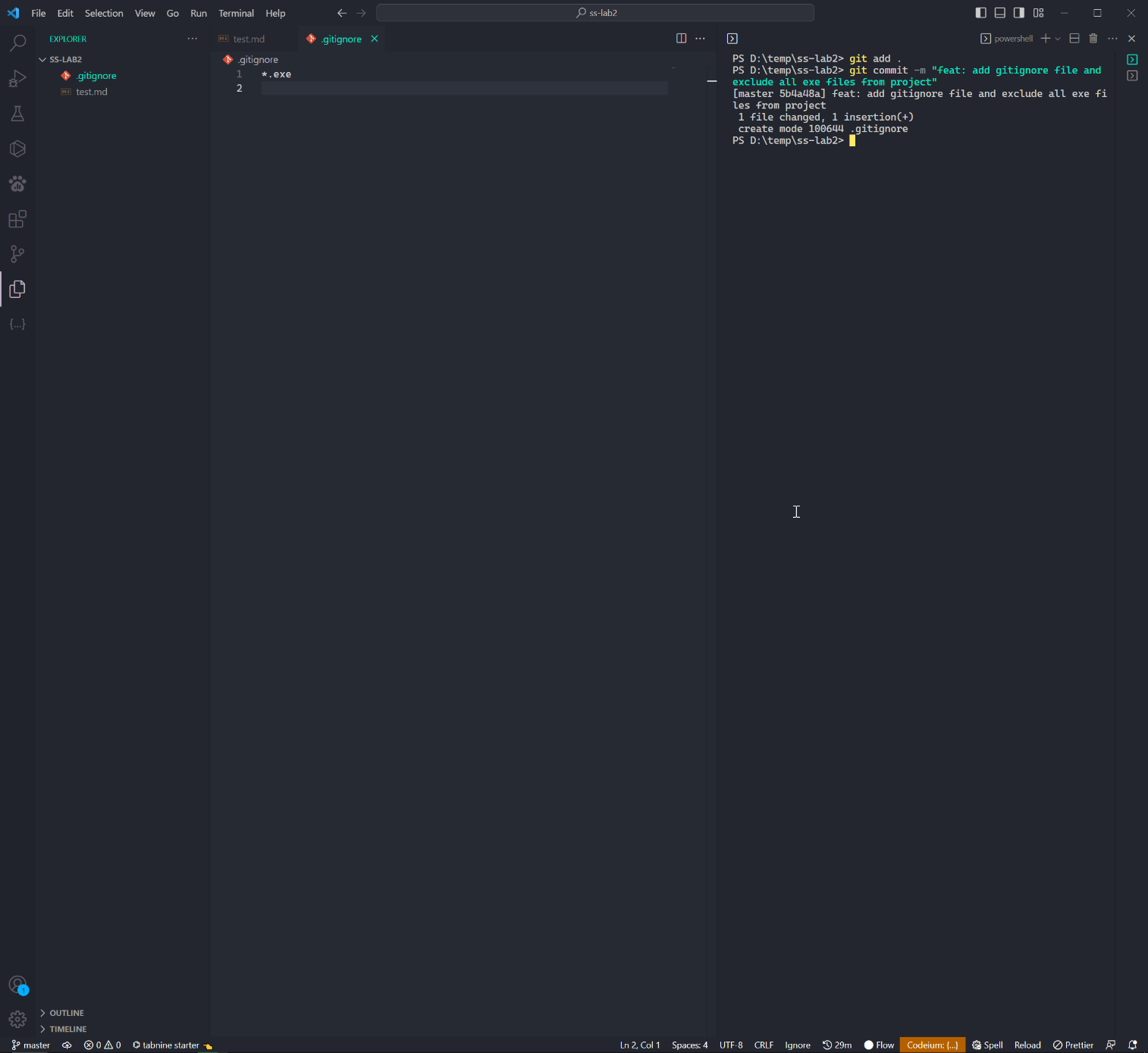
2.3.16 Використовуючи систему керування версіями Git (для пунктів 2.3.16-2.3.25), створити репозиторій на основі одного з раніше розроблених проектів.



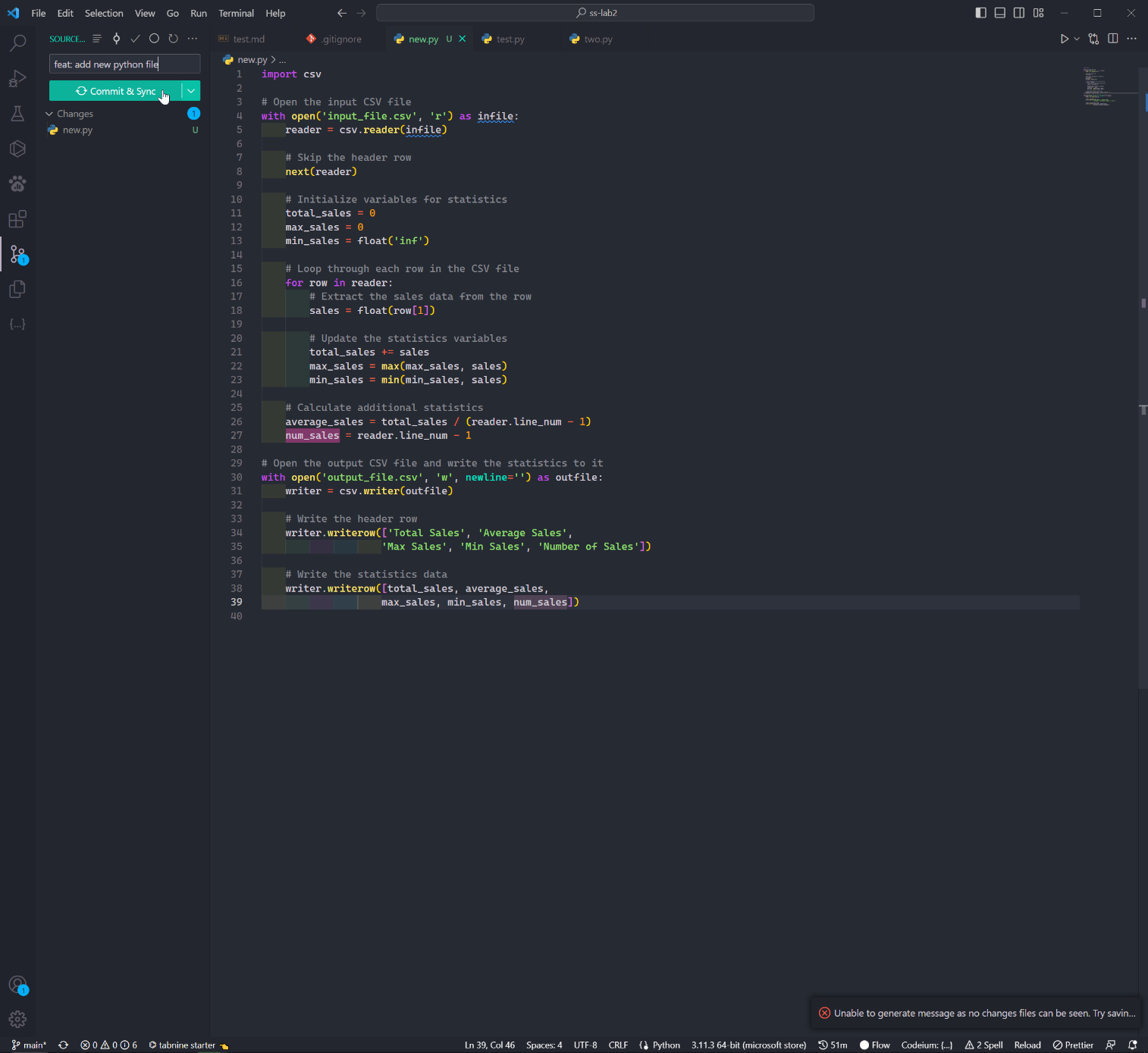
2.3.17 Заборонити автоматичне додання в репозиторій файлів з розширенням \*.exe.

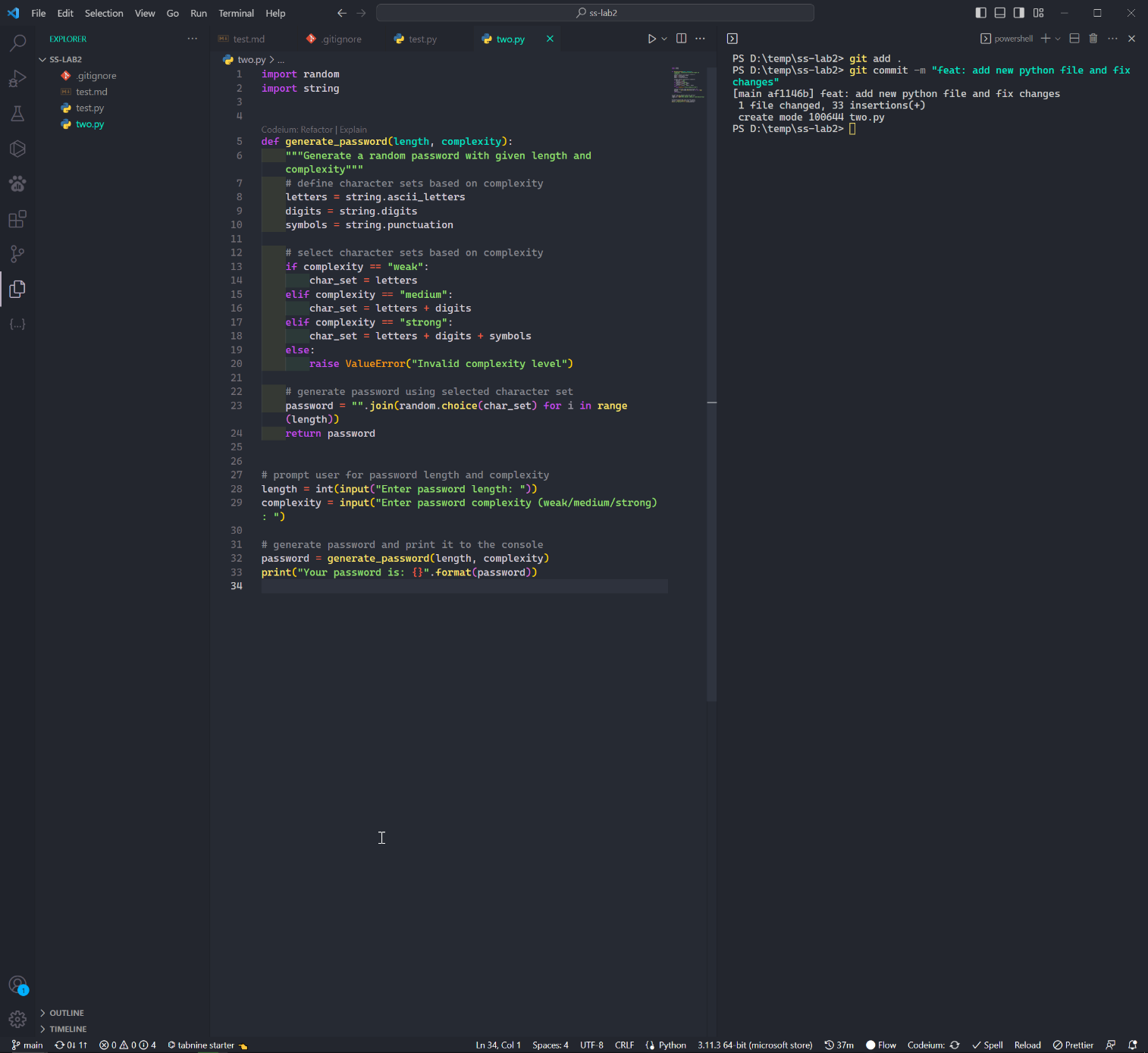




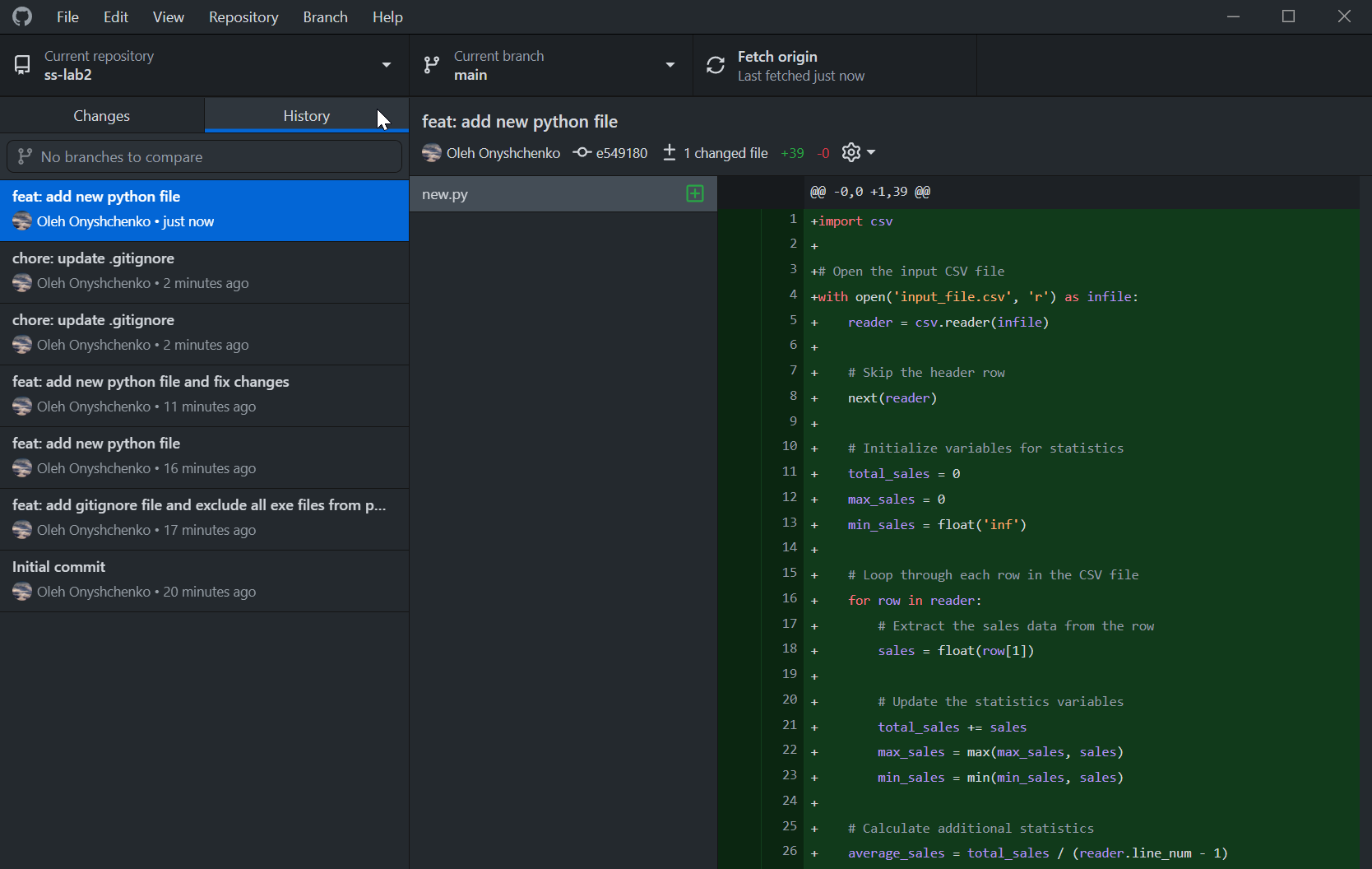


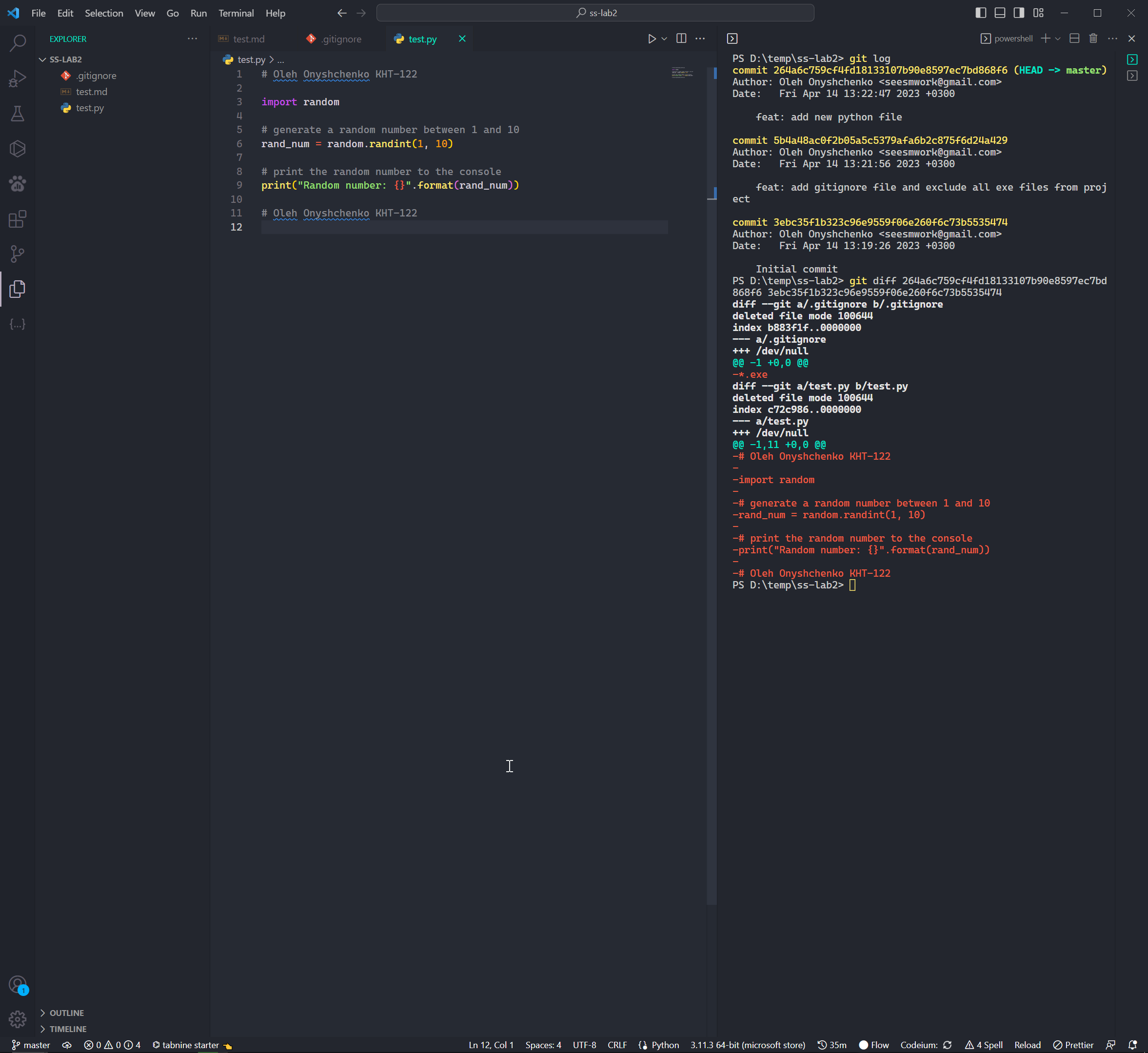
2.3.18 Внести зміни в текст проекту та зафіксувати їх, для підпису використовуючи власні дані.



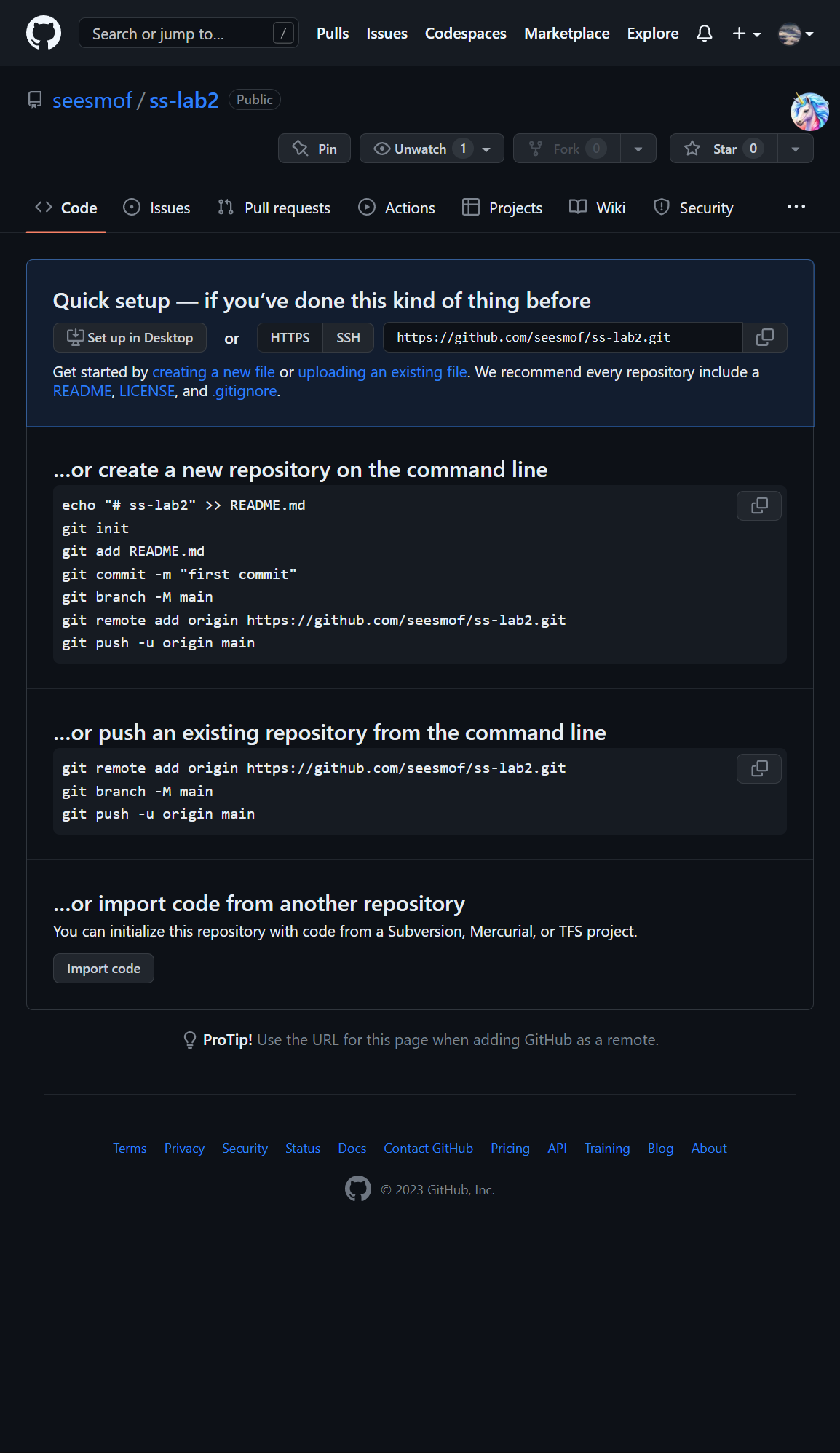


2.3.19 Переглянути різницю між новою версією проекту та початковою.

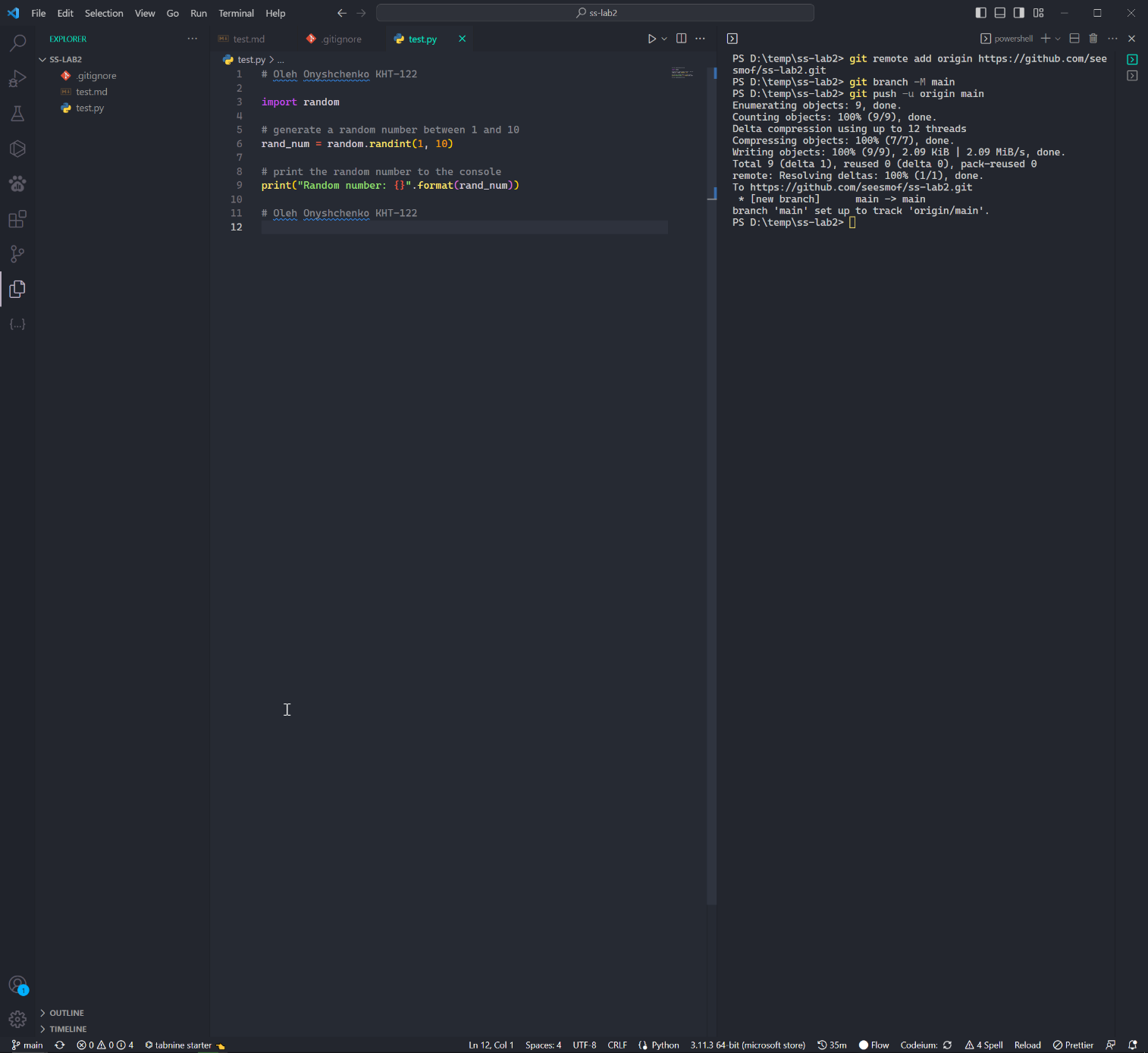




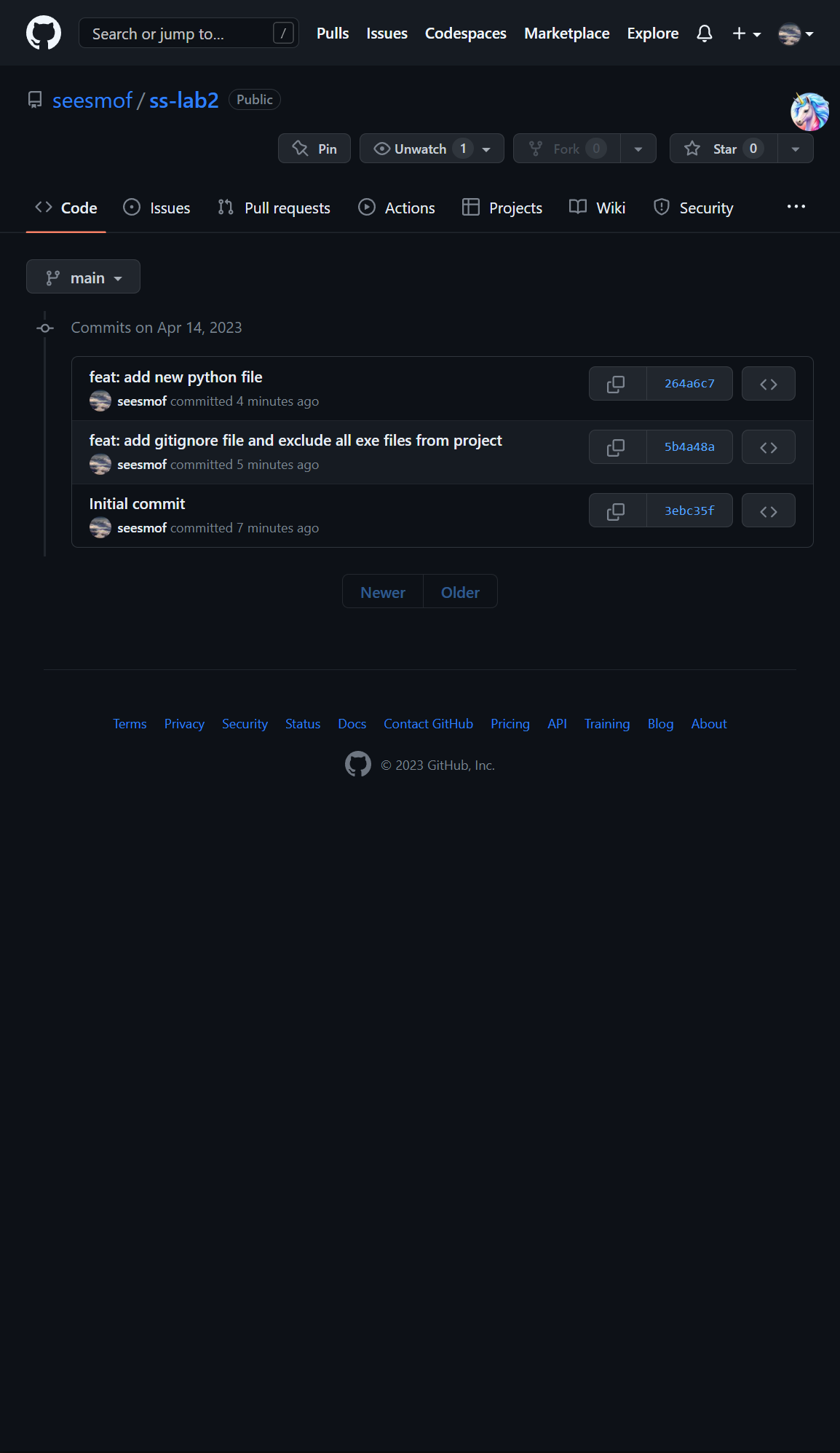
2.3.20 Створити власне віддалене сховище, використовуючи сервіс GitHub.



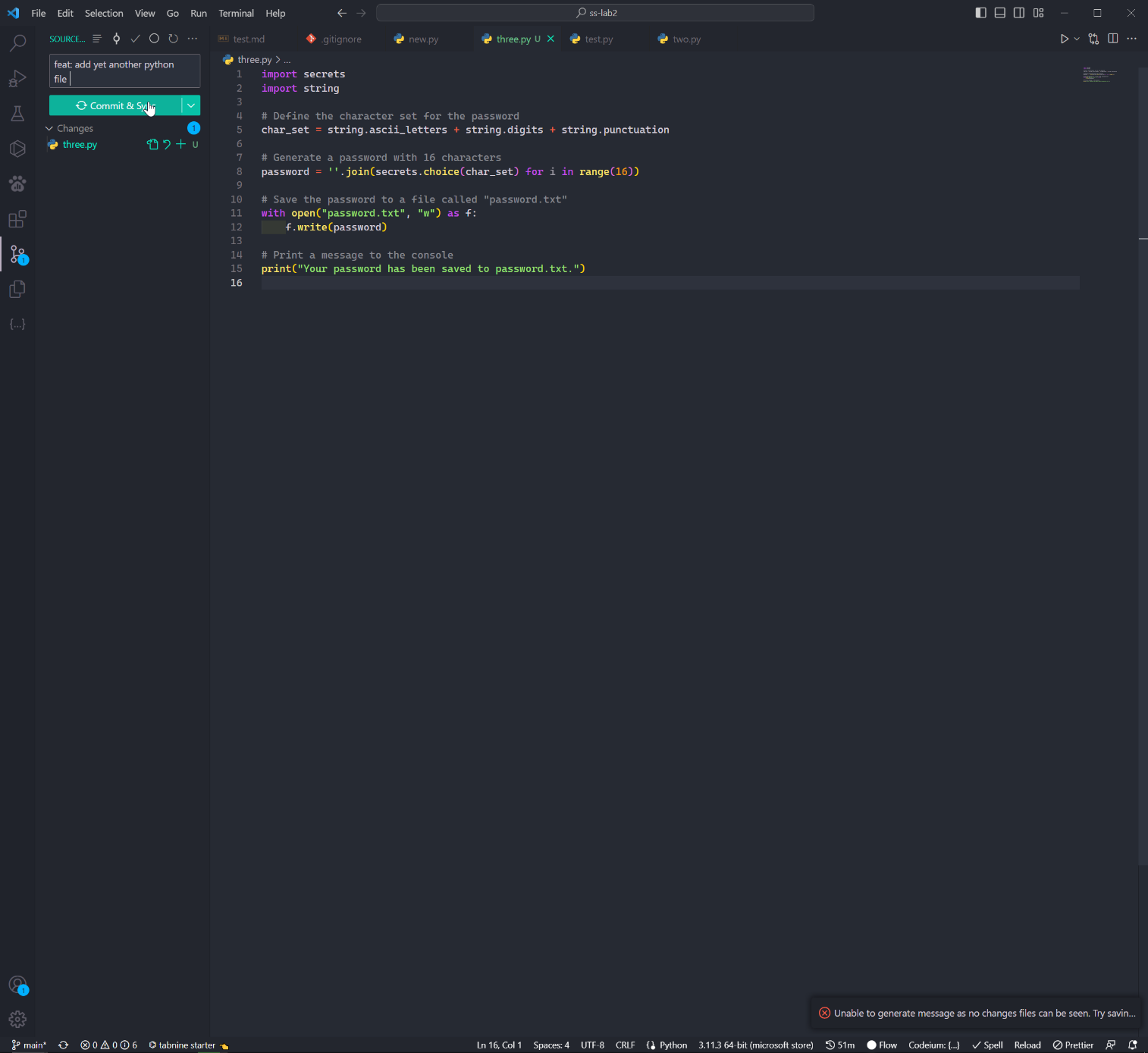
2.3.21 Налаштувати локальне сховище для синхронізації з віддаленим та відправити локальну версію на сервер.

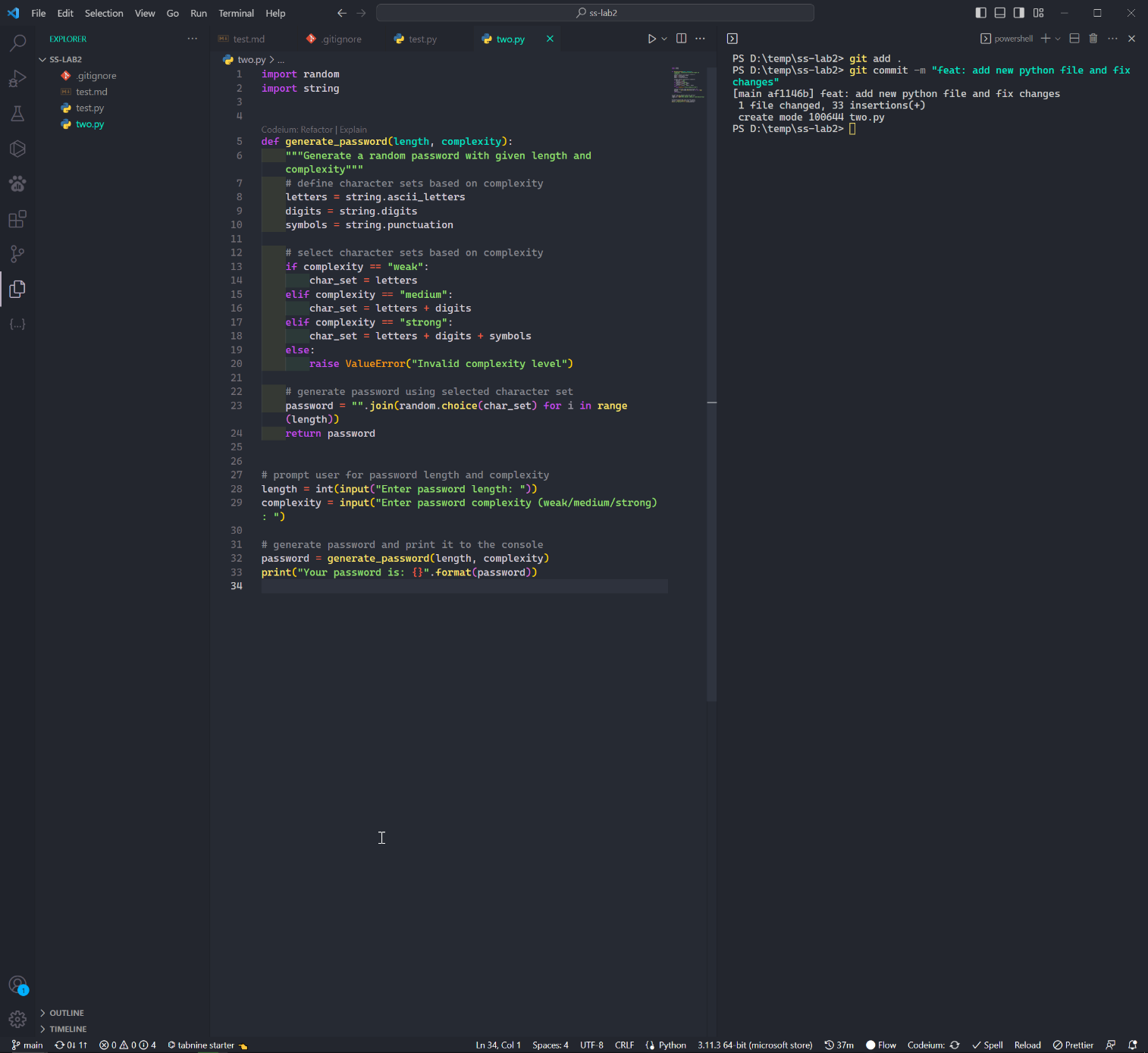


2.3.22 Переглянути історію проекту та сторінку проекту через web-інтерфейс.

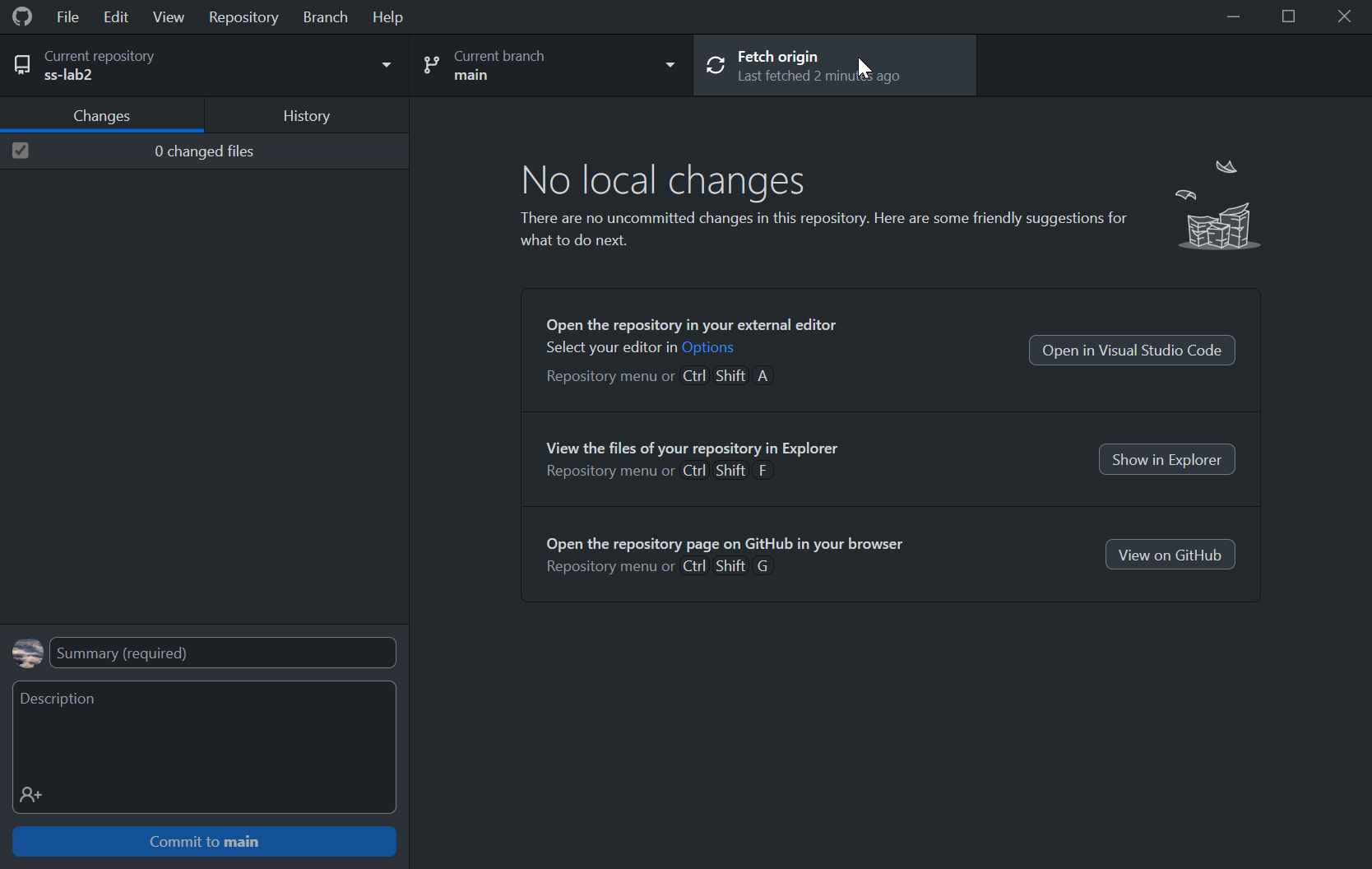


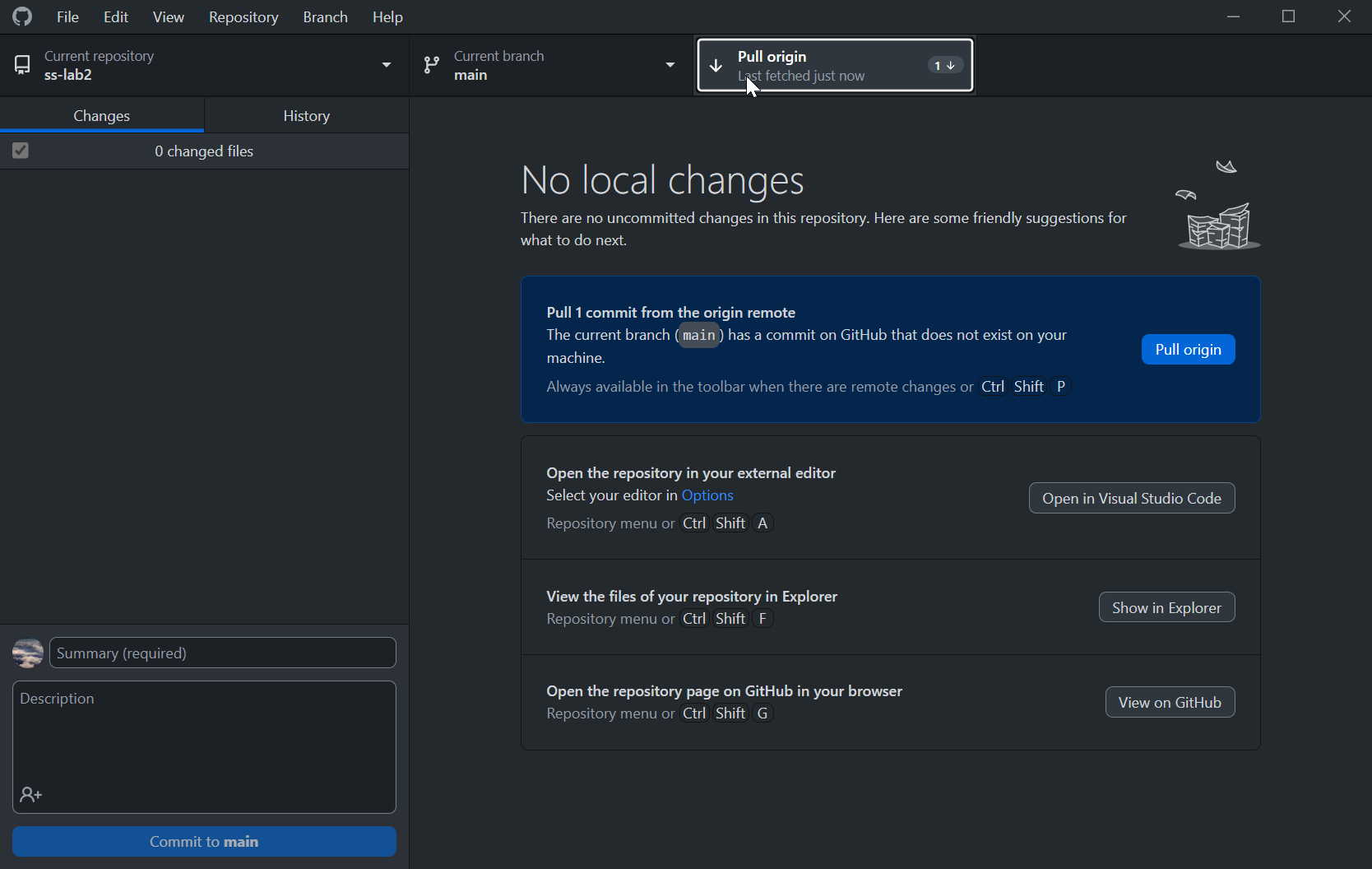
2.3.23 Внести зміни в текст проекту та зафіксувати їх.

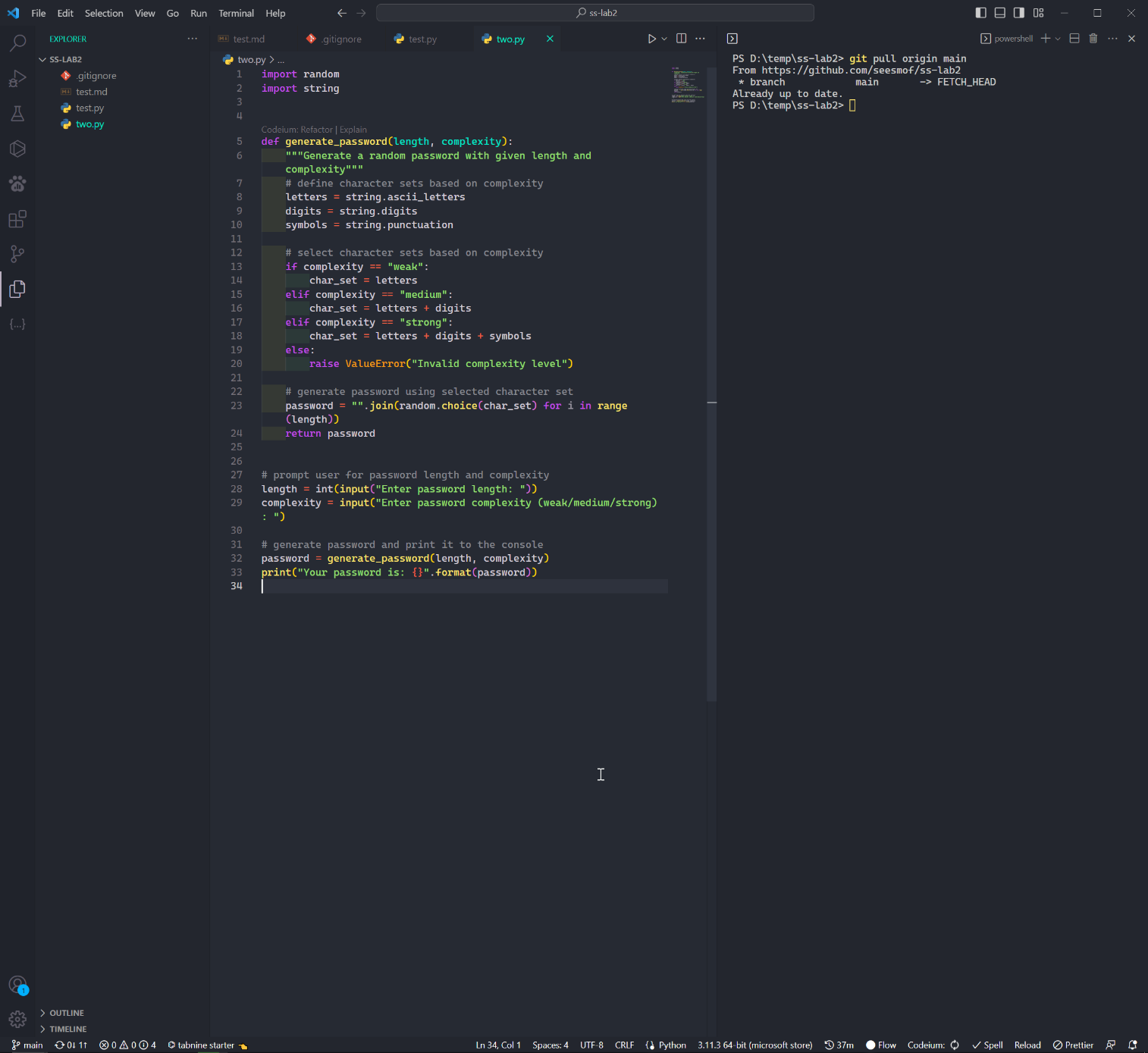




2.3.24 Узгодити локальну версію репозиторію з сервером.







Висновки

В цій роботі ми навчилися використовувати можливості систем керування версіями Subversion та Git для групової роботи.

2.5.1 Що таке система контролю версій?

Система контролю версій (СКВ) - це програмний інструмент, який допомагає вам керувати змінами в наборі файлів з плином часу. VCS дозволяє відстежувати історію змін у ваших файлах, співпрацювати з іншими розробниками та керувати кількома версіями вашої кодової бази. За допомогою VCS ви можете легко повернутися до попередньої версії коду, якщо це необхідно, переглянути зміни, внесені різними розробниками, та об'єднати зміни з різних джерел.

2.5.2 Які існують варіанти організації сховищ у Subversion?

Subversion надає декілька варіантів організації сховищ, зокрема

* Структура "стовбур-гілки-теги" (TBT): Це найпоширеніша структура, яка використовується у Subversion, де основна лінія розробки (стовбур) відокремлена від відгалужень (гілок) та версій (тегів).
* Структура одного проекту: Ця структура може бути використана для невеликих проектів, де структура TBT є непотрібною. Усі файли проекту зберігаються у кореневому каталозі сховища.
* Структура з декількома проектами: Ця структура може бути використана для проектів, які складаються з декількох пов'язаних між собою проектів. Кожен проект має власний каталог у сховищі.

Модульна структура: Ця структура може бути використана для великих проектів, які складаються з декількох модулів. Кожен модуль має власний каталог у сховищі.

2.5.3 Які типи файлів можна зберігати у сховищі?

Subversion може зберігати будь-які типи файлів, включаючи файли коду, файли зображень, аудіофайли, відеофайли і текстові файли. Однак, зазвичай рекомендується зберігати у Subversion лише текстові файли, оскільки двійкові файли (наприклад, зображення, аудіо, відео) можуть швидко роздути сховище і ускладнити керування ним. Якщо вам потрібно зберігати бінарні файли у Subversion, рекомендується використовувати розширення SVN Git Large File Storage (LFS) або спеціальну систему керування бінарними артефактами.

2.5.4 Що таке сховище Subversion?

Сховище Subversion - це центральне місце, де зберігаються всі файли проекту Subversion. Сховище містить всю історію версій проекту, включно з усіма змінами, внесеними до файлів з плином часу. Сховище зазвичай розміщується на сервері і до нього можуть мати доступ декілька розробників для спільної роботи над проектом.

2.5.5 Скільки сховищ може використовувати Subversion для зберігання даних?

Subversion може використовувати будь-яку кількість сховищ для зберігання даних, залежно від ваших потреб. Кожне сховище є незалежним і містить власний набір файлів та історію версій. Зазвичай використовується один сховище для одного проекту, але ви також можете використовувати декілька сховищ для одного проекту, якщо це необхідно. Кількість сховищ, які слід використовувати, залежить від вимог вашого проекту та складності вашої кодової бази.