

Міністерство освіти і науки України

Національний технічний університет «Дніпровська політехніка»



Звіт

Лабораторна робота №2

З дисципліни «Аналіз програмного забезпечення»

Виконала:

студентка групи 122-22-5

Алексєєнко Є.Д.

Перевірив:

Мінєєв Олександр Сергійович

Юлія Олександрівна Шевченко

м. Дніпро

2025 рік

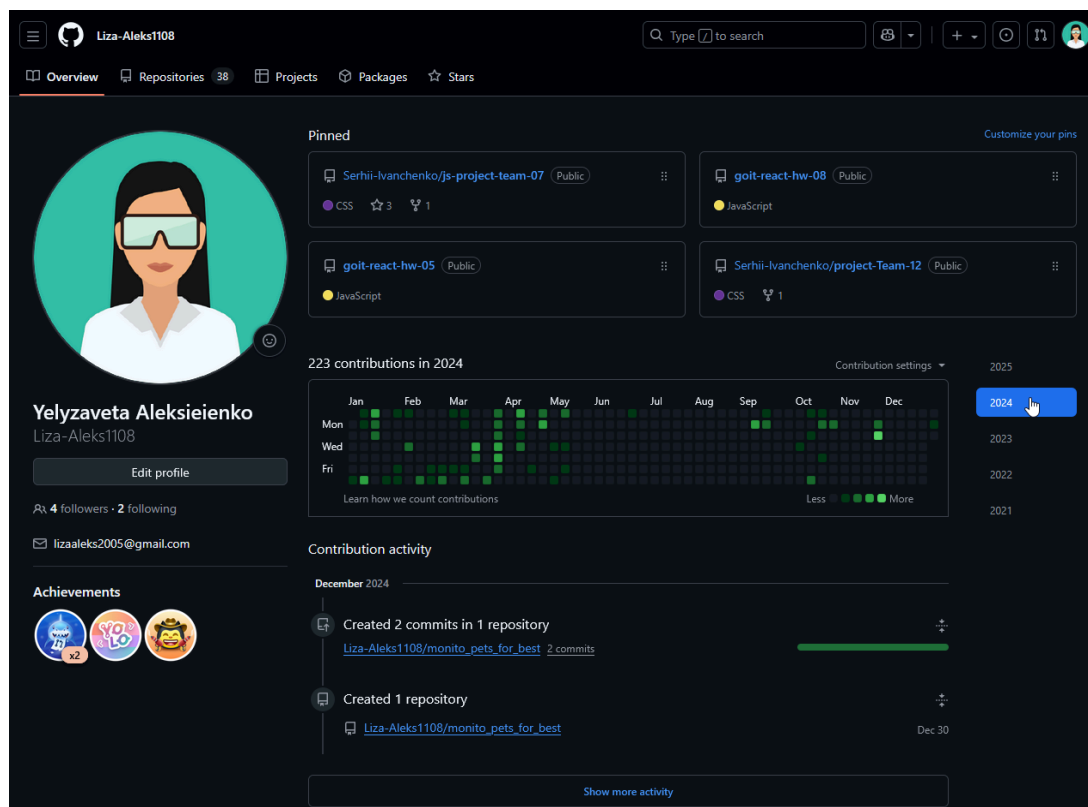
Тема: Створення і налаштування профілю у системі Git.

Мета: Набування навичок при реєстрації та налаштування облікового запису (account) на хостингу GitHub.

Очікувані результати навчання: уміння створити, налаштувати та підтримувати власний профіль на найбільшому хостингу для сховищ Git.

Завдання. Створити власний репозиторій в GitHub. В подальшому усі результати своїх практичних робіт необхідно завантажувати у цей репозиторій. В репозиторії створити для кожної практичної роботи окрему папку і розмістити звіт. Якщо ви ще не маєте обліковий запис, будь ласка, в професійному світі, запис повинен виглядати, як поєднання першої літери ім'я та прізвище в повному виді. Наприклад Микола Єфремов буде «myefremov», а Тарас Шевченко «tshevchenko».

В мене вже є робочий акаунт у системі GitHub:



Посилання: <https://github.com/Liza-Aleks1108/yaleksieienko>

Далі необхідно створити новий проект (репозиторій) для розміщення всіх практичних робіт в ньому:

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

1

General

Owner *

Liza-Aleks1108

Repository name *

yaleksieienko

✔ yaleksieienko is available.

Great repository names are short and memorable. How about [vigilant-octo-telegram](#)?

Description

В подальшому усі результати практичних робіт з дисципліни "Аналіз програмного забезпечення" необхід

135 / 350 characters

2

Configuration

Choose visibility *

Choose who can see and commit to this repository

Public

Start with a template

Templates pre-configure your repository with files.

No template

Add README

READMEs can be used as longer descriptions. [About READMEs](#)

Off

Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

No .gitignore

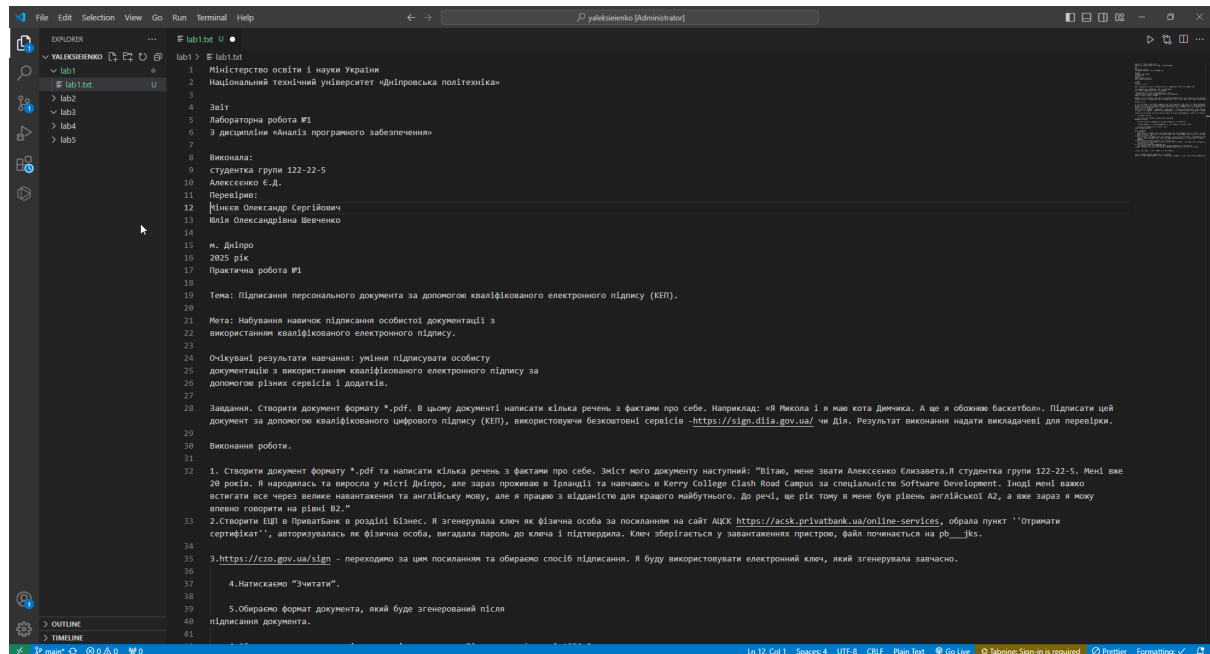
Add license

Licenses explain how others can use your code. [About licenses](#)

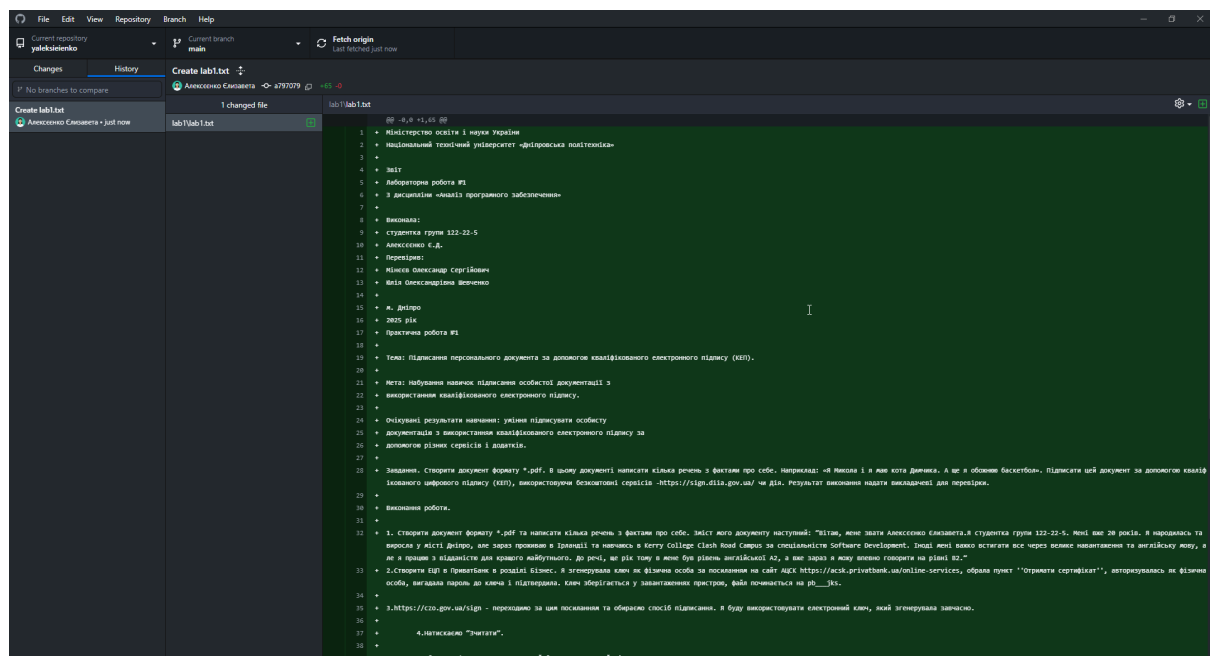
No license

Create repository

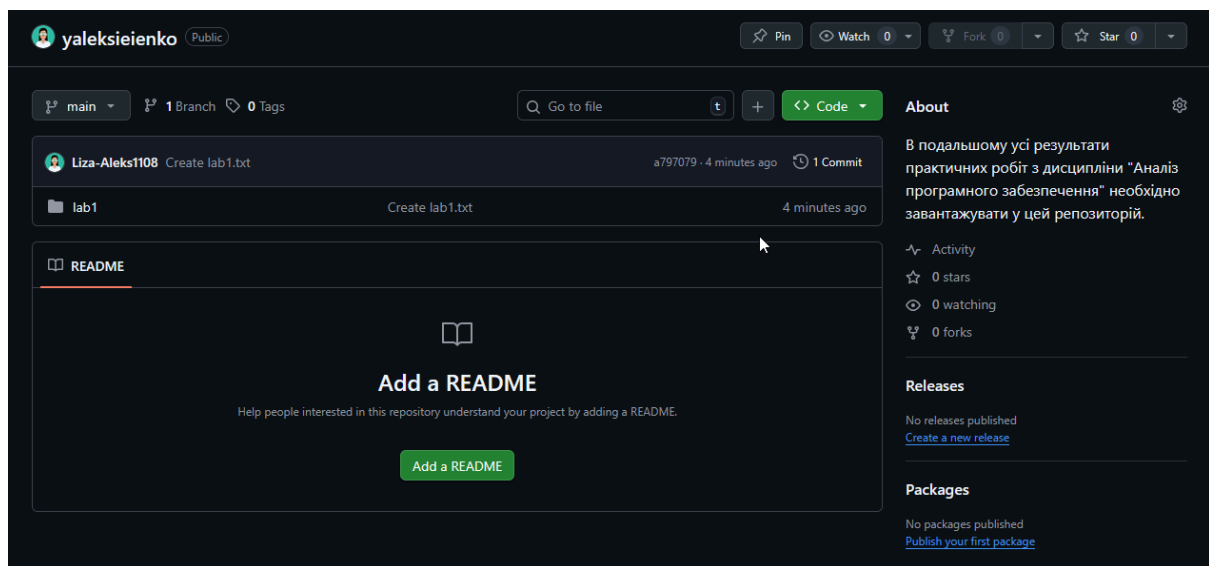
Створимо для кожної практичної роботи окрему папку та розмістимо там звіт:



Зробимо commit to main через GitHub Desktop:



Результат:



1. Що таке Git?

Git — це система для керування та контролю версіями. До основних завдань Git належить збереження коду та історії змін, збереження інформації про користувачів, які змінюють код, можливість відкрити код до будь-якої версії, можливість об'єднувати різні версії, зміни версій, підготовка кінцевого коду до релізу.

2. Що таке репозиторій у Git?

Репозиторій — це місце, де зберігається сам проєкт — усі файли, коди, документація, зображення і будь-які інші файли, які стосуються проєкту і вся історія змін цього проєкту.

3. Які переваги використання Git?

Можна повернутись до попередньої версії, бачити, хто що змінив, працювати в команді без плутанини й не боятись втратити

дані або якщо нова версія містить помилки, тоді проєкт легко можна відкотити до попередньої версії.

4. Яка мова використовується в Git?

Використовуються команди, які вводяться в терміналі, тобто це мова командного рядка.

5. Як можна створити репозиторій у Git?

Найпростіше — через GitHub, натиснути “New repository” і вказати назву та характеристики репозиторія. Також можна створити репозиторій через термінал через VS Code.

6. Яка команда використовується для видалення гілки?

Щоб видалити гілку, треба написати в консолі `git branch -d назва_гілки`.

7. Що таке контроль версій у Git?

Це коли всі зміни в коді фіксуються, і при потребі можна подивитись старі версії або повернутись до них.

8. Як можна виправити несправний комміт?

Можна скористатись командою `git commit --amend`, щоб підправити останній комміт, або зробити новий із виправленням.

9. Як дізнатись у Git, чи гілку вже об'єднано в master?

Вводжу команду `git branch --merged`, і якщо моя гілка там є — значить, її вже об'єднали. А подивитися в самому GitHub.

Висновок.

У ході виконання роботи я ознайомилась із системою Git та сервісом GitHub, навчилась налаштовувати власний профіль та репозиторій. Також отримала практичні навички завантаження файлів і роботи з коммітами через GitHub Desktop.