**Державний вищий навчальний заклад**

**Ужгородський національний університет**

**Факультет інформаційних технологій**

**ЛАБОРАТОРНА РОБОТА № 2**

**Тема:** “Мова гіпертекстової розмітки (**HTML**)”

Виконав студент І курсу

спеціальності «Інженерія

програмного забезпечення»

Романюк Артем Романович

**Ужгород-2024**

***Мета:*** Ознайомитись з мовою гіпертекстової розмітки (HTML) та навчитись працювати з системою контролю версій GIT.

***Вивчити питання:***що таке GIT? Його основи. Хмарне сховище Github.

***Завдання до роботи:***

1. Відкрити або встановити один з редакторів або IDE:
   * [**Webstorm**](https://www.jetbrains.com/webstorm/) (безкоштовна ліцензія для студентів)
   * [VS Code](https://code.visualstudio.com/) (як альтернатива, безкоштовний)
   * [Sublime](https://www.sublimetext.com/) + [Sourcetree](https://www.sourcetreeapp.com/)/[GitHub Desktop](https://desktop.github.com/download/)
2. Робота з GitHub:
   * Проаналізуйте документацію за посиланням [Git](https://git-scm.com/book/uk/v2)
   * !!! особливо розібратися з [**цим розділом**](https://git-scm.com/book/uk/v2/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%B8-Git-%D0%A1%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F-Git-%D1%80%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D1%96%D1%8F) !!!
   * Додатково, можна почитати [статтю](https://www.theodinproject.com/lessons/foundations-git-basics) (on English)
   * Зареєструвати [**GitHub**](https://github.com/signup?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home) акаунт на університетську електронну адресу (ваша електронна адреса домену uzhnu.edu.ua)
   * Ім’я користувача (nickname) встановити зрозумілим, на кшталт rbuchuk, romanbuchuk, buchukroman…Тобто повинно складатися Вашого імені та прізвища.
   * Створити два репозиторії (онлайн: у вашому GitHub), один з назвою **web\_basics,** другий – [назва вашого проєкту]
   * Додайте до ваших репозиторіїв клоабораторів ***romanbuchuk*** та **saibertuzhnu**
   * Локально (на вашому ПК) створити директорію (папку), яка міститиме Ваші лабораторні роботи та проєкт.
   * У консольному застосунку (cmd для Windows / terminal для Mac) перейти у вашу директоріюA screenshot of a computer

     Description automatically generated

*На скріншоті моя директорія* ***test***

* + Скопіювати посилання новоствореного репозиторію **web\_basics** в буфер обміну **(**може бути HTTPS або SSH посилання)A blue box with black text

    Description automatically generated
  + Клонувати вибраний репозиторій (в консолі виконати команду **git clone** <url>). Де замість <url> вставити щойно скопійоване посилання на репозиторій. Більш детальну інформацію клонування можна знайти в документації, яку Ви [прочитали вище](https://git-scm.com/book/uk/v2/______-Git-_________-Git-___________)
  + Перейти в папку з вашим проектом та додати [git remote](https://git-scm.com/docs/git-remote) (відповідну інструкцію Ви могли бачити коли створили новий репозиторій на GitHub)A screenshot of a computer

    Description automatically generated  
      
    Ціль даного завдання спробувати кілька варіантів роботи з репозиторіями, як з існуючим так і створення чи клонування нового.

1. Якщо у Вас існує папка з лабораторною роботою 1, перенесіть її чи скопіюйте в папку **web\_basics** на вашому ПК
2. Використовуючи набуті знання, зробити коміт та залити зміни на відповідний репозиторій (**необхідно використати команди git add, git pull, git commit та git push**)
3. Створіть у **web\_basics** папку “**Lab2\_HTML**”.
4. Створіть 2 html файли (**index.html, author.html**) що матимуть базову розмітку:

<html lang=ua>

  <head>

    <meta charset=utf-8>

   …

  </head>

  <body>

    …

  </body>

</html>

Обидва файли повинні використовувати семантичну коректну розмітку (**параграф, заголовок першого рівня, другого рівня, курсив, жирний, зображення, списки, посилання, розрив рядка** якщо дійсно необхідно).

**Інструкції щодо першого файлу: index.html:**

index.html повинен виглядати наступним чином (зверніть увагу на заголовок таби):

A screenshot of a computer

Description automatically generated

Якщо є бажання, можна додати ще додаткову інформацію, використовуючи відомі Вам таги. Єдина вимога – документ повинен симантично бути коректно написаний.

1. !!! Перевірити, чи все симантично виконано коректно, використавши для цього [ресурс](https://validator.w3.org/#validate_by_input) !!!

Файл index міститиме 2 посилання: перше - на сторінку “Про автора” а друге - на зовнішній ресурс - сайт УжНУ (при кліку повинен відкритись у новому вікні).

**Інструкції щодо другого файлу: author.html:**

Файл author міститиме 1 посилання на головну сторінку (index.html). Крім того треба використати 2 зображення: на перше встановити ширину не більше 300px (краще - своє фото або довільне зображення), а друге - шириною не більше 32px іконка будиночка (<https://upload.wikimedia.org/wikipedia/commons/b/ba/Home_icon_orange.png> ).

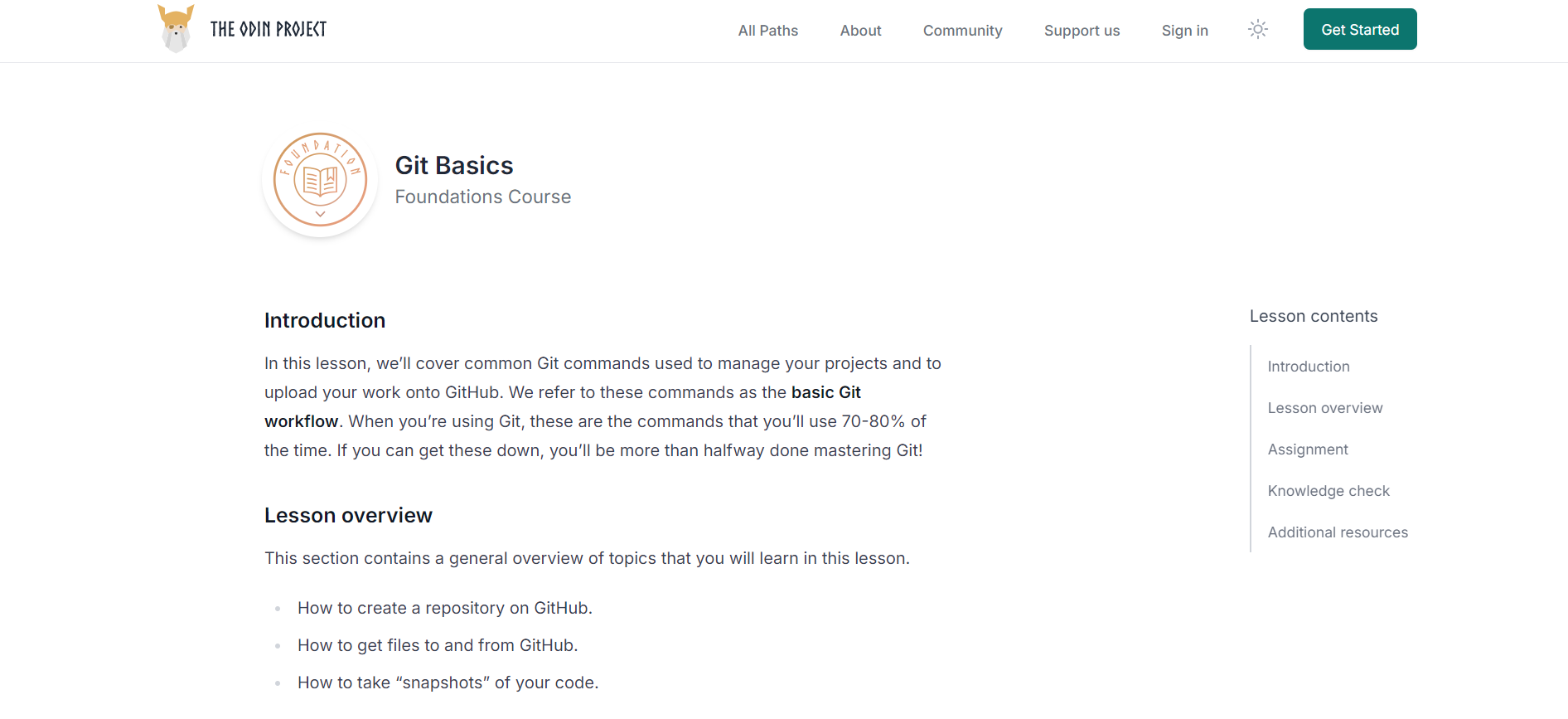
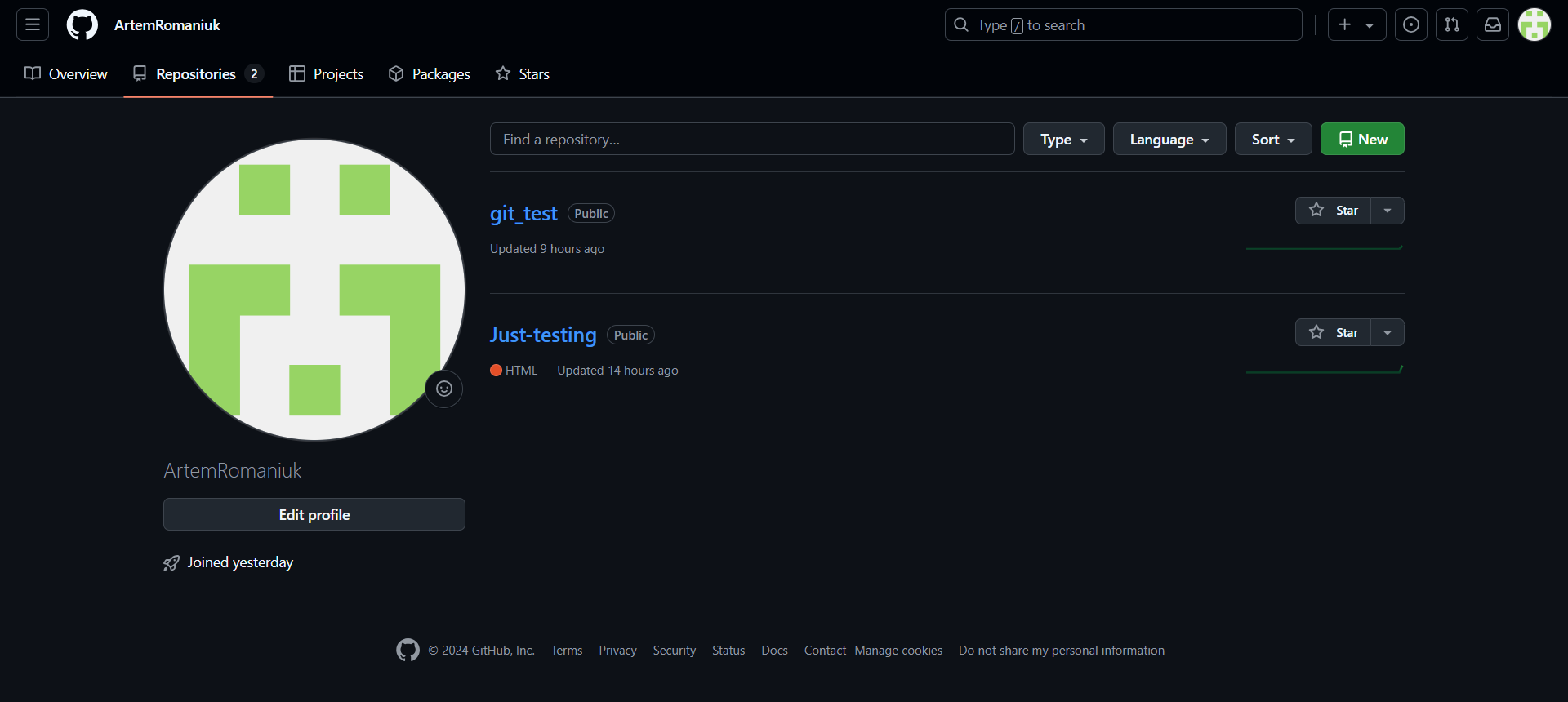
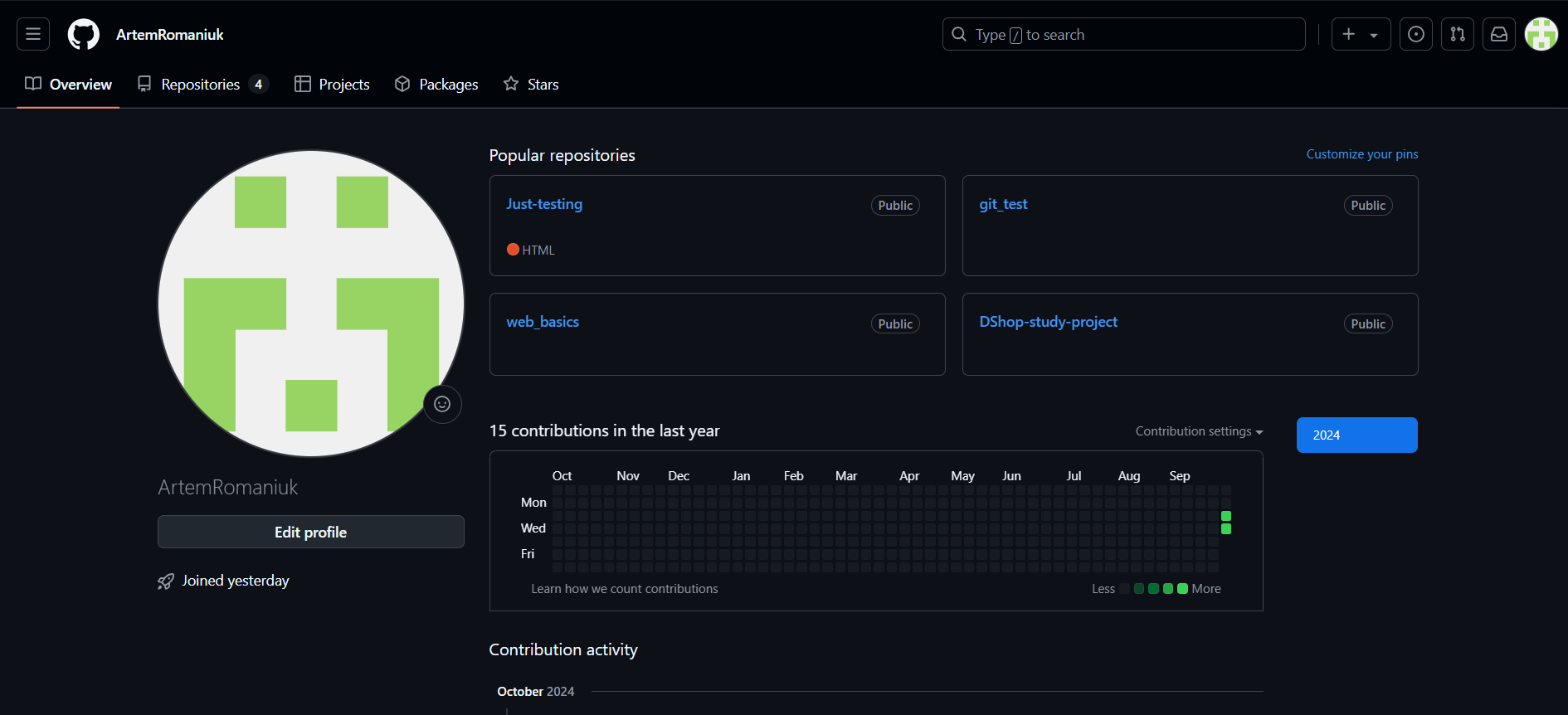
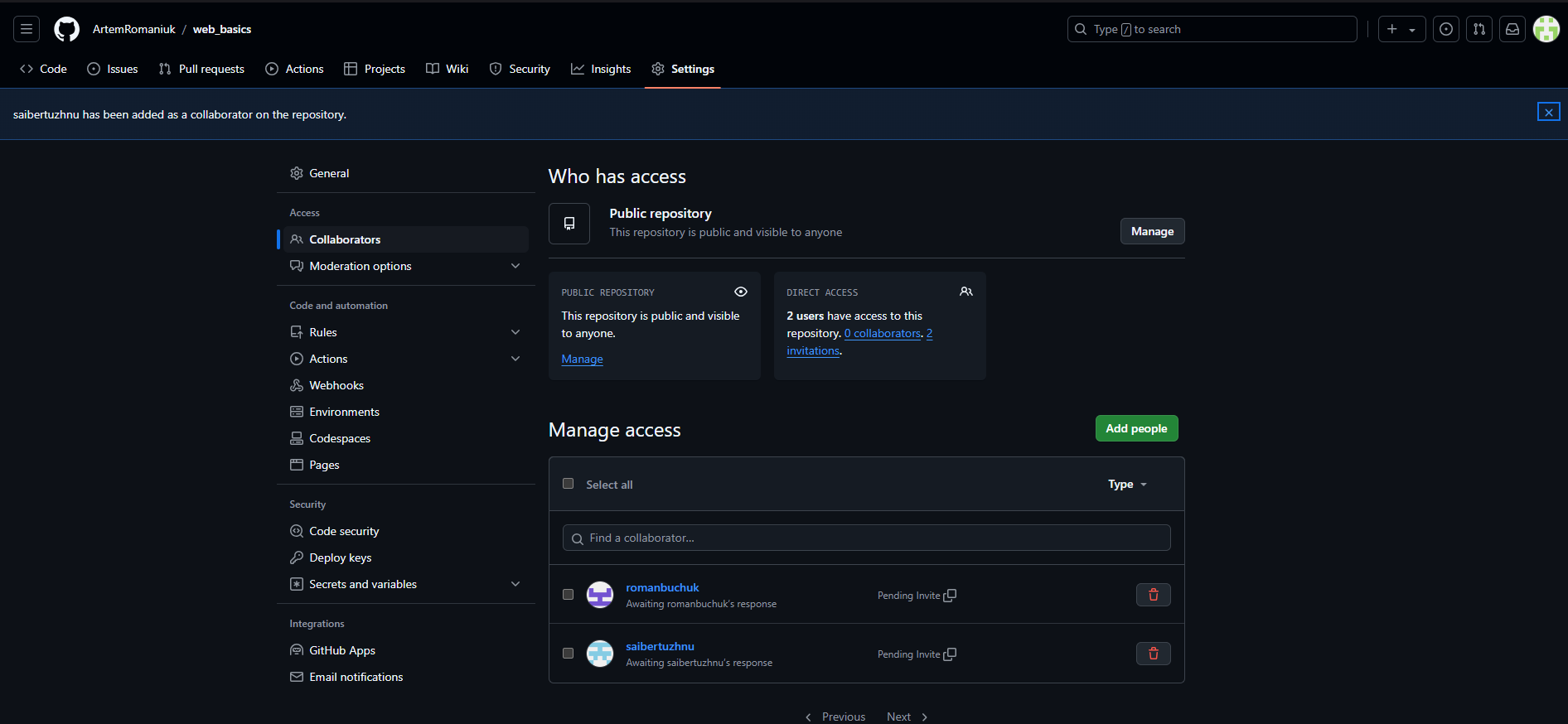
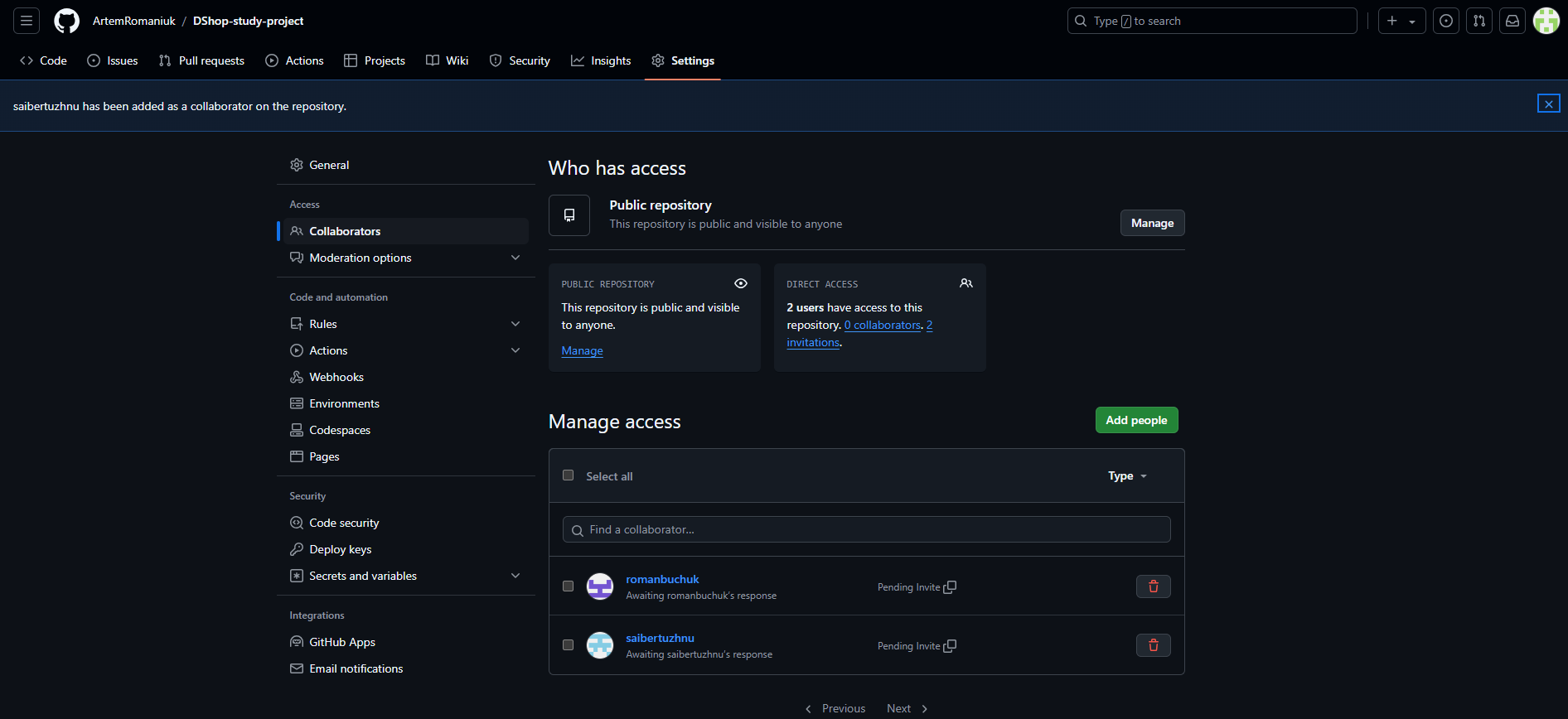
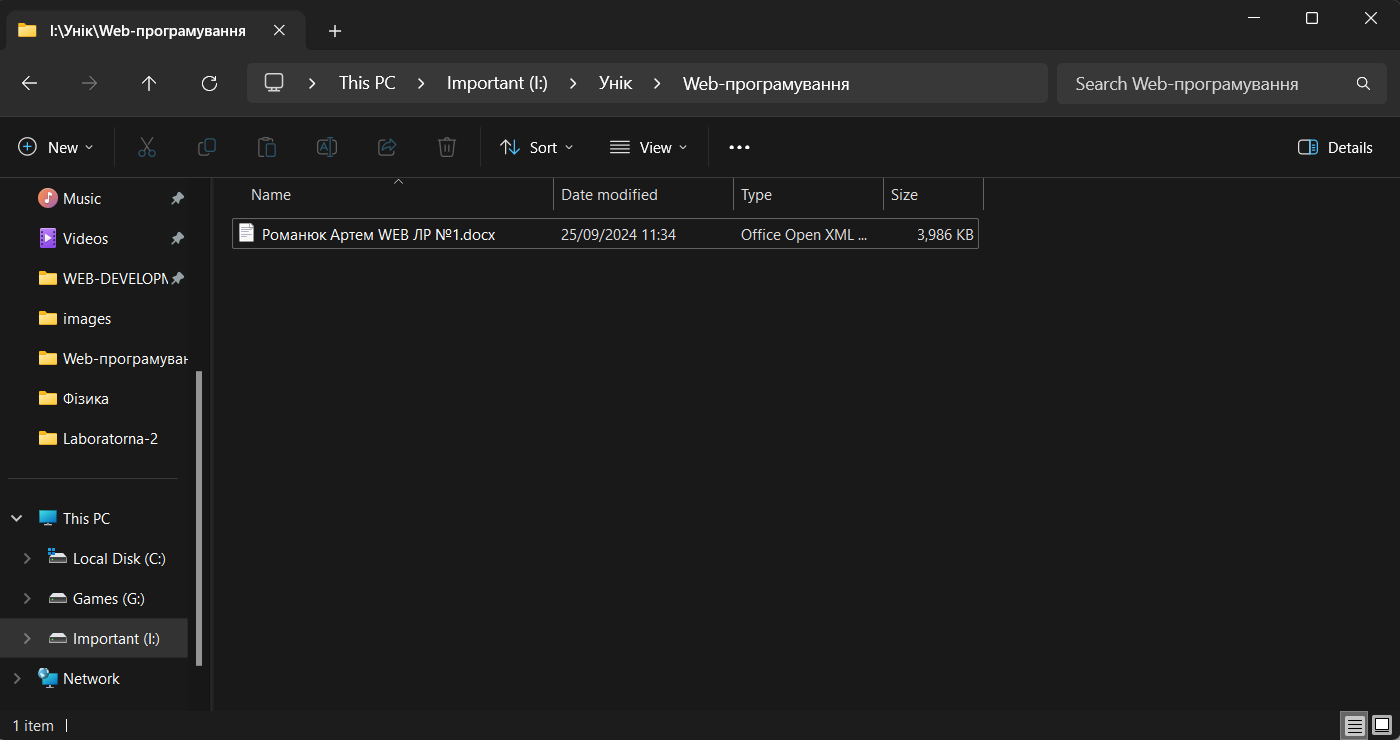
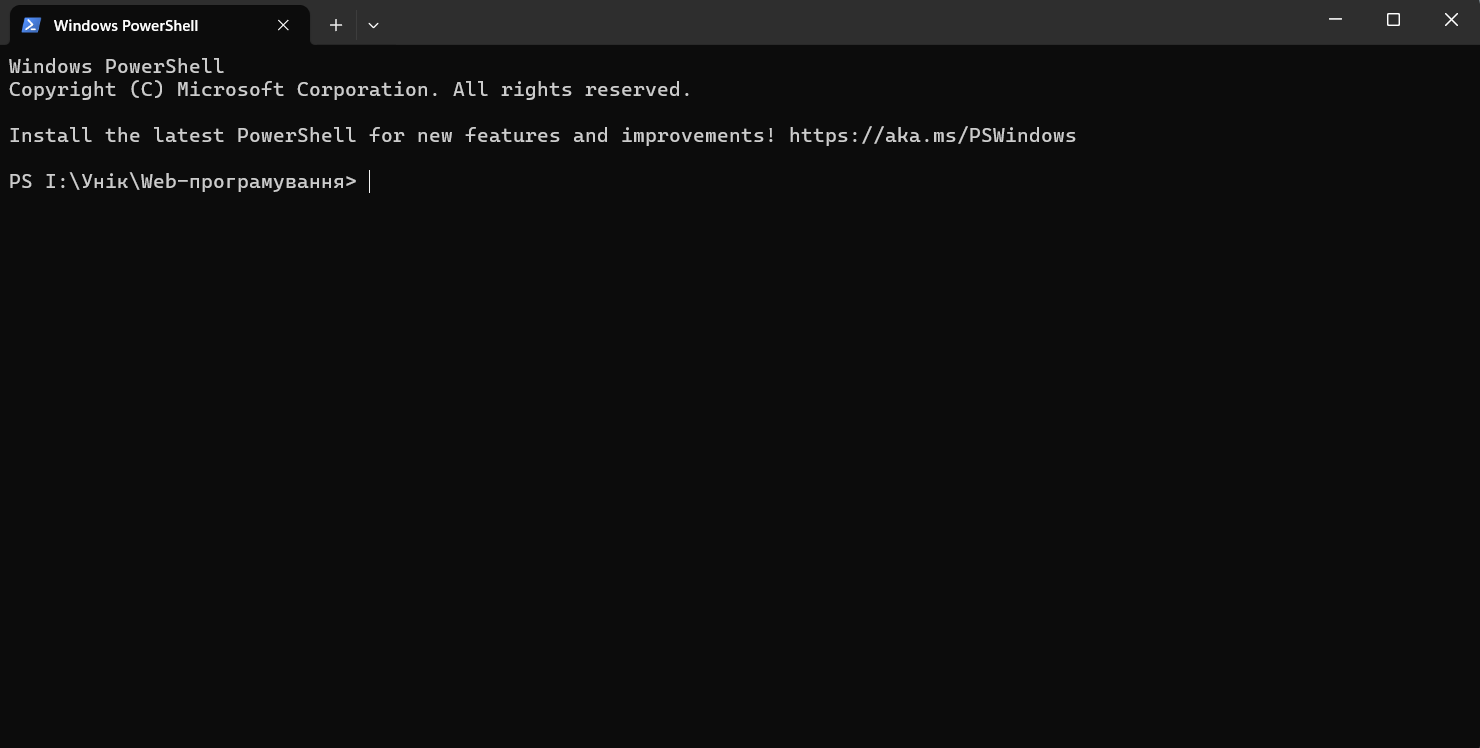
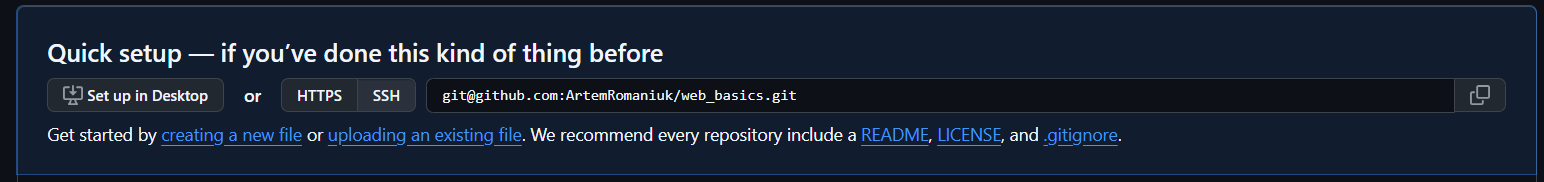
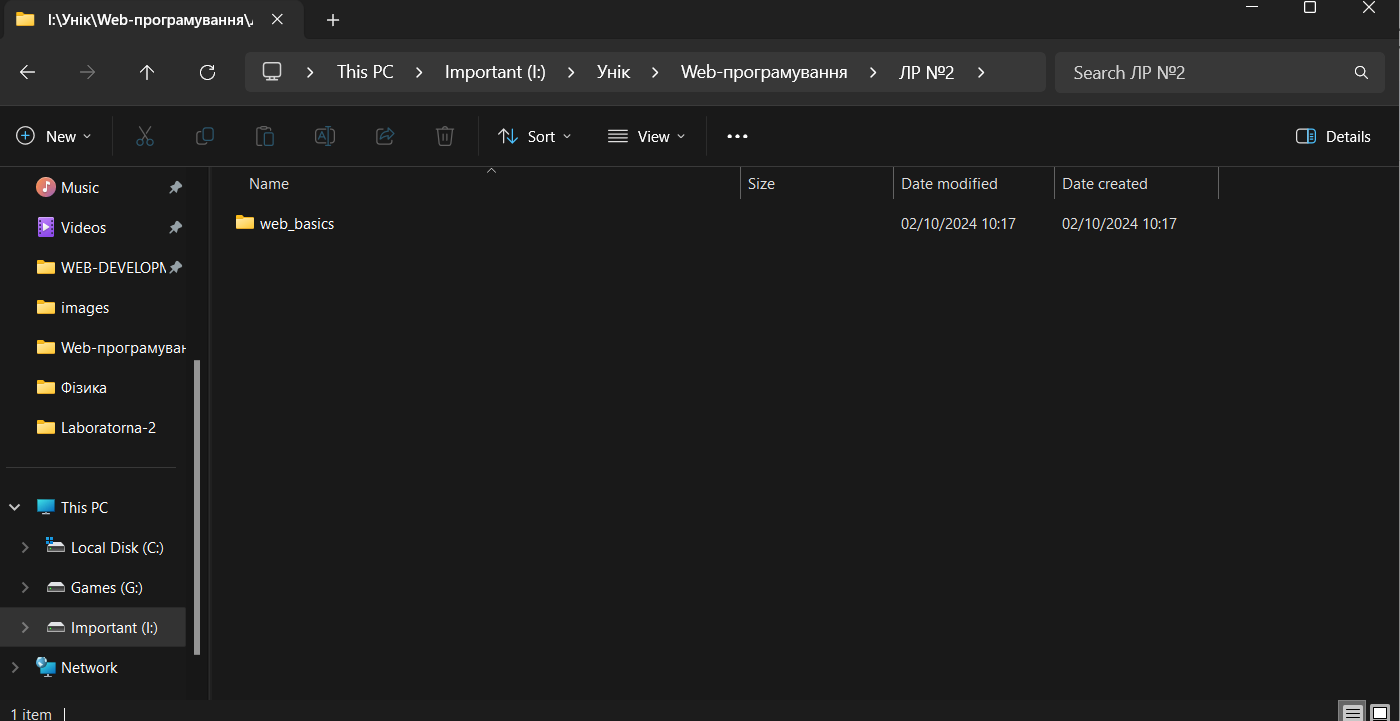
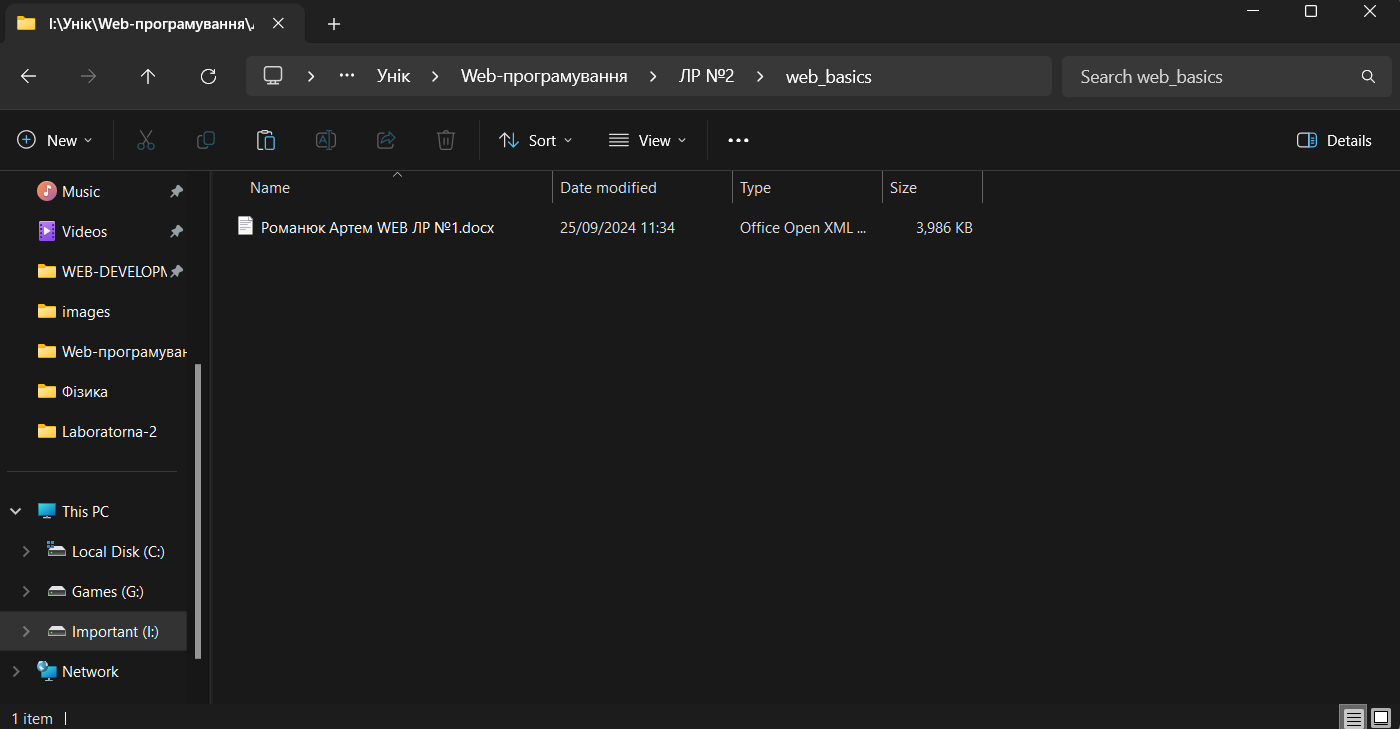
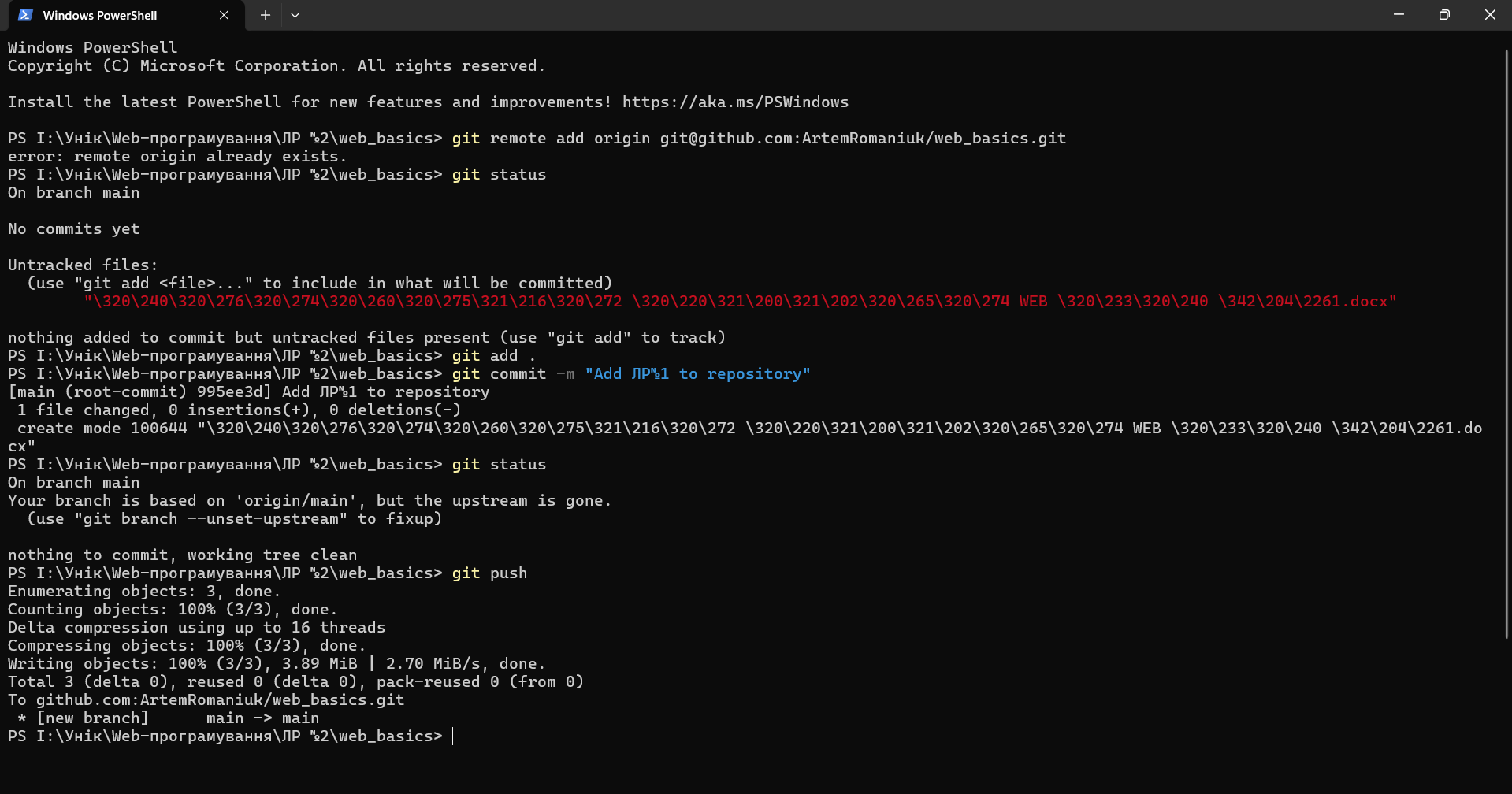
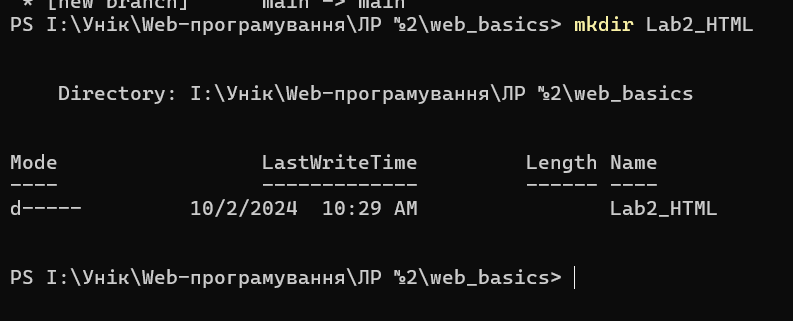
*Підказка: для зображень можна додати атрибут width=300 для встановлення ширини*

author.html повинен виглядати наступним чином (зверніть увагу на заголовок таби):A child writing on a book

Description automatically generated

1. Додайте кілька коментарів до HTML-розмітки щоб позначити умовні  розділи розмітки.
2. Зробіть так, щоб при наведенні миші на іконку будиночка показувався текст “Перейти на головну сторінку”.
3. Додайте до документів зовнішні стилі у файлі styles.css, можете змінити колір тексту, розташування.
4. Результат роботи закомітити та опублікувати зміни (зробити пуш) (як у попередніх пунктах за допомогою git)
5. Оформити звіт роботи. Зробити відповідні скріншоти (на Ваш вибір, щоб ефективно представити звіт Вашої проробленої роботи), скріншоти вставляти у звіт.

**Хід роботи**

1. Проаналізував офіційну документацію GIT  
   
2. Додатково почитав сторонню статтю  
   
3. Зареєстрував **Github** аккаунт на університетську електронну адресу  
   
4. Створив два репозиторії, один з назвою **web\_basics,** другий – **DShop-study-project**  
   
5. Додав до своїх репозиторіїв клоабораторів ***romanbuchuk*** та **saibertuzhnu  
     
   **
6. Локально створив директорію, яка міститиме лабораторні роботи та проект  
   
7. У консольному застосунку перейшов у новостворену директорію  
   
8. Скопіював посилання новоствореного репозиторію **web\_basics** в буфер обміну  
   
9. Клонував вибраний репозиторій  
   
10. Скопіював звіт ЛР №1 в папку **web\_basics**  
    
11. Використовуючи набуті знання, зробив коміт та залив зміни на відповідний репозиторій  
    
12. Створив у **web\_basics** папку “**Lab2\_HTML**”  
    

13. Створив 2 html файли (**index.html, author.html**) що мають базову розмітку:  
  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

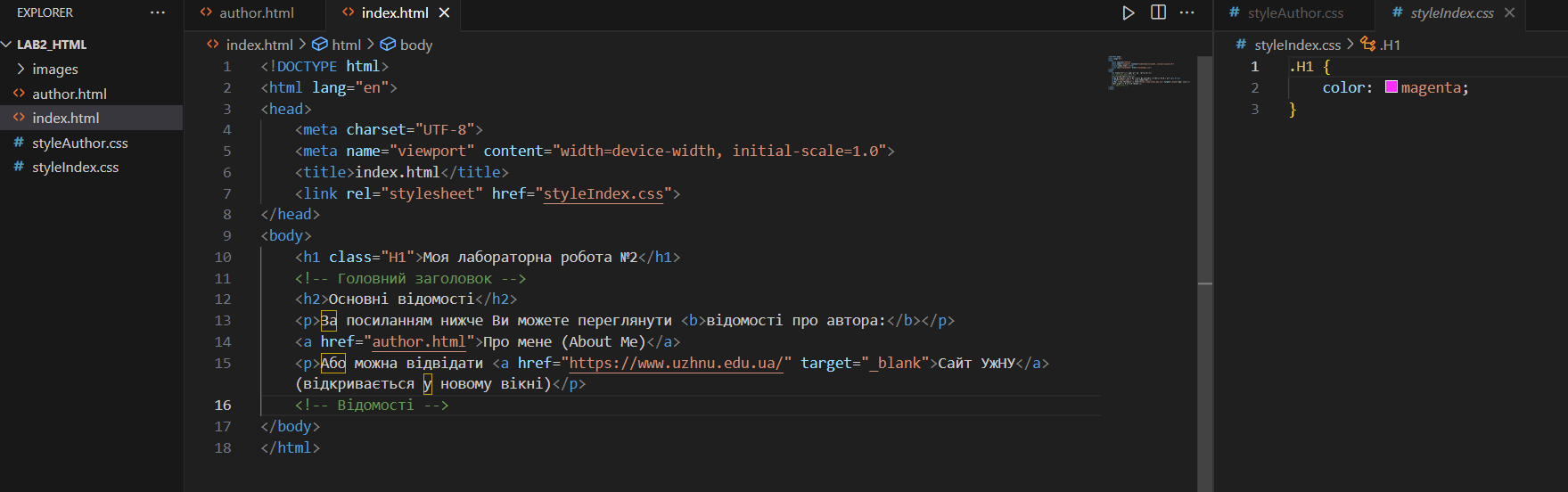
<title>index.html</title>

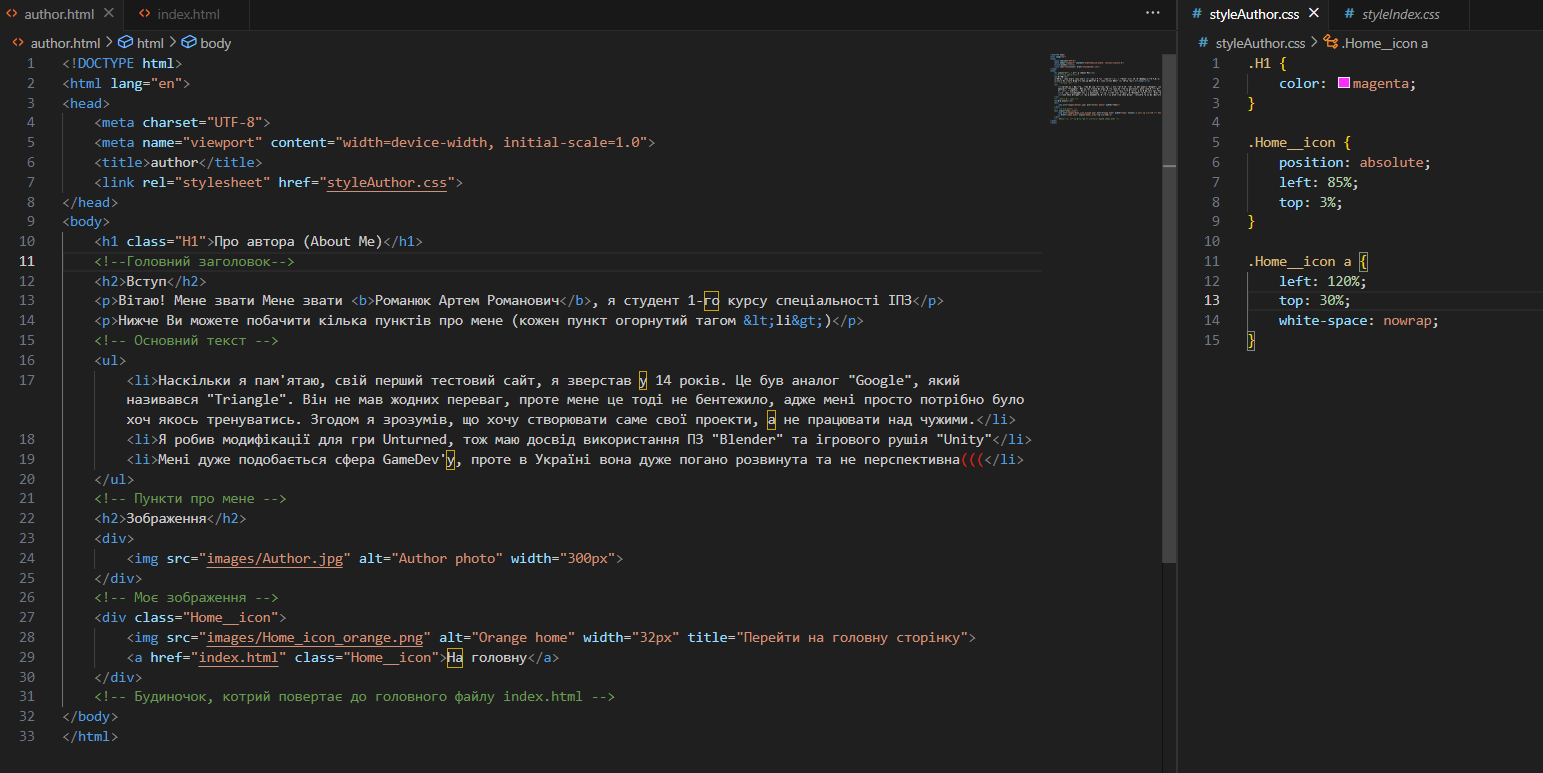
</head>

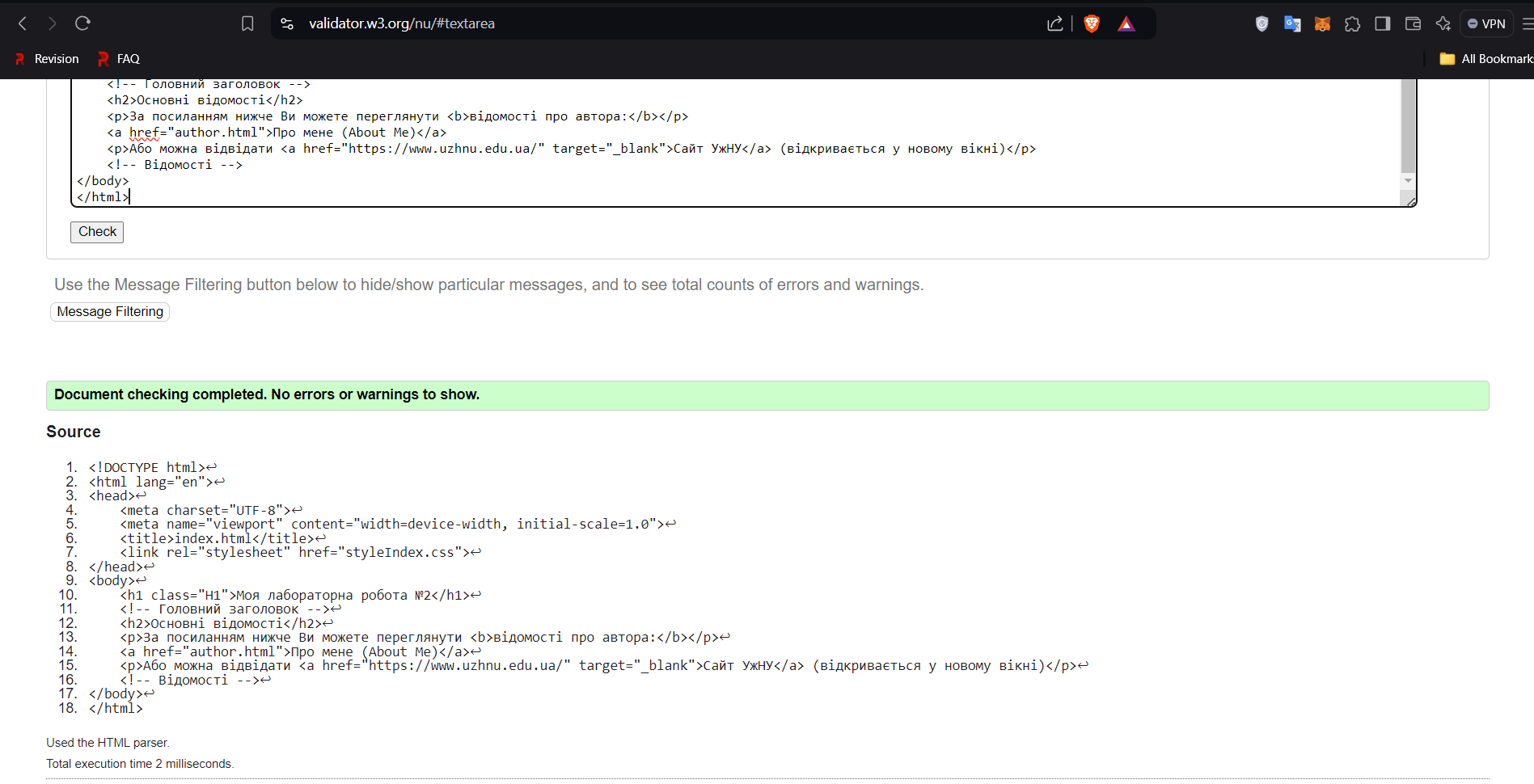
<body>

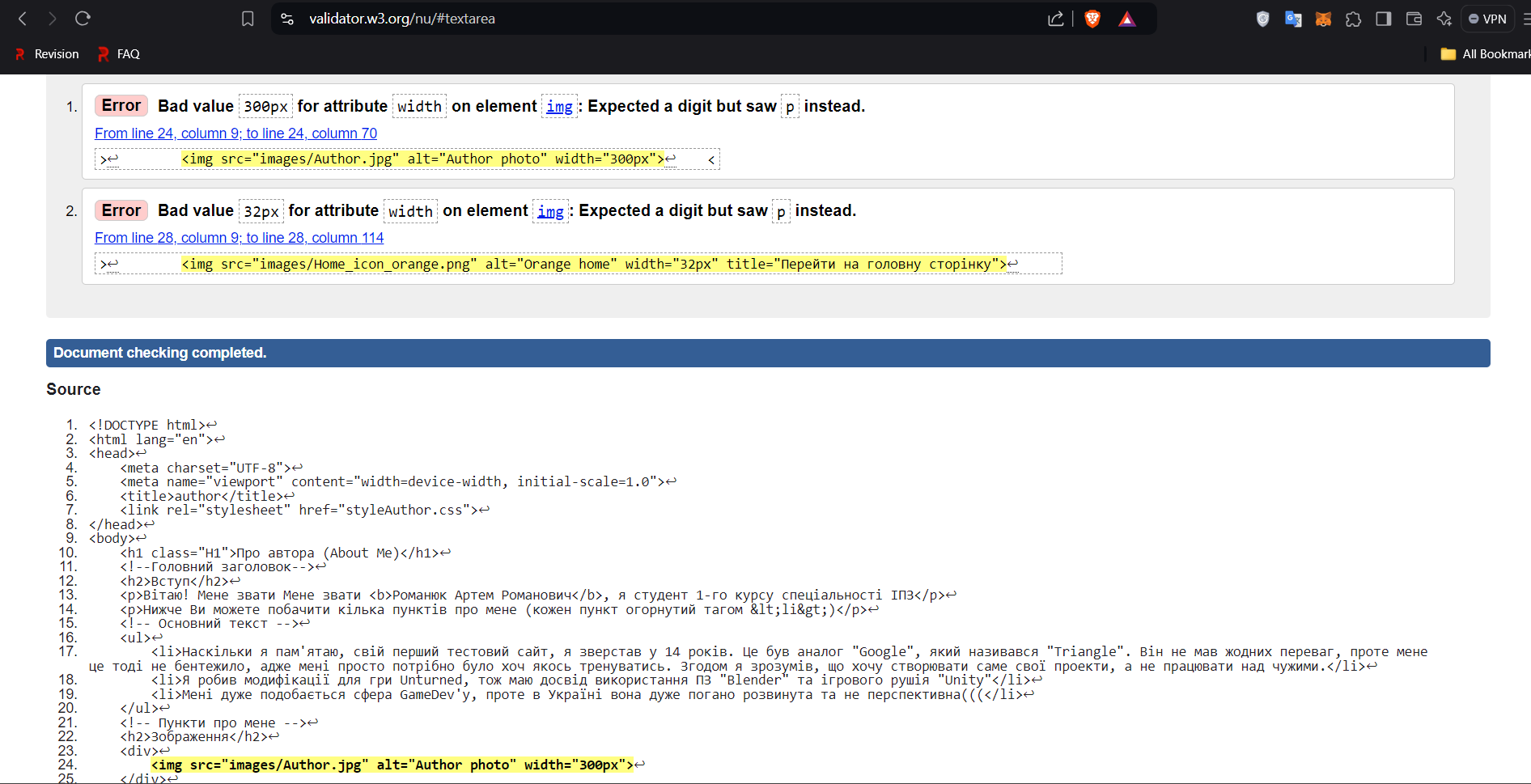
</body>

</html>



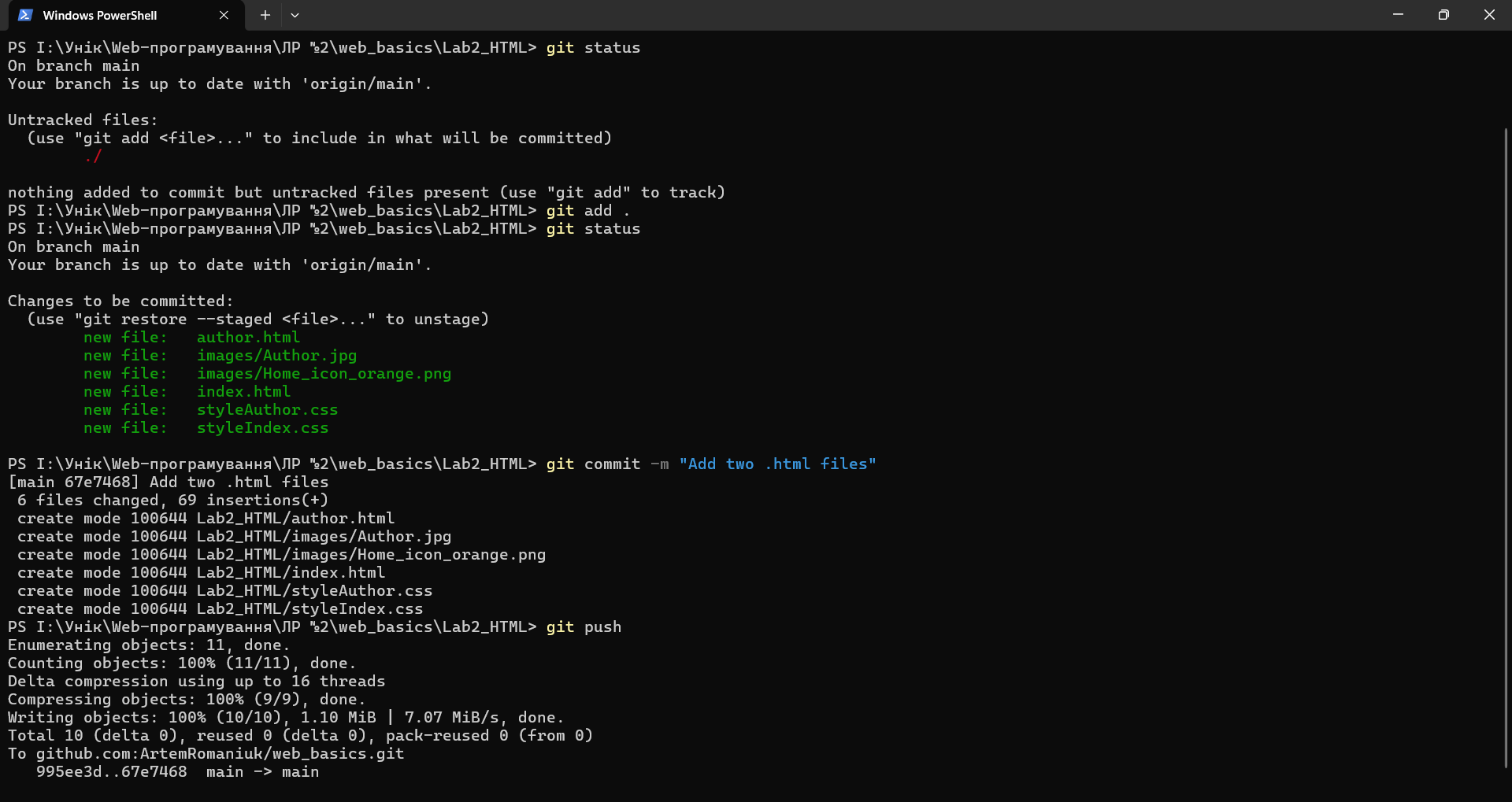


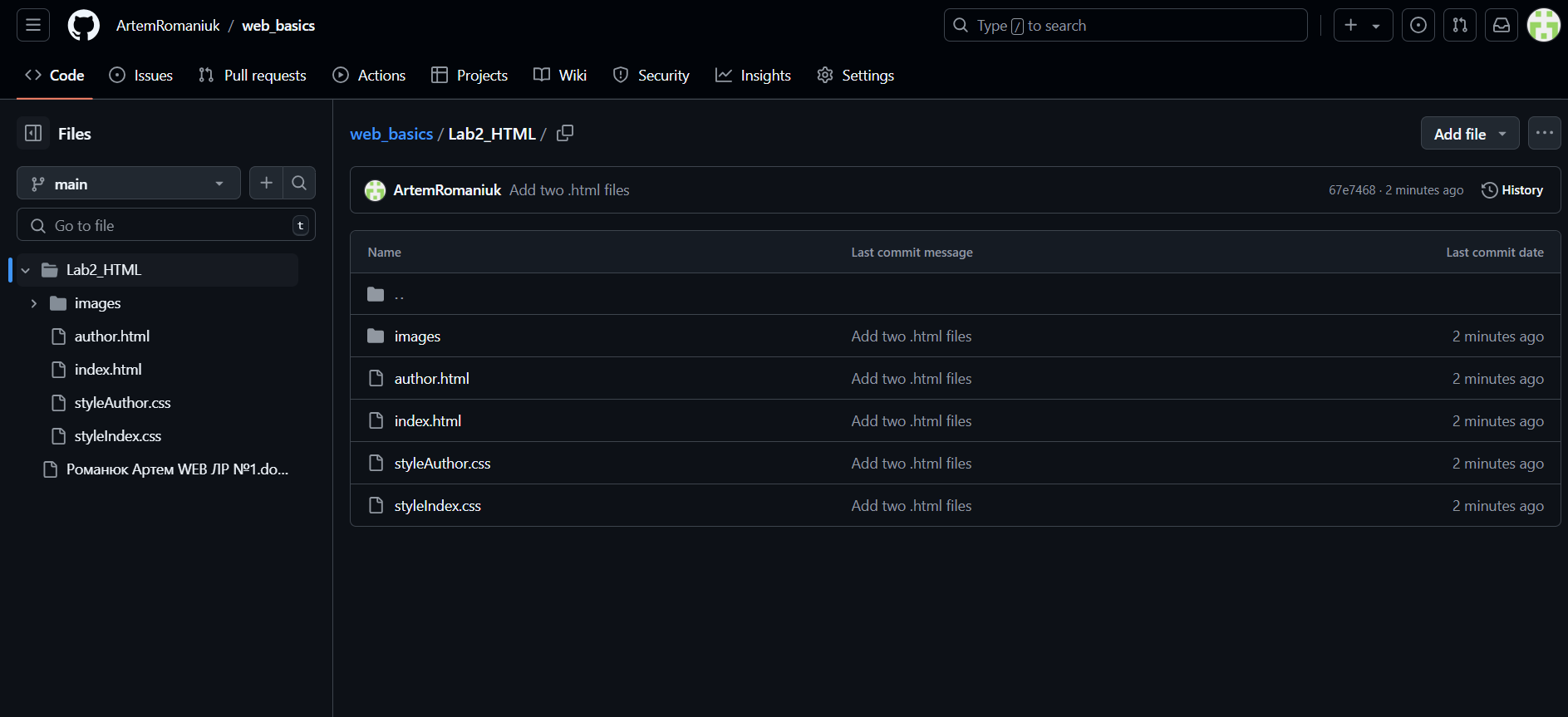
14. Перевірив чи все семантично виконано коректно, використавши для цього [ресурс](https://validator.w3.org/#validate_by_input)  




Веб-ресурс знайшов помилки у файлі **author.html**, проте значення ширини картинки були задані в умові, тому фактично помилок немає

15. Результати роботи закомітив та опублікував на Github





**Висновок:**

В ході даної лабораторної роботи було опрацьовано теоретичний матеріал по системі керування версіями файлів та спільної роботи - **GIT**.

Для роботи з **GIT** використовували термінали Windows PowerShell, Git Bash та графічну інтеграцію **GIT** у **Visual Studio Code**. Для збереження файлів використовували веб-сервіс **Github**.  
 Загалом, ми навчилися користуватися найпопулярнішою системою керування версіями файлів - **GIT** та веб-сервісом **Github**. Розглянули **можливості та функціонал**:

1. Створення, клонування та додавання репозиторіїв за допомогою консольних команд **git init**, **git clone (Посилання з github)** та **git remote** відповідно

2. Додавання та коміт файлів - **git add (файл)** та **git commit -m “повідомлення”**

3. Перевірка статусу репозиторію та перегляд логів - **git status** та **git log**

4. Передання файлів на репозиторій - **git push**

5. Команда для відображення, створення або видалення гілок - **git branch**

6. Команда для поєднання гілки в активну - **git merge**

7. та створення гілок та перемикання між ними - **git checkout (або git switch)**