**Лабораторна робота № 2**

**Каскадні таблиці стилів**

**Мета роботи** – навчитися створювати власні таблиці стилів.

**Завдання.** Розробити декілька пов‘язаних між собою веб сторінок відповідно до варіанту роботи. Оформлення веб сторінок описати у окремих CSS файлах.

**Порядок виконання роботи**

* Створити головну сторінку з параметрами відповідно до варіанту (таблиця 1.1).
* Використовуючи теги заголовка H1 і H2 вивести у верхній частині документа відповідно назва дисципліни і номер лабораторної роботи з вирівнюванням по центру.
* За допомогою тегів списків DL, UL і OL вивести список, який включає підсписок відповідно до варіанту(таблиця 1.2).
* Один з елементів списку зробити посиланням на нову сторінку.
* Створити нову сторінку з описом окремого пункту зі списку. У верхній частині документа поставити посилання на першу сторінку. Внизу другої сторінку поставити посилання «початок», що веде на початок документа.
* Вставити до другої сторінки два зображення та зробити їм обтікання по лівому та правом краю.
* Одне з зображень перетворити на посилання на третю сторінку.
* Вставити до другої сторінки текст з трьох абзаців які оформлені відповідно до варіанту (табл1.3).
* Зробити малюнок фоновим зображенням у третьої сторінки.
* Створити для кожної з трьох веб-сторінок власті файли зі стильовим оформленням та підключити їх.
* Разом з результатами навести результати валідації, у результатах не повинно бути помилок, можливо лише декілька попереджень.

**Теоретичні відомості**

**4.1 СSS основи**

CSS (Cascading Style Sheets), або каскадні таблиці стилів, описують правила форматування окремого елемента веб-сторінки. Створивши стиль один раз, його можна застосовувати до будь елементів сторінки скільки завгодно разів.

Визначення стилю складається з двох основних частин: самого елемента веб-сторінки - селектора, і команди форматування - блоку оголошення.

Селектор повідомляє браузеру, який саме елемент форматувати, в блоці оголошення перелічено властивості команд форматування і їх значення (рис. 4.1).



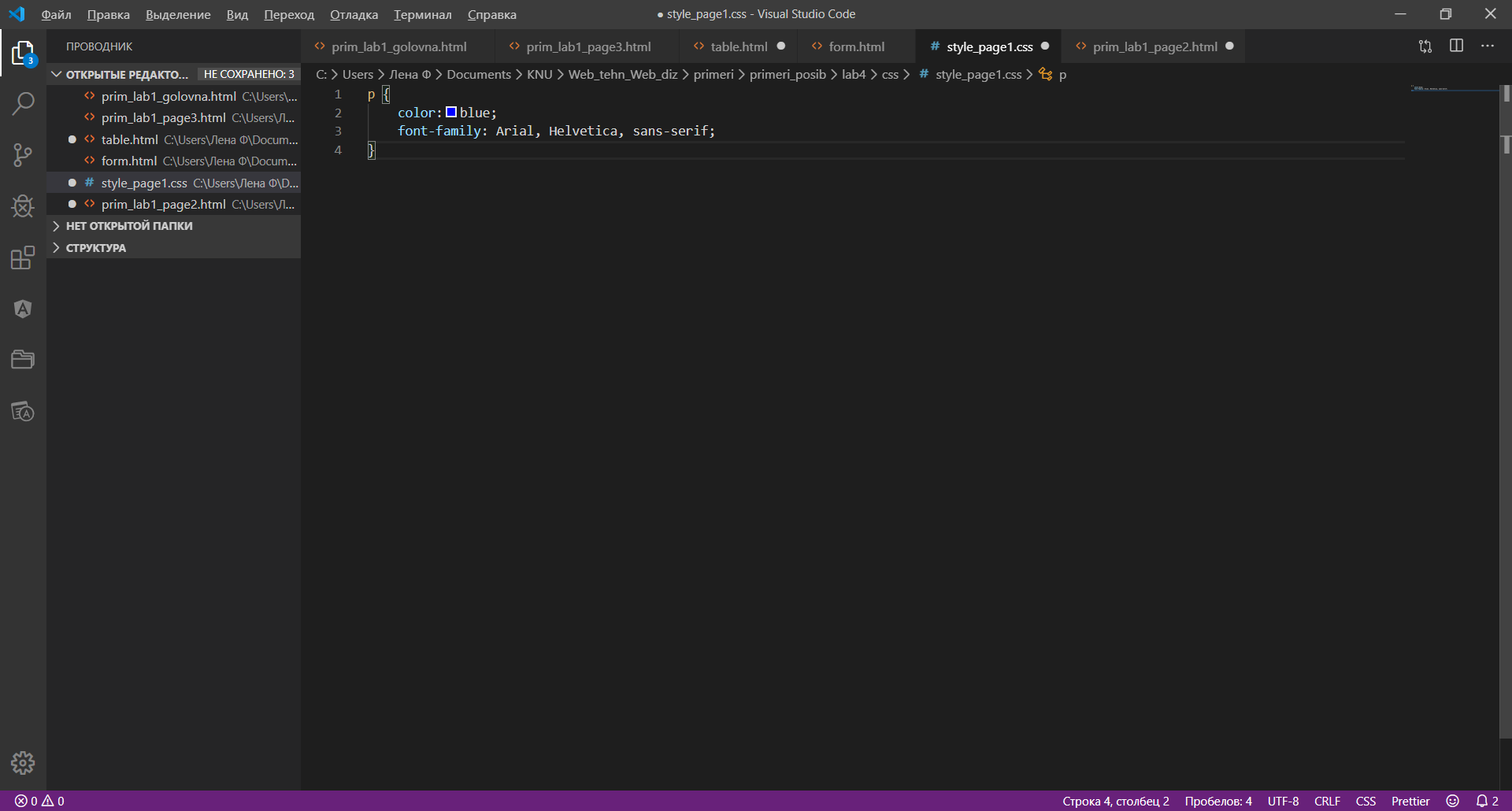


Рисунок 4.1 Приклад CSS

Селектор *p* означає, що дані стилі застосовуються для всіх тегів *<p>* у html документі будуть встановлене визначене форматування, а саме властивість *color* встановлює колір тексту у значення *blue*, а властивість *font-family* встановлює перелік значень шрифтів.

Таблиці стилів до у html документу можуть підключатися наступним чином:

* Вбудовані таблиці стилів. Представляють набір стилів, що є частиною коду веб-сторінки, знаходяться між тегами *<style> ... </ style>* і розташовані всередині елемента *<head>*. Вбудовані стилі мають пріоритет перед глобальними, але поступаються стилям, що вказані в тегу. На одній сторінці можна розміщувати довільну кількість вбудованих стилів
* Внутрішні текстові таблиці стилів. Не використовують селектори, присвоєння стилю відбувається безпосередньо до html-елементу через атрибут *style*. Мають найвищий пріорітет.
* Правило @import. Дозволяє завантажити зовнішню таблицю стилів. Щоб директива @import працювала, вона повинна розташовуватися всередині тегу *<style>* перед іншими правилами.
* Тег *<link>*. Зовнішня таблиця стилів представляє текстовий файл з розширенням .css, в якому знаходиться весь набір стилів CSS. Він не містить HTML-код, тому його не потрібно вкладати всередину тегів *<style> ... </ style>.* Задані в файлі стилі будуть працювати для всіх сторінок веб-сайту.
  1. **Селектори**

За допомогою селекторів створюються CSS-правила для форматування елементів веб-сторінки. В якості селекторів використовуються не тільки самі елементи та їхні класи і ідентифікатори, а також псевдокласи і псевдоелементи.

Розглянемо більш докладніше основні види селекторів:

* Універсальний селектор. Універсальний селектор відповідає будь-якому елементу, наприклад, наступний запис усім відступам для усіх елементів веб-сайту присвоїть значення 0.

*\*{Margin: 0;}*

* Селектор Елемента. Селектори елементів використовуються для визначення стилів елементів для всіх сторінок веб-сайту, наприклад, стиль заголовків *h1* або загальний стиль абзаців(рис.4.1):
* Селектор класу. Селектори класу використовуються для визначення стилів для кількох елементів одного типу, розміщених у різних частинах або на різних сторінках веб-сайту. Перед назвою класу у css файлі ставиться крапка, а сам клас підключається до тегу за допомогою атрибуту *class*.
* Селектор ідентифікатора. Селектори ідентифікатора використовуються для привласнення стилю одному конкретному елементу. Ідентифікатор *id* елемента можна використовувати в документі лише один раз, оскільки він виділяє унікальний елемент. Перед ідентифікатор ставиться знак #.
* Селектор нащадка. До нащадків елемента відносяться його дочірні елементи. Селектори нащадків дозволяють стилізувати всі вкладені елементи, які перераховуються через проміжок.
* Дочірній селектор. Дочірній тег є прямим нащадком тегу, що його містить. Тобто, відносини "батьки-діти" існують між елементами і тими елементами, які містяться безпосередньо всередині них. В одного елемента може бути кілька дочірніх елементів, а батьківський елемент може бути в кожного елемента тільки один. Дочірній елемент вказується через знак >.
* Сестринський селектор. Сестринські відносини виникають між елементами, що мають загального батька. Селектори сестринських елементів дозволяють вибрати теги з групи елементів одного рівня. Для запису правил з використанням сестринських елементів використовуються знаки + та ~.

Селектор Атрибута. Селектори атрибутів дозволяють форматувати елементи на основі вибірки атрибутів або значень атрибутів, атрибути записуються наступним чином:

*селектор [атрибут]*

* Псевдоклас. Псевдокласи - це класи, що фактично не прикріплені до тегів html-коду. Вони викликають CSS-правила при вчиненні тої чи іншої події або підкоряються певному правилу.

Також усі селектори можна комбінувати та групувати. При групуванні селектори записуються через кому.

* 1. **Успадкування та каскад**

Успадкування та каскад є двома фундаментальними поняттями мови CSS, які тісно пов‘язані між собою. Успадкування, це принцип за яким елементи успадковують css властивості своїх батьківських елементів, тобто елементів в які вони вкладені. Каскад визначає принцип за яким різні таблиці стилів застосовуються до html документу та як правила, що конфліктують перевизначають один одного.

Каскадування - це механізм, який керує кінцевим результатом в ситуації, коли до одного елементу застосовуються різні CSS-правила. Існує три критерії, які визначають порядок застосування властивостей:

* Правило !important. Вагу будь якого css правила можна задати за допомогою ключового слова *!important*, яке додається відразу після значення властивості. Правило необхідно розміщувати в кінець оголошення перед закриваючою фігурною дужкою, без пробілу. Таке оголошення буде мати пріоритет над усіма іншими правилами.
* Специфічність. Для кожного правила браузер обчислює специфічність селектора, і якщо у елемента є конфліктуючі оголошення властивостей, до уваги береться правило, має найбільшу специфічність. Значення специфічності складається з чотирьох частин: 0, 0, 0, 0. Специфічність селектора визначається наступним чином:
* для *id* додається 0, 1, 0, 0;
* для *class* додається 0, 0, 1, 0;
* для кожного елемента і псевдоелемента додається 0, 0, 0, 1;
* для вбудованого стилю, доданого безпосередньо до елементу - 1, 0, 0, 0;
* універсальний селектор не має специфічності.

В результаті до елементу застосуються ті правила, специфічність яких більше.

* Порядок підключених таблиць. Якщо в різних таблицях будуть зустрічатися різні значення властивостей одного елемента, то в результаті до елементу, застосовується правило, що знаходиться в таблиці стилів, що йде в списку нижче.

**Приклад виконання**

**Завдання**

Розробимо веб-застосунок для виконання наступного завдання:

Створити класифікатор казкових істот відповідно до книг Джоан К. Роулінг, та

описати більш докладніше одну з них. Колір фону-блідо рожевий, колір тексту –

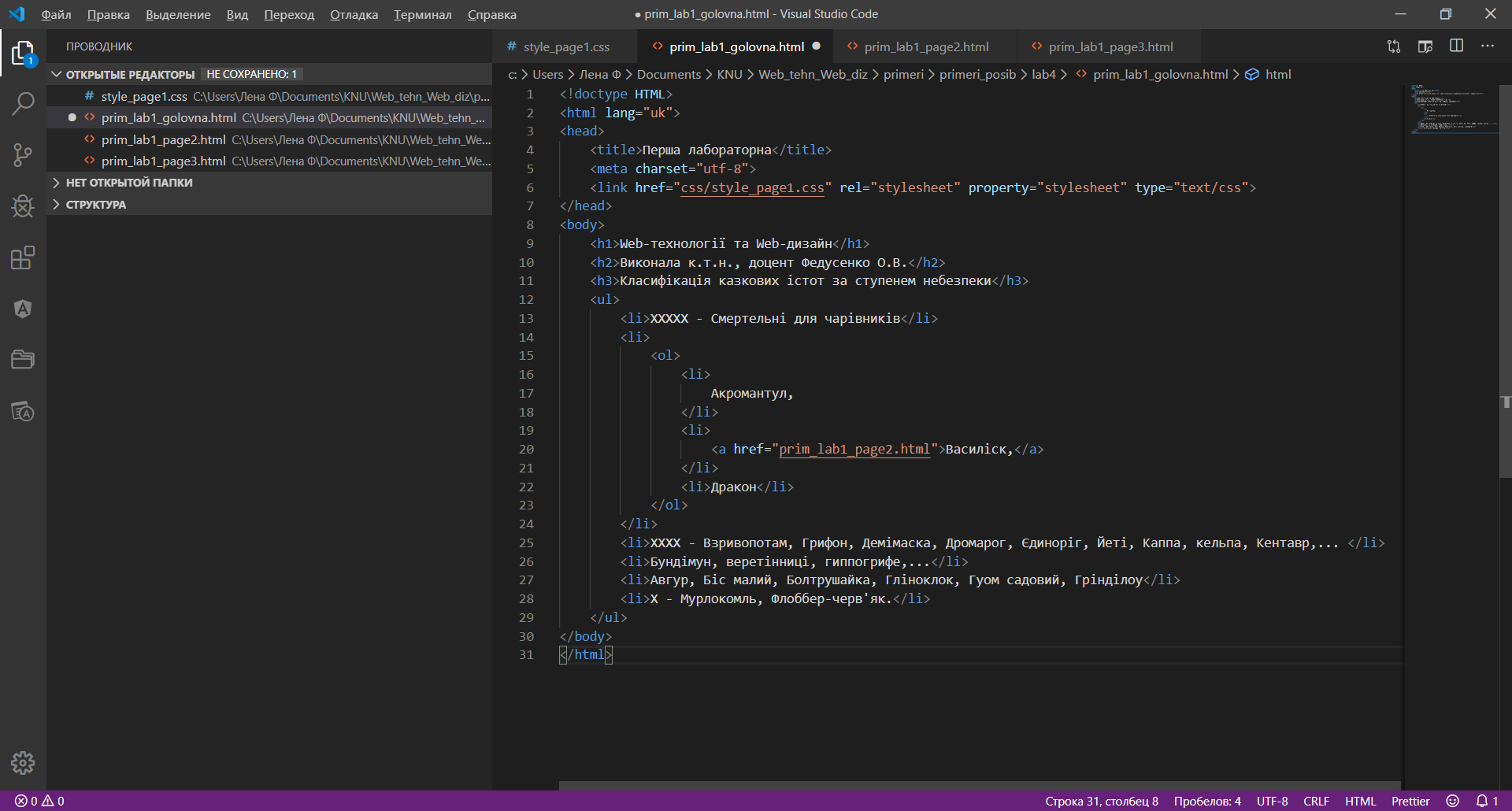
синій, колір посилань – зелений.

Оформлення абзаців другої сторінки: перший абзац – напівжирний колір

зелений, другий підкреслений курсив червоний колір, третій – сьоме слово

виділити жирним.

1. Спочатку розробимо першу сторінку, а вже потім напишемо її стильове оформлення у файлі css. Вона буде містити ненумерований список, заголовок першого рівня та заголовок другого рівня та гіперпосилання на другу сторінку.. Головна сторінка наведена на рис.4.2.
2. Тепер створимо окремий css файл, для опису усіх стилів головної сторінки, він буде виглядати наступним чином рис. 4.2. Для того, щоб таблиця стилів запрацювала необхідно підключити її до html файлу за допомогою тегу *<link>*(рис. 4.2). Як ми бачимо зовнішній вигляд головної сторінки не змінився (рис. 4.3), але тепер вона проходить валідування без помилок та попереджень.



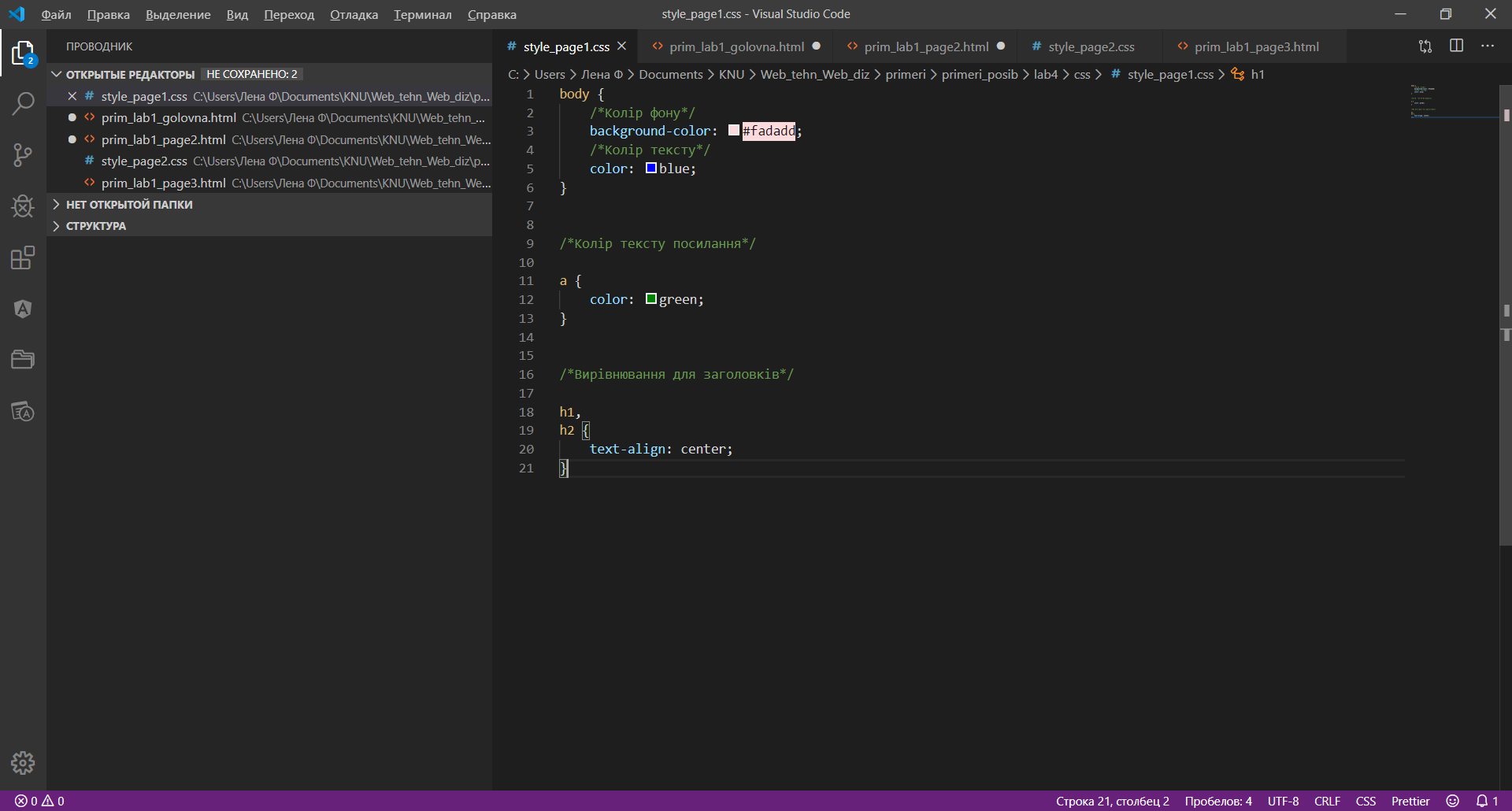


Рисунок 4.2 Html та css першої сторінки

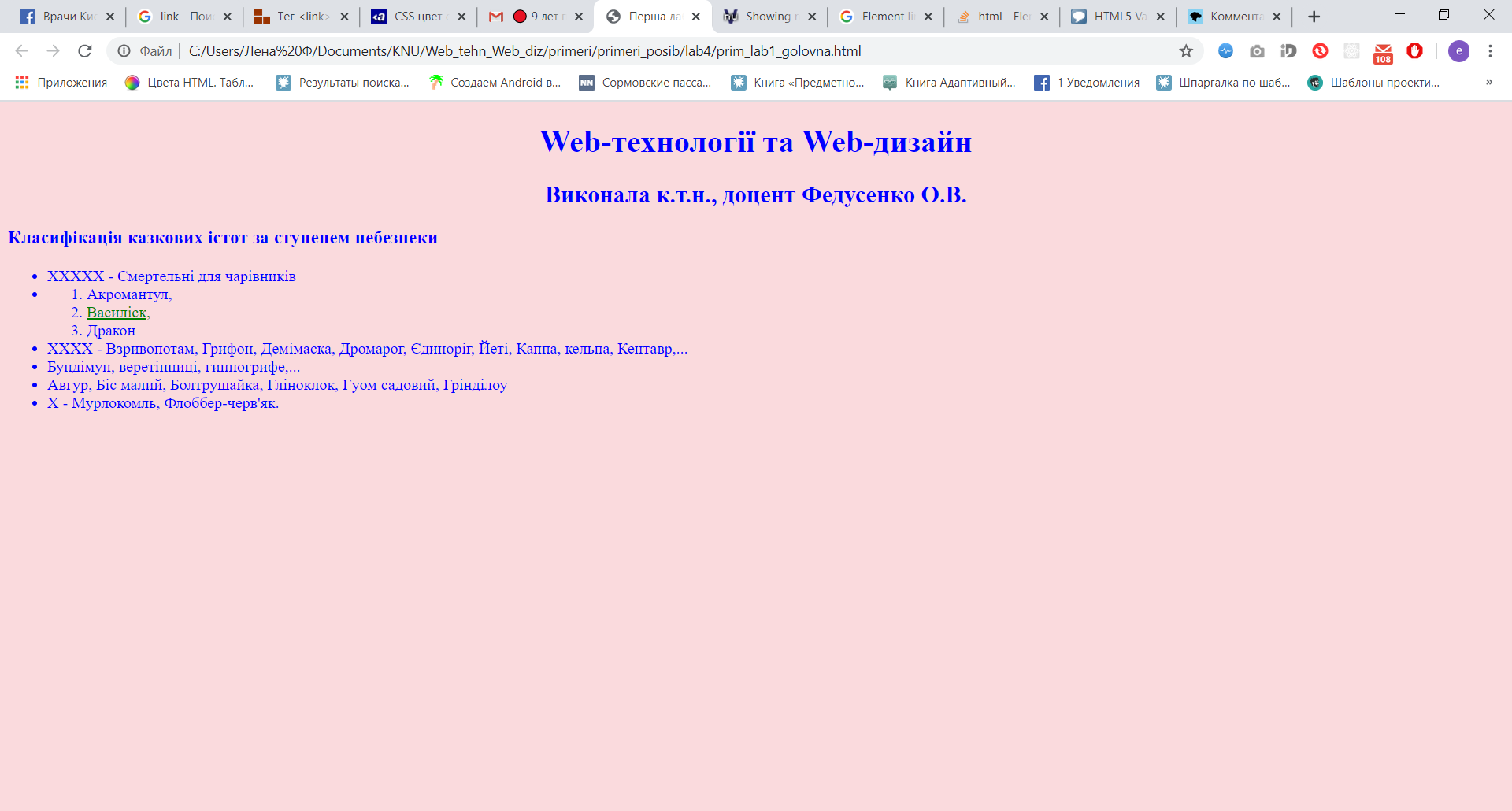
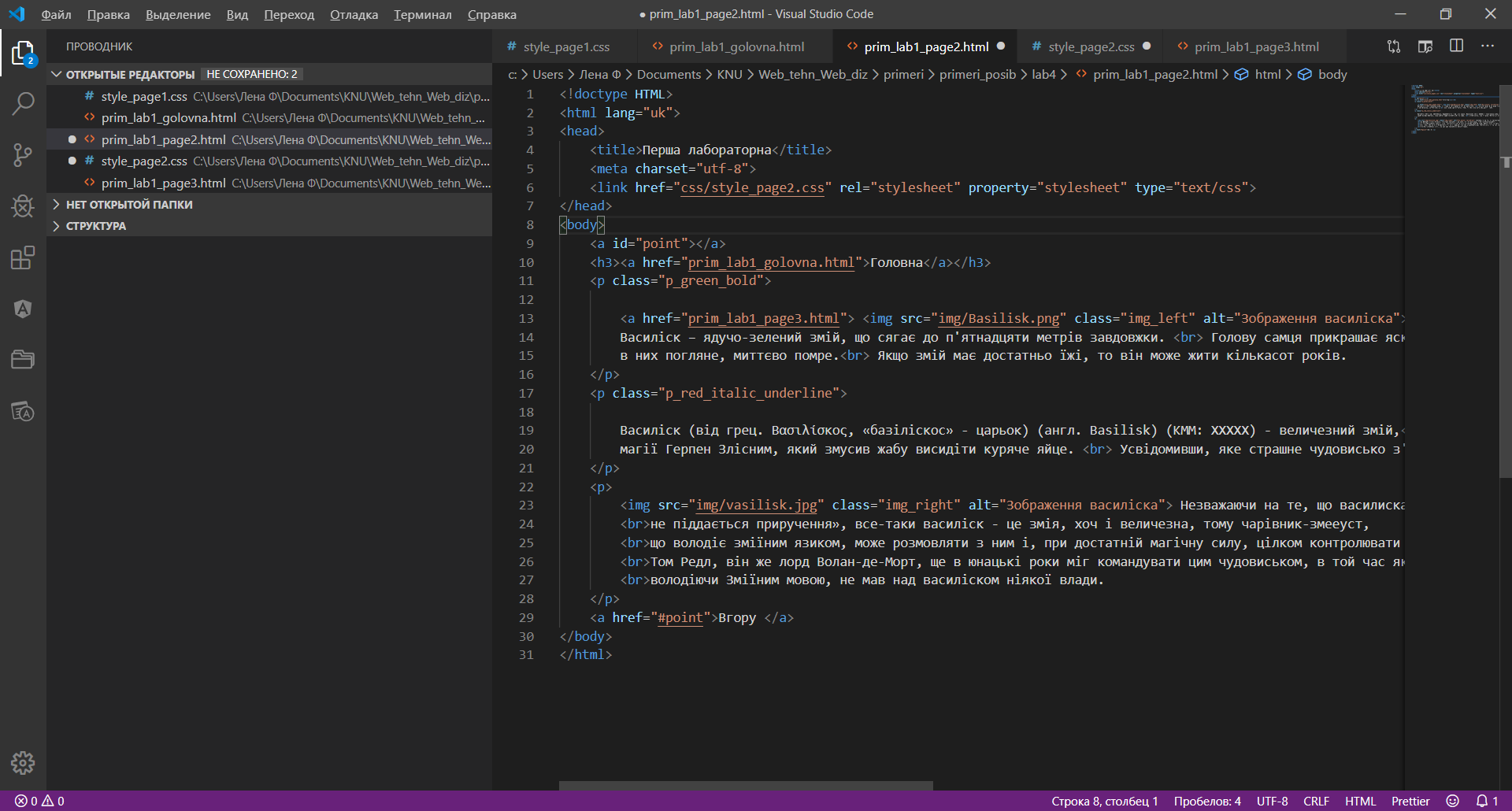


Рисунок 4.3 Перша сторінка лабораторної роботи

1. Розробимо другу сторінку відповідно до варіанту. Перехід вгору сторінки зробимо наступним чином: зверху сторінки створим наступне посилання: *<a name="point"></a>*, а знизу вставимо тег посилання на цю мітку *<a href="#point">Вгору</a>*. Html файл другої сторінки наведено на рис.4.4. Створимо css файл для другої сторінки, але оскільки в завданні необхідно задавати колір та властивості шрифту для кожного абзацу, то необхідно крім селекторів тегів використовувати селектори класів (рис 4.4). Для того, щоб виокремити одне слово у останньому абзаці необхідно використати тег *<span>* та встановити відповідну властивість у css файлі. Для завдання обтікання картинки будемо використовувати *float*. Зовнішній вигляд другої сторінки наведено на рис. 4.5.



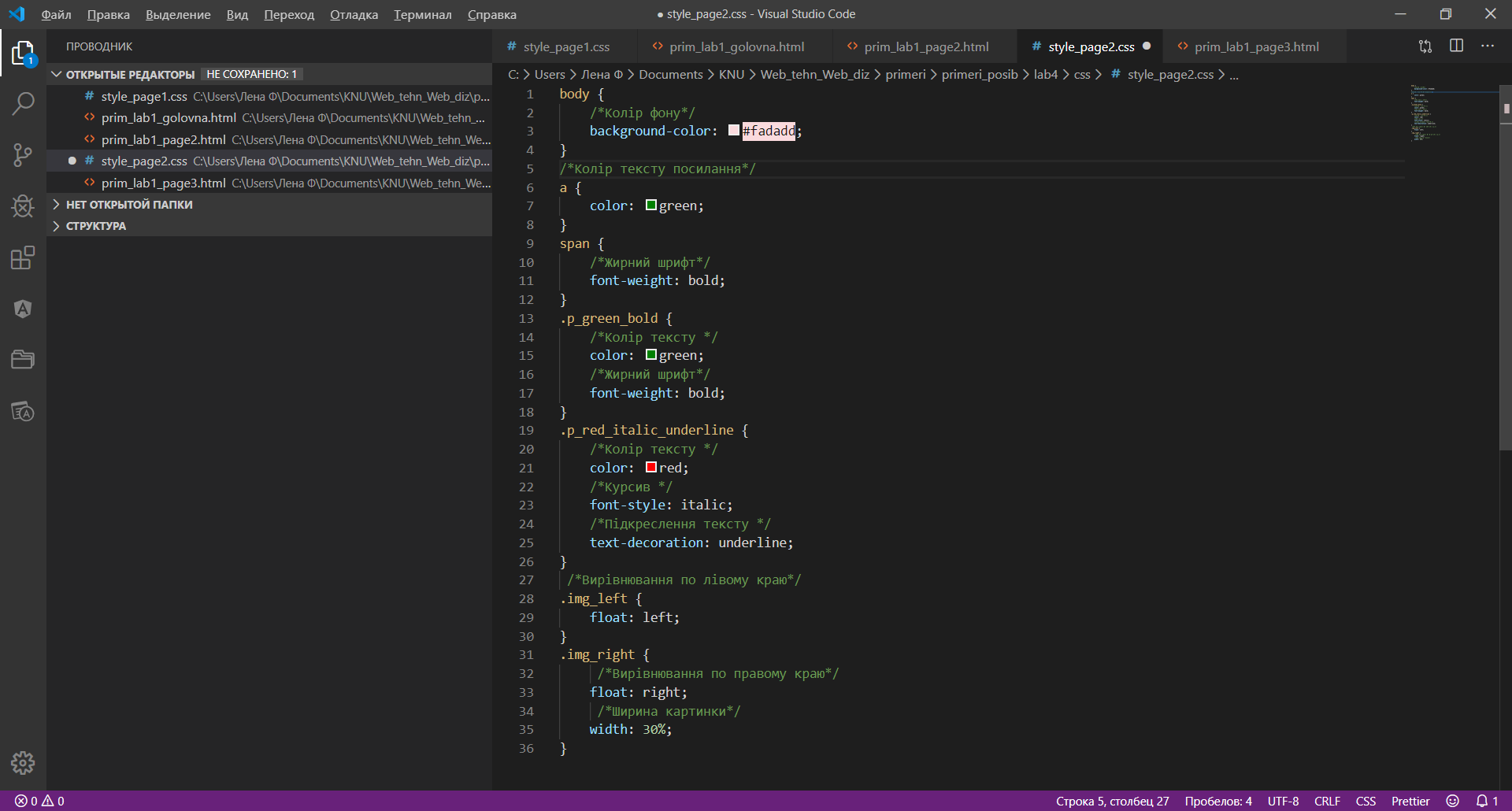


Рисунок 4.4 Html та css другої сторінки

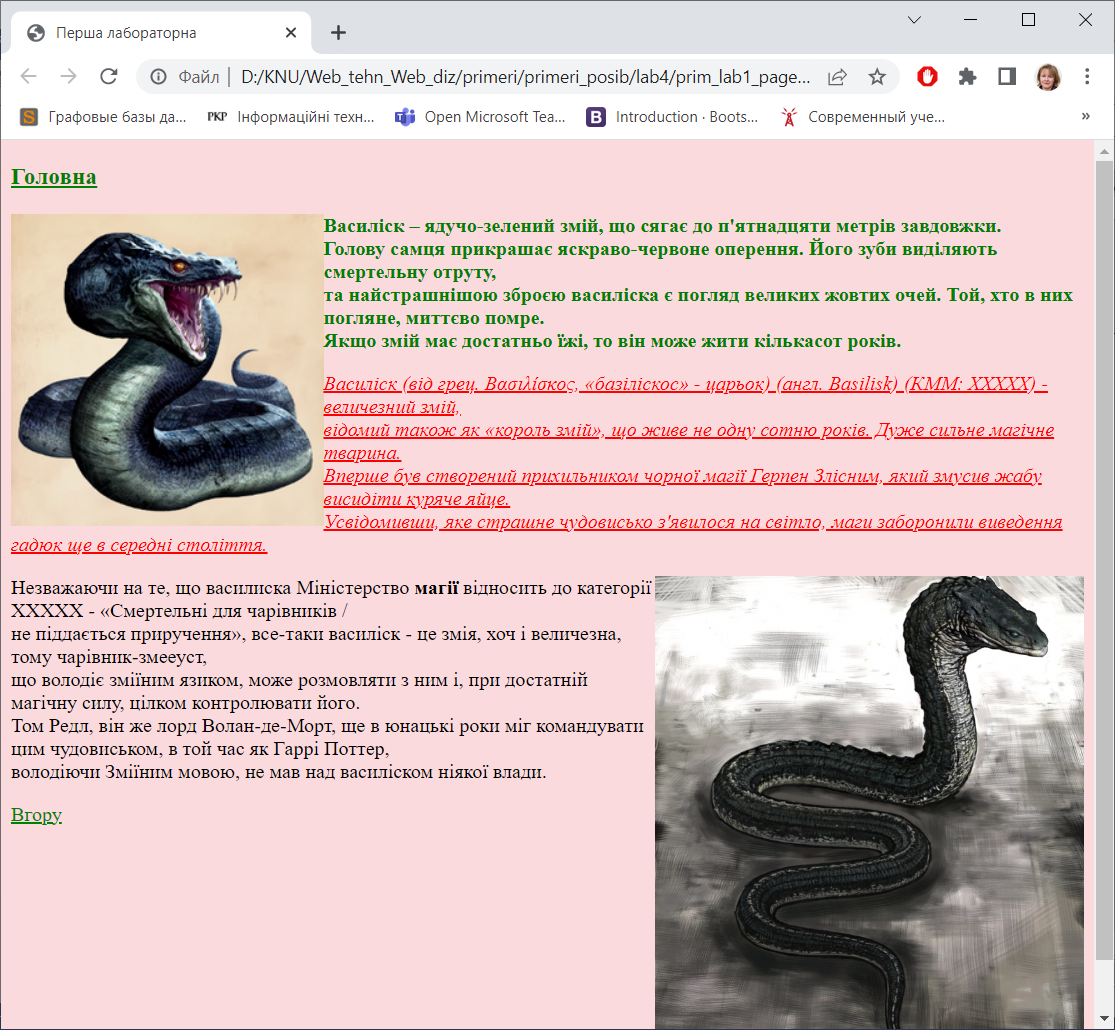


Рисунок 4.5 Зовнішній вигляд другої сторінки

1. Третя сторінка буде мати лише основні теги.Далі необхідно створити css файл для третьої сторінки він буде вміщувати лише один селектор *body* призначений для виведення зображення ( рис. 4.6.). Зовнішній вигляд третьої сторінки наведено на рис. 4.7.

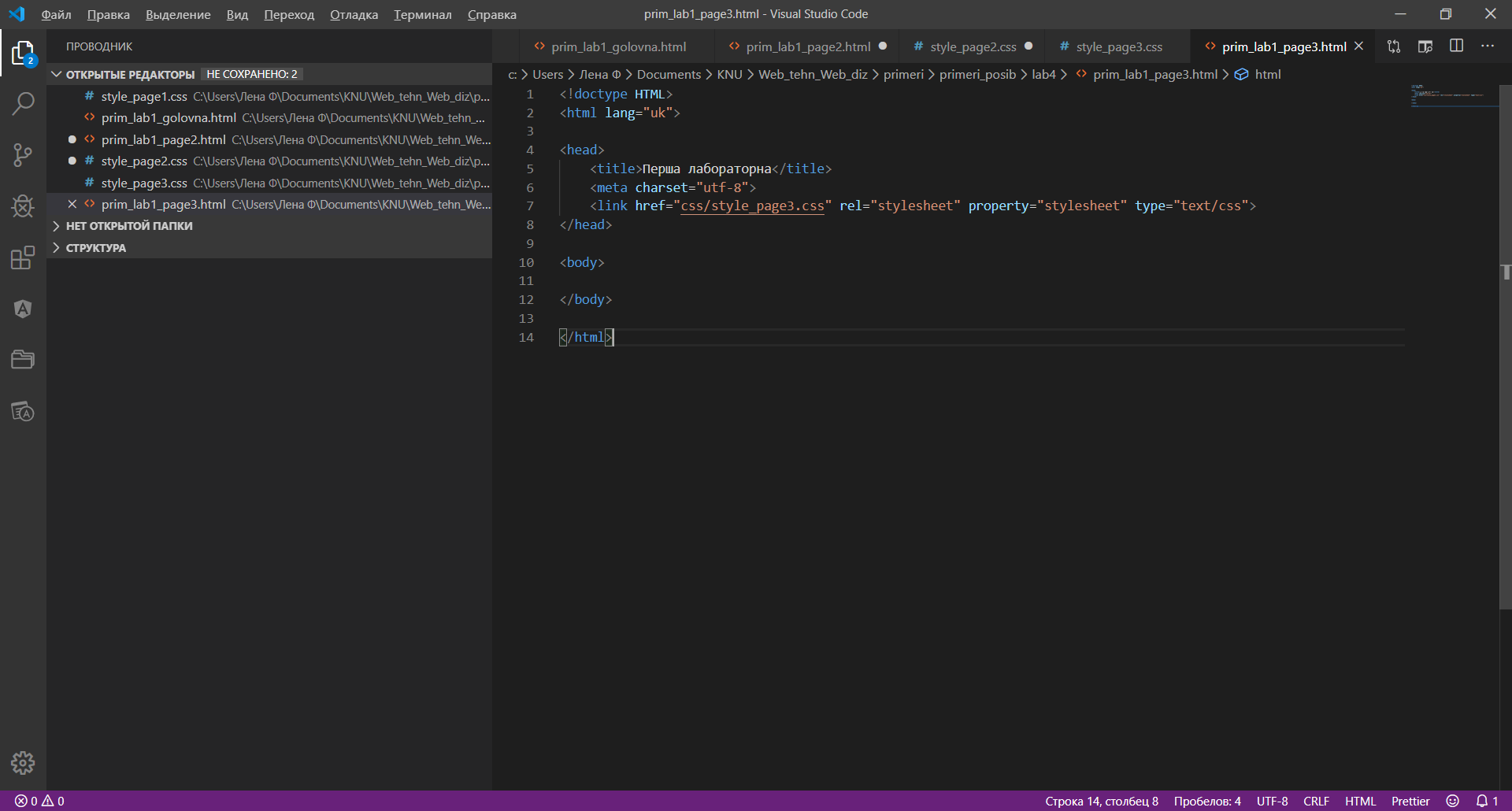
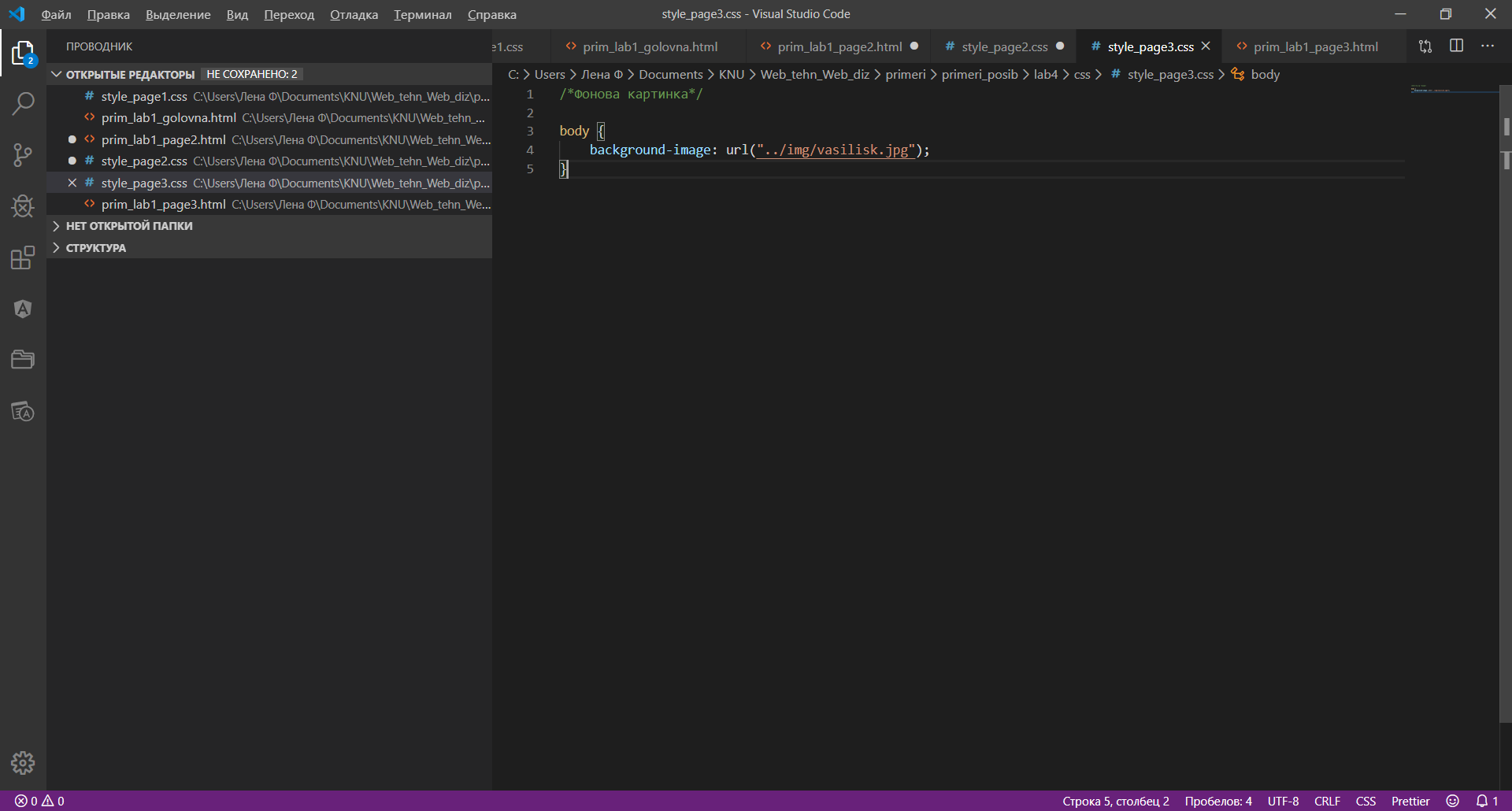
 

Рисунок 4.6 Html та css третьої сторінки

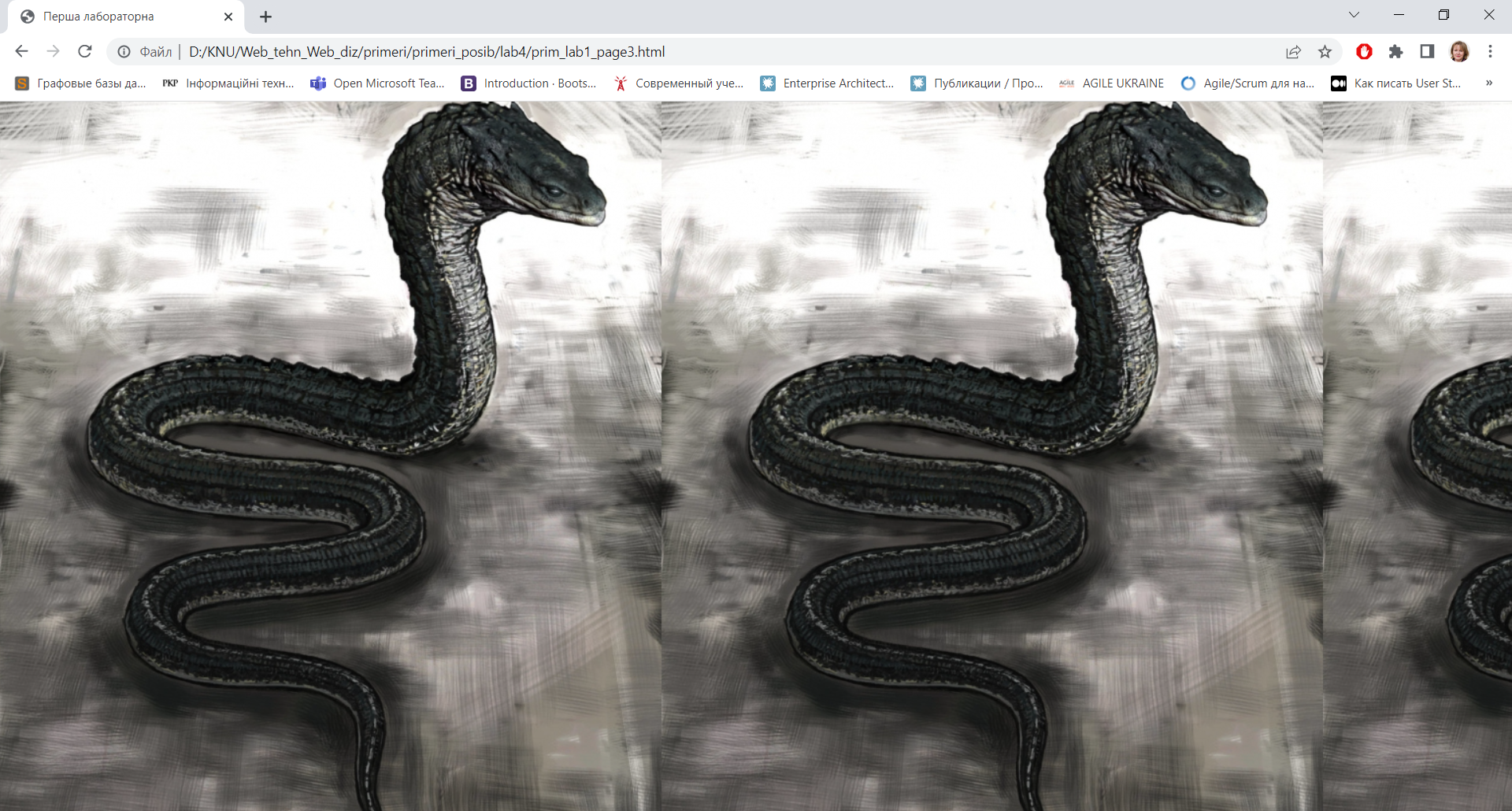


Рисунок 4.7 Зовнішній вигляд код третьої сторінки

5. Проведемо валідування розроблених сторінок за допомогою валідатору <https://validator.w3.org/>. Результати валідування наведено на рис.4.8.

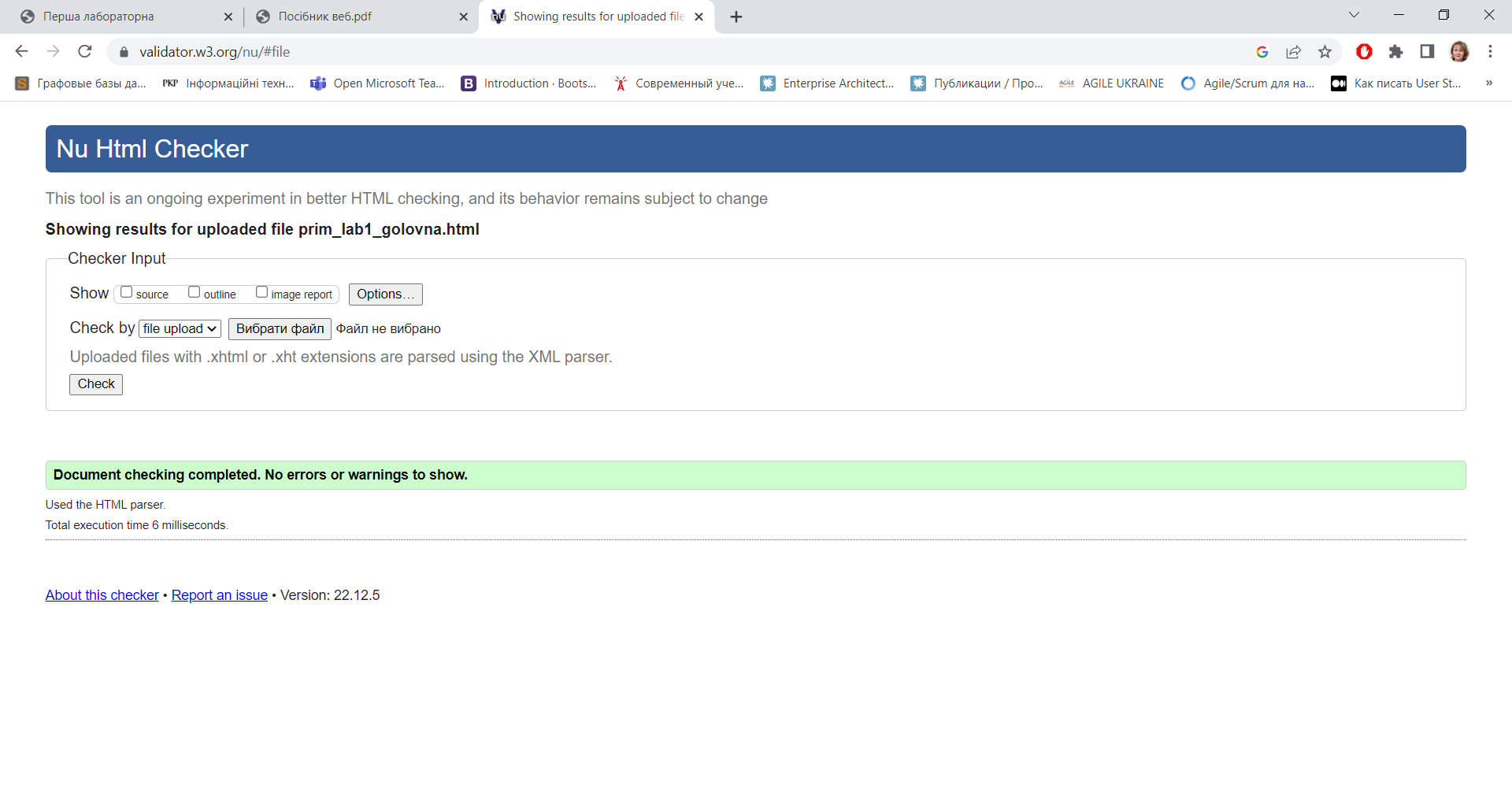
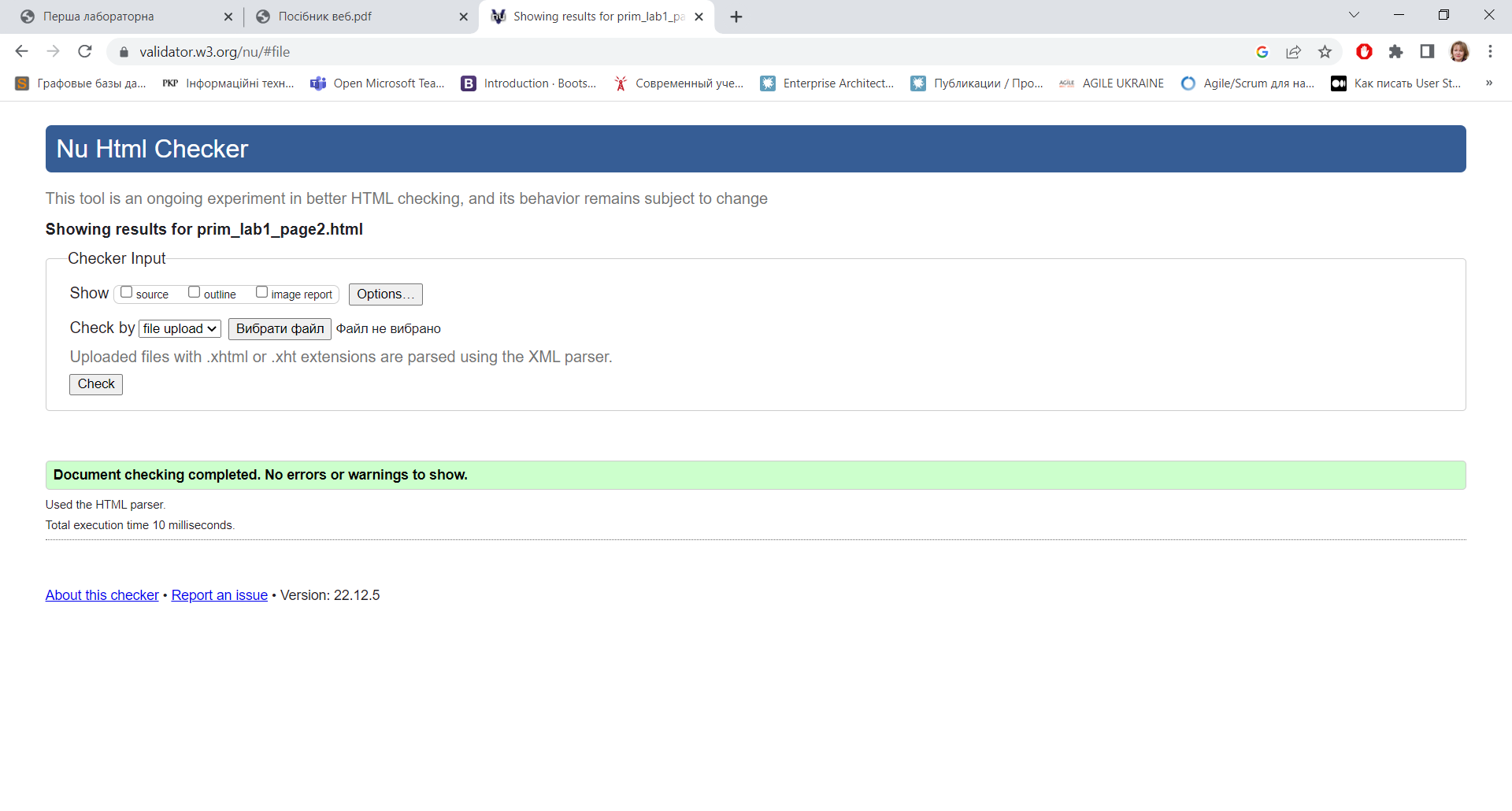
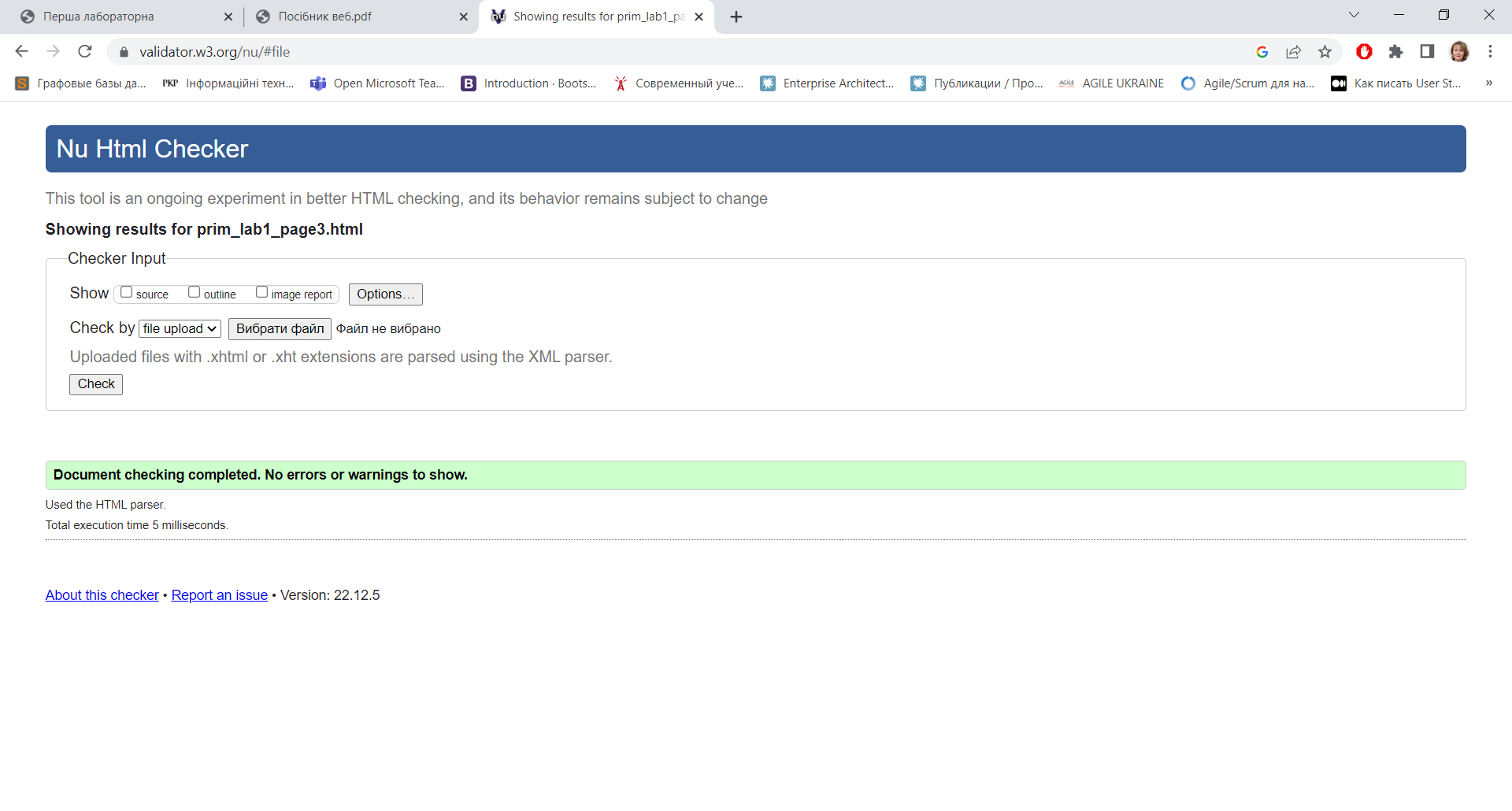
  

Рисунок 4.8 Результати валідування

**Варіанти завдань**

Варіанти завдань аналогічні завданням на лабораторну роботу №1.

**Варіанти завдань**

Таблиця 1.1 Оформлення першої сторінки

|  |  |  |  |
| --- | --- | --- | --- |
| Варіант | Параметри | | |
| Колір фону | Колір посилань | Колір тексту |
| 1 | Світло-сірий | коричневий | Темно-сірий |
| 2 | Темно-сірий | Світло-зелений | Світло-зелений |
| 3 | Світло-зелений | Темно-зелений | Блакитний |
| 4 | Темно-зелений | Жовтий | Бордовий |
| 5 | Бордовий | Блакитний | Жовтий |
| 6 | Блакитний | Бордовий | Синій |
| 7 | Зелений | Сірий | Коричневий |
| 8 | Фіолетовий | Шоколадний | Білий |
| 9 | Сірий | Червоний | Зелений |

Таблиця 1.2 Класифікатори

|  |  |
| --- | --- |
| Варіант | Список |
|  | Класифікатор грибів |
|  | Класифікатор лікарських рослин |
|  | Класифікатор диких тварин |
|  | Класифікатор домашніх тварин |
|  | Класифікатор квитів |
|  | Класифікатор посад на підприємстві |
|  | Класифікатор дерев |
|  | Класифікатор коней |
|  | Класифікатор рослин |

Таблиця 1.3 Оформлення другої сторінки

|  |  |  |  |
| --- | --- | --- | --- |
| Варіант | Параметри | | |
| Абзац 1 | Абзац2 | Абзац3 |
| 1 | Світло-сірий,  напівжирний | курсив | Останнє слово виокремить синім |
| 2 | Темно-сірий,  курсив | напівжирний | Перше слово зробити червоним |
| 3 | Світло-зелений,  підкреслений | підкреслений | Передостаннє слово зробити курсивним та червоним |
| 4 | Темно-зелений,  напівжирний | курсив | Останнє речення зробити зеленим |
| 5 | Жовтий  курсив,  підкреслений | напівжирний | Друге речення виділити блакитним |
| 6 | Блакитний,  курсив | підкреслений | Останнє слово виокремить синім |
| 7 | Бордовий,  напівжирний | курсив | Перше слово зробити червоним |
| 8 | Синій,  підкреслений | напівжирний | Останнє речення зробити зеленим |
| 9 | коричневий,  курсив | підкреслений | Передостаннє слово зробити курсивним та червоним |