**Мета роботи**– Вивчення правил формування HTML-документа, створення  
списків, посилань, таблиць та форм.  
**Завдання**  
Розробити декілька пов‘язаних між собою веб сторінок відповідно до варіанту роботи.  
Всі сторінки повинні мати єдине оформлення і вміщувати "шапку", що містить ПІБ студента, номер групи і номер варіанту.

1. Головна сторінка повинна вміщувати наступну інформацію:  
Використовуючи теги заголовка H1 і H2 вивести у верхній частині документа відповідно назва дисципліни і номер лабораторної роботи.  
За допомогою тегів списків DL, UL і OL вивести список, який включає посилання на сторінки з іншими завданнями.

2. Друга сторінка з таблицею повинна вміщувати наступну  
інформацію:  
"шапку", що містить ПІБ студента, номер групи і номер варіанту та посилання для повернення на головну сторінку;  
опис завдань відповідно до варіанту;  
розроблену таблицю відповідно до варіанту.

3. Третя сторінка з формою повинна вміщувати наступну інформацію  
"шапку", що містить ПІБ студента, номер групи і номер варіанту та посилання для повернення на головну сторінку;     
опис завдань відповідно до варіанту;  
розроблену форму взаємодії з користувачем, дані у формі повинні бути згруповані за певними категоріями. На формі обов‘язково повинна бути кнопка відправлення даних.

**Лабораторна робота №1 Основи мови HTML**

**Мета роботи** – Вивчення правил формування HTML-документа, створення списків, посилань, таблиць та форм.

**Завдання**

Розробити декілька пов‘язаних між собою веб сторінок відповідно до варіанту роботи.

Всі сторінки повинні мати єдине оформлення і вміщувати "шапку", що містить ПІБ студента, номер групи і номер варіанту.

1. Головна сторінка повинна вміщувати наступну інформацію:

* Використовуючи теги заголовка H1 і H2 вивести у верхній частині документа відповідно назва дисципліни і номер лабораторної роботи.
* За допомогою тегів списків DL, UL і OL вивести список, який включає посилання на сторінки з іншими завданнями.

2. Друга сторінка з таблицею повинна вміщувати наступну інформацію:

* "шапку", що містить ПІБ студента, номер групи і номер варіанту та посилання для повернення на головну сторінку;
* опис завдань відповідно до варіанту;
* розроблену таблицю відповідно до варіанту.

3. Третя сторінка з формою повинна вміщувати наступну інформацію

* "шапку", що містить ПІБ студента, номер групи і номер варіанту та посилання для повернення на головну сторінку;
* опис завдань відповідно до варіанту;
* розроблену форму взаємодії з користувачем, дані у формі повинні бути згруповані за певними категоріями. На формі обов‘язково повинна бути кнопка відправлення даних.

**Теоретичні відомості**

**Структура HTML документу**

Будь який HTML документ повинен мати наступну структуру (рис. 1.1).

Розглянемо докладно, деякі теги описані в структурі.

*Теги* – службові символи, що являють собою команди HTML, які вказує браузеру, як слід відображати Web-сторінку.

Теги починаються зі знака <, за яким потрібно писати ключове слово, і  
закінчується знаком > .

Теги можна «вкладати» усередину інших тегів. Оскільки мова HTML побудована на базі тегів, документи цього формату можна створювати простим розміщенням тегів навколо тексту.

*<!doctype html>* - елемент говорить браузеру, якою мовою розмітки і його версії написаний документ. Для версії HTML5 використовується саме цей запис.

Як видно з наведеного коду, HTML документ завжди поміщається в тег *<html>* і складається з двох частин: заголовка документа, що визначається тегом *<head>*, тіла документа, що визначається елементом *<body>*.

У заголовку HTML-документа наводиться інформація про документ, яка не відображається у вікні браузера. Винятком є ​​тег *<title>*, вміст якого виводиться в заголовку вікна браузера і використовується для ідентифікації документа користувачем або пошуковою машиною.  
Кожен HTML документ повинен мати назву.

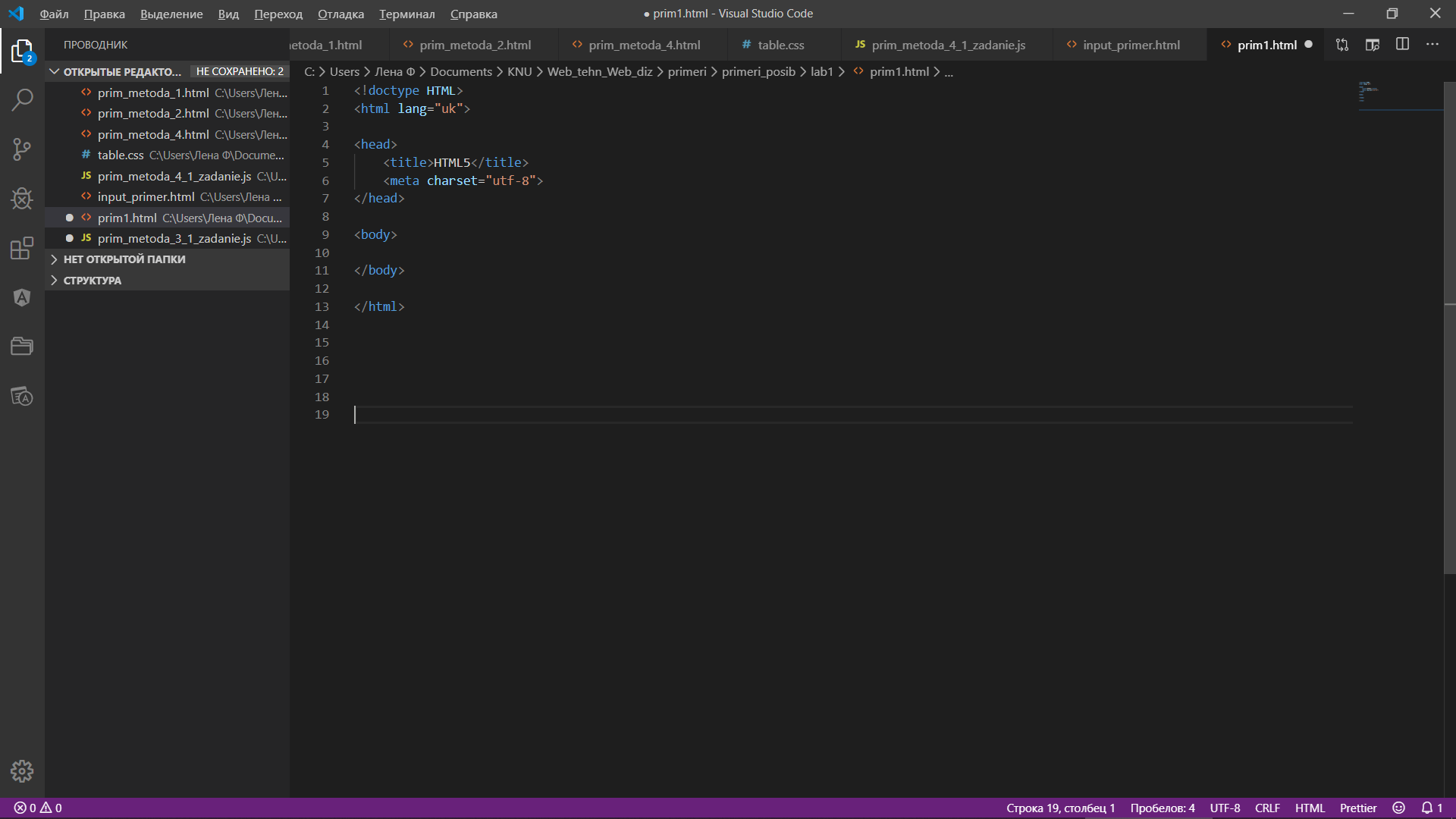


Рисунок 1.1 Структура Html документу

Необов'язковим тегом розділу *<head>* є одинарний тег *<meta>*. З його допомогою можна задати опис вмісту сторінки і ключові слова для пошукових машин, автора HTML-документа і інші властивості метаданих. Елемент *<head>* може містити кілька елементів *<meta>*, тому що в залежності від використовуваних атрибутів вони несуть різну інформацію.

Елементи, що знаходяться всередині тега <html>, утворюють дерево документа, так звану об'єктну модель документа, DOM (document object model). При цьому елемент *<html>* є кореневим елементом.

* Предок - елемент, який містить в собі інші елементи.  
  Нащадок - елемент, розташований усередині одного або більше типів елементів.
* Батьківський елемент - елемент, пов'язаний з іншими елементами нижчого рівня, і що знаходиться на дереві вище їх.
* Дочірній елемент - елемент, безпосередньо підлеглий іншому елементу більш високого рівня.
* Сестринський елемент - елемент, який має загальний батьківський елемент з даним, так звані елементи одного рівня.

**Теги елементів тексту**

* Теги заголовків: *<h1 ... h6>*
* Теги для форматування тексту: *<b>, <em>, <i>, <small>, <strong>, <sub>, <sup>, <ins>, <del>.*
* <ins> Виділяє текст підкреслюючи його.
* <del> Перекреслює текст. Використовується для виділення тексту, видаленого з документа.
* Теги для введення «комп'ютерного» тексту: *<code>, <kbd>, <samp>, <var>, <pre>*
* *<code>* Служить для виділення фрагментів програмного коду. Відображається моноширінним шрифтом.
* *<kbd>* Зазначає фрагмент як вводиться користувачем з клавіатури. Відображається моноширінним шрифтом.
* *<samp>* Застосовується для виділення результату, отриманого в ході виконання програми. Відображається моноширінним шрифтом.
* *<var>* Виділяє імена змінних, відображаючи курсивом.
* *<pre>* Дозволяє вивести текст на екран, зберігши початкове форматування.
* Теги для оформлення цитат і визначень: <*abbr>, <bdo>, <blockquote>, <q>, <cite>, <dfn>*
* *<abbr>* Застосовується для форматування абревіатур. Браузером зазвичай підкреслюється пунктирною лінією. Розшифровка скорочення здійснюється за допомогою атрибута *title*, вона з'являється при наведенні курсору миші на текст.
* *<bdo>* Використовується для заміщення поточного напрямку тексту, тобто текст в тезі відображається дзеркально.
* *<blockquote>* Виділяє цитати всередині документа, виділяючи його відступами і переносами рядків.
* *<q>* Використовується для виділення коротких цитат. Браузерами полягає в лапки.
* *<cite>* Застосовується для виділення цитат, назв творів, виносок на інші документи*.*
* *<dfn>* Дозволяє виділити текст як визначення. Незважаючи на наявність спеціального тега, рекомендується виділяти текст силами CSS.
* Абзаци, засоби перенесення тексту: *<p>, <br>, <hr>.*

**Атрибути тегів**

HTML-атрибути повідомляють браузеру, яким чином повинен відображатися той чи інший елемент сторінки. Атрибути дозволяють зробити більш різноманітними зовнішній вигляд інформації, що додається за допомогою однакових тегів.

Значення атрибута полягає в лапки "". Назви і значення атрибутів не чутливі до регістру, але, тим не менш, рекомендується набирати їх в нижньому регістрі.

Атрибути можна поділити на дві групи:

* + Глобальні атрибути , наприклад *id,class,lang*
  + Локальні атрибути, наприклад, *alt.*

Розглянемо деякі з атрибутів певних тегів більш докладніше.

*<font size=?></font>* Атрибут *size* тегу *font* установлює відносний розмір шрифту. Список можливих значень складається з додатних і від’ємних чисел від 0 до 7.

*<font color=?></font>* Атрибут *color* тегу *font* установлює кольори тексту. Колір може бути зазначений за допомогою за допомогою назви або шістнадцяткового коду.  
*<p></p>* Тег *p* установлює початок і кінець абзацу.

*<p align=?>* Атрибут *align* тегу *p* установлює режим вирівнювання тексту: уліво (*LEFT*), вправо (*RIGHT*) або по центру (*CENTER*).

Слід зазначити, що більшість з текстових тегів та їх атрибутів радять не використовувати у HTML5, оскільки за зовнішній вигляд відповідає CSS. Тому HTML5 використовуються лише теги *<p>, <br>, <hr>* без атрибутів, що відповідають за зовнішній вигляд.

**Списки, посилання, зображення**

У HTML існує три види списків в

* 1. Ненумеровані списки. Створюються з використанням тегів *<ul>* та *<li>*
  2. Нумеровані списки. Створюються з використанням тегів *<ol>* та *<li>*
  3. Список-визначення *<dl>* складається з терміну *<dt>* та його визначення *<dd>*

За допомогою посилань реалізується зв’язок між декількома документами один з яких є поточним документом.

*<a href=URL></a>* Тег встановлює зв’язок з деякою точкою усередині того ж HTML-документа або з іншим URL (гипертекстове посилання). Має наступні основні атрибути:

* Атрибут *href* тегу *a* описує об’єкт, що представляє собою текст або малюнок усередині HTML-документа, або текст або малюнок у зовнішньому документі.

Для вставки в HTML-документ зображень призначений непарний тег *<img>,*  базові можливості якого визначаються наступними атрибутами:  
*src* - відносний або абсолютний URI зображення (обов'язковий)  
*alt* - короткий опис (відображається при відключеному графіку)  
*width і height* - ширина і висота відповідно, в пікселях або відсотках

У HTML 5, на відміну від попередньої версії, атрибут alt не є обов'язковим, але рекомендується до заповнення. Якщо не вказані ширина і висота, то картинка відобразиться в оригінальному розмірі, або, якщо графіка в браузері відключена, то елемент прийме такий розмір, щоб в нього помістилася *alt*-напис. Вказавши лише один з цих атрибутів, ви отримаєте зображення необхідної ширини або висоти зі збереженням оригінальних пропорцій. Для досягнення точних розмірів вказуйте обидва значення, але пам'ятайте, що якщо задані атрибути не співпадають з оригіналом зображення, то масштабування призведе до втрати його якості.

* Атрибут *align* тегу *img* установлює режим вирівнювання зазначеного зображення. Можливе вирівнювання вліво *(LEFT)*, вправо *(RIGHT)*, по центру *(CENTER)*, униз *(BOTTOM)*, нагору *(TOP)*, посередині *(MIDDLE)*.
* Атрибут *border* тегу *img* установлює режим відображення рамки навколо зазначеного зображення. Малюнок з гіперпосиланням за замовчуванням має рамку, кольори якого збігається з кольорами, використовуваними для гіперпосилань.
* Атрибут *alt* тегу *img* установлює режим відображення зазначеного тексту (як альтернатива) у випадку, якщо браузер не може відобразити графічний файл.

Атрибути *align, width, height* та *border* радять не використовувати у HTML5, оскільки за зовнішній вигляд відповідає CSS.

**Таблиці**

HTML-таблиці впорядковують і виводять на екран дані за допомогою рядків або стовпців. Таблиці складаються з клітинок, що утворюються при перетині рядків і стовпців. Клітинки таблиць можуть містити будь-які HTML-елементи, такі як заголовки, списки, текст, зображення, елементи форм, а також інші таблиці. Кожній таблиці можна додати пов'язаний з нею заголовок, розташувавши його перед таблицею або після неї.

Таблиця створюється за допомогою парного тегу *<table> </table>.* Даний тег є контейнером для елементів таблиці і всі елементи повинні знаходитися всередині нього. Розглянемо докладніше основні теги для створення таблиці (рис. 1.2).

Теги *<caption>,<tbody>,<tfoot>,<thead>,<colgroup>* не є обов‘язковими. Таблиця може описуватися лише з використанням тегів для створення рядків *<tr>* та клітинок у них *<td>*.

Атрибути *colspan* і *rowspan* об'єднують клітинки таблиці. Атрибут *colspan* задає кількість клітинок, об'єднаних по горизонталі, а *rowspan* - по вертикалі.

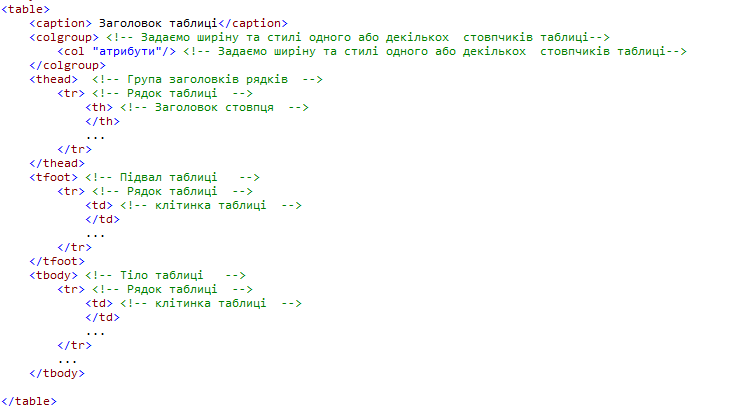


Рисунок 1.2 Повний опис таблиці

**HTML-форми**

HTML-форми є елементами управління, які застосовуються для збору інформації від відвідувачів веб-сайту.

Веб-форми складаються з набору текстових полів, кнопок, списків і інших елементів управління, які активізуються клацанням миші. Технічно форми передають дані від користувача до віддаленого сервера.

Форми створюються за допомогою тега *<form> …</form>.* Для тега *<form>* бажано вказувати наступні атрибути:

* *action* - цей атрибут визначає URL-адресу скрипту, який отримає дані з цієї форми. Наприклад, *action="avto.php"*. Якщо вказати "порожню" адресу (тобто *action=""*), то браузер відправить дані форми файлу, в якому і створена форма.
* *method* - атрибут задає спосіб відправки даних на сервер. При значенні *"get"* дані будуть додані до URL-адресу скрипту, до якого і відправляються дані. В цьому випадку вони будуть видимі в адресному рядку браузера, що не дуже добре, оскільки їх зможе побачити користувач. Але ще є обмеження на довжину даних що передаються, обумовлені допустимою кількістю символів в адресному рядку браузера. При значенні *"post"* дані будуть відправлені у вигляді блоку даних, який не буде видимий для користувача. І в цьому випадку об'єм даних практично ні чим не обмежений. Тому дуже часто використовують саме метод *"post"*.
  1. **Мітки на формі**

HTML тег label пов'язує елемент форми *(<input>, <select>, <textarea>)* з міткою, наприклад текстом, це буває дуже корисно при проектуванні веб-інтерфейсів.

Елемент *<label>* не робить нічого особливого для користувача. Він просто забезпечує краще юзабіліті для користувачів, тому, що, якщо користувач клацає по тексту, що виступає міткою, то це автоматично впливає на форму (наприклад ставить прапорці).

Мітка прив'язується до елемента за допомогою атрибута "for", або шляхом розміщення елемента всередині елемента *<label>*.

*<form>*

*<input id="ідентифікатор“ …>*

*<label for="ідентифікатор">Текст</label>*

*</form>*

або

*<form>*

*<label >Текст*

*<input id="ідентифікатор“ type=“…” >*

*</label>*

*</form>*

* 1. **Групування елементів форми**

HTML тег *<fieldset>* використовується для групування пов'язаних елементів в формі. Таке групування полегшує роботу з формами, що містять велику кількість даних, наприклад, один блок може бути призначений для введення текстової інформації, а інший - для прапорців.

Тег *<fieldset>* малює рамку навколо пов'язаних елементів. Її вигляд залежить від операційної системи, а також браузера.

Для кожної групи може бути вказана назва за допомогою HTML тегу *<legend>.*

*<form>*

*<fieldset>...</fieldset>*

*</form>*

* 1. **Тег input**

HTML теґ *<input>* створює поле в яке користувач може ввести дані

Головним чином *<input>* призначений для створення текстових полів, різних кнопок, перемикачів і прапорців.

Поле *input* може мати різний вигляд - залежить від атрибуту *type*. Даний атрибут може приймати наступні значенння.

* *button* - створює кнопку.
* *checkbox* - перетворює поле введення під прапорець, який можна встановити або очистити.
* *color* - генерує палітри кольорів, даючи користувачам можливість вибирати значення кольорів в шістнадцятковому форматі.
* *date* - дозволяє вводити дату в форматі дд.мм.рррр.
* *datetime-local* - дозволяє вводити дату і час, розділені великої англійської буквою Т за шаблоном дд.мм.рррр гг: хх.
* *email* - браузери, які підтримують даний атрибут, чекатимуть, що користувач введе дані, відповідні до синтаксису адреси електронної пошти.
* *file* - дозволяє завантажувати файли з комп'ютера користувача.
* *hidden* - приховує елемент управління, такий елемент не відображається браузером і не дає користувачеві змінювати значення за замовчуванням.
* *image* - створює кнопку, дозволяючи замість тексту на кнопці вставити зображення.
* *month* - дозволяє користувачеві вводити рік і номер місяця за шаблоном рррр-мм.
* *number* - призначене для введення цілих значень. Його атрибути *min*, *max і step* задають верхню, нижню межу і крок між значеннями.
* *password* - створює текстові поля у формі, при цьому символи, що вводяться користувачем замінюються на зірочки, маркери, або інші, встановлені браузером значки.
* *radio* - створює перемикач - елемент управління у вигляді невеликого кола, який можна включити або виключити.
* *range* - дозволить створити такий елемент інтерфейсу, як повзунок, що має атрибути *min / max*, які дозволять встановити діапазон вибору
* *reset* - створює кнопку, яка очищає поля форми від введених користувачем даних.
* *search* - позначає поле пошуку, за дане поле введення має прямокутну форму.
* *submit* - створює стандартну кнопку, що збирає інформацію з форми і відправляє її для обробки.
* *text* - створює текстове поле у формі.
* *time* - дозволяє вводити час в 24-годинному форматі за шаблоном гг: хх Воно відображається як елемент управління у вигляді числового поля введення і допускає введення тільки значень часу.
* *url* - поле призначене для введення URL-адреси.
* *week* - дозволяє користувачеві вибрати один тиждень на рік.
  1. **Текстові поля вводу**

HTML теґ *<textarea>* визначає область в яку можна вводити декілька рядків тексту.

На відміну від теґа *<input>* у текстовій області *<textarea>* допустимо робити переноси рядків і вони будуть збережені при відправці даних на сервер.

Розмір текстової області може бути визначений за допомогою атрибутів cols та rows, але краще це робити за допомогою CSS властивостей height і width.

*<textarea> текст </textarea>*

Наприклад

*<textarea rows="4" cols="50">*

*Поле виситою 4 і довжиною 50*

*</textarea>*

* 1. **Списки**

Тег *<select>* дозволяє створити елемент інтерфейсу у вигляді списку, що розкривається, а також список з одним або множинним вибором.

Кожен пункт створюється за допомогою тегу *<option>*, який повинен бути вкладений в контейнер *<select>*.

Якщо планується відправляти дані списку на сервер, то потрібно помістити елемент *<select>* всередину форми. Це також необхідно, коли до даних списку йде звернення через скрипти.

*<select>*

*<option>Пункт 1</option>*

*<option>Пункт 2</option>*

*</select>*

* 1. **Кнопки**

Елемент *<button> ... </ button>* створює клікабельні кнопки.

На відміну від кнопок, створених *<input>* всередину елемента *<button>* можна помістити контент - текст або зображення.

Для коректного відображення елемента *<button>* різними браузерами потрібно вказувати атрибут *type,* наприклад,

*<button type = "submit"> Відправити</ button>.*

Атрибут *type* може приймати наступні значення:

* *button* - клікабельна кнопка.
* *reset* - кнопка скидання, повертає первісне значення елементам форми.
* *submit* - кнопка для відправки даних форми.

Кнопки дозволяють користувачам передавати дані в форму, очищати вміст форми або робити які-небудь інші дії. Можна змінювати фон і вирівнювати текст на кнопці.

**Приклад виконання**

**Завдання**

Розробимо застосунок для виконання наступного завдання.

*Друга сторінка:*

Необхідно описати веб-сторінку для розкладу уроків у школі. Розклад виглядає наступним чином рис. 1.3. Малюнки у таблицю можна не вставляти.



Рисунок 1.3 Приклад складної таблиці

*Третя сторінка:*

Необхідно розробити форму для введення книги у веб-систему бібліотеки. Необхідно заповнити наступні данні:

• дані про книгу, включають назву книги, жанр (вибирається зі списку), рік випуску, кількість екземплярів у бібліотеці;

• дані про автора книги, включають ПІБ автору, дату народження, дату смерті, біографію автора, стать автору.

**Хід виконання**

1. Розробимо головну сторінку яка буде вміщувати «шапку» з заголовками та список з пунктами меню з посиланнями на інші сторінки. Код сторінки наведено на рис. 1.4, а зовнішній вигляд на рис. 1.5.

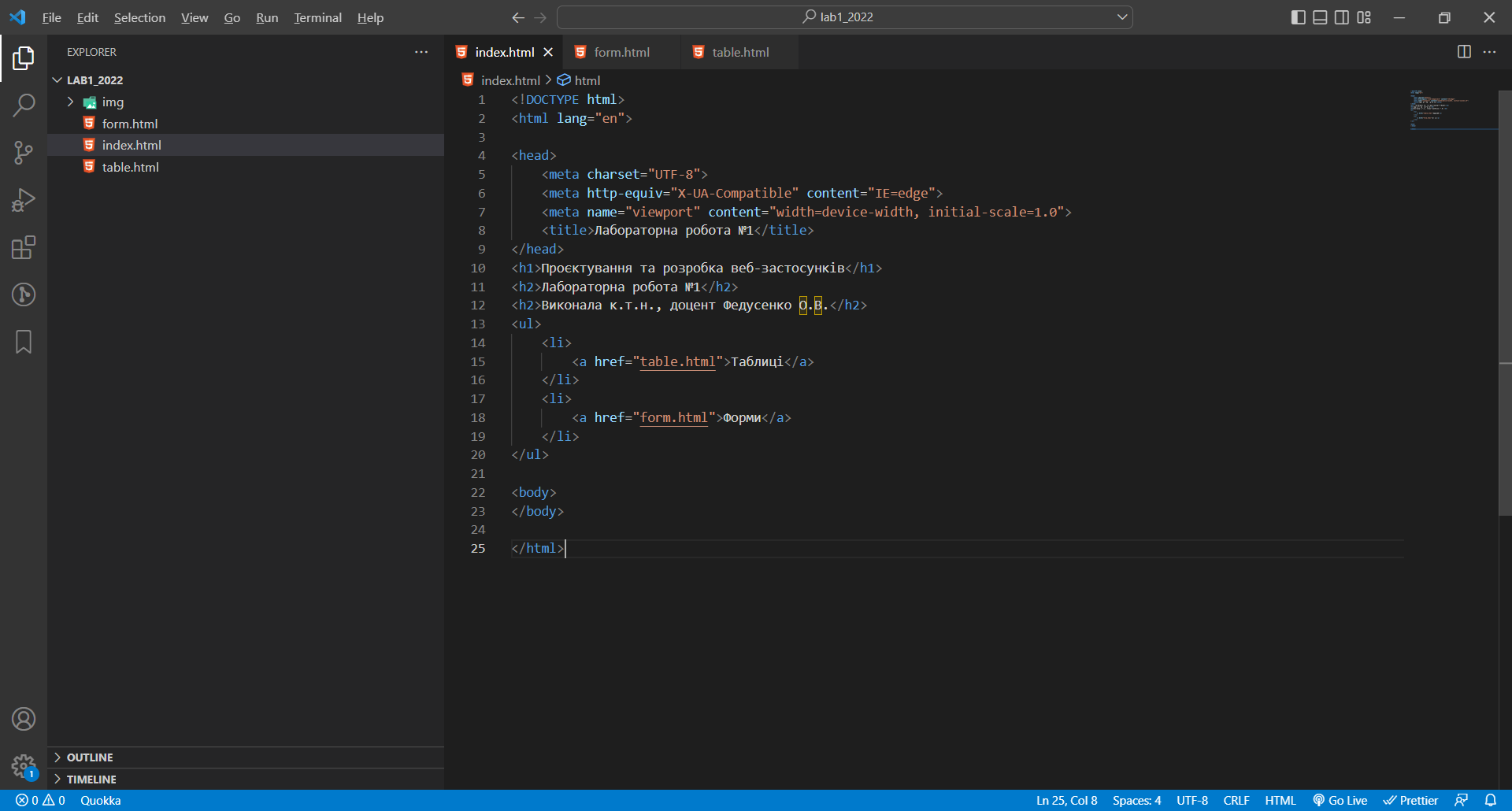


Рисунок 1.4 Html код головної сторінки

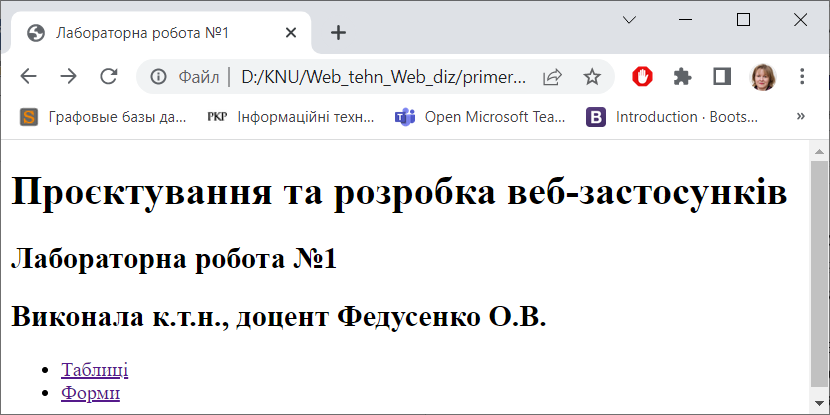


Рисунок 1.5 Зовнішній вигляд головної сторінки

1. Розробимо спочатку шаблон другої веб сторінки яка буде містить шапку, посилання на головну та опис завдання (рис. 1.6 та рис. 1.7).

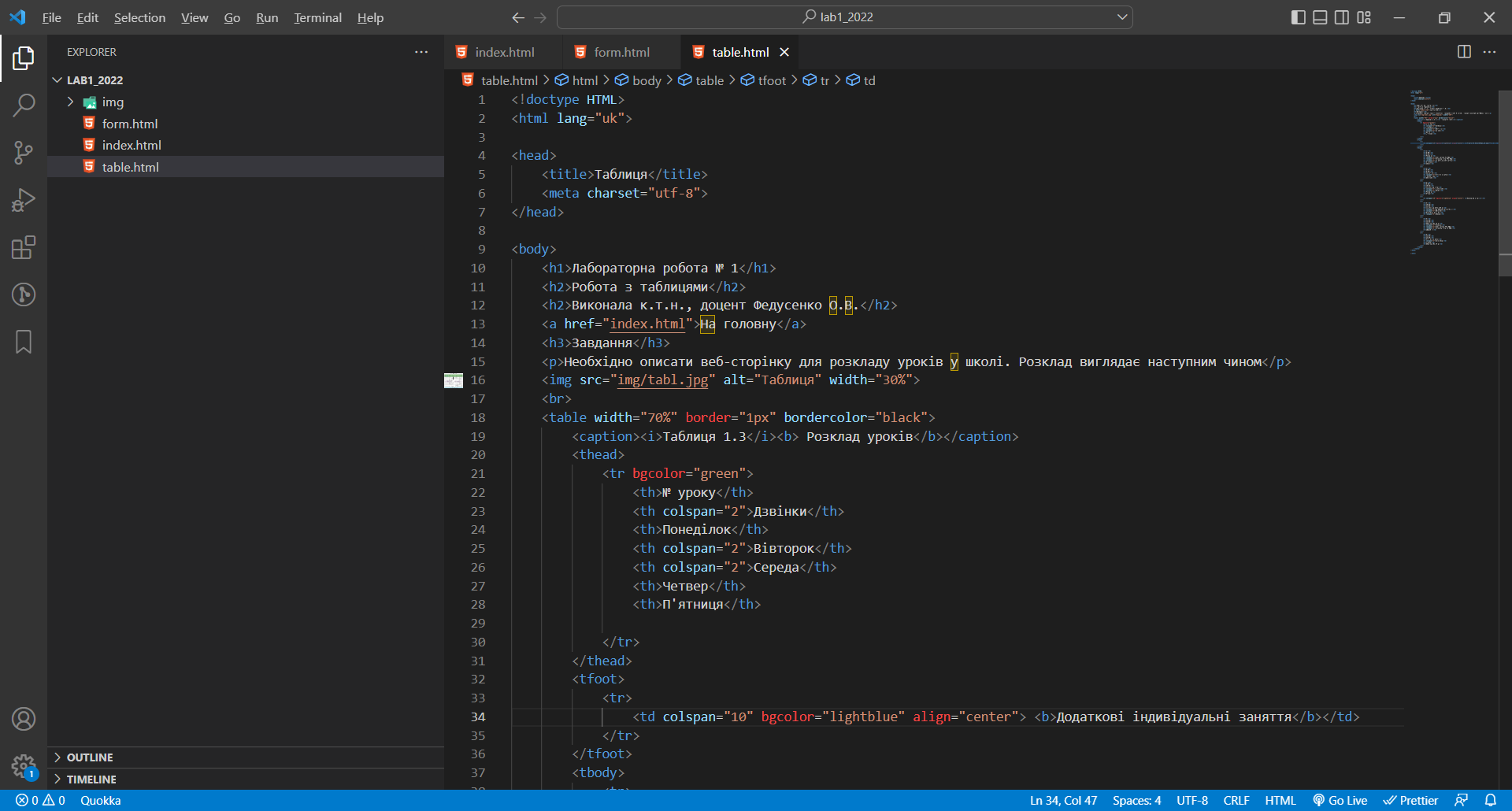


Рисунок 1.6 Шаблон html сторінки з таблицею

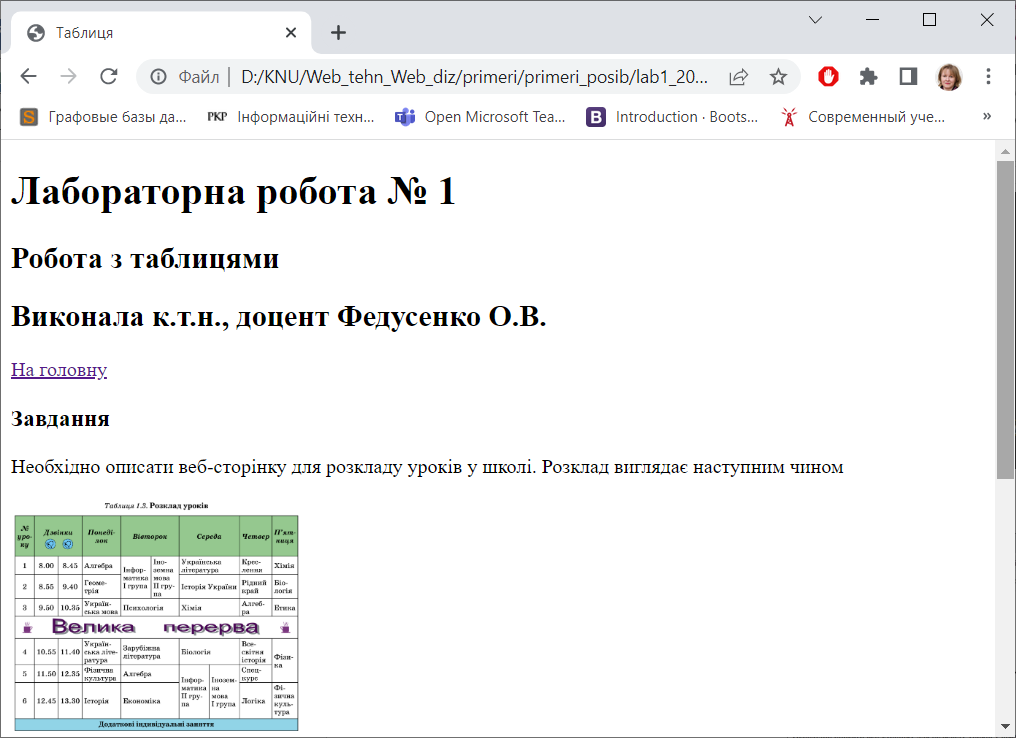


Рисунок 1.7 Зовнішній вигляд html сторінки

1. Створимо таблицю з використанням тегу *<table>* та опишемо її заголовок тегом*<caption>,* заголовок рядків тегом*<thead>* та підвал тегом *<tfoot>* (рис. 1.8). Наша таблиця в цілому буде складатися з 10 стовпчиків та 9 рядків.

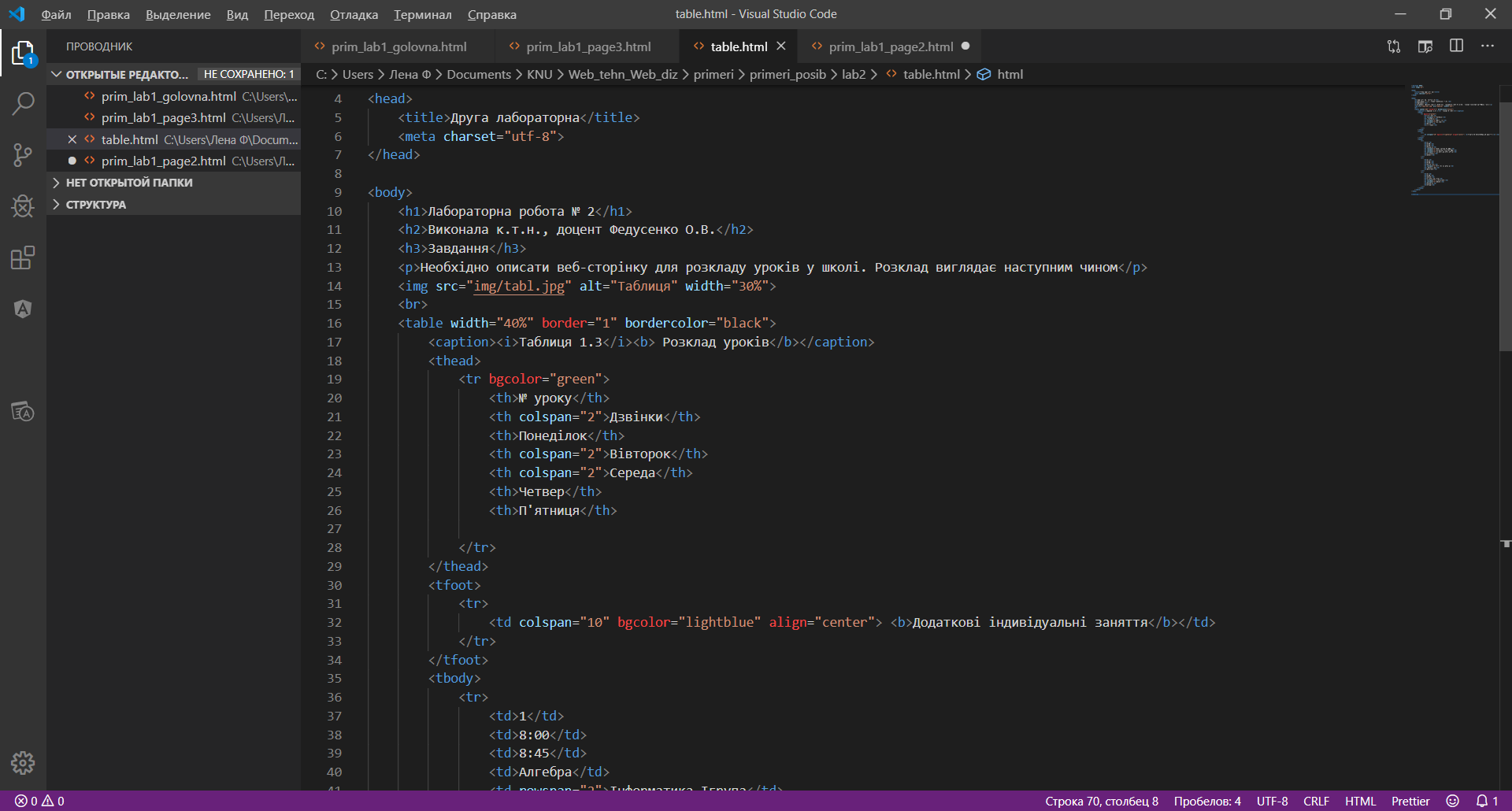


Рисунок 1.8 Опис заголовку та підвалу таблиці

1. Опишемо верхню частину таблиці до рядка з написом «Велика перерва» (рис. 1.9). Опис у тегу *<tbody>* вставимо відразу після опису підвалу таблиці.

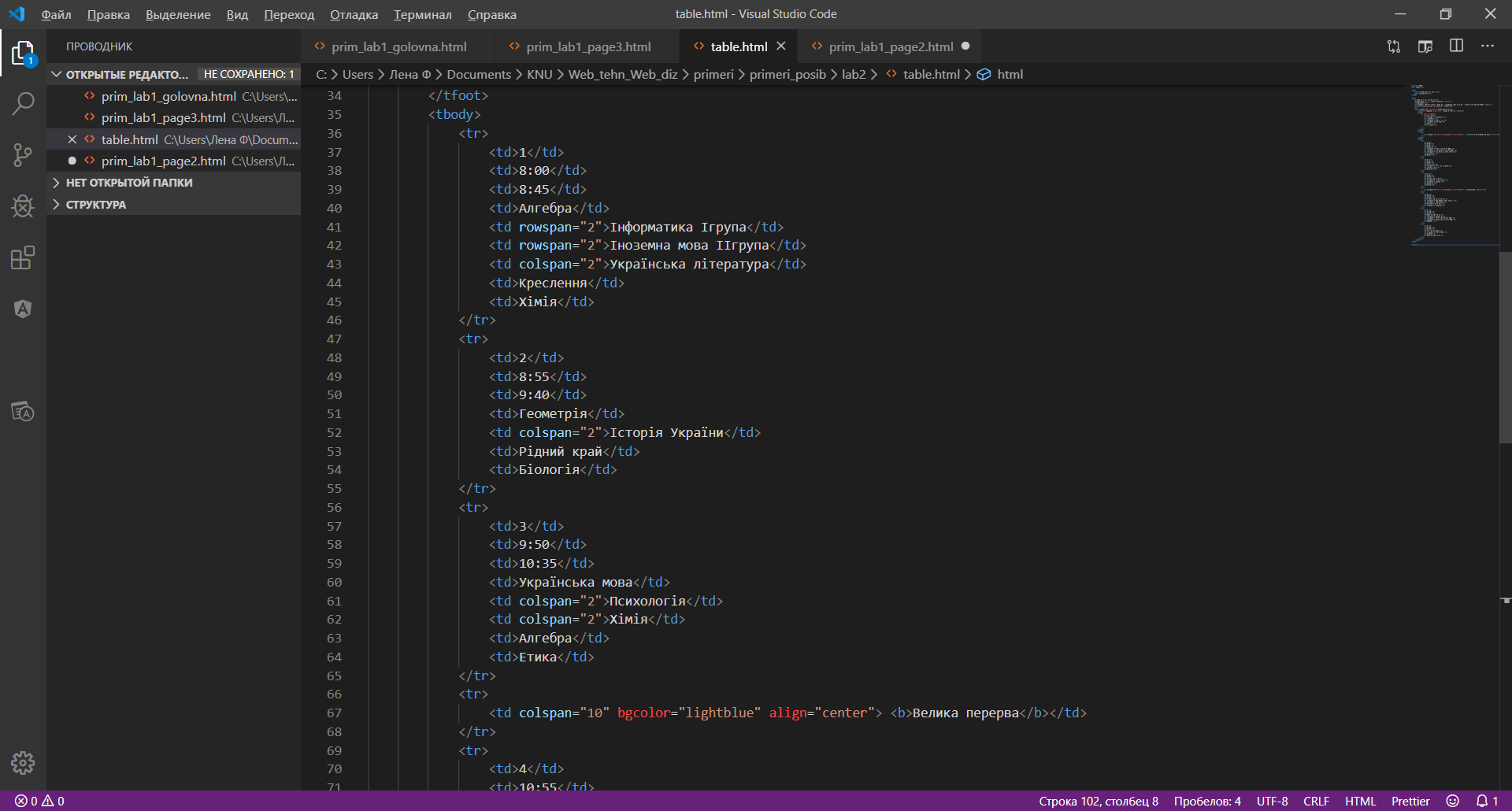


Рисунок 1.9 Опис трьох перших рядків таблиці

1. Аналогічним чином опишімо рядки, які залишилися. Вигляд результуючої таблиці наведено на рис.1. 10.

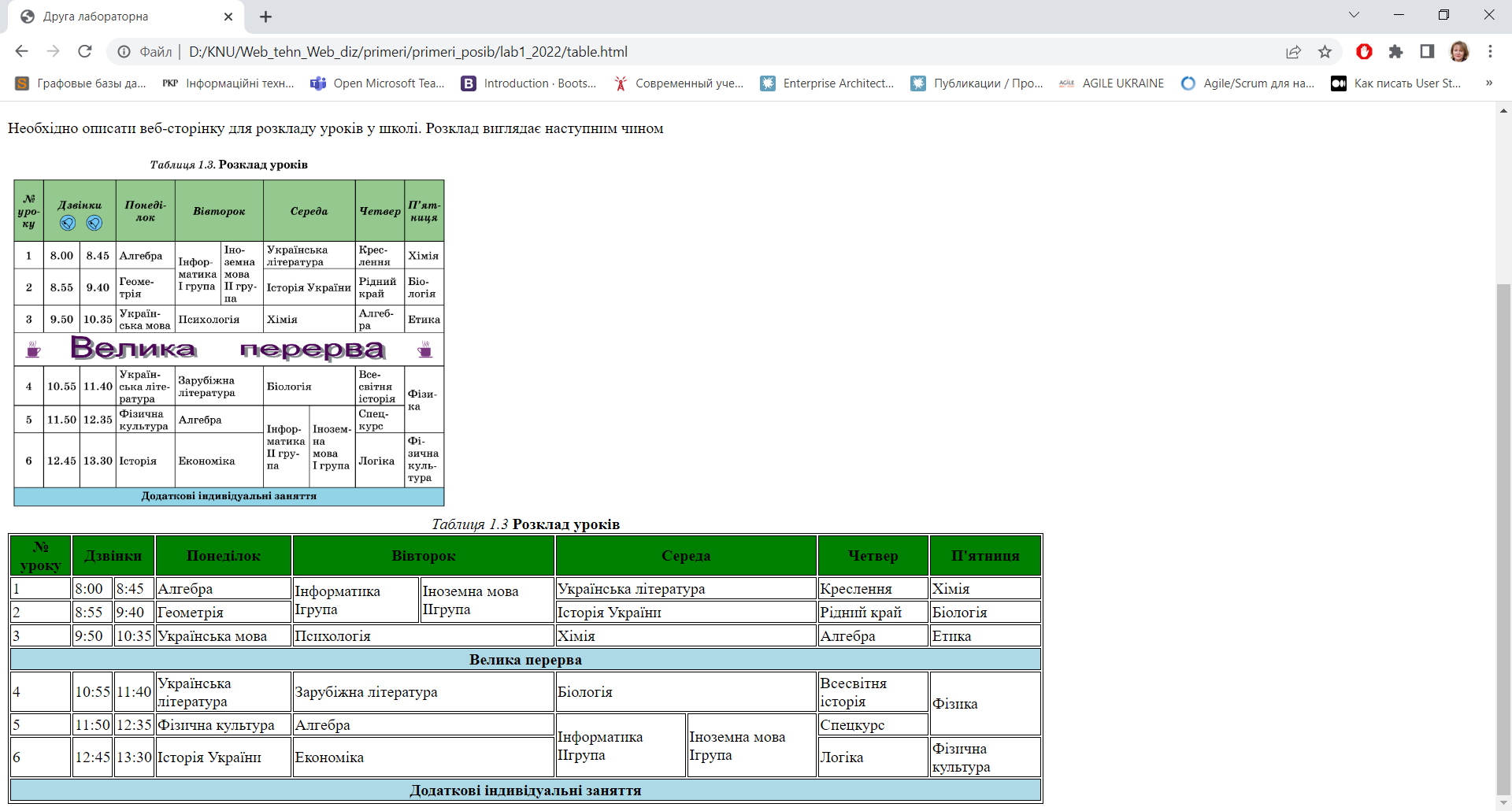


Рисунок 1.10 Результат виконання html коду

1. Розробимо html файл для виконання завдання по роботі з формами. По структурі він буде схожий на файл другої роботи, але необхідно розробити власну форму. Форма буде мати дві групи елементів «Книга» та «Автор». Спочатку розробимо першу частину форми, html код сторінки наведено на рис. 1.11, а зовнішній вигляд сторінки показано на рис. 1.12.

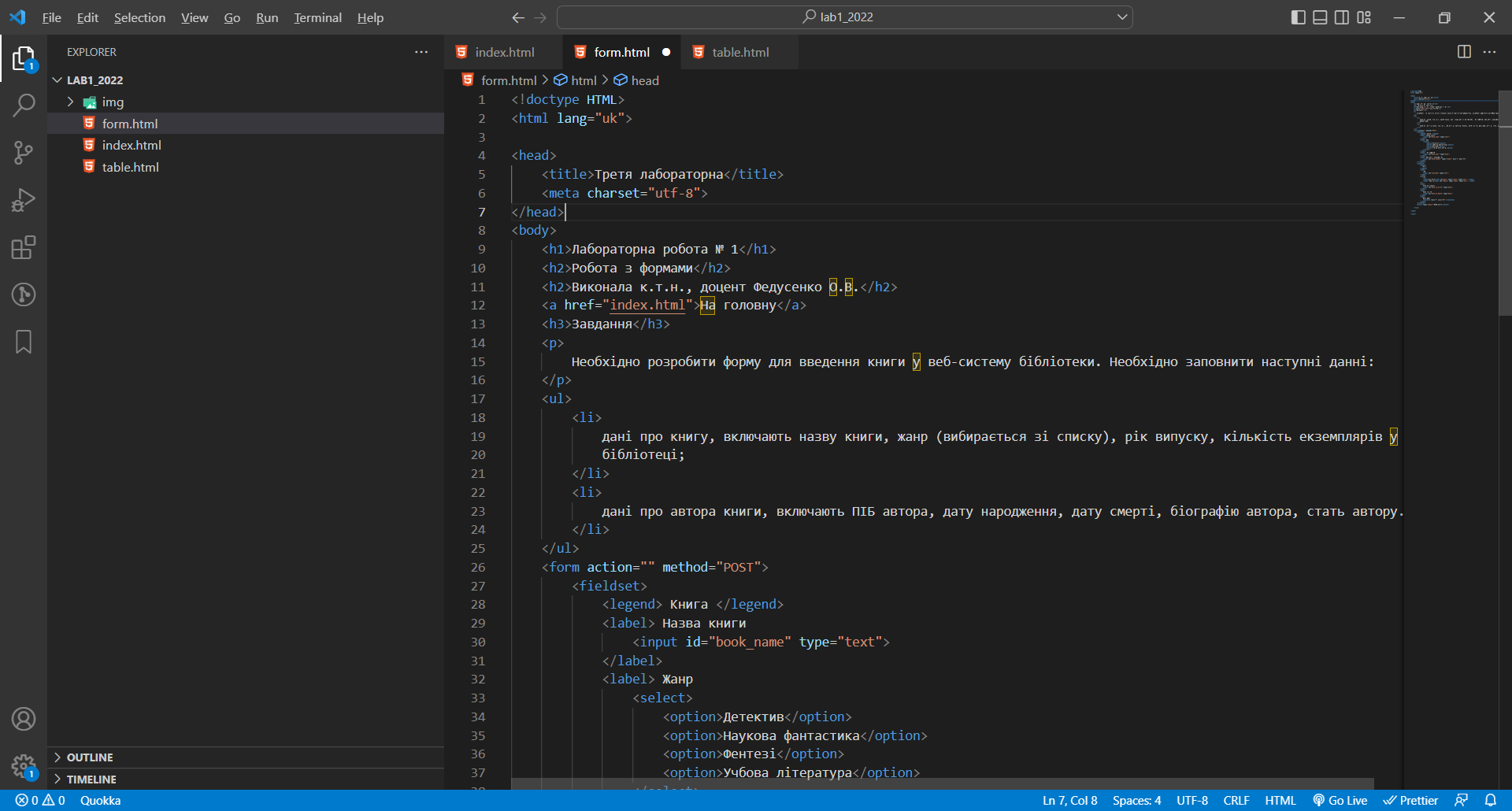
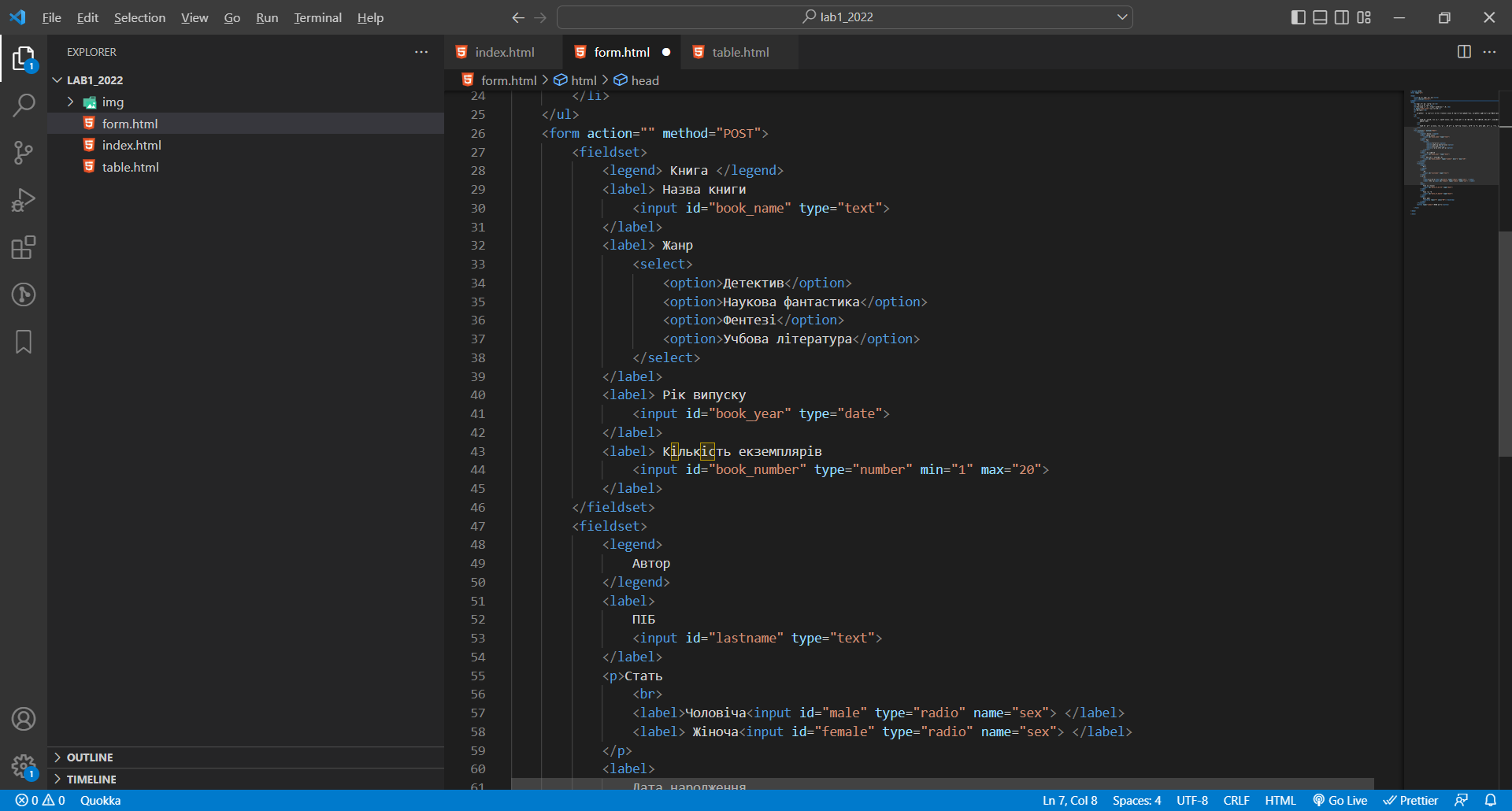
 

Рисунок 1.11 Html код першої частини завдання

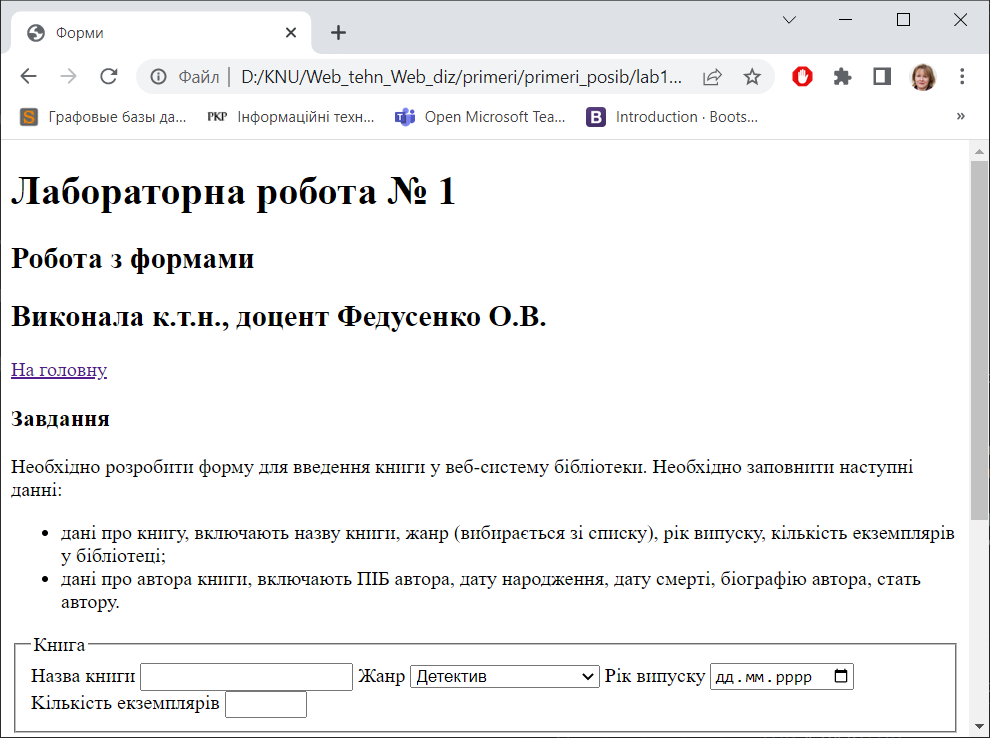


Рисунок 1.12 Сторінка з завданням

1. Розробимо другу частину форми з даними про автора та кнопку відправити на сайт. Для того, щоб при виборі статі автору користувач мав можливість орати лише один варіант необхідно дати тегам *<input>* з атрибутом *type*, який буде дорівнювати значенню *radio* дати однакове ім‘я. Тобто присвоїти однакове значення атрибуту *name*. Фрагмент html коду для опису другої частини форми наведено на рис. 1.13, а зовнішній вигляд сторінки з усією формою показано на рис. 1.14., а результати валідування за допомогою валідатору https://validator.w3.org/ наведено на рис. 1.15. Як ми бачимо по результатам валідування наша робота вміщує одну помилки, пов‘язану з тим, що ми не вказали значення атрибуту *action* тегу *form,* оскільки у нас ще не має скрипту обробки цієї форми.

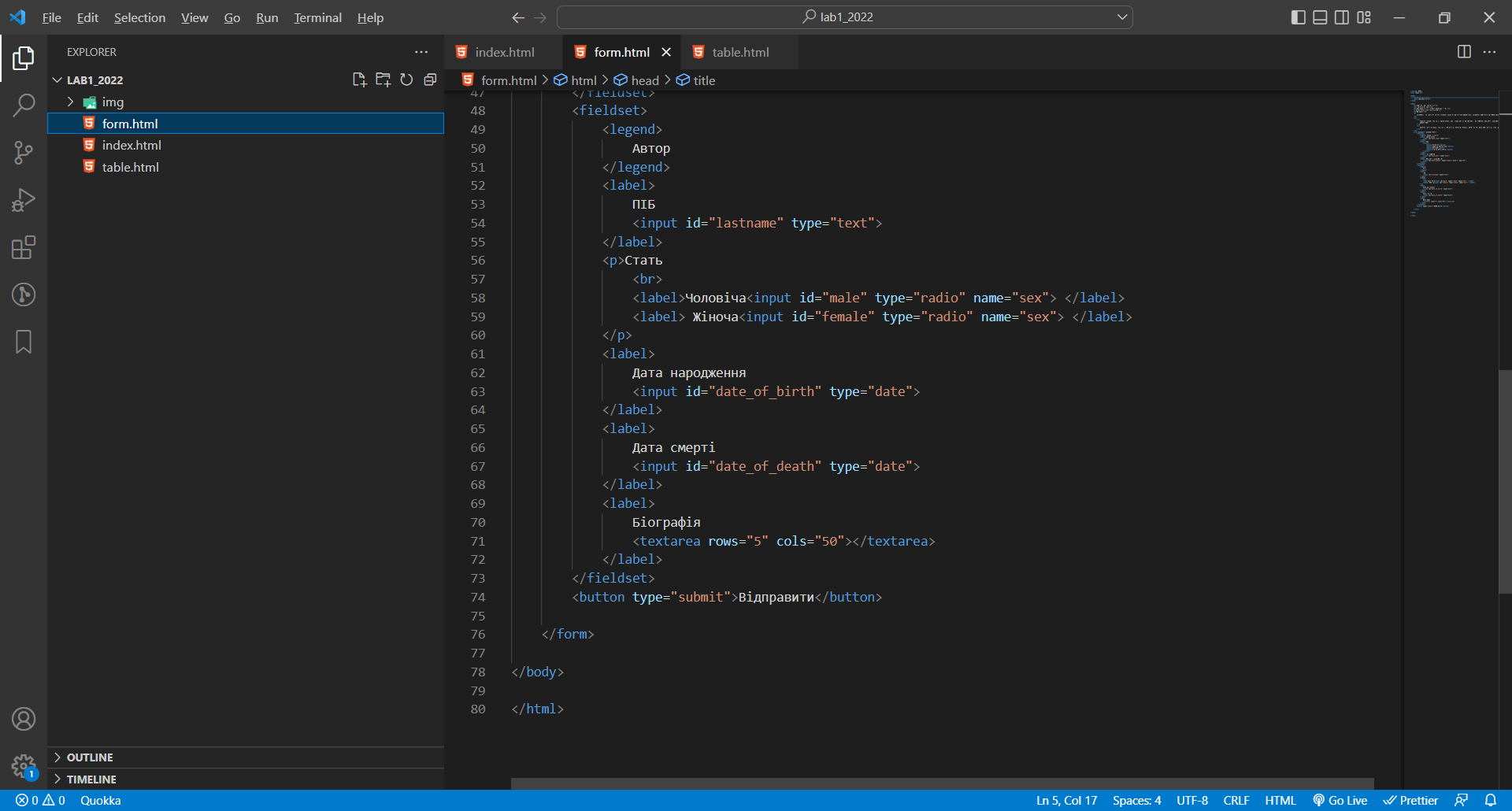


Рисунок 1.13 Html код другої частини завдання

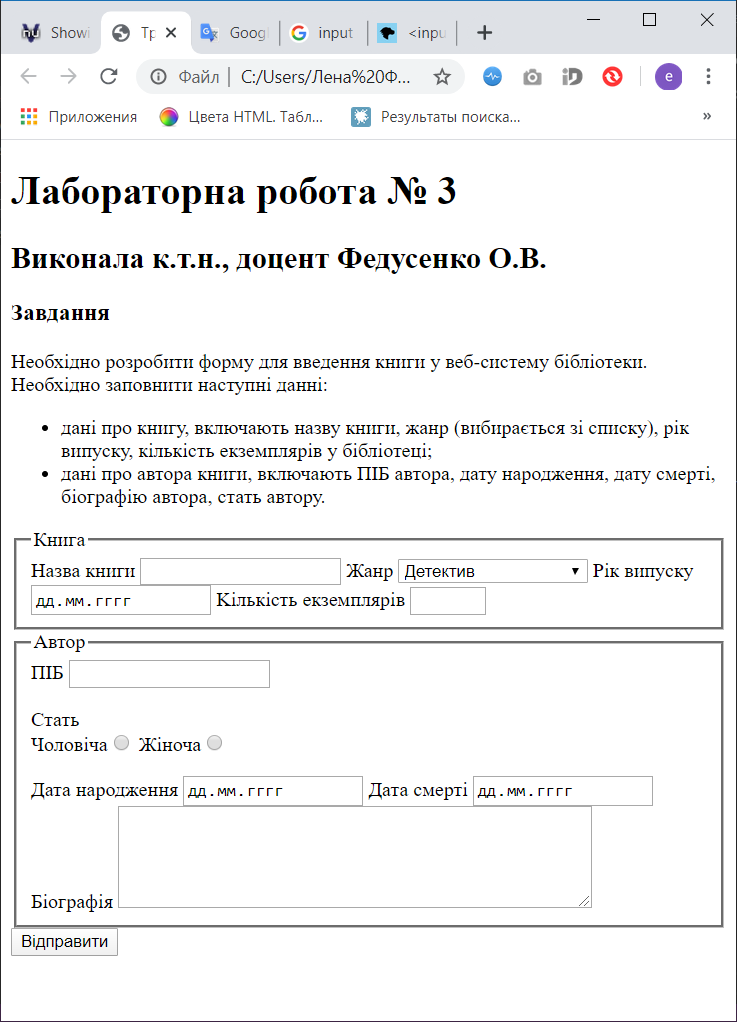


Рисунок 1.14 Результат виконання лабораторної роботи

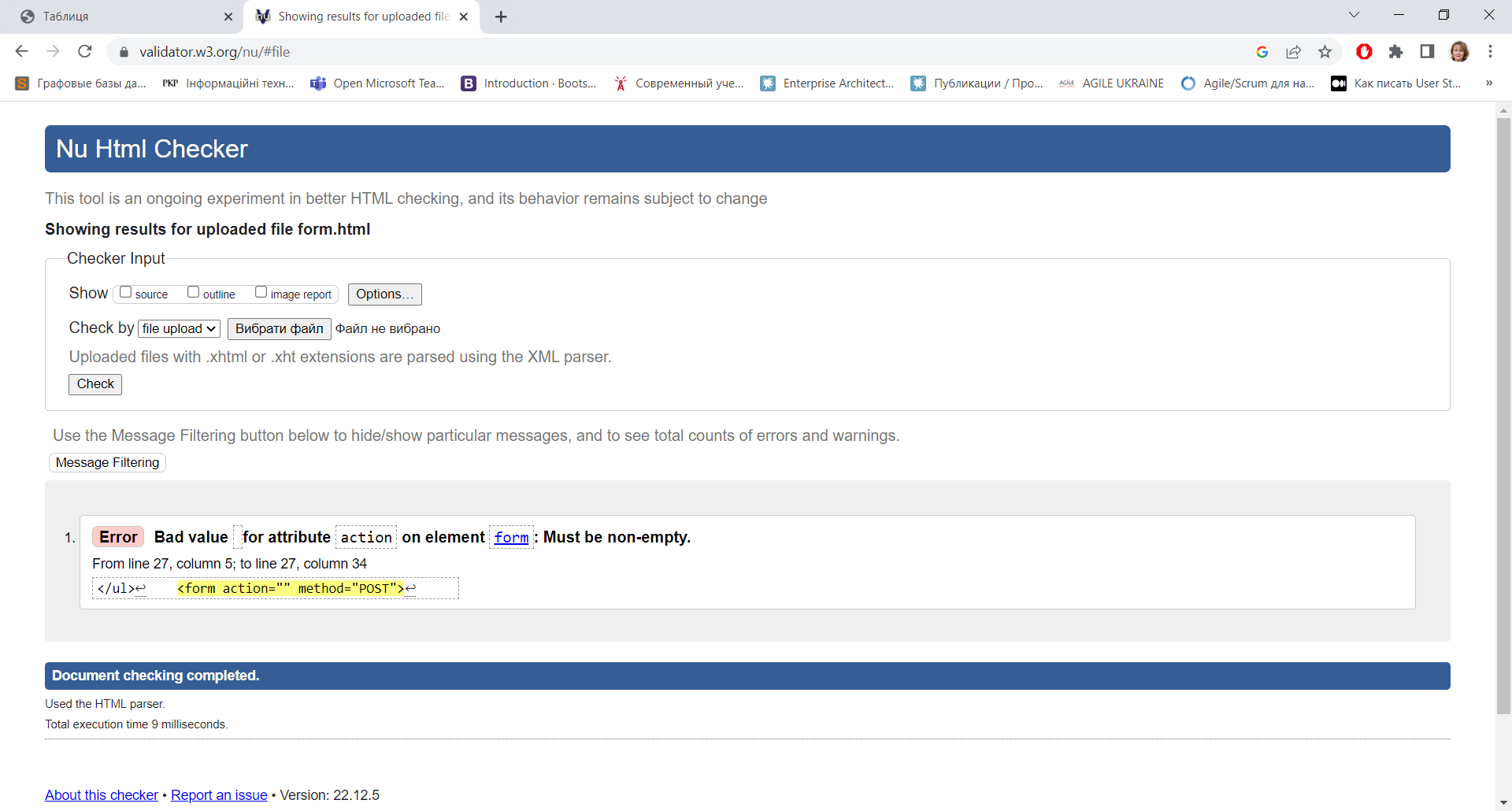
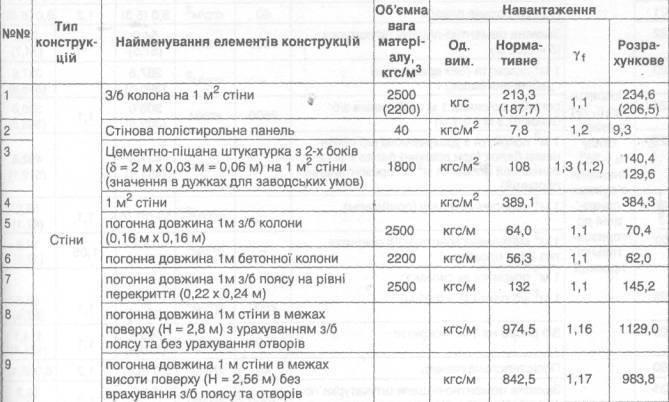


Рисунок 1.15 Результат валідуваня лабораторної роботи

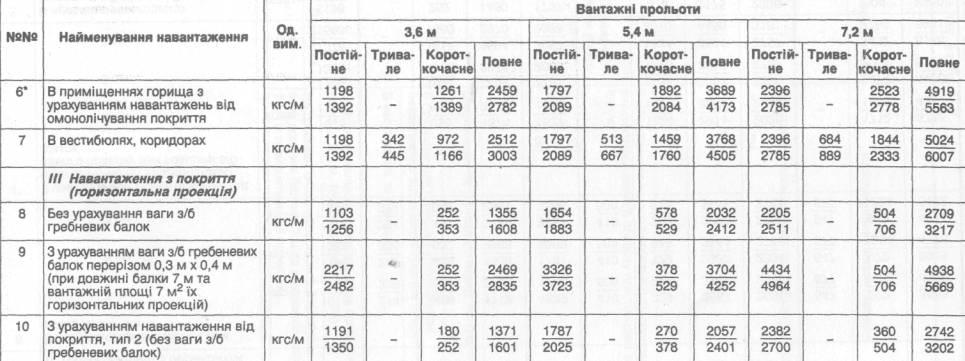
**Варіанти завдань**

**Таблиці**

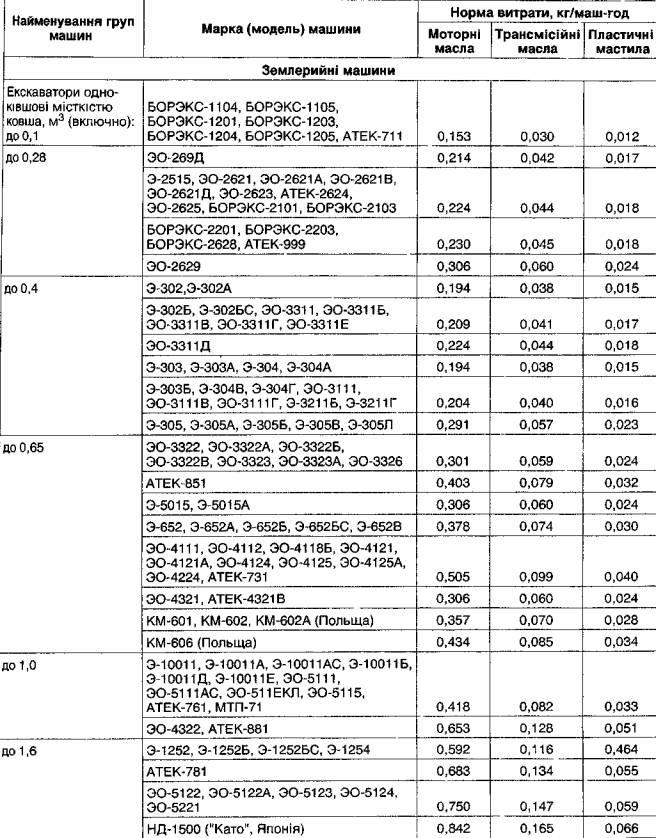
1



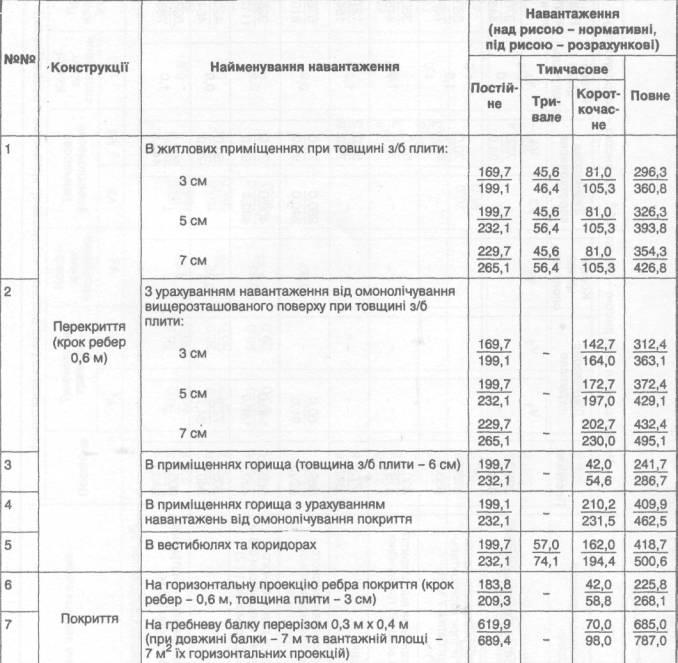
2



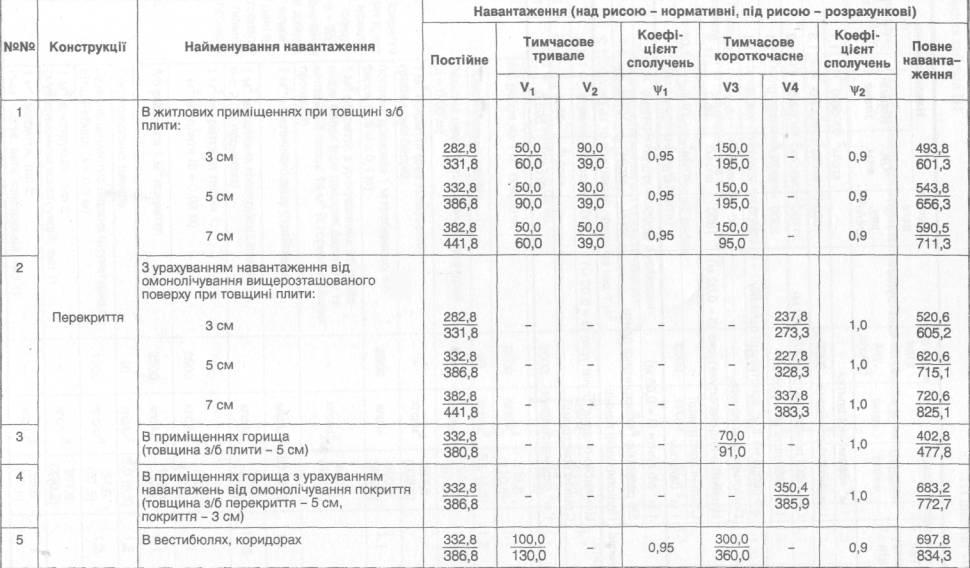
3.



4.



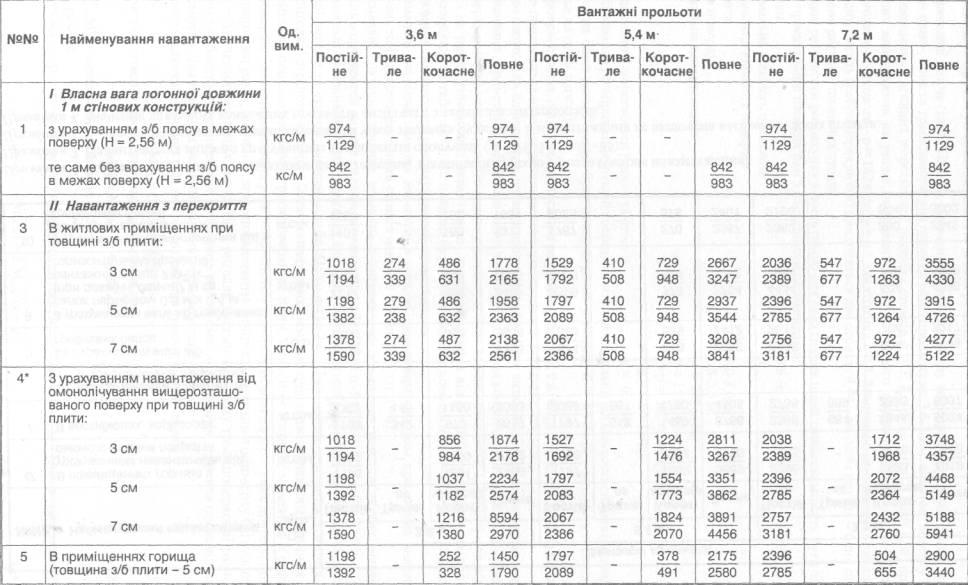
5.



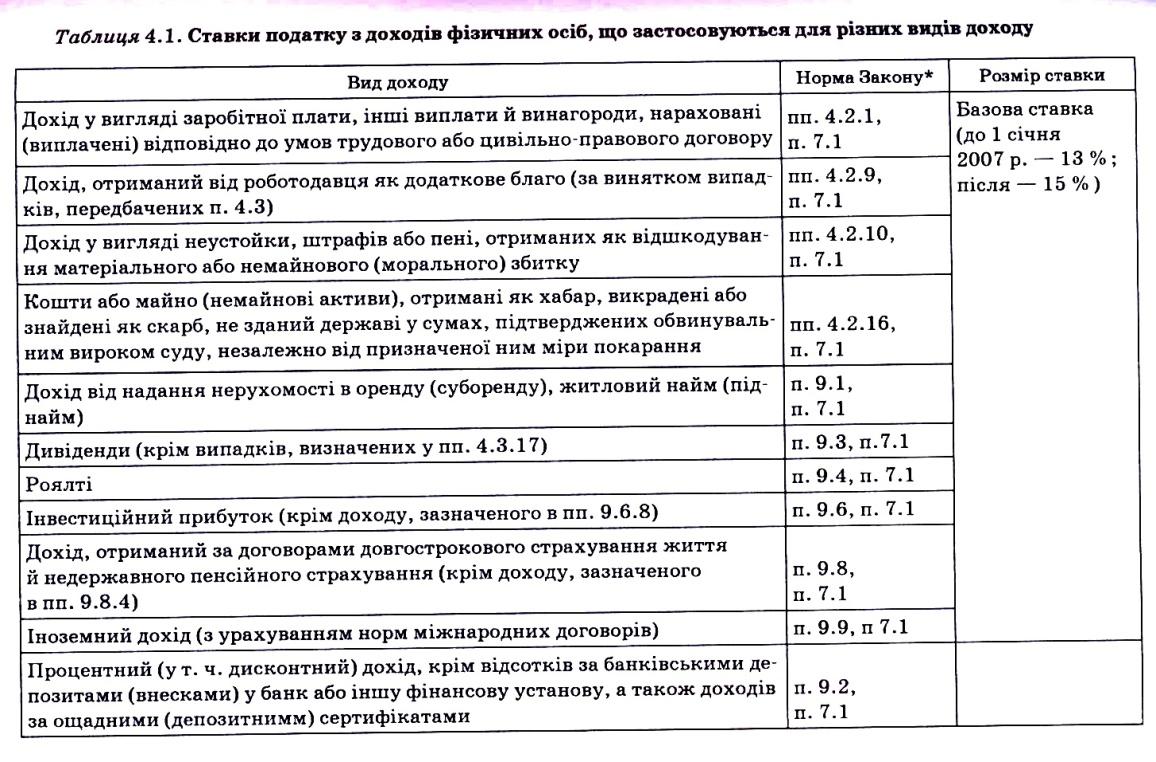
6.



7.



8.



9.



**Форми**

1. Необхідно розробити форму для введення результатів поточної сесії у веб-систему. Необхідно заповнити наступні данні:

* дані про студента, включають ПІБ, назву групи (вибирається із списку), оцінку за дисципліну (лише ціле значення від 0 до 100), дату проведення заліку або іспиту;
* дані про залік або іспит, включають назву дисципліни(вибирається із списку), перемикач це залік чи іспит, прізвище викладача.

1. Необхідно розробити форму для реєстрації студента у веб-системі. Необхідно заповнити наступні данні:

* основні дані про студента, включають ПІБ, назву групи (вибирається із списку), форму навчання(бюджет або контакт) реалізується у вигляді перемикача, коротку інформацію про студента, електрона адреса студента;
* особисті дані студента, включають дані паспорта, ідентифікаційний код, рік народження, стать (реалізується у вигляді перемикача).

1. Необхідно розробити форму для реєстрації учасника художнього форуму. Необхідно заповнити наступні данні:

* дані про реєстрацію, включають логін, пароль, електрону адресу, улюбленого художника (обирається зі списку);
* особисті дані, включають улюблений колір, посилання на веб-сторінку з власним портфоліо, улюблений жанр у живописі (обирається зі списку),докладний опис власних художніх вподобань.

1. Необхідно розробити форму для введення товару до веб-системи. Необхідно заповнити наступні данні:

* дані про товар, включають назву товару, тип товару(вибирається зі списку), дату виробництва, ціну, кількість на складі (ціле число від 1 до 100000);
* дані про виробника, включають назву, юридичну адресу, електрону пошту, посилання на веб-сайт, коментарі.

5. Необхідно розробити форму для введення інформації про фільм до БД фільмів. Необхідно заповнити наступні данні:

• дані про фільм, включають назву фільму, жанр(вибирається зі списку), дату виходу на екрани, опис фільму, картинку з афішою;

• дані про режисера, включають ПІБ, дату народження, біографію, посилання на веб-сайт.

6. Необхідно розробити форму для введення товару до інтернет-магазину. Необхідно заповнити наступні данні:

• дані про товар, включають назву товару, тип товару(вибирається зі списку), опис, поточну ціну, ціну зі знижкою;

• дані про виробника, включають назву, електрону пошту, посилання на веб-сайт, коментарі.

7. Необхідно розробити форму для введення інформації про виставу. Необхідно заповнити наступні данні:

• дані про виставу, включають назву вистави, жанр(вибирається зі списку), дата вистави, час вистави, опис вистави, картинку з афішою;

• дані про квиток, включають тип розміщення (вибирається зі списку, наприклад, партер, ложа бенуар, балкон і т.п.), номер місця, вартість квитка.

8. Необхідно розробити форму для введення інформації про тварину. Необхідно заповнити наступні данні:

• дані про тварину, включають кличку тварини, тип(вибирається зі списку, наприклад, собака кішка і т.п.), дата народження, колір;

• дані про власника тварини, включають ПІБ, телефон, електрону адресу, вік (ціле число від 18 до 100).