This lab session covers the basics of HTML, CSS, and Django template tags. By the end of this session, students should have a very good understanding of how to build a basic website that has several html files with static files and a simple menu.

**Pre-lab Preparation:**

1. Having bookmodule and usermodule in apps directory, each containing separate URL configurations.
2. Template directory located in ‘apps’ with necessary settings.

**Lab Activities:**

**Task 1: Build your first HTML file with Fundamental HTML Document and HTML tags (Lecture 4 – part1)**

**Step 1:** from any location, for example, in the home directory, create a folder “CS471\_lab”:

> mkdire ~/CS471\_lab

**Step 2:** Create four html files (index.html, aboutus.html, list\_books.html, and one\_book.html) within *CS471\_lab* folder, and add the following content:

|  |
| --- |
| **~/CS471\_lab/index.html** |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>My First Index HTML Page</title>  <meta name="keywords" content="Books, recent, Learning, library">  <meta name="description" content="Books website">  <meta name="author" content="author name">  <link rel="icon" href="qu-icon.png">  <meta charset="UTF-8">  </head>  <body>  <div>  <a href="index.html">Home</a>  <a href="list\_books.html">List Books</a>  <a href="aboutus.html">About</a>  </div>  <h1>This is heading 1 (Library)</h1>  <p>This is a paragraph text </p>  <a href="https://www.w3schools.com/html/default.asp">Learn more about HTML</a>  <hr>  <!-- This is a comment in HTML, not render in the bowser -->  <h2>This is heading 2</h2>  <p>This is some other text. <br> with newline <br> and another line</p>  <hr>    <h2>This is heading 2</h2>  <p>This is some other text.</p>  </body>  </html> |

|  |
| --- |
| **~/CS471\_lab/aboutus.html** |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>About Us</title>  <meta name="keywords" content="Books, recent, Learning, library">  <meta name="description" content="Books website">  <meta name="author" content="author name">  <link rel="icon" href="qu-icon.png">  <meta charset="UTF-8">  </head>  <body>    <div>  <a href="index.html">Home</a>  <a href="list\_books.html">List Books</a>  <a href="aboutus.html">About</a>  </div>    <h1>About us</h1>  <p>This is a paragraph text This is a paragraph text This is a paragraph text This is a paragraph text </p>    <p>To write to <a href="mailto:email@domain.com">our store</a>, click the link and your default email client will open an email message and address it to us. </p>  </body>  </html> |
| **~/CS471\_lab/list\_books.html** |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>List of Books</title>  <meta name="keywords" content="Books, recent, Learning, library">  <meta name="description" content="Books website">  <meta name="author" content="author name">  <link rel="icon" href="qu-icon.png">  <meta charset="UTF-8">  </head>  <body>    <div>  <a href="index.html">Home</a>  <a href="list\_books.html">List Books</a>  <a href="aboutus.html">About</a>  </div>      <h1>List of Books</h1>  <a href="one\_book.html"><h2>Internet & World Wide Web How to Program</h2></a>  <a href="one\_book.html"><img src="book1.jpg" alt="Book1 img desc" width="200" height="200"></a>  <hr>    <h2>2) C++ How to Program, Late Objects Version</h2>  <img src="book2.jpg" alt="Book2 img desc" width="200" height="200">  <hr>    <h2>3) Images in Another Folder</h2>  <img src="book3.jpg" alt="Book3 img desc" width="200" height="200">  <hr>    </body>  </html> |
| **~/CS471\_lab/one\_book.html** |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>Show one Book</title>  <meta name="keywords" content="Books, recent, Learning, library">  <meta name="description" content="Books website">  <meta name="author" content="author name">  <link rel="icon" href="qu-icon.png">  <meta charset="UTF-8">  </head>  <body>    <a href="list\_books.html">Go Back</a>    <h1>Details of Book1</h1>  <h2>Internet & World Wide Web How to Program</h2>  <h2>Author: author name</h2>  <p><strong> Internet & World Wide Web How to Program, 5/e </strong> provides a clear, simple, engaging and entertaining introduction to Internet and web programming. It’s appropriate for both introductory and intermediate-level client-side and server-side programming courses.</p>    <img src="book1.jpg" alt="Book1 img desc" width="200" height="200">    </body>  </html> |

**Step 3:** Download four images into the same directory as the html files, and rename them according to the following:

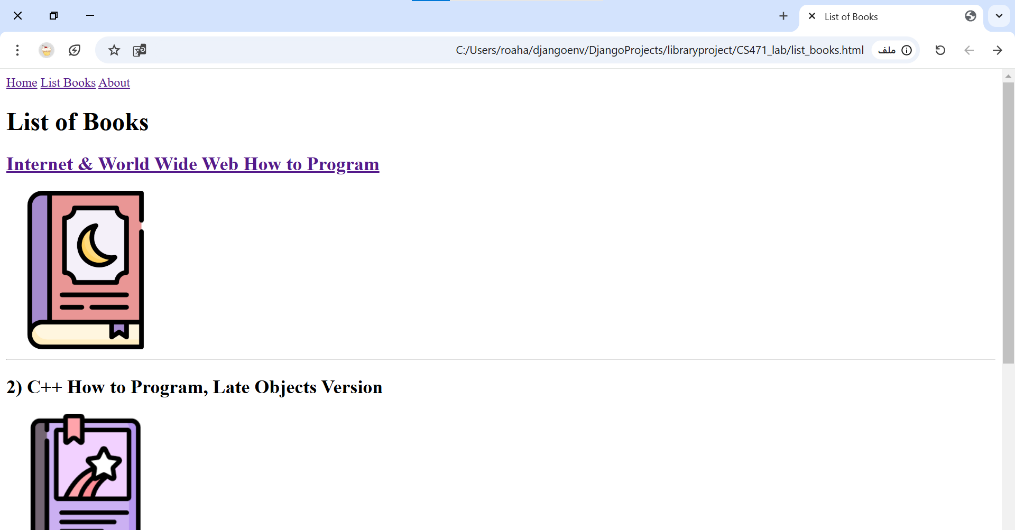
* The first image is for the webpage icon, with the name: qu-icon.png
* The other three images are for the three example books: book1.jpg, book2.jpg, and book3.jpg

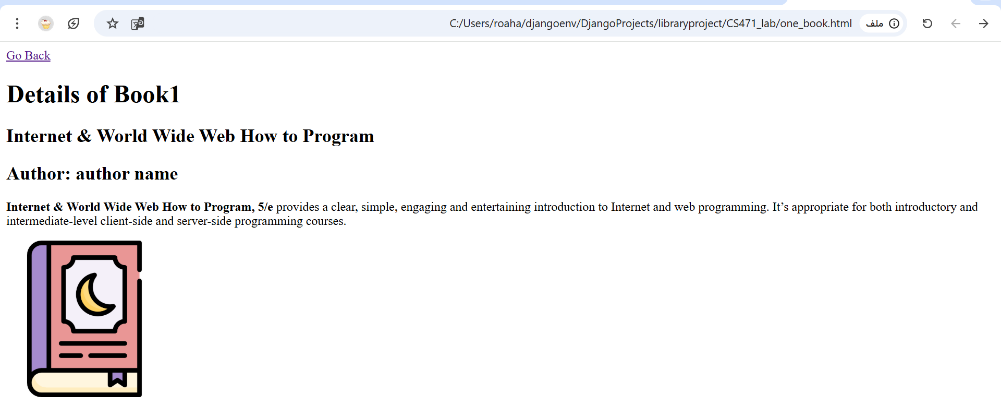
Note that these images are located in the same directory as the HTML5 document, so only the image’s file name is required. This is known as a relative path—the image is stored relative to the location of the document. Also, make sure that the image formats are exactly as needed (PNG or JPEG)[[1]](#footnote-2). Also, it is possible to use other formats, but you must modify the relative path accordingly in img and link tags.

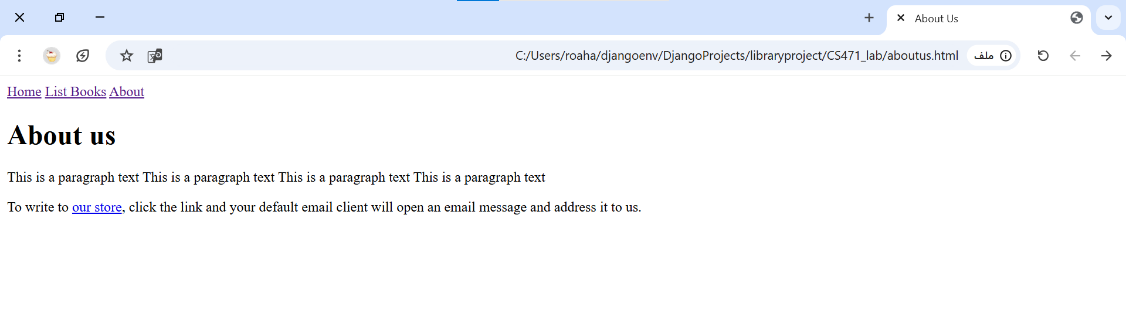
**Step 4:** To open an HTML5 examples from this task, open the folder where you saved the html files, and double click any file (index.html, for example), to open it in your default web browser.

**صورة تحتوي على نص, لقطة شاشة, برمجيات, الخط

تم إنشاء الوصف تلقائياً**

****

****

****

**Task 2: Style the previous HTML files with CSS in three ways (inline CSS, internal style element, and separated CSS file)**

**Step 1:** update the html files, to include CSS style:

|  |
| --- |
| **~/CS471\_lab/index.html** |
| . . .  <head>  . . .    <link rel="stylesheet" href="styles.css"> <!-- include css file -->  </head>  <body>    <div class="topnav"> <!-- add class attribute -->  <a class="active" href="index.html">Home</a>  <a href="list\_books.html">List Books</a>  <a href="aboutus.html">About</a>  </div>    <h1 style="color: blue;">This is heading 1 (Library)</h1> <!-- add inline css -->  <p style="color: white;background-color: black;">This is a paragraph text </p> <!-- add inline css -->    . . . |
| **~/CS471\_lab/aboutus.html** |
| . . .  <head>  . . .    <link rel="stylesheet" href="styles.css"> <!-- include css file -->  </head>  <body>    <div class="topnav"> <!-- add class attribute -->  <a href="index.html">Home</a>  <a href="list\_books.html">List Books</a>  <a class="active" href="aboutus.html">About</a>  </div>  . . . |
| **~/CS471\_lab/list\_books.html** |
| . . .  <head>  . . .    <style> /\* add internal css \*/  h1{  color: green;  }    h2{  color: lightskyblue;  }    p {  font-family: verdana;  font-size: 20px;  }    </style>    <link rel="stylesheet" href="styles.css"> <!-- include css file -->  </head>  <body>    <div class="topnav"> <!-- add class attribute -->  <a href="index.html">Home</a>  <a class="active" href="list\_books.html">List Books</a>  <a href="aboutus.html">About</a>  </div>  . . . |
| **~/CS471\_lab/one\_book.html** |
| . . .  <head>  . . .    <link rel="stylesheet" href="styles.css"> <!-- include css file -->  </head>  <body>  . . . |

**Step 3:** Create a new CSS file “styles.css” in the same directory as the html files, and insert the following content into the CSS file:

/\* Add a black background color to the top navigation \*/

.topnav {

background-color: #eeeeee;

width: 100%;

}

/\* Style the links inside the navigation bar \*/

.topnav a {

color: #000000;

text-align: center;

margin-right: 25px;

text-decoration: none;

font-size: 17px;

}

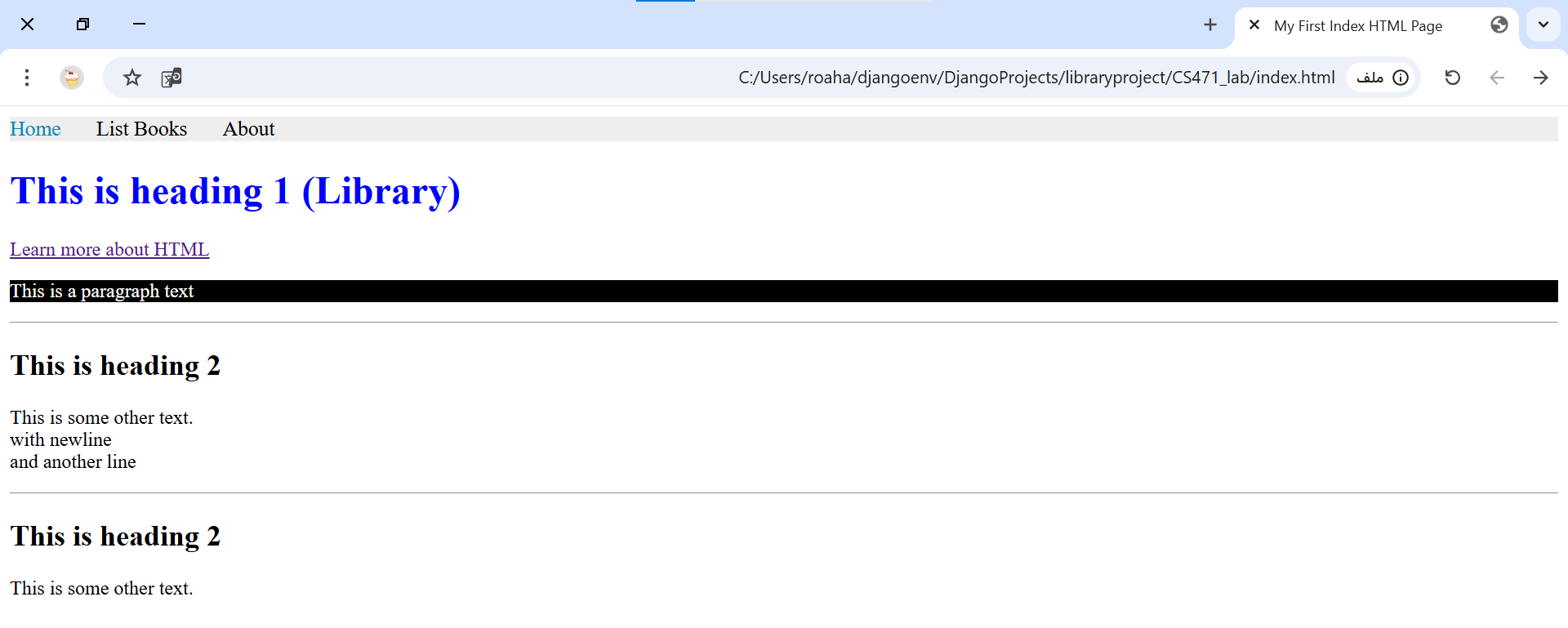
/\* Add a color to the active/current link \*/

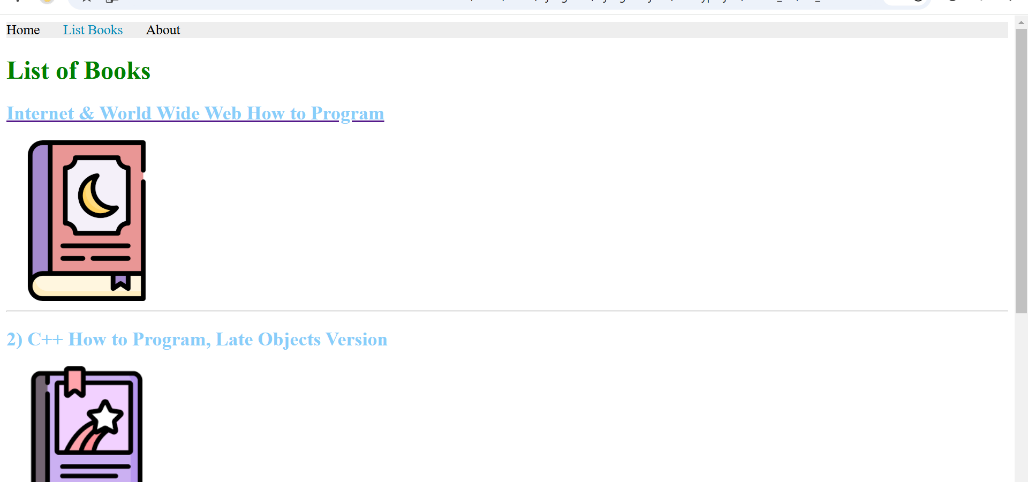
.topnav a.active{

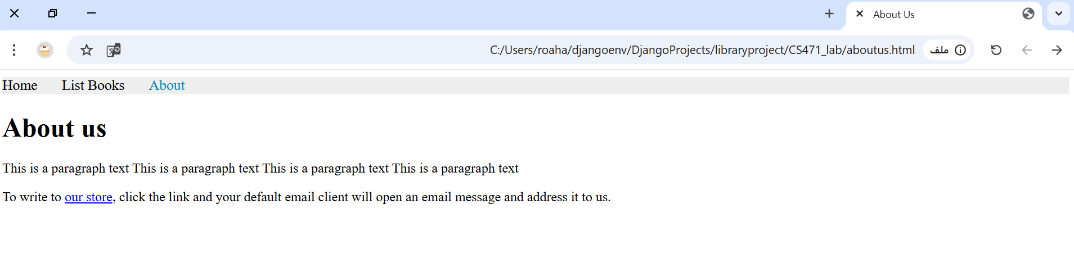
color: #008CBA;

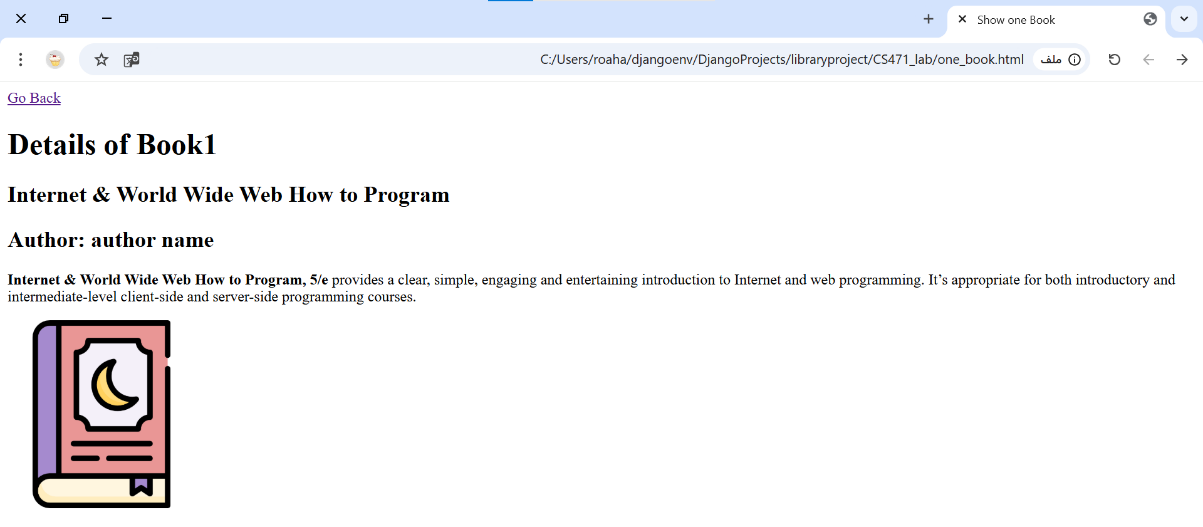
}

**Step 4:** Now double click on index.html, to open it in your default web browser.







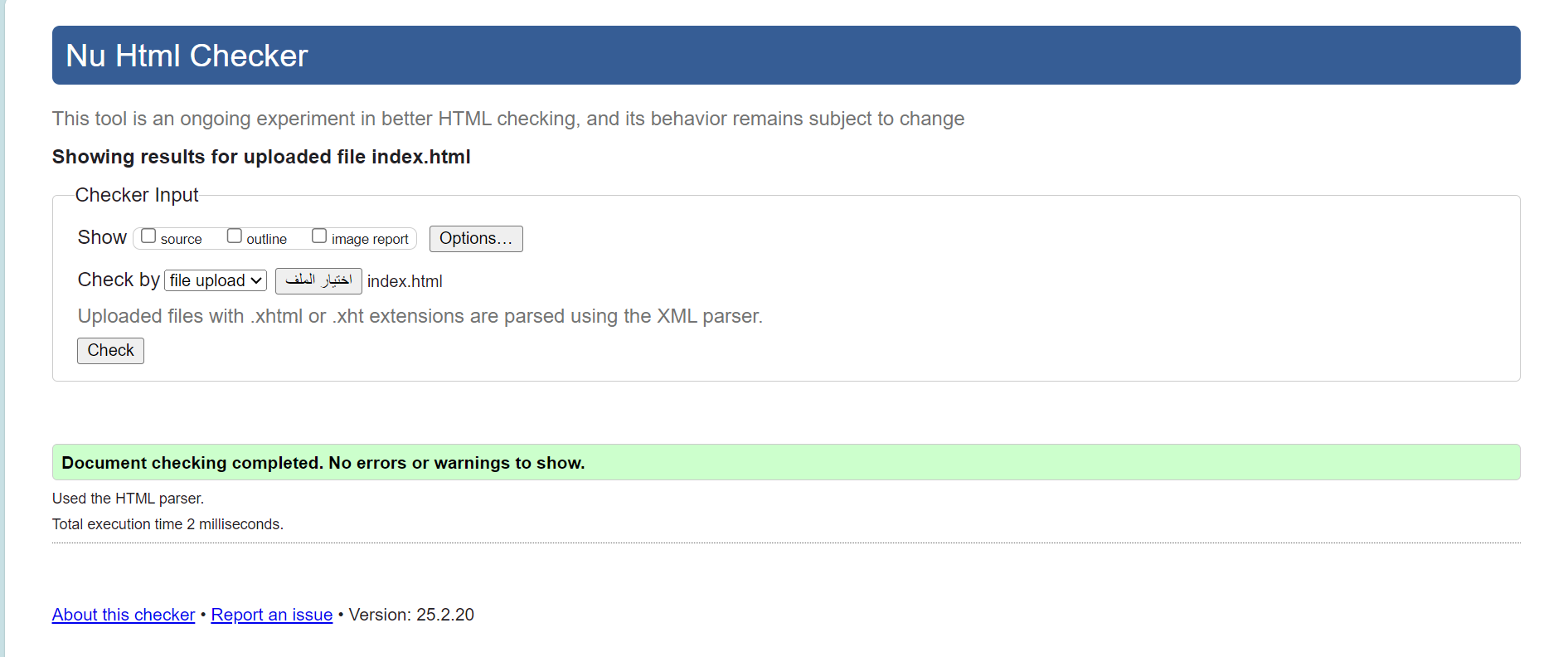


**Task 3: W3C HTML5 Validation Service[[2]](#footnote-3) to use proper HTML5 syntax to ensure that browsers process your documents properly.**

**Step 1:** Visit W3C HTML5 validation service at <https://validator.w3.org/#validate_by_upload>

**Step 2:** Click the Choose File button to select an HTML file from your computer to validate.

**Step 3:** Click the Check button to validate your document.

****

**Task 4: Implementation and management of the previous HTML and static files (e.g. CSS and images) in Django framework.**

**Step 1:** In the apps folder, create ‘static’ folder

**Step 2:** Update main/settings.py with static location to store CSS, images, and future static files

. . .

STATIC\_URL = 'static/'

STATICFILES\_DIRS = [( os.path.join(BASE\_DIR, "apps/static"))]

. . .

**Step 3:** Move the style.css and four image files to ‘apps/static‘ folder

**Step 4:** From the root folder, navigate to ‘apps/bookmodule/view.py’, then, in views.py, add the following view functions:

def index(request):

return render(request, "bookmodule/index.html")

def list\_books(request):

return render(request, 'bookmodule/list\_books.html')

def viewbook(request, bookId):

return render(request, 'bookmodule/one\_book.html')

def aboutus(request):

return render(request, 'bookmodule/aboutus.html')

**Step 5:** From the root folder, navigate to ‘apps/bookmodule/urls.py’, add a map to the view functions with names to the urlpatterns list:

urlpatterns = [

path('', views.index, name= "books.index"),

path('list\_books/', views.list\_books, name= "books.list\_books"),

path('<int:bookId>/', views.viewbook, name="books.view\_one\_book"),

path('aboutus/', views.aboutus, name="books.aboutus"),

]

**Step 6:** Move HTML files to ‘apps/templates/bookmodule‘ folder, update URLs (for hyperlink, style, and images) with URL or Static Django template tag, this is an example for only index.html file, and you should do the same for the rest of html files:

<!DOCTYPE html>

<html>

{% load static %} <!-- load static -->

<head>

<title>My First Index HTML Page</title>

<meta name="keywords" content="Books, recent, Learning, library">

<meta name="description" content="Books website">

<meta name="author" content="author name">

<link rel="icon" href="{% static 'qu-icon.png' %}"> <!-- static template tag -->

<meta charset="UTF-8">

<link rel="stylesheet" href="{% static 'styles.css' %}"> <!-- static template tag -->

</head>

<body>

<div class="topnav">

<a class="active" href="{% url "books.index" %}">Home</a> <!-- URL template tag -->

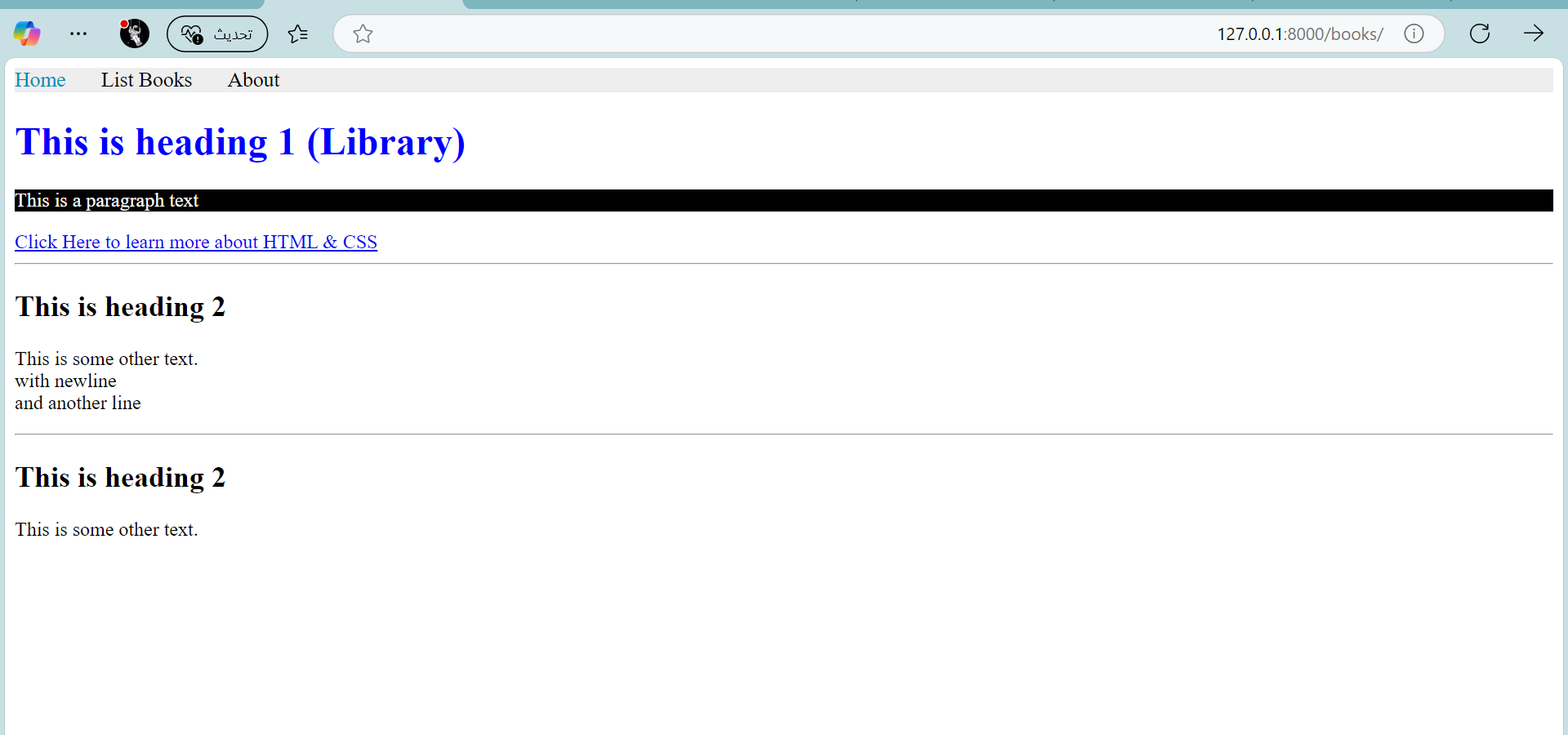
<a href="{% url "books.list\_books" %}">List Books</a> <!-- URL template tag -->

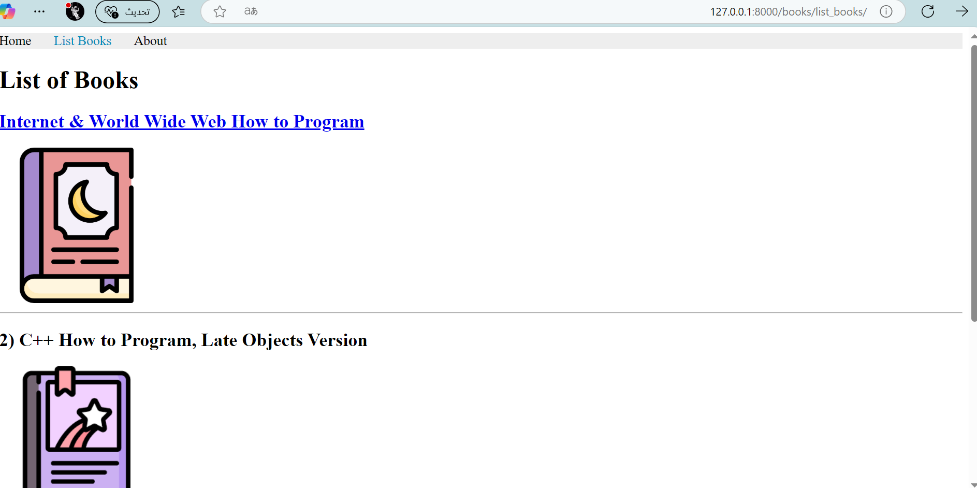
<a href="{% url "books.aboutus" %}">About</a> <!-- URL template tag -->

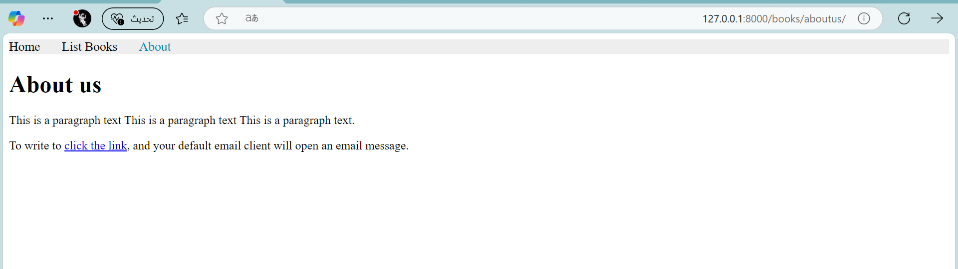
</div>

. . .

**Step 7:** Run the server if you haven’t, open the web browser, and navigate to <http://127.0.0.1:8000/books/>.







**Task 5: Additional implementation of Django tags to build one layout instead of repeating the layout for every HTML file.**

**Step 1:** In ‘apps/templates‘ folder, add these new folders and files:

templates/

…

├── includes/

│ ├── header.html

│ ├── footer.html

└── layouts/

│ ├── base.html

**Step 2:** In base.html, add the following html content:

<!DOCTYPE html>

<html>

{% load static %}

<head>

<title>{% block title %} {% endblock title %}</title>

<meta name="keywords" content="Books, recent, Learning, library">

<meta name="description" content="Books website">

<meta name="author" content="author name">

<link rel="icon" href="{% static 'qu-icon.png' %}">

<meta charset="UTF-8">

<link rel="stylesheet" href="{% static 'styles.css' %}"> <!-- include css file -->

{% block stylesheets %}{% endblock stylesheets %}

</head>

<body>

{% include "includes/header.html" %}

{% block content %} {% endblock content %}

{% include "includes/footer.html" %}

</body>

</html>

**Step 3:** In header.html, add the following html content:

{% with request.resolver\_match.url\_name as url\_name %}

<div class="topnav">

<a class="{% if url\_name == 'books.index' %}active{% endif %}" href="{% url "books.index" %}">Home</a>

<a class="{% if url\_name == 'books.list\_books' %}active{% endif %}" href="{% url "books.list\_books" %}">List Books</a>

<a class="{% if url\_name == 'books.aboutus' %}active{% endif %}" href="{% url "books.aboutus" %}">About</a>

</div>

{% endwith %}

**If you want** a simple html content for header.html without class attribute in a tag, then add this instead:

**<div class="topnav">**

**<a href="{% url "books.index" %}">Home</a>**

**<a href="{% url "books.list\_books" %}">List Books</a>**

**<a href="{% url "books.aboutus" %}">About</a>**

**</div>**

**Step 4:** update index.html to:

{% extends "layouts/base.html" %}

{% load static %}

{% block title %} My First Index HTML Page {% endblock title %}

{% block content %}

<h1 style="color: blue;">This is heading 1 (Library)</h1> <!-- add inline css -->

<p style="color: white;background-color: black;">This is a paragraph text </p> <!-- add inline css -->

<a href="https://www.w3schools.com/html/default.asp">Click Here to learn more about HTML & CSS</a>

<hr>

<!-- This is a comment in HTML, not render in the bowser -->

<h2>This is heading 2</h2>

<p>This is some other text. <br/> with newline <br/> and another line</p>

<hr>

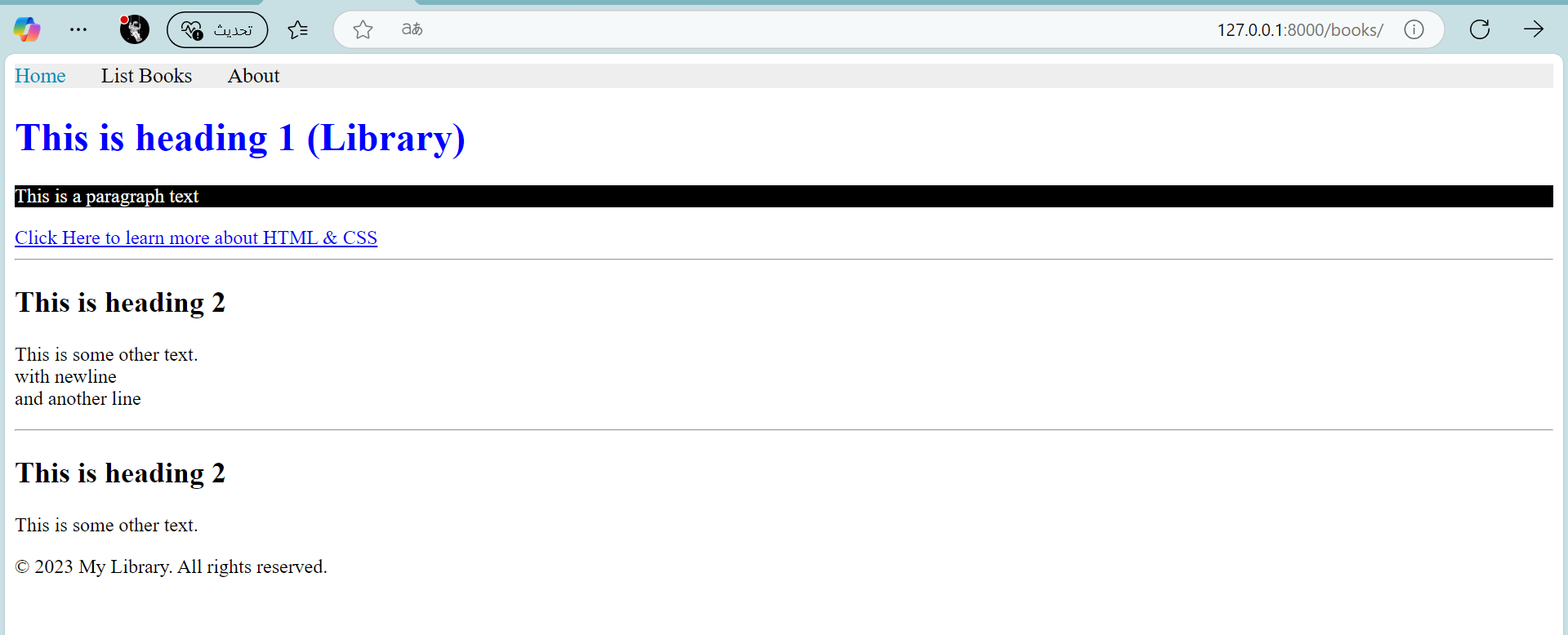
<h2>This is heading 2</h2>

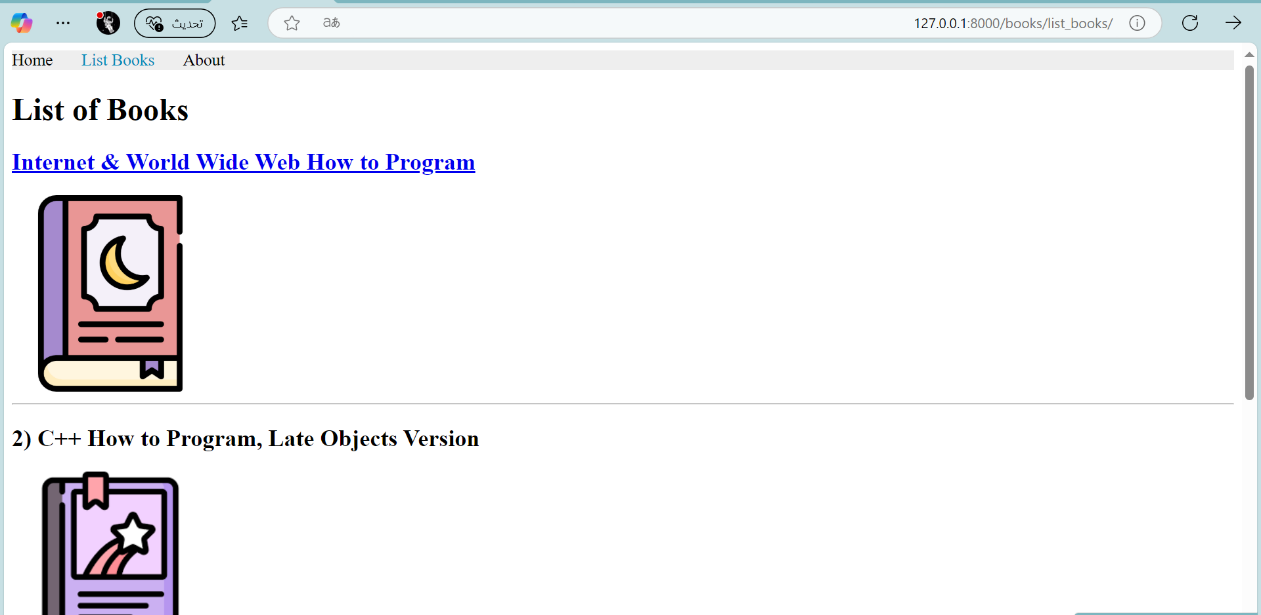
<p>This is some other text.</p>

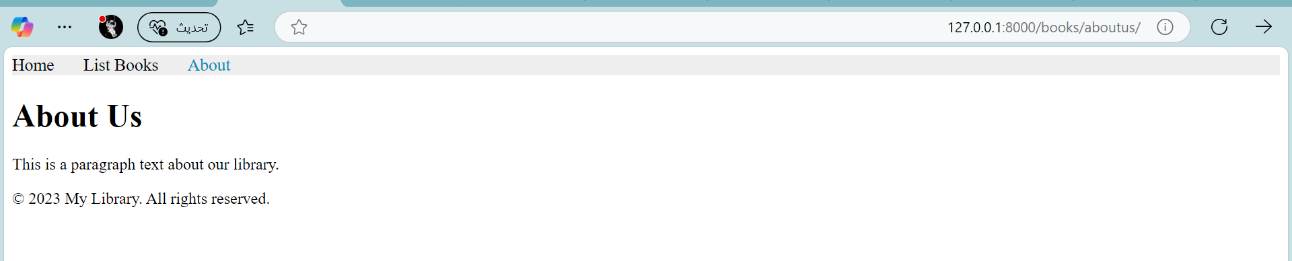
{% endblock content %}

**Step 5:** Update the remaining HTML files (aboutus.html, list\_books.html, and one\_book.html), similar to index.html, with the necessary changes according to Django template tags.

**Step 6:** Now from the browser navigate to <http://127.0.0.1:8000/books/>





****

1. The most popular image formats used by web developers are PNG (Portable Network Graphics) and JPEG (Joint Photographic Experts Group), also recently SVG (Scalable Vector Graphics) [↑](#footnote-ref-2)
2. The World Wide Web Consortium (W3C) provides a validation service (at validator.w3.org) for checking a document’s syntax [↑](#footnote-ref-3)