

# Table of Contents

[Table of Contents 2](#_Toc196256181)

[Introduction 2](#_Toc196256182)

[HTML Structure Breakdown 3](#_Toc196256183)

[Document Declaration & Head Section 3](#_Toc196256184)

[Hero Section (Welcome Banner) 5](#_Toc196256185)

[Login Section 5](#_Toc196256186)

[Book Search Section 6](#_Toc196256187)

[Main Content Section (Search Results) 7](#_Toc196256188)

[Wishlist Section 7](#_Toc196256189)

[Footer Section 8](#_Toc196256190)

[JavaScript Integration 8](#_Toc196256191)

[Conclusion for HTML 9](#_Toc196256192)

[CSS JavaScript Report for "WORLD OF BOOKS" 10](#_Toc196256193)

[Introduction 10](#_Toc196256194)

[Book Data Structure 11](#_Toc196256195)

[Search Functionality 12](#_Toc196256196)

[Display Results 12](#_Toc196256197)

[User Login System 13](#_Toc196256198)

[Wishlist Feature 14](#_Toc196256199)

[DOM Interactions & Event Handling 15](#_Toc196256200)

[Conclusion 16](#_Toc196256201)

[References 16](#_Toc196256202)

# Introduction

The process of developing an interactive web application for a bookshop involves more than simply creating code; it also involves moulding a user experience that is both intuitive and entertaining. As part of this project, I intended to develop a website as part of my assessment for a bookshop by utilising HTML, CSS, and JavaScript The objective of the website is to provide users with the ability to log in, search for books, save their favourites to a wish list, and peruse the results, all within a layout that is both visually appealing and responsive. The path, decisions, and problem-solving that went into the development of the final product are all included in this report, in addition to the product itself.

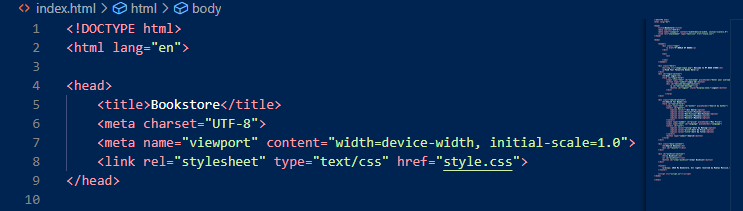
This report presents an in-depth and user-friendly analysis of the HTML and CSS structure that was utilised in the construction of the "WORLD OF BOOKS" online bookshop. Within this page, each component of the HTML and CSS code is broken down and explained in detail, including its structure, its purpose, and the manner in which it contributes to the overall operation of the web application.

# **HTML Structure Breakdown**

## **Document Declaration & Head Section**

The document begins with the HTML5 declaration, ensuring compatibility with modern

Browsers and improving rendering consistency:

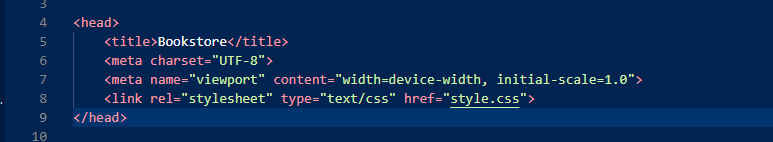


<!DOCTYPE html> tells the browser that this is an HTML5 document.

<html lang="en"> sets the language of the page to English.

**Head Section**

The section contains critical metadata and resource links needed for the website's functionality and presentation. Key components include:



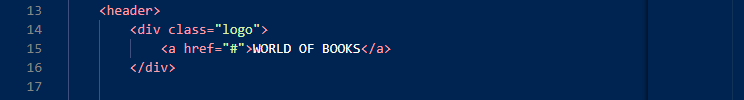
The <head> section includes metadata about the document:

* <title> sets the name as Bookstore shown in the browser tab.
* <meta charset="UTF-8"> ensures the website can display all characters properly.
* <meta name="viewport"...> makes the website responsive on mobile devices.
* <link rel="stylesheet"...> links the external CSS file as “style.css” for page styling.

**Head Section (Welcome Banners)**

This section contains the logo and navigation menu, providing users with quick access to key sections of the website:

That introduces visitors to the website with a welcoming message and a visually appealing logo:

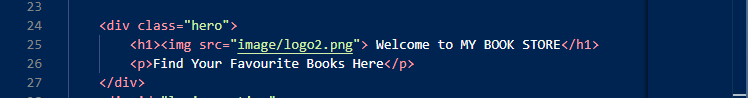


Explanation:

The <header> is used to hold the top section of the site, including branding and navigation.

<div class="logo"> contains a clickable logo that links to the homepage.

## Hero Section (Welcome Banner)

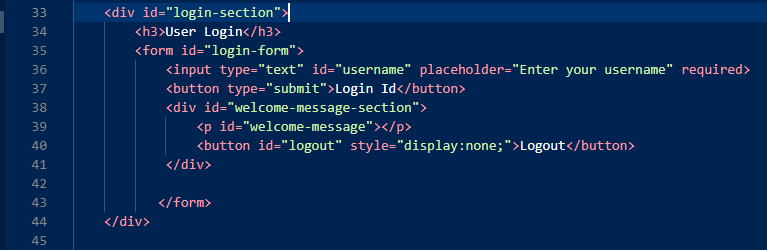


Explanation:

* This <div class="hero"> creates a prominent welcome banner.
* The <img> displays a logo next to the heading.
* <h1> shows the main welcome message as “Welcome to My BOOK STORE”.
* <p> gives users a short description of what the site offers.

## Login Section

This section facilitates user authentication, enhancing personalisation and access to user-specific features:



Features:

This section lets users log in by entering their username.

<form> handles the login inputs.

<input> accepts the username.

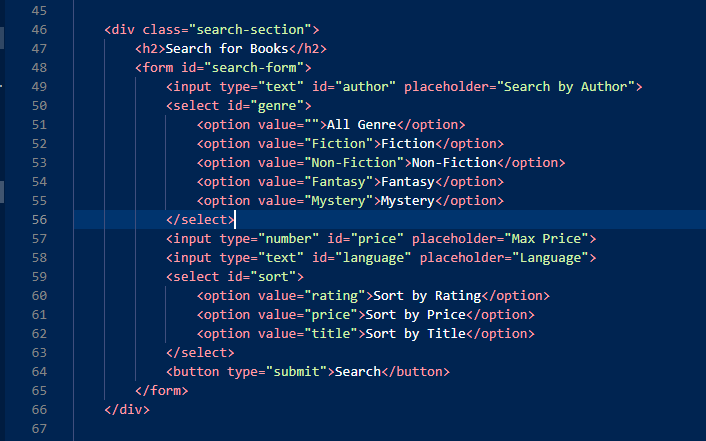
<button type="submit"> triggers the login logic in JavaScript.

<p id="welcome-message"> displays a greeting like “Welcome, [Name]”.

<button id="logout"> allows users to log out and is hidden by default using style="display:none;".

## Book Search Section

The search functionality is implemented using filters and inputs, enabling users to find books based on various criteria:

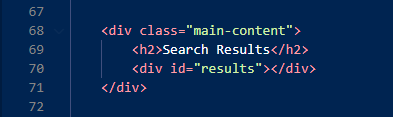


Key Elements:

* This form allows users to filter books by multiple criteria:
* Author name
* Genre (dropdown)
* Maximum price
* Language
* Sorting option (rating, price, or title)
* <button type="submit"> triggers the search function in the JavaScript file.

## Main Content Section (Search Results)

A dedicated area displays search results dynamically:

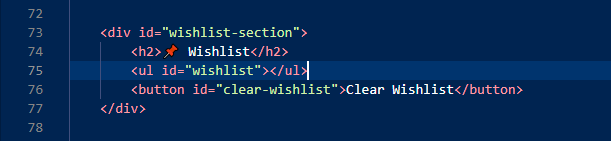


Purpose:

* This section displays the books found during the search.
* <div id="results"> is dynamically filled with book cards using JavaScript.

## Wishlist Section

The wishlist feature allows users to save and manage a list of desired books:

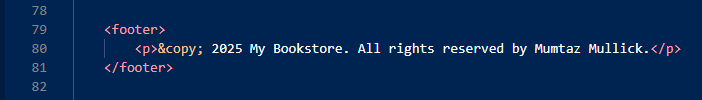


Explanation:

* Users can save their favorite books to this section.
* <ul id="wishlist"> lists saved books, which are stored in the browser using localStorage.
* <button id="clear-wishlist"> lets users delete all wishlist items with one click.

## Footer Section

The footer includes copyright information and credits:

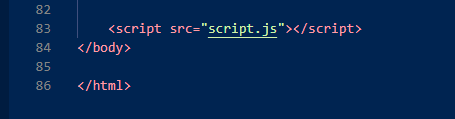


Explanation:

* The footer shows copyright.
* &copy; is an HTML entity used for the © symbol.

## JavaScript Integration

The HTML file references an external JavaScript file:



Explanation:

* This line connects the HTML page to the JavaScript file named script.js.
* All login, search, wishlist, and dynamic interactions are handled by JavaScript.

**Key Features Implemented:**

* Semantic HTML for accessibility
* Login/logout system
* Book search with multiple filters
* Wishlist with persistence using localStorage
* Responsive design support

**Recommendations:**

* **Add Navigation Links:** Add Home, Books, Contact, etc. in the header.
* **Form Validation:** Add error messages or alerts if fields are left empty.
* **Accessibility Enhancements:** Use aria-label, role, and screen-reader support.
* **Image Handling:** Confirm all image files exist and have descriptive alt text

## Conclusion for HTML

An online bookshop may be built on top of the "My Bookshop" HTML framework, which incorporates key functions including wishlist management, product search, and user identification. Accessibility, functionality, and a user-friendly experience are guaranteed by the careful application of semantic components, responsive design techniques, and outside resources. This framework has the potential to develop into a dynamic and captivating e-commerce platform when combined with CSS style and JavaScript capabilities.

With features like book search, user login, and wishlist management, this script offers the "My Bookshop" website full functioning. In addition to improving interaction, dynamic DOM manipulation guarantees data permanence. This script's modular design and organisation provide a solid basis for an easy-to-use e-commerce site.

The online bookshop program effectively incorporates a wishlist system, book search features, and user login functions. The project gives customers a dynamic and captivating experience by skilfully integrating HTML, CSS, and JavaScript.

Future enhancements could include:

* Including a toggle for dark mode to increase accessibility.
* Using keyword matching to increase the search system's accuracy.
* To improve security, user authentication should be implemented.

# **CSS JavaScript Report for "WORLD OF BOOKS"**

## **Introduction**

This report explains the functionality and structure of the script.js file used in the "WORLD OF BOOKS" interactive web application. The purpose of the JavaScript file is to provide interactivity and dynamic behavior to the website, such as book searching, user login, and a personalized wishlist. The script interacts with the HTML elements on the page using the Document Object Model (DOM) and stores user data locally using local Storage.

## **Book Data Structure**

At the beginning of the script, a books array is defined. Each book is stored as an object containing key details: title, author, genre, price, language, rating, and an image path. This array acts as the database for all search and display operations throughout the site.



## **Search Functionality**

The searchBooks() function is triggered when a user submits the search form. It filters the books array based on user input in the following fields:

* Genre
* Author
* Maximum price
* Language

All filters are case-insensitive, and partial matches are allowed for author and language. The function returns all matching books. Additionally, it finds the best-rated book in the results and highlights it as the "Best Match".



## Display Results

The displayResults() function is responsible for visually showing the filtered results in the HTML. It creates a new div element for each book and populates it with the book's image, title, author, price, and rating. If the book is the best match, a special label is added.



### **User Login System**

The script allows a user to log in by entering a username, which is then stored in the browser's localStorage. Once logged in, a welcome message is displayed. Logging out removes the username from storage and hides the welcome message.

The relevant functions include:

* displayUser() – Shows or hides the welcome message based on login status.
* logout() – Clears the stored username and updates the interface.

This simple login system does not require a password and is purely for personalization and feature access like the wishlist.



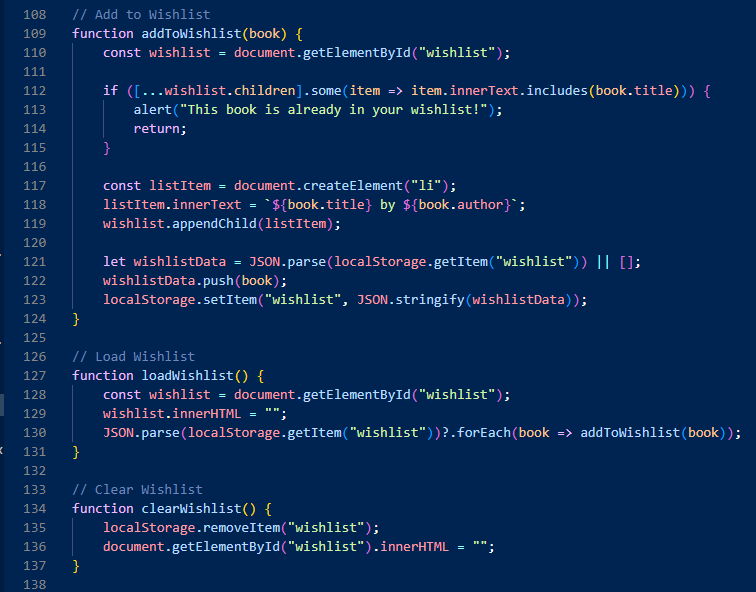
## **Wishlist Feature**

Users can add books to a wishlist, which is displayed as a list on the page. When the "Add to Wishlist" button is clicked, the selected book is added both to the HTML list and to the local Storage.

Key points:

* Duplicate books are not allowed in the wishlist.
* Data persistence is achieved with localStorage, so the wishlist is remembered between sessions.
* On page load, the load Wishlist() function retrieves any saved wishlist data and repopulates the list.

There is also a "Clear Wishlist" button, which clears the list both from the page and from storage.

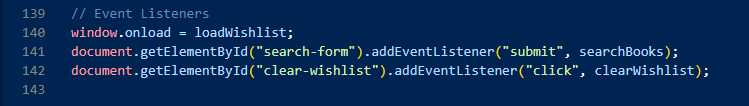


## **DOM Interactions & Event Handling**

Several event listeners are attached to form submissions and buttons:

* When the user submits the search form, searchBooks() is called.
* When the user logs in using the login form, displayUser() is triggered.
* When the page loads, loadWishlist() is called automatically using window.onload.
* A "Clear Wishlist" button is linked to the clearWishlist() function.

These interactions create a smooth user experience and allow real-time updates without reloading the page.



## **Conclusion**

Overall, the script.js file is a key component of the "WORLD OF BOOKS" application. It adds important interactivity by enabling users to search for books, log in, and manage a wishlist. The use of localStorage provides basic data persistence, and the code is structured clearly with appropriate functions and event listeners. While the login system is simple, it meets the needs of a user-friendly web application for browsing and saving favorite books.

The script could be further improved by adding user feedback messages, form validation, or even integrating with a real backend in the future. However, as it stands, it effectively demonstrates JavaScript functionality, DOM manipulation, and client-side storage.

# References

The Code and Report assessed by Mumtaz Mullick. I create a code in Visual Studio