



**Informatics Institute of Technology**  
**Department of Computing**

BSc(Hons) Artificial Intelligence and Data Science

**Module: CM1605 Web Technology**

**Module Coordinator: Ms. Janani Harischandra**

**Coursework Report**

**RGU Student ID : 2425460**

**IIT Student ID : 20240875**

**Student Name : D. A. K. Chamath Hesara Gunawardhana**

## Table of contents

### Contents

1. Introduction.....	4
2. Technical Discussion .....	4
2.1 Fetch XML data .....	4
2.2 Cart Management using localStorage .....	6
3. Discussion of UX/UI principles/Applications/Justifications .....	8
3.1 Navigation Techniques.....	8
3.2 Color balance/selection/consistency .....	10
3.3 Color Contrast Test.....	13
3.4 Typography / consistency.....	14
3.5 Accessibility .....	15
3.5.1 Text Accessibility Techniques.....	15
3.5.2 Image Accessibility Techniques .....	16
3.5.3 Form Accessibility Techniques .....	16
3.6 Accessibility Test.....	17
3.6.1 Home Page .....	17
3.6.2 Book Detail Page.....	17
4. Validation Reports.....	18
4.1 Home Page .....	18
4.2 Book Detail Page.....	18
5. Self-Reflection.....	19
5.1 Figma Wireframe .....	19
5.1.1 Before Feedback .....	19
5.1.2 After Feedback.....	20
5.1.3 Adjustments after Feedback .....	20
5.2 Develop using html & css.....	21
5.2.1 Before Feedback .....	21
5.2.2 After Feedback.....	22
5.2.3 Adjustments after Feedback .....	22
6. References .....	23
Appendix: Screenshots .....	24

## List of figures

Figure 1: XML Structure .....	4
Figure 2: Book-Detail HTML Structure .....	5
Figure 3: Fetching XML using JavaScript Part 1 .....	5
Figure 4: Fetching XML using JavaScript Part 2 .....	6
Figure 5: Initialize Cart function .....	6
Figure 6: Add to cart functionality Part 1 .....	7
Figure 7: Add to cart functionality Part 2 .....	7
Figure 8: Navigation Menu System .....	8
Figure 9: HTML of Navigation Bar .....	8
Figure 10: CSS of Navigation Bar .....	9
Figure 11: Breadcrumb Technique .....	9
Figure 12: HTML of Breadcrumb .....	10
Figure 13: CSS of Breadcrumb .....	10
Figure 14: Color Contrast .....	11
Figure 15: CSS of color contrast .....	11
Figure 16: Logo & Title .....	11
Figure 17: CSS of logo & title .....	12
Figure 18: Contact Details .....	13
Figure 19: HTML of contact us .....	13
Figure 20: Color contrast checker .....	13
Figure 21: Typography .....	14
Figure 22: HTML of Typography .....	14
Figure 23: CSS of Typography .....	15
Figure 24: Text Accessibility Techniques .....	15
Figure 25: Image Accessibility Techniques .....	16
Figure 26: Form Accessibility Techniques .....	16
Figure 27: Accessibility Test Home Page .....	17
Figure 28: Accessibility Test Book Detail Page .....	17
Figure 29: Validation Test Home Page .....	18
Figure 30: Validation Test Book Detail Page .....	18
Figure 31: Figma Wireframe Before Feedback .....	19
Figure 32: Figma Wireframe After Feedback .....	20
Figure 33: HTML & CSS Before Feedback .....	21
Figure 34: HTML & CSS After Feedback .....	22
Figure 35: Fiction Page .....	24
Figure 36: My Cart Page .....	25
Figure 37: Checkout Page .....	26

## 1. Introduction

This report contains the development details of a bookstore application based on CM1605 web technology coursework case study 5. The main goal of the web site is to provide the best quality products for the users. This web application contains many book categories such as fiction, best seller, new arrival .... etc. It also has navigation features, filtering features, purchasing books and many more. Let's go through with the types of strategies and ways of implementation used in the application.

## 2. Technical Discussion

This web application contains many functionalities which are developed by JavaScript. JavaScript is a world-famous programming language used to develop web applications. It is mainly used to create dynamic content for web sites. In this web application JavaScript is used to cart management and fetch XML data.

### 2.1 Fetch XML data

```
<?xml version="1.0" encoding="UTF-8"?>
<books>
  <book id="1">
    <image>../../images/book1.jpg</image>
    <title>Harry Potter and the Half Blood Prince</title>
    <price>4325.00</price>
    <stock>In Stock</stock>
    <language>English</language>
    <author>J.K. Rowling</author>
    <publisher>Bloomsbury Publishing PLC</publisher>
    <isbn>1408586172</isbn>
    <category>Children</category>
    <rating>3.9</rating>
    <description>A compelling continuation of Harry's journey as he faces the darker facets of the wizarding world.</descr
  </book>
</books>
```

Figure 1: XML Structure

Extensible Markup Language (XML) is used to store the details of the books. The screenshot is a part of a reviews.xml file. In the structure of the xml, the root element is books which represent the collection of books and the child element bookstores the specification of the book. Like that 80 books are stored in the XML and each of them has a unique book id.

```

<div class="book-information"
  data-id=""
  data-category=""
  data-title=""
  data-price="">
  <h1 id="book-title"></h1>
  <div class="price" id="book-price"></div>
  <div class="stock" id="book-stock"></div>
  <ul class="info">
    <li><strong>Language</strong>: <span id="book-language"></span></li>
    <li><strong>Author</strong>: <span id="book-author"></span></li>
    <li><strong>Publisher</strong>: <span id="book-publisher"></span></li>
    <li><strong>ISBN</strong>: <span id="book-isbn"></span></li>
    <li><strong>Category</strong>: <span id="book-category"></span></li>
    <li><strong>Rating</strong>: <span id="book-rating"></span></li>
  </ul>

```

Figure 2: Book-Detail HTML Structure

Class="book-information" is a container in book detail.html which contains the book details. There are 4 main attributes such as id, category, title, price. This structure is used to fetch data dynamically.

```

1 document.addEventListener("DOMContentLoaded", function () {
2   // Get the book ID from the URL query parameter
3   const urlParams = new URLSearchParams(window.location.search);
4   const bookId = urlParams.get('id'); // e.g., book-detail.html?id=1
5
6   if (!bookId) {
7     console.error("Book ID not found in URL.");
8     return;
9   }
10
11   // Fetch the XML file
12   fetch("../XML/reviews.xml")
13     .then(response => {
14       if (!response.ok) {
15         throw new Error("Network response was not ok");
16       }
17       return response.text();
18     })
19     .then(data => {
20       const parser = new DOMParser();
21       const xmlDoc = parser.parseFromString(data, "text/xml");
22
23       // Find the book with the specified ID
24       const book = xmlDoc.querySelector(`book[id="${bookId}"]`);
25

```

Figure 3: Fetching XML using JavaScript Part 1

In the above Js, line 3 and 4 extract the 'id' from a specific URL which means we can call an id from another html file. As an example, in the home page.html there is anchor tag "<a

href="book detail.html?id=19">" which calls the details of the book where id is 19. Line 10-17 fetches the reviews.xml file and if the response is invalid it will throw an exception. Line 21 convert the xml data into a structured manner and line 24 finds a specific <book> element where id matches to book id in xml.

```
26     if (book) {  
27         // Extract book details  
28         const title = book.querySelector("title").textContent;  
29         const price = book.querySelector("price").textContent;  
30  
31         // Populate the book details in the HTML  
32         document.getElementById("book-title").textContent = title;  
33         document.getElementById("book-price").textContent = `LKR ${price}`;  
34     }
```

Figure 4: Fetching XML using JavaScript Part 2

If the particular book exists in the XML this code will extract content of each element (title, price, language.... etc.) and populate each element in html. The textContent property is used to update the text content like title, price, .... etc. Overall the js code fetches XML data, matches the correct book and updates the web page dynamically. Using java script is a must in scenarios like this because if java script not used I would have to create 80 static web pages for all the books and it is difficult and time consuming.

## 2.2 Cart Management using localStorage

```
1  document.addEventListener('DOMContentLoaded', function () {  
2      // Load cart items and counters from localStorage  
3      let cartItems = JSON.parse(localStorage.getItem('cartItems')) || [];  
4      let cartCounter = parseInt(localStorage.getItem('cartCounter')) || 0;  
5  
6      // Reset cartCounter to 0 if there are no items in the cart  
7      if (cartItems.length === 0) {  
8          cartCounter = 0;  
9          localStorage.setItem('cartCounter', cartCounter); // Save to localStorage  
10     }  
11 }
```

Figure 5: Initialize Cart function

Initialize an empty array if there are no elements. By using JSON.parse can convert JSON string back into an array. The key cartItem is used to retrieve stored items in the array. Like that created cartCounter and passed integers by parseInt. From line 7 onwards the if condition checks whether the counter is 0. If so, reset counter to 0 and update the counter respectively.

```

86 // Event listeners for add to cart buttons
87 const cartButtons = document.querySelectorAll('.cart-btn');
88 if (cartButtons.length > 0) {
89     cartButtons.forEach(button => {
90         button.addEventListener('click', function () {
91             const bookElement = this.closest('.book-categories'); // Use closest() to find the book container
92             if (bookElement) {
93                 const book = {
94                     id: bookElement.getAttribute('data-id'),
95                     category: bookElement.getAttribute('data-category'), // Include category
96                     title: bookElement.getAttribute('data-title'),
97                     price: parseFloat(bookElement.getAttribute('data-price'))
98                 };
99                 addToCart(book);
100             }
101         });
102     });
103 }

```

Figure 6: Add to cart functionality Part 1

Line 87 retrieves all the html elements tagged with class = 'cart-btn'. Condition checks whether there are elements with this class name in the html pages if true the code block executes. 89 & 90 lines assign specific logic to each button if clicked. If clicked, `closest('.book-categories')`: Searches for the nearest parent element with the class `book-categories`, which represents the container holding the book's details. After that a constant variable created as `book` which contains key-value pairs. If `id` is the key the value of it will load by `bookElement`. Lastly, `addToCart` function is called with the argument `book`. The `addToCart` function is given below.

```

64 // Function to add item to cart
65 function addToCart(book) {
66     const existingItem = cartItems.find(item => item.id === book.id && item.category === book.category);
67     if (existingItem) {
68         existingItem.quantity += 1;
69     } else {
70         cartItems.push({ ...book, quantity: 1 });
71     }
72     cartCounter += 1;
73     updateCartCounter();

```

Figure 7: Add to cart functionality Part 2

The function `addToCart` adds items to the cart. This function has one parameter as 'book', which was define as an object with `id`, `title`, `price` and `category`. Line 66 there is an iteration in `cartItem` array for matching `id` and `category` by using `find()`. If the item found it is stored in `existingItem` and if it already exists it will increment the quantity by 1. If the item is not found it will push to the array as a new item. The `CartCounter` is incremented by one and updated.

### 3. Discussion of UX/UI principles/Applications/Justifications

#### 3.1 Navigation Techniques

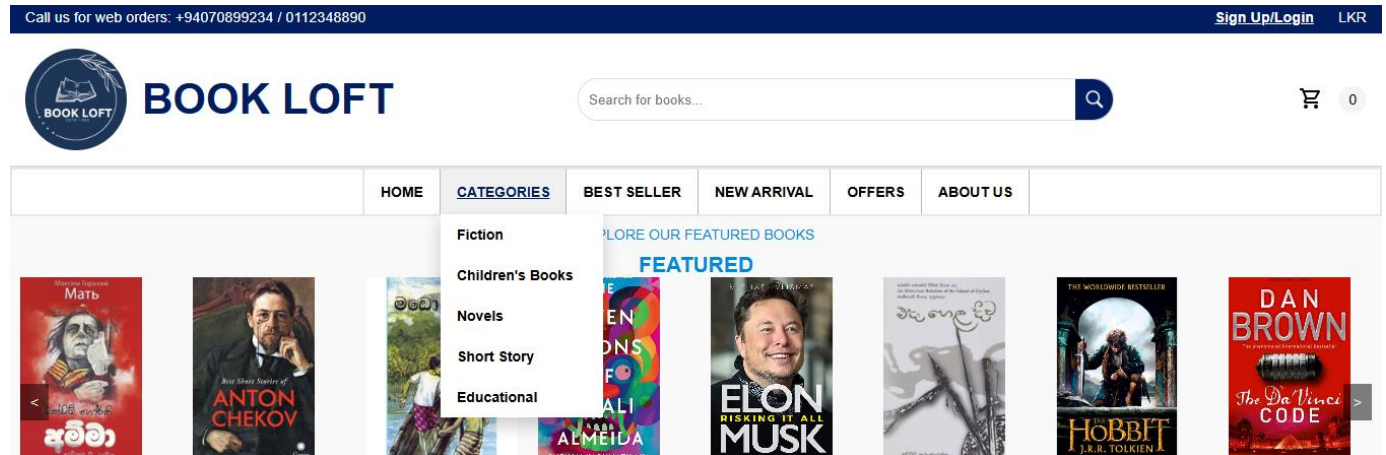


Figure 8: Navigation Menu System

The web application contains a fully functional navigation system with navigation elements such as home, best seller, new arrival, offers, about us and a dropdown menu for categories. This navigation bar enables the user to navigate and manage the web page easily.

```
44 <div class="nvbar">
45
46 <ul>
47 <li><a href="Home Page.html">HOME</a></li>
48 <li class="dropdown">
49 <a href="javascript:void(0)" class="dropbtn">CATEGORIES</a>
50 <div class="dropdown-content">
51 <a href="Fiction.html">Fiction</a>
52 <a href="Children's books.html">Children's Books</a>
53 <a href="Novels .html">Novels</a>
54 <a href="Short Story .html">Short Story</a>
55 <a href="Educational .html">Educational</a>
56 </div>
57 </li>
58 <li><a href="best seller.html">BEST SELLER</a></li>
59 <li><a href="new arrival.html">NEW ARRIVAL</a></li>
60 <li><a href="offers.html">OFFERS</a></li>
61 <li><a href="About Us.html">ABOUT US</a></li>
62 </ul>
63 </div>
```

Figure 9: HTML of Navigation Bar



Initialize a class as nvbar to contain the navigation menu system. Used an unordered list to organize the elements in the class. The ‘dropdown’ class has a drop-down menu. `<a href="javascript:void(0)" class="dropbtn">CATEGORIES</a>` creates a clickable button and prevent the default action for the link , allowing the dropdown behaviour to manage via Js or CSS. Anchor tags are used to link the html pages.

```

129  /* CSS for Naviagation bar */
130  .nvbar ul {
131      list-style-type: none;
132      padding:0;
133      overflow: hidden;
134      border: 2px solid #e7e7e7;
135      background-color: white;
136      display: flex;
137      justify-content: center;
138      font-weight: bold;
139  }
140  .nvbar {
141      position: sticky;
142      top: 0;
143      z-index: 1000;
144  }
145  .nvbar li {
146      float: left;
147  }
148  .nvbar li:first-child{
149      border-left:2px solid #e7e7e7 ;
150  }
151
152  li a, .dropbtn {
153      display: inline-block;
154      color: black;
155      text-align: center;
156      padding: 14px 16px;
157      text-decoration:none;
158  }

```

Figure 10: CSS of Navigation Bar

These are some css added to the navigation bar to make it more attractive. The ul elements are align in a row without the default bullets. Also, navigation is fixed and stays at the top without overlap issuers. Line 145 to 150 makes the list items are align horizontally next to other and add a vertical border line to the first element. The drop-down content displayed as inline blocks vertically, text is black and cantered. Element ‘padding: 14px 16px;’ adds spacing in the buttons in four sides (14px on top and bottom, 16px on the sides).

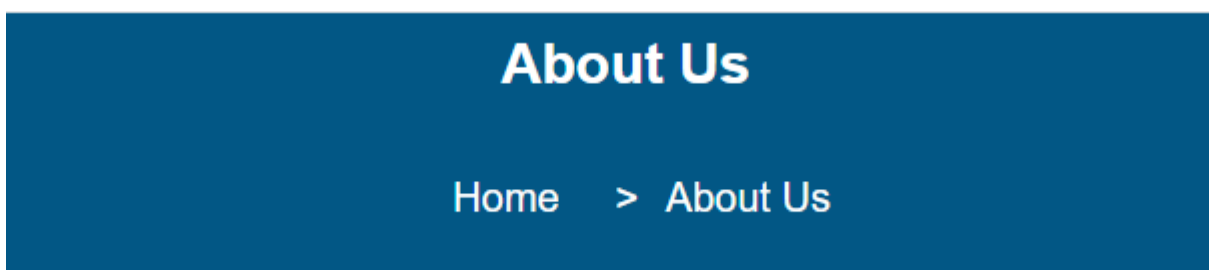


Figure 11: Breadcrumb Technique

Breadcrumb is a type of navigation which shows the user where he/she is within the website.

```
207      <h1>About Us</h1>
208      <ul class="breadcrumb">
209          <li><a href="Home Page.html">Home</a></li>
210          <li>About Us</li>
211      </ul>
212  </div>
```

Figure 12: HTML of Breadcrumb

So, the ul class ‘breadcrumb’ represents the unordered items in the breadcrumb. <li> tag can be connected to html pages using anchor tags like it is connected to home page as in the screenshot.

```
454  /* CSS for breadcrumb */
455  ul.breadcrumb {
456      padding: 5px 1px 5px 1px;
457      list-style: none;
458      background-color: #025785;
459  }
460  ul.breadcrumb li {
461      display: inline;
462      font-size: 20px;
463  }
464  ul.breadcrumb li+li:before {
465      padding: 8px;
466      color: white;
467      content: ">\00a0";
```

Figure 13: CSS of Breadcrumb

In the ul element top and bottom padding are 5px meanwhile left and right are 1px. Also default bullet style is removed from line 454. Font size of the list is 20px and it is displayed inline horizontally. Style ‘content: ">\00a0";’ define the separator symbol “>” and add proper spaces before & after the symbol.

### 3.2 Color balance/selection/consistency

The color theme of the web application is blue and white mainly. Most of the time if the background color is blue the text inside that background is white. Most of the world-famous web applications use this theme because these two colors are blended neatly and are not difficult for eyes.

## Our Mission

To satisfy clients by offering high-quality services founded on fundamental principles and actively assisting the nation's youth in developing their skills, knowledge, and attitudes. Create a strategy to address the issues facing the sector both individually and collectively while maintaining the excellent reputation Book Loft (Pvt) Ltd has built as the industry pioneers.

Figure 14: Color Contrast

Like previous discussed, in the picture the text inside the box is easily readable. Because of the color combination it is very clear. The css of the above output is given below.

```
background-color: #021E60;  
margin-top: 50px;  
color: white;  
font-size: 18px;
```

Figure 15: CSS of color contrast

The background color is dark blue, and the text color is white with font size 18px. This combination makes the output clear.



# BOOK LOFT

---

Figure 16: Logo & Title

Both logo and the title of the web site is attached together in a horizontal way. The logo was

taken from the internet, and it was dark blue and had to be matched the title text colour.

```
48  /* CSS for logo & shop title */
49  .name-logo{
50      gap: 5px;
51      display: flex;
52      align-items: center;
53  }
54  .logo {
55      width: 100px;
56      height: 100px;
57      overflow: hidden;
58      display: flex;
59      margin-right: 10px;
60      border-radius: 50px;
61  }
62  }
63
64  .logo img {
65      width: 100%;
66      height: 100%;
67      object-fit: cover;
68  }
69  }
70  .shop-name{
71      font-family: arial;
72      font-size: 2em;
73      font-weight: bold;
74      color: #021E60;
75      margin: 0;
76  }
```

Figure 17: CSS of logo & title

Using the above css, the logo and the title are displayed side by side with a small gap. Overflow and object fit styles ensure that the unwanted parts of the logo are cropped and removed. Border radius makes the logo shape circle. Shop title text is an arial and the text size is 2 times than the default font size. Set the color to dark blue and match the color of the logo.

This web application includes many icons such as cart, location, phone, mail, credit card icons ...etc.

## Contact Details

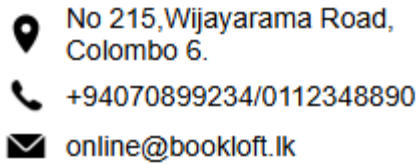


Figure 18: Contact Details

This section in the footer gives the details of the contact information of the company. As you can see, the location icon given to address, phone icon is given to telephone numbers and letter icon is given to email. The icons are in '.png' format and they are downloaded from <https://icons8.com/>.

```
<div class="contact-info">
  <h4 style="margin-bottom: 0px;">Contact Details</h4>
  <ul>
    <li>No 215,Wijayarama Road,<br>Colombo 6.</li>
    <li></i>+94070899234/0112348890</li>
    <li>online@bookloft.lk</li>
  </ul>
```

Figure 19: HTML of contact us

Using the below css, the list items are arranged horizontally include icons or images (e.g., a phone or email icon) alongside the text. The design ensures a clean and visually balanced presentation.

### 3.3Color Contrast Test

## Colour Contrast Check

Date created: January 11, 2005  
Date last modified: January 11, 2015

Foreground Colour:	Background Colour:	Results
#FFFFFF	#021E60	This is example text. Some of it bolded. Some of it italicized.
Red: <input type="range"/>	Red: <input type="range"/>	Brightness Difference: ( $\geq 125$ ) <input type="text" value="225.84"/>
Green: <input type="range"/>	Green: <input type="range"/>	Colour Difference: ( $\geq 500$ ) <input type="text" value="637"/>
Blue: <input type="range"/>	Blue: <input type="range"/>	Are colours compliant? <input type="text" value="YES"/>
Hue (*): <input type="range"/>	Hue (*): <input type="range"/>	Contrast Ratio <input type="text" value="15.473"/>
Saturation (%): <input type="range"/>	Saturation (%): <input type="range"/>	WCAG 2 AA Compliant <input type="text" value="YES"/>
Value (%): <input type="range"/>	Value (%): <input type="range"/>	WCAG 2 AA Compliant (18pt+) <input type="text" value="YES"/>
		WCAG 2 AAA Compliant <input type="text" value="YES"/>
		WCAG 2 AAA Compliant (18pt+) <input type="text" value="YES"/>

Figure 20: Color contrast checker

This color contrast checker ensures that the combination of white and blue is a suitable choice for web applications. Since the white text appeared in a dark background the text is more visible and easy to read also.

### 3.4 Typography / consistency

The web application contains many text sections which serve different kinds of distributions. All the text is in arial font type. Set a default text size as 14px for body content and the background color of the web site is similar to white.

```
1
2  * {
3    margin: 0;
4    padding: 0;
5    box-sizing: border-box;
6    font-family: 'Arial';
7  }
8  body {
9    font-size: 14px;
10   background-color: #f8f8f8;
11 }
12
```

Figure 21: Typography

This is a part of the html of the terms and conditions page which has headings & paragraph elements.

```
<div class="main-content-information">
  <h2>Introduction</h2>

  <p>This website is operated by Book Loft & Co. (Pvt.) Ltd. Throughout the site, the terms "we", "us" and "our" refer to B

  <h2>Copyright Infringement</h2>

  <p>You agree not to reproduce, duplicate, copy, sell, resell or exploit any portion of the Website, use of the Website, o

  <h2>Payments</h2>
  <p>We accept payments via Visa, Master. All our payments are processed through PayHere Service.All network communications

  <h2>Waiver of Rights</h2>

  <p>These Terms of Service and any policies or operating rules posted by us on this site or in respect to The Service cons

</div>
```

Figure 22: HTML of Typography

<h2> elements indicate that these are headings and <p> elements indicate that these are simple content text. Second largest heading element is used to increase the font size to show

that this is a headline so the user can clearly identify the heading and the paragraph.

```
435 .main-content-information p {
436     line-height: 1.6;
437     margin:10px 10px 10px 10px;
438     color: □black;
439     text-align: justify;
440 }
441 .main-content-information h2 {
442     margin-top: 24px;
443     margin-bottom: 12px;
444     color: □black;
445 }
```

Figure 23: CSS of Typography

This css makes the headings black and adjusts the margins from the bottom and top. Line size style makes the space between the line 1.6 times than the font size. It is also margin 10px from bottom. These all-margins styles make proper spaces between the content in the page. Also, the text is justified.

### 3.5 Accessibility

The term "web accessibility" refers to the creation of tools, technologies, and websites that are usable by individuals with disabilities. This web application contains text accessibility techniques, image accessibility techniques, form accessibility techniques.

#### 3.5.1 Text Accessibility Techniques

To ensure the text accessibility, this application follows the principle of hierarchy. The elements h1, h2, h4 are used to display main heading, section heading and subheadings respectively. Normal font size is 14px so it is eye catching. Used <alt> text tag to describe the images. Also using anchor link tags and button tags helps the user to navigate those interactive elements.

```

<p><a href="mailto: online@bookloft.lk">online@bookloft.lk</a></p>
<button id="proceed-to-checkout">Proceed to Checkout</button>
```

Figure 24: Text Accessibility Techniques

### 3.5.2 Image Accessibility Techniques

All the images in the web site have alt tag text to accessibility for visually impaired users. For example, the logo image has the alt attribute alt="Book Loft Logo", which provides context for screen readers.

```
<div class="logo">  
    
</div>
```

Figure 25: Image Accessibility Techniques

### 3.5.3 Form Accessibility Techniques

All the forms in the web site are developed using the form tag. For input fields the label tag is used. By using the label tag, it ensures that the purpose of the specific input field is trackable to screen readers.

```
<div class="form">  
  <form id="contact-form">  
    <div class="row">  
      <div class="col-25">  
        <label for="full_name">Full Name</label>  
      </div>  
      <div class="col-75">  
        <input type="text" id="contact-full-name" name="full_name">  
        <span class="error" style="display: none; color: red;" id="contact-name-error">Full Name is required.</span>  
      </div>  
    </div>  
  </form>  
</div>
```

Figure 26: Form Accessibility Techniques



## 3.6 Accessibility Test

### 3.6.1 Home Page

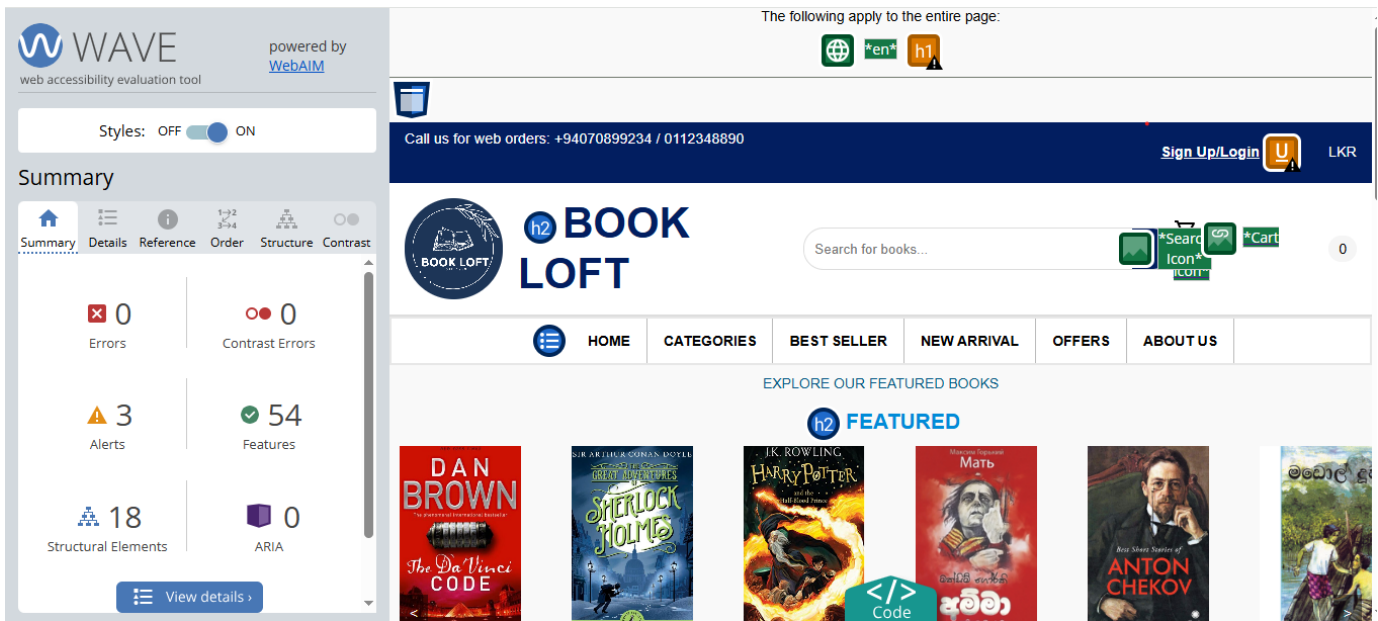


Figure 27: Accessibility Test Home Page

### 3.6.2 Book Detail Page

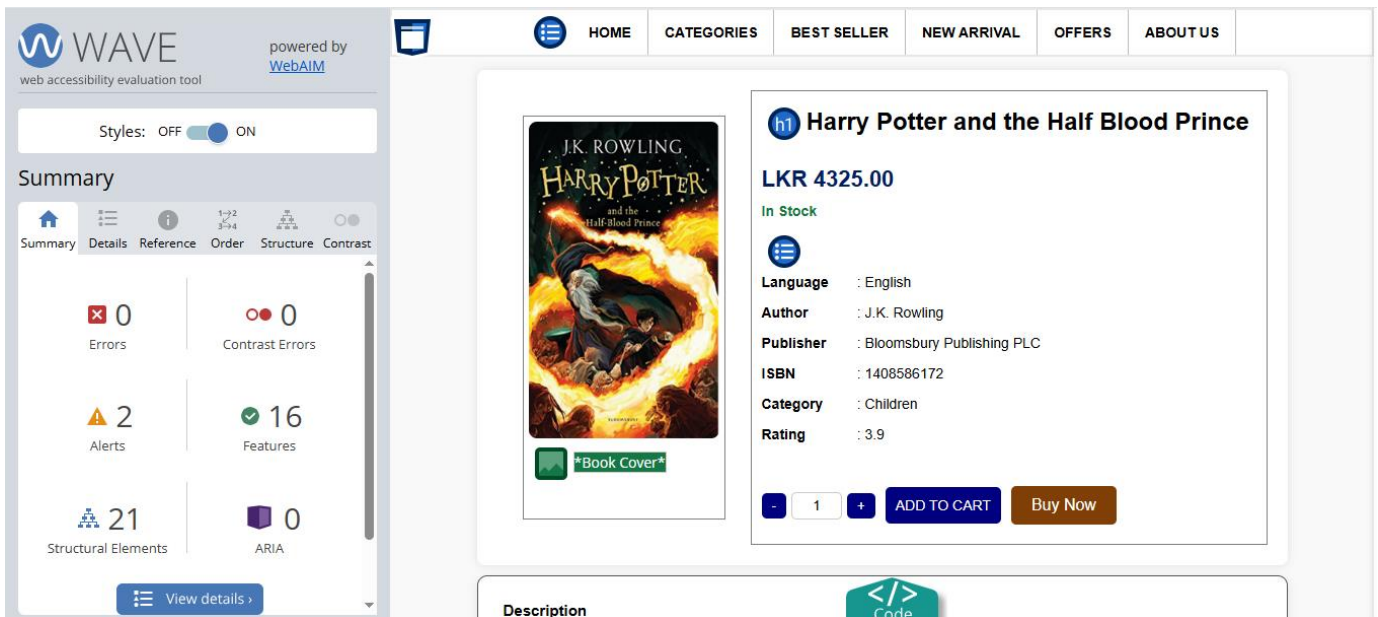


Figure 28: Accessibility Test Book Detail Page

## 4. Validation Reports

### 4.1 Home Page

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for uploaded file Home\_Page.html

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by   No file chosen

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.

Figure 29: Validation Test Home Page

### 4.2 Book Detail Page

#### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for book\_detail.html

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by   No file chosen

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.

Figure 30: Validation Test Book Detail Page

## 5. Self-Reflection

### 5.1 Figma Wireframe

#### 5.1.1 Before Feedback

## HOME PAGE

HEADER	Email Address	Sign Up/Login			
Site Logo	Search Form	Q	♥	●	🛒
<u>HOME</u>	<u>CATEGORIES</u>	<u>BEST SELLER</u>	<u>NEW ARRIVAL</u>	<u>OFFERS</u>	<u>ABOUT US</u>
FEATURE BOOK ITEMS					
Book Items 01	Book Items 02	Book Items 03	Book Items 04	Book Items 05	
NEW ARRIVALS					
Book Items 01	Book Items 02	Book Items 03	Book Items 04	Book Items 05	
FOOTER	<u>Categories</u>	<u>Quick Links</u>	<u>Information</u>	<u>Contact Details</u>	

Figure 31: Figma Wireframe Before Feedback

### 5.1.2 After Feedback

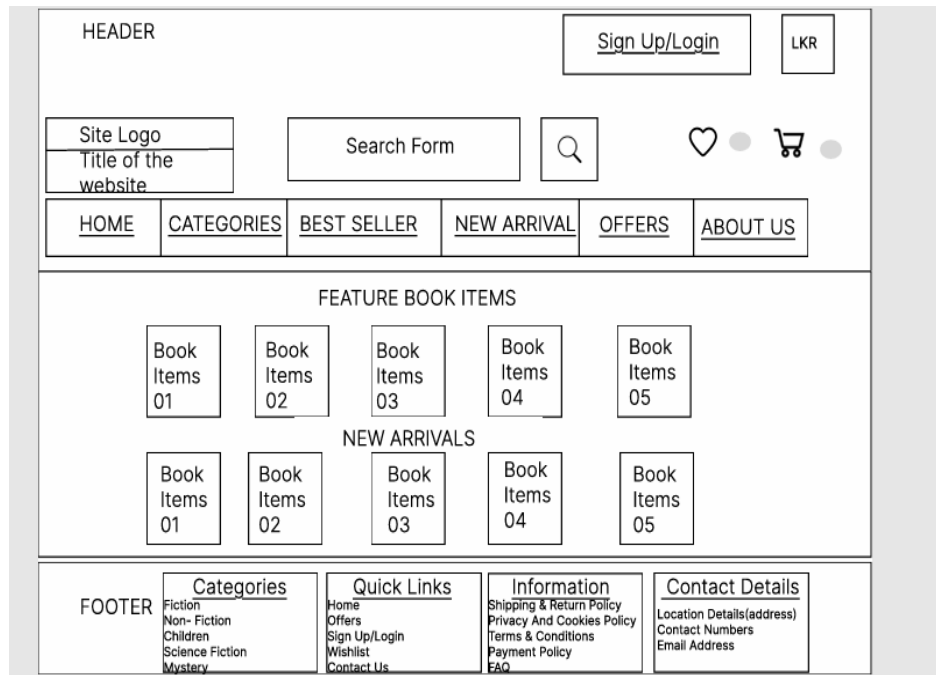


Figure 32: Figma Wireframe After Feedback

### 5.1.3 Adjustments after Feedback

- Removed email address from the header and insert it into the footer section.
- Added the shop title next to the logo of the page.
- Previously the footer section was not detail wise so inserted meaningful information and links to the footer.

5.2 Develop using html & css

5.2.1 Before Feedback

2. HOME PAGE

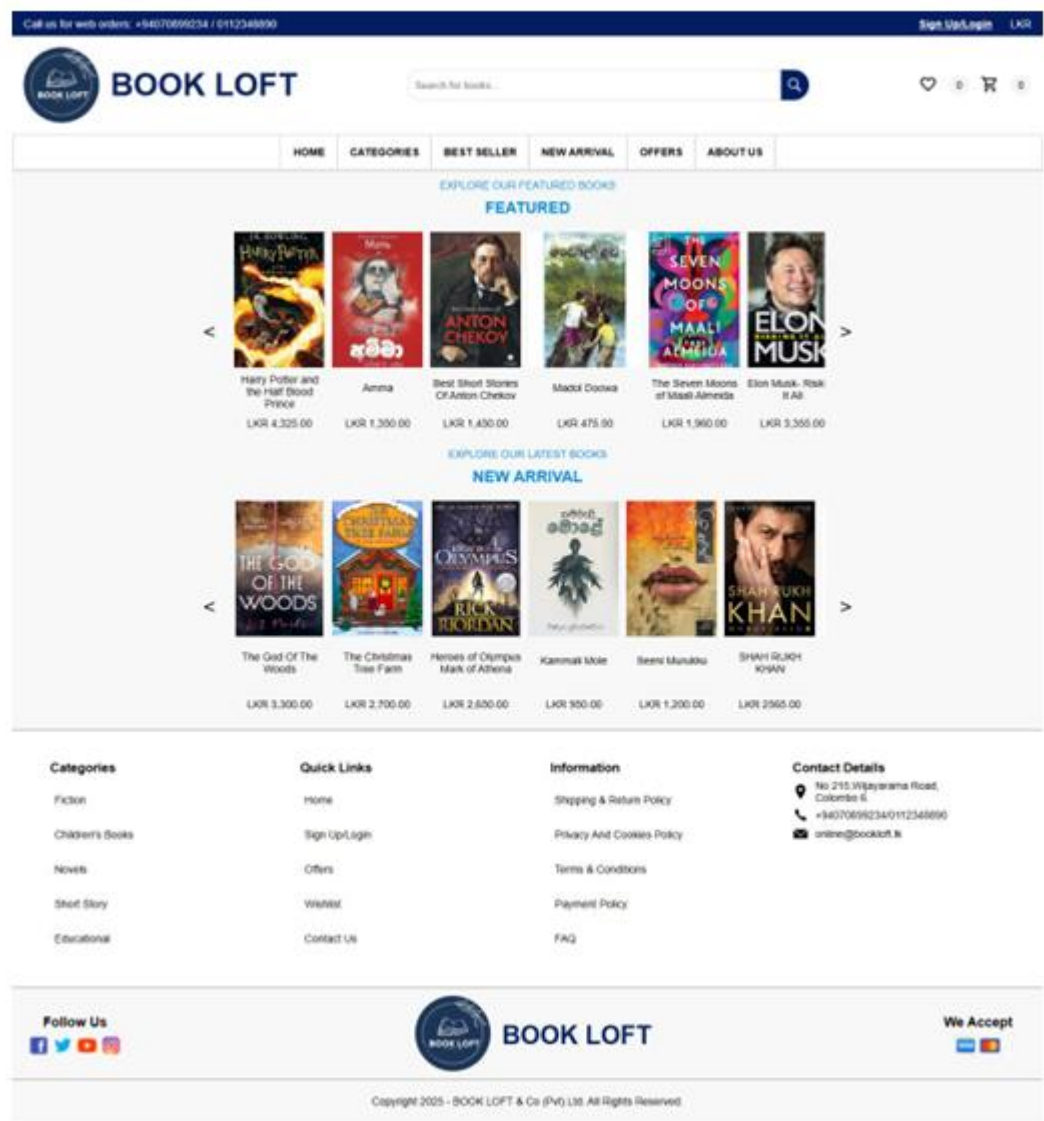


Figure 33: HTML & CSS Before Feedback

### 5.2.3 Adjustments after Feedback

- Previously the navigation bar was not fixed. So, it had to be fixed because when the user is at the bottom of the web page the user doesn't need to scroll up to access it.
- Reduced the area of the footer section by replacing 'follow us and we accept'

sections below the contact information.

- Overall, the feedback from the tutors was really helpful to develop this web application.

## 6. References

- All the images of the books' covers are taken from the <https://www.sarasavi.lk/> .



## Appendix: Screenshots

Fiction

Home > Fiction

Sort By

☐ Price: Low to High
 ☐ Price: High to Low
 ☐ Rating

**Project Hail Mary**  
Price: LKR 2560  
Rating: 4.5  
Add to Cart

**INVISIBLE**  
Price: LKR 2538  
Rating: 3.3  
Add to Cart

**TALE OF THE HEART QUEEN**  
Price: LKR 4000  
Rating: 3.2  
Add to Cart

**EMBASSY CLA - 1984**  
Price: LKR 2800  
Rating: 4.6  
Add to Cart

**Dream Town**  
Price: LKR 2550  
Rating: 4.0  
Add to Cart

**Story of my Life**  
Price: LKR 2700  
Rating: 2.8  
Add to Cart

**DAMSEL**  
Price: LKR 3200  
Rating: 4.7  
Add to Cart

**Song of the West**  
Price: LKR 1500  
Rating: 3.6  
Add to Cart

**To The Lighthouse**  
Price: LKR 4890  
Rating: 3.5  
Add to Cart

**NEEDFUL THINGS**  
Price: LKR 6500  
Rating: 2.1  
Add to Cart

**TRAIL BY FIRE**  
Price: LKR 3500  
Rating: 4.3  
Add to Cart

**SIDDHARTHA**  
Price: LKR 1200  
Rating: 3.3  
Add to Cart

Figure 35: Fiction Page



[HOME](#)
[CATEGORIES](#)
[BEST SELLER](#)
[NEW ARRIVAL](#)
[OFFERS](#)
[ABOUT US](#)



### Harry Potter and the Half Blood Prince

**LKR 4325.00**

**In Stock**

**Language** : English

**Author** : J.K. Rowling

**Publisher** : Bloomsbury Publishing PLC

**ISBN** : 1408586172

**Category** : Children

**Rating** : 3.9

-

1

+

ADD TO CART

Buy Now

**Description**

A compelling continuation of Harry's journey as he faces the darker facets of the wizarding world.

Shopping Cart

Harry Potter and the Half Blood Prince (Children) - LKR 4325.00 x 1

**Total: LKR 4325.00**

View Cart

Close

Figure 36: Adding a book to the cart

My Cart

Home > My Cart

My Cart

PRODUCT	PRICE	QUANTITY	SUBTOTAL
Project Hail Mary	LKR 2560.00	<div>- 1 +</div>	LKR 2560.00
LIFE OF PI	LKR 1800.00	<div>- 1 +</div>	LKR 1800.00
Python for Data Analysis	LKR 2600.00	<div>- 1 +</div>	LKR 2600.00
DISNEY PIXAR	LKR 2800.00	<div>- 2 +</div>	LKR 5600.00

**Total: LKR 12560.00**

Proceed to Checkout

Figure 37: My Cart Page

Checkout

Home > Checkout

Billing Details

Full Name \*

Email \*

Telephone \*

Address \*

Postal Code

Payment Details

Name on Card \*

Card Number \*

Expiry Date \*

CVV \*

My Bill

Total: LKR 12560.00

Place Order

Figure 38: Checkout Page