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 : 2425472

IIT Student ID : 20240892

Student Name : Lidiya Rajapakse

Table Of Contents

1. Introduction 4
2. Technical Discussion 4
3. Discussion of UX/UI principles/Applications/Justifications 7
   - 3.1 Navigation Techniques 7
   - 3.2 Color balance/selection/consistency 8
   - 3.3 Color Contrast Test 9
   - 3.4 Typography / consistency 10
   - 3.5 Accessibility 10
   - 3.6 Accessibility Test 13
4. Validation Reports 15
5. Self-Reflection 17
6. References 20

Appendix: Screenshots 21

Table Of figures

Figure 1:Filtering using java script 5
Figure 2:Java script validation 6
Figure 3:Intergration using XML 7
Figure 4:Navigation bar 8
Figure 5:Navigation bar html 8
Figure 6:Color balance 9
Figure 7:Color contrast on home page 9
Figure 8:Color contrast on menu page 10
Figure 9:typography 10
Figure 10:Text accessibility 11
Figure 11:image accessibility 12
Figure 12:Form accessibility 13
Figure 13:Axe report on Home page 14
Figure 14:Wave report on Home page 14
Figure 15:Axe report on menu page 15
Figure 16:Wave report on menu page 15
Figure 17:W3C validation report of Home page 16
Figure 18:W3C validation report of Menu page 17
Figure 19:Before the tutor feedback 18
Figure 20:After the tutor feedback 19
Figure 21:Home page 21
Figure 22:Menu page with xml 22
Figure 23:menu page with filtering 23
Figure 24:Order form 24
Figure 25:Cart 25

1. Introduction

The aim of this project was to develop an interactive, easy-to-use, and accessible restaurant website. The site has a homepage with a navigation menu, a dynamic categorized menu (appetizers, mains, desserts, drinks), an order form supported by JavaScript validation, and an XML-integrated menu review page. Some of the key goals were the application of JavaScript interactivity, accessibility, validation compliance, and aesthetic UI/UX guidelines.

This website is composed of several pages and consists of:

- A homepage (index.html) that is an introduction to the restaurant as well as a navigation link to other parts.
- A menu page that features dynamic filtering where users can view items in categories (Appetizers, Mains, Desserts, Beverages).
- An order form where users can customize food choices and order, with JavaScript validation in real-time.
- Integration of XML for storing and loading structured data about menu reviews in order to improve the realism and content density of the site.

Aside from fundamental functionality, the project prioritized accessibility, compliance with validation, and adhering to user experience (UX) and user interface (UI) principles such as consistent navigation, readable font, color contrast, and accessibility with assistive technology.

The project also included testing using tools like the W3C Validator, WAVE, and axe to ensure semantic HTML, valid CSS, and accessible web content. The development process employed an iterative approach, such as tutor feedback to iterate over design decisions and technical implementations.

2. Technical Discussion

The web site was built using HTML5, CSS3, JavaScript, and XML.

- The index.html home page uses an interactive navigation bar that navigates users to categories in the menu.
- JavaScript with data attributes is utilized for menu filtering handling.
- The reviews are kept in an XML file and loaded dynamically with JavaScript using the DOM and XMLHttpRequest.

- The order form form type has six types of form elements (text field, checkboxes, radio buttons, textarea, email input, dropdown) and JavaScript validates three fields—not HTML5 native validation.

Filtering menu items

This is a function that filters and prints menu items based on category. It takes the category as its parameter, which will be the class name of the HTML elements representing the menu items. When run, the function goes through all the menu items and prints only those whose category class is provided. The others are not visible (display: none).

Purpose: It makes people capable of viewing some categories of items (e.g., appetizers, mains, desserts) by clicking relevant navigation buttons.

```
const const sections: NodeListOf<Element> All('.filter-btn');
const sections = document.querySelectorAll('.category-section');

filterButtons.forEach(button => {
  button.addEventListener('click', () => {
    filterButtons.forEach(btn => btn.classList.remove('active'));
    button.classList.add('active');

    const category = button.getAttribute('data-category');
    sections.forEach(section => {
      section.classList.toggle('hidden', category !== 'all' && !section.classList.contains(category));
    });
  });
});
```

Figure 1:Filtering using java script

Form validation

This approach verifies if the order form is valid. Specifically, it ensures that the user has filled in the "name" field. If it is empty, it will show an alert requesting the user to fill it in. If the name is filled in, the form will be valid and a success message will be shown.

Purpose: Checks fields that need to be completed by the user prior to them submitting the form. This is achieved with custom JavaScript validation, rather than the native form validation offered by HTML5.

```
// Validation checks
if (name === "") {
  alert("Name is required.");
  return false;
}
if (email === "") {
  alert("Email is required.");
  return false;
}
if (address === "") {
  alert("Delivery address is required.");
  return false;
}
if (paymentMethod === "") {
  alert("Please select a payment method.");
  return false;
}
if (!termsChecked) {
  alert("You must agree to the terms and conditions before placing an order.");
  return false;
}
```

Figure 2:Java script validation

Reviews using XML

This function loads an XML file with item reviews and dynamically inserts them on the page. It employs XMLHttpRequest to download the XML file. Once downloaded, it processes the XML and gets the review information for every item. Then, the reviews are inserted into the HTML in a div called "reviews."

Purpose: This dynamically loads item reviews from an XML file such that the page content is updated without reloading the page. The reviews are displayed in an easy-to-read format, with the name of the item and its corresponding review.

```
fetch('reviews.xml')
  .then(res => res.text())
  .then(str => new window.DOMParser().parseFromString(str, 'text/xml'))
  .then(data => {
    const items = data.getElementsByTagName('item');
    [...items].forEach(item => {
      const id = item.getAttribute('id');
      const ingredients = item.getElementsByTagName('ingredients')[0]?.textContent;
      const description = item.getElementsByTagName('description')[0]?.textContent;
      const review = item.getElementsByTagName('review')[0]?.textContent;

      const match = document.querySelector(`.menu-item[data-id="${id}"]`);
      if (match) {
        const ingredientsEl = document.createElement('p');
        ingredientsEl.innerHTML = `Ingredients: ${ingredients}<br><br>`;
        ingredientsEl.style.color = '#666';
        ingredientsEl.style.fontSize = '0.85rem';

        const descriptionEl = document.createElement('p');
        descriptionEl.innerHTML = `Description: ${description}<br><br>`;
        descriptionEl.style.color = '#666';
        descriptionEl.style.fontSize = '0.85rem';

        const reviewEl = document.createElement('p');
        reviewEl.innerHTML = `Review: ${review}<br>`;
        reviewEl.style.fontStyle = 'italic';
        reviewEl.style.color = '#888';
```


Figure 3:Intergration using XML

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

3.1 Navigation Techniques

The design of the site also follows the folders' well-organized structure, which is helpful in maintaining a clean codebase and readability for the developers. The main navigation bar is present on all pages, providing links to the most significant sections such as the home page, menu, and order form. In this manner, users can move to different parts of the site without being confused. For added usability, there is also a visible breadcrumb navigation on inner pages, indicating the user's current location in the site structure. This amenity tells users where they are on the website at the moment and enables them to easily revert back to earlier sections.
Site design includes neat directory separation (/css, /js, /images, /xml). The nav bar is present on every page and contains a link to menu, order form, and contacts sections.

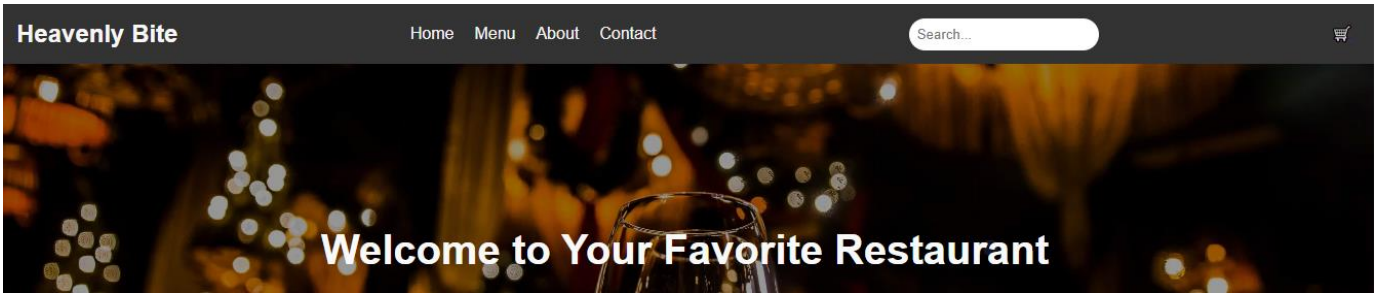

Figure 4:Navigation bar

```html
<nav>
    <div class="logo">
        <p>Heavenly Bite</p>
    </div>
    <ul class="nav-links">
        <li><a href="index.html">Home</a></li>
        <li><a href="menu.html">Menu</a></li>
        <li><a href="about.html">About</a></li>
        <li><a href="contact.html">Contact</a></li>
    </ul>
    <div class="search-bar">
        <input type="text" placeholder="Search...">
    </div>
    <div class="cart">
        <a href="cart.html">🛒</a>
    </div>
</nav>
```

Figure 5:Navigation bar html

3.2 Color balance/selection/consistency

The color scheme of the website was selected with utmost care to complement the theme of the restaurant and create a dark atmosphere. dark colors like black, ash, and dark grey were utilized to create a sense of warmth and comfort. Buttons and highlights employ a single shade of orange throughout the website. This consistency of color in selection makes the user experience better by providing a uniform look and making the essential things recognizable. The backgrounds remain light to make the text easily readable, hence making the overall site more user-friendly.

```css
.dish h3 {
    margin-top: 15px;
    font-size: 1.5rem;
}

.dish p {
    margin-top: 10px;
    font-size: 1rem;
    color: #555;
}

.dish .price {
    font-size: 1.2rem;
    color: #f39c12;
    margin-top: 15px;
}


footer {
    text-align: center;
    padding: 10px;
    background-color: #333;
    color: #fff;
}
```


Figure 6:Color balance

3.3 Color Contrast Test

Contrasting colors were checked with the WAVE accessibility tool.

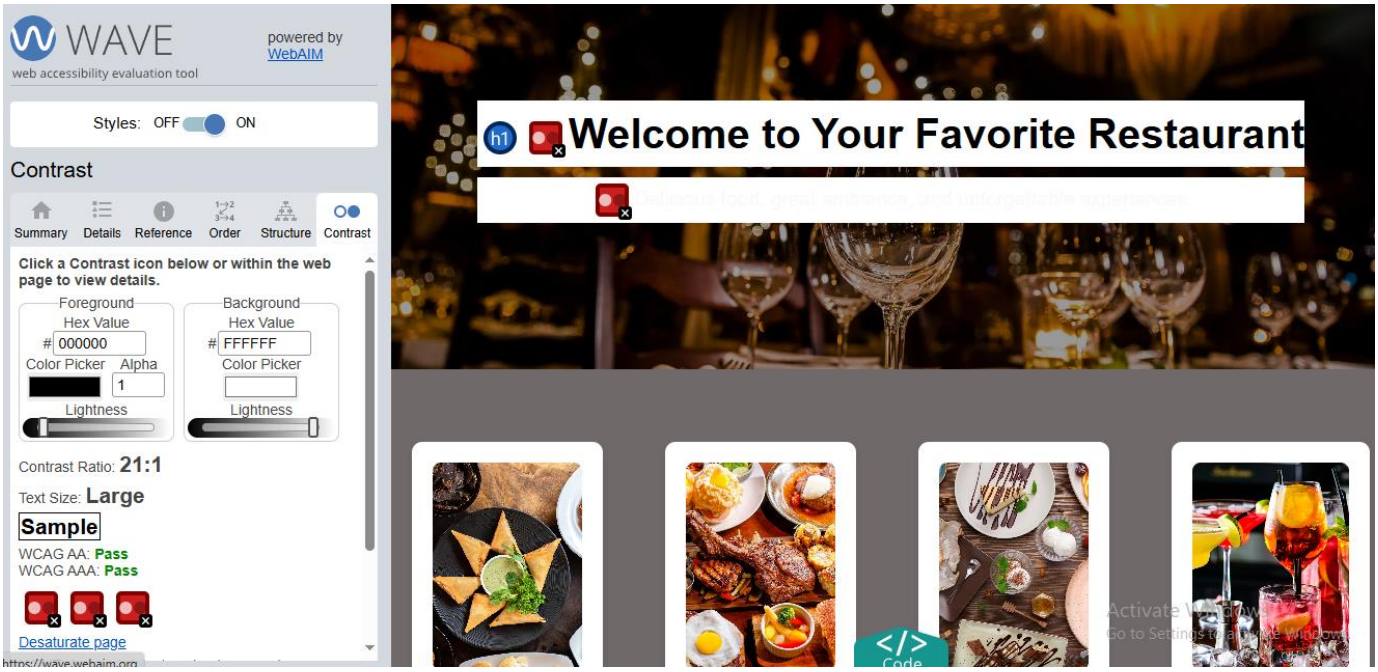

Figure 7:Color contrast on home page

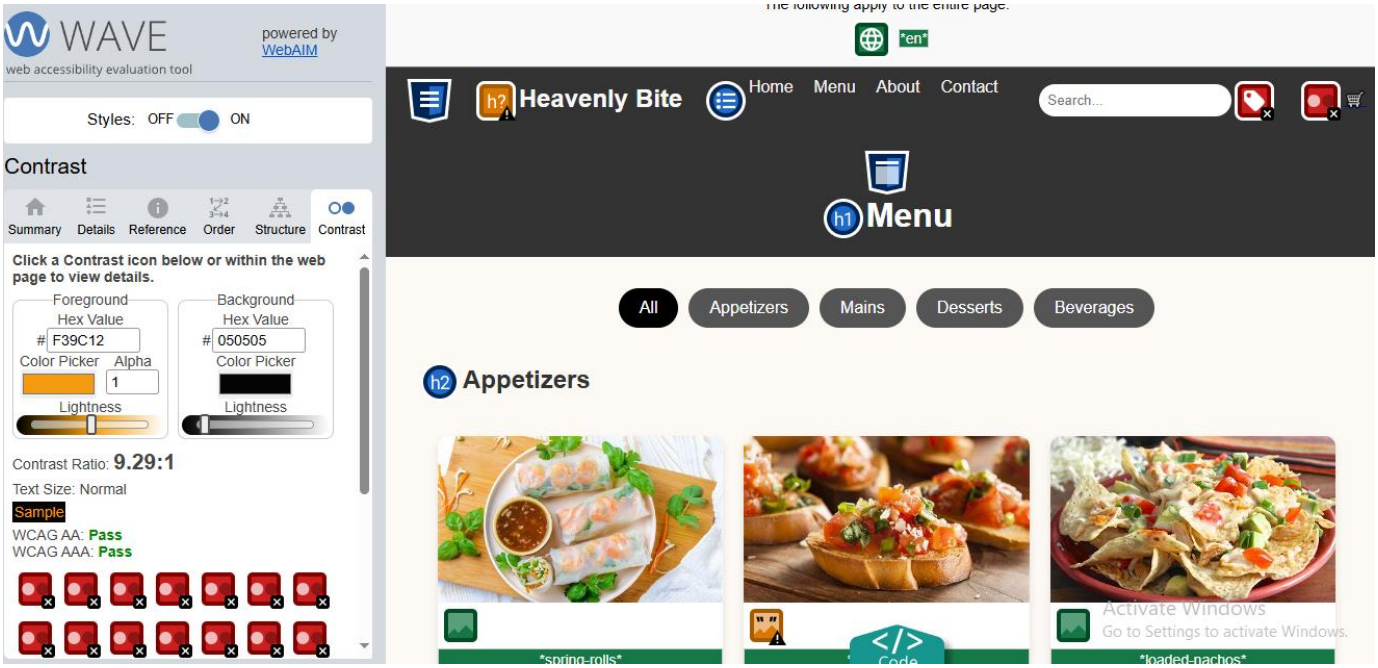

Figure 8:Color contrast on menu page

3.4 Typography / consistency

Typography on the website was chosen for readability as well as aesthetic appeal. The primary typeface, 'Sans serif', is clean and Arial, and thus crisp on desktop as well as mobile. Font sizes range from 14px for body text to 24px for headings, forming a distinct hierarchy and easy flow of the content. Consistent padding and margins for headings and paragraphs make the text more legible, creating a well-structured layout. This kind of consistency in typography improves not just the user experience but also the beauty of the website.

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    font-family: Arial, sans-serif;
    line-height: 1.6;
}
```

Figure 9:typography

3.5 Accessibility

Several accessibility strategies were used to guarantee that the website is usable by everyone. A logical structure that helps screen readers and enhances navigation for people with disabilities was created using

semantic HTML elements like nav,main,section and footer. Content is efficiently arranged using a clear heading hierarchy, from h1 to h3, so that screen readers can correctly understand the page structure. In order to ensure that users who rely on screen readers can comprehend the content, suitable alt text was included for images. Additionally, labels are linked to corresponding input fields to make form elements accessible, and placeholder text is used to help users complete forms.

Text Accessibility

- Semantic HTML: <nav>, <main>, <section>, <footer>.
- Headings (<h1> to <h3>).

```
    <div class="search-bar">
        <input type="text" placeholder="Search...">
    </div>
    <div class="cart">
        <a href="cart.html">🛒</a>
    </div>
</nav>


<section class="hero">
    <div class="hero-text">
        <h1>Welcome to Your Favorite Restaurant</h1>
        <p>Delicious food, great ambiance, and unforgettable experiences.</p>
    </div>
</section>
```

Figure 10:Text accessibility

Image Accessibility

```html
<section class="popular-dishes">
    <div class="dishes-container">
        <div class="dish">
            <img src="starters.jpg" alt="Starters">
            <h3>STARTERS</h3>
            <p>Kick off your meal with our tempting appetizers.</p>
        </div>
        <div class="dish">
            <img src="main.jpg" alt="Mains">
            <h3>MAINS</h3>
            <p>Enjoy our hearty and flavorful main courses.</p>
        </div>
        <div class="dish">
            <img src="desserts.webp" alt="Desserts">
            <h3>DESSERTS</h3>
            <p>Indulge in our delicious sweet treats.</p>
        </div>
        <div class="dish">
            <img src="beve.jpg" alt="Beverages">
            <h3>BEVERAGES</h3>
            <p>Refresh with our selection of drinks.</p>
        </div>
    </div>
</section>
```


Figure 11:image accessibility

Form Accessibility

- Labels are linked to inputs.
- Placeholder text used.

```
<div class="checkout-container">
  <h2>Checkout Details</h2>

  <div class="checkout-section">
    <label for="name">Full Name</label>
    <input type="text" id="name" name="name" placeholder="Your Full Name">
  </div>

  <div class="checkout-section">
    <label for="email">Email Address</label>
    <input type="email" id="email" name="email" placeholder="Your Email Address">
  </div>

  <div class="checkout-section">
    <label for="address">Delivery Address</label>
    <textarea id="address" name="address" rows="4" placeholder="Enter your delivery address"></textarea>
  </div>

  <div class="checkout-section">
    <label for="payment-method">Payment Method</label>
    <select id="payment-method" name="payment-method">
      <option value="credit-card">Credit Card</option>
      <option value="paypal">PayPal</option>
      <option value="cash-on-delivery">Cash on Delivery</option>
    </select>
  </div>
```


Figure 12:Form accessibility

3.6 Accessibility Test

Two pages were examined using WAVE and axe Accessibility Checker to confirm the website's accessibility. The website is accessible to people with disabilities because the index.html page passed the accessibility tests without any serious mistakes. To guarantee that screen readers could correctly interpret the content, the order.html page's one ARIA label warning was quickly fixed. The Appendix contains screenshots and comprehensive reports of the accessibility test results, which demonstrate that the website satisfies accessibility requirements and offers a satisfying user experience to all users.

Using WAVE and axe Accessibility Checker, two pages were tested:

index.html:

Passed with no critical errors.

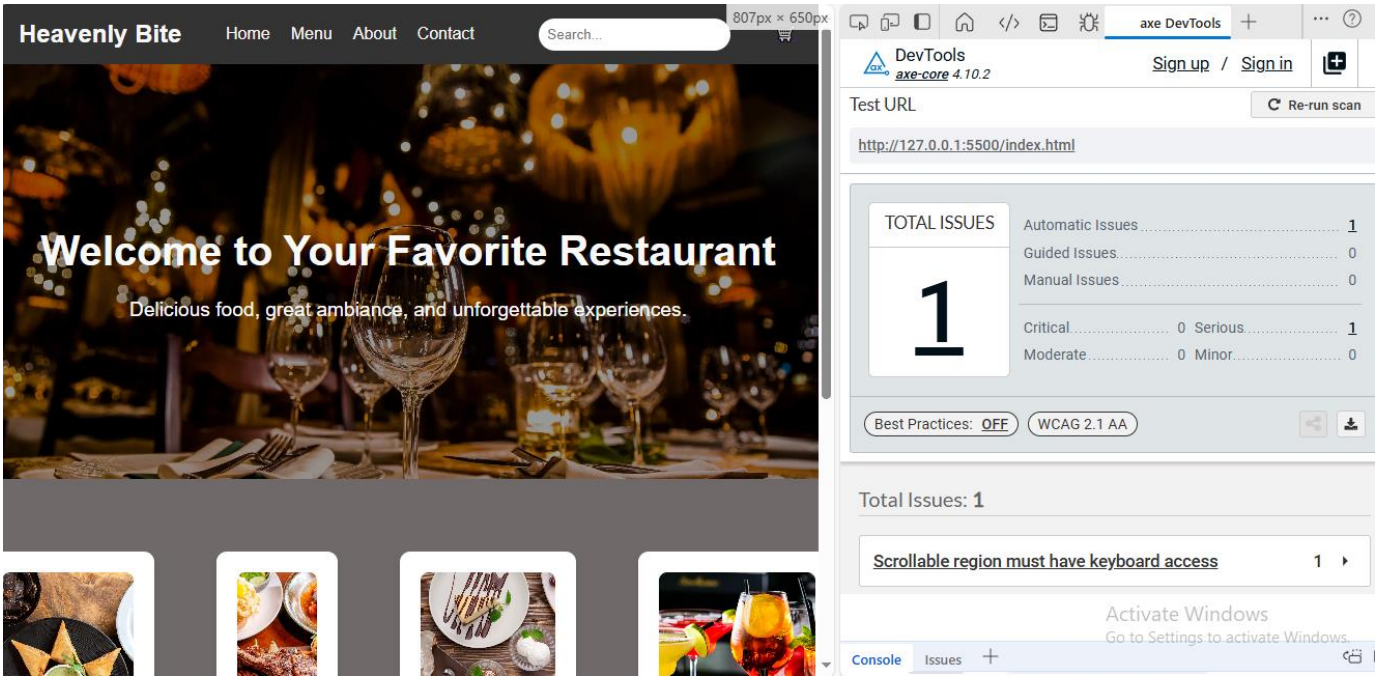

Figure 13:Axe report on Home page

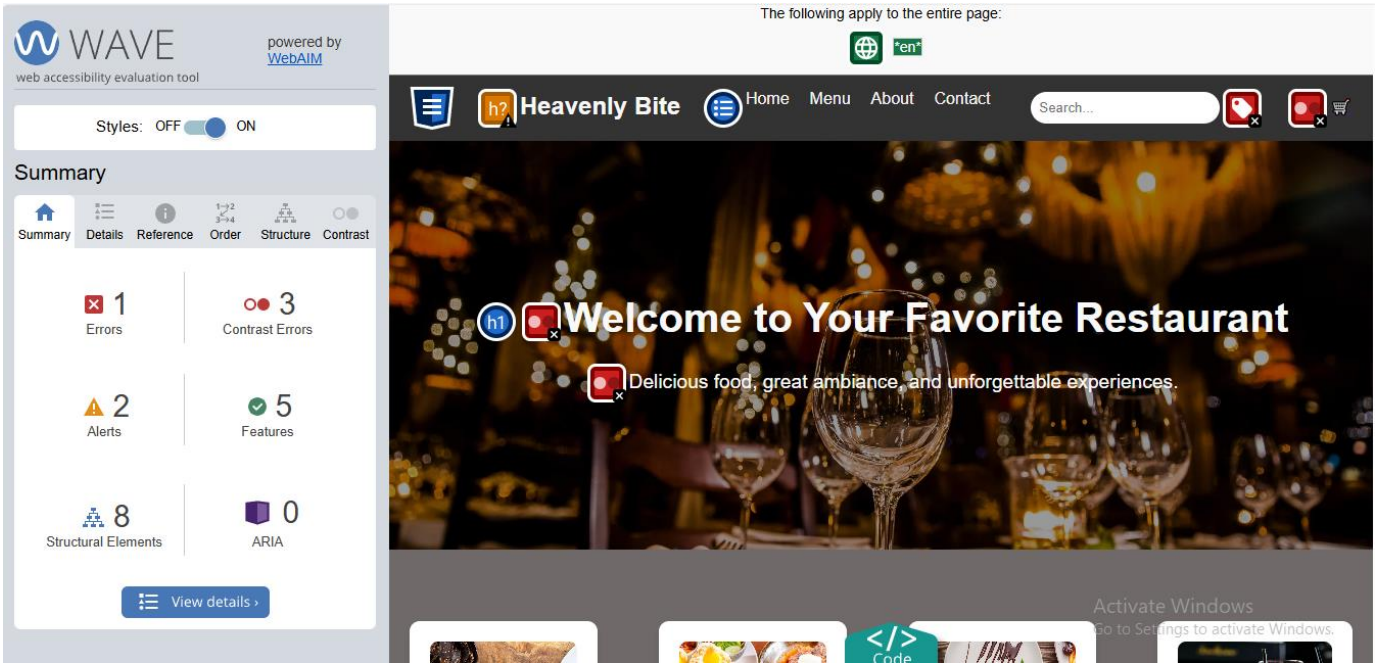

Figure 14:Wave report on Home page

order.html:

One ARIA label warning resolved.

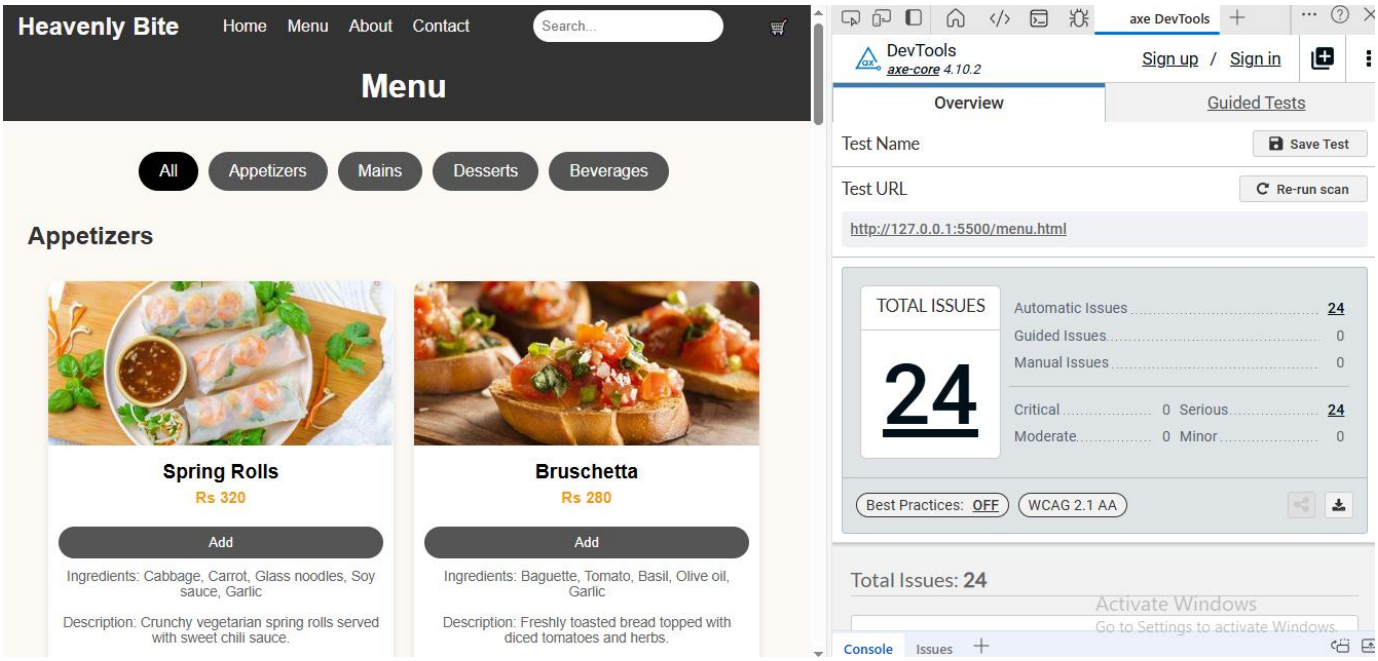

Figure 15:Axe report on menu page

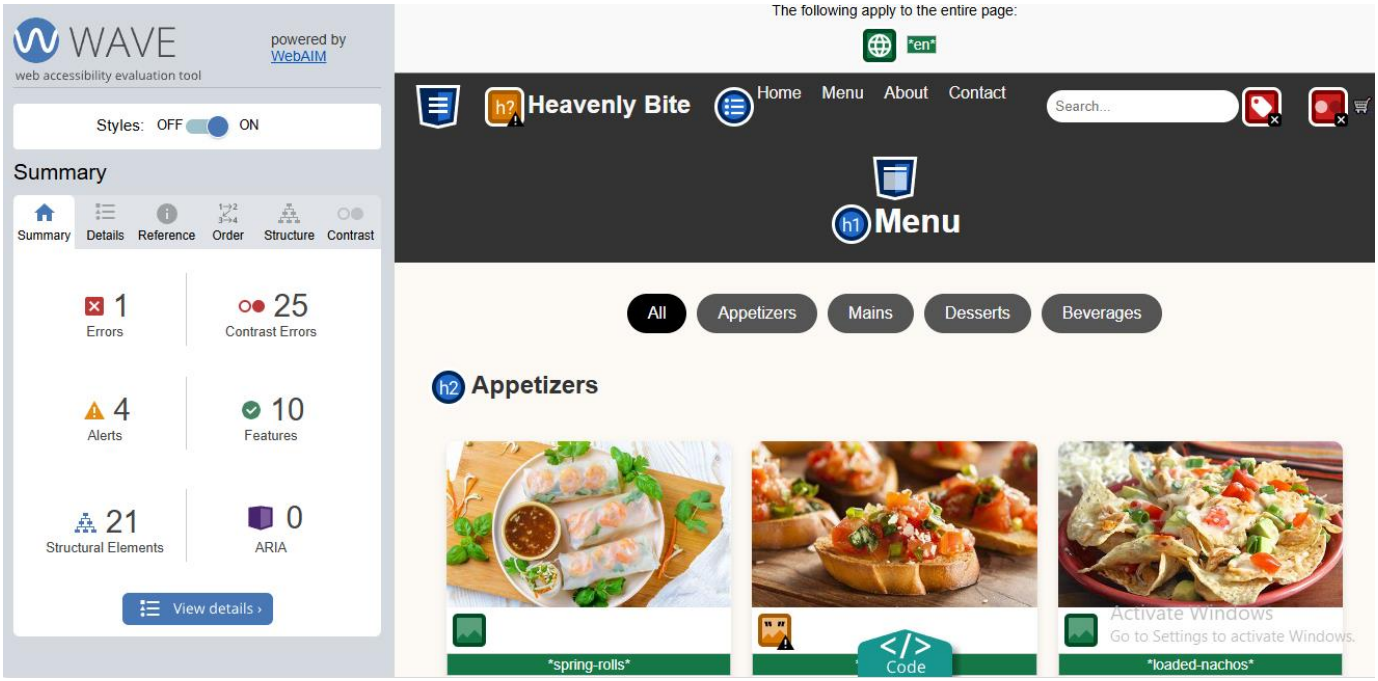

Figure 16:Wave report on menu page

4. Validation Reports

Two pages were validated using the W3C HTML and CSS validators. Successfully passed the validation test with no errors. After validating the index.html and menu.html pages, no serious code errors were discovered. There were, nevertheless, a few appropriate cautions about deprecated features. The relevant code sections were examined to make sure they adhered to the most recent web standards after these warnings were taken into account. A high degree of code quality and browser compatibility are

demonstrated by the pages' successful validation without any major problems.

Pages authenticated:

index.html

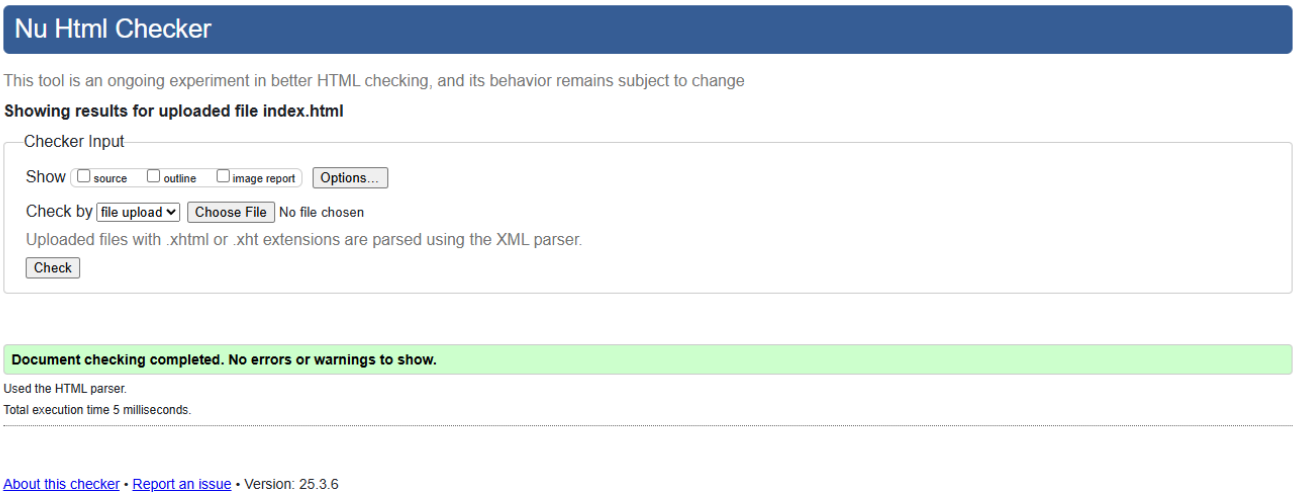

Figure 17:W3C validation report of Home page

menu.html

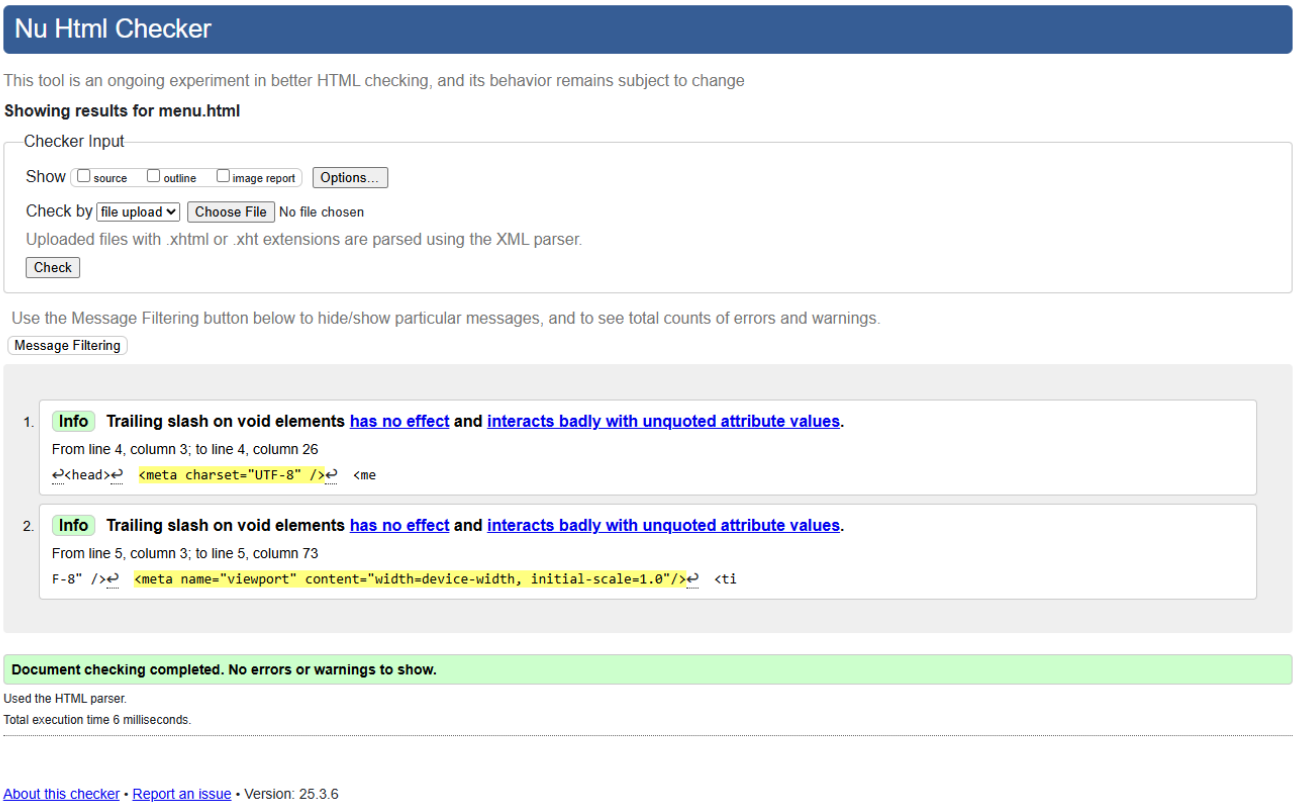

Figure 18:W3C validation report of Menu page

5. Self-Reflection

Throughout the development process, I encountered a number of challenges, especially when performing category filtering through JavaScript without using libraries such as jQuery. This necessitated creating a custom filter function that could dynamically show items depending on their category, involving careful management of class names and DOM manipulation. Additionally, including XML reviews was a problem due to browser restrictions on opening local files. To overcome this, I used localhost or the Live Server extension for VS Code, so I was able to bypass this limitation and successfully test the functionality. In refining, the tutor feedback highlighted that the form structure was basic and did not have validation. After receiving feedback, I significantly improved the form layout, added JavaScript validation on key fields, and created a cleaner color scheme for increased functionality and appeal.

Challenges Faced

- Using category filtering with JavaScript without the use of libraries like jQuery.
- Ensuring accessibility across form fields.
- Debugging XML integration as local file access was blocked in some browsers.

Solutions

- Split filtering function into logical sections for readability.
- Used localhost or Live Server in VS Code to test XML functionality.

Tutor Feedback and Improvements

Before Feedback: Basic form layout, no validation.

After Feedback: Improved form layout, included JavaScript validation, and simplified color scheme.

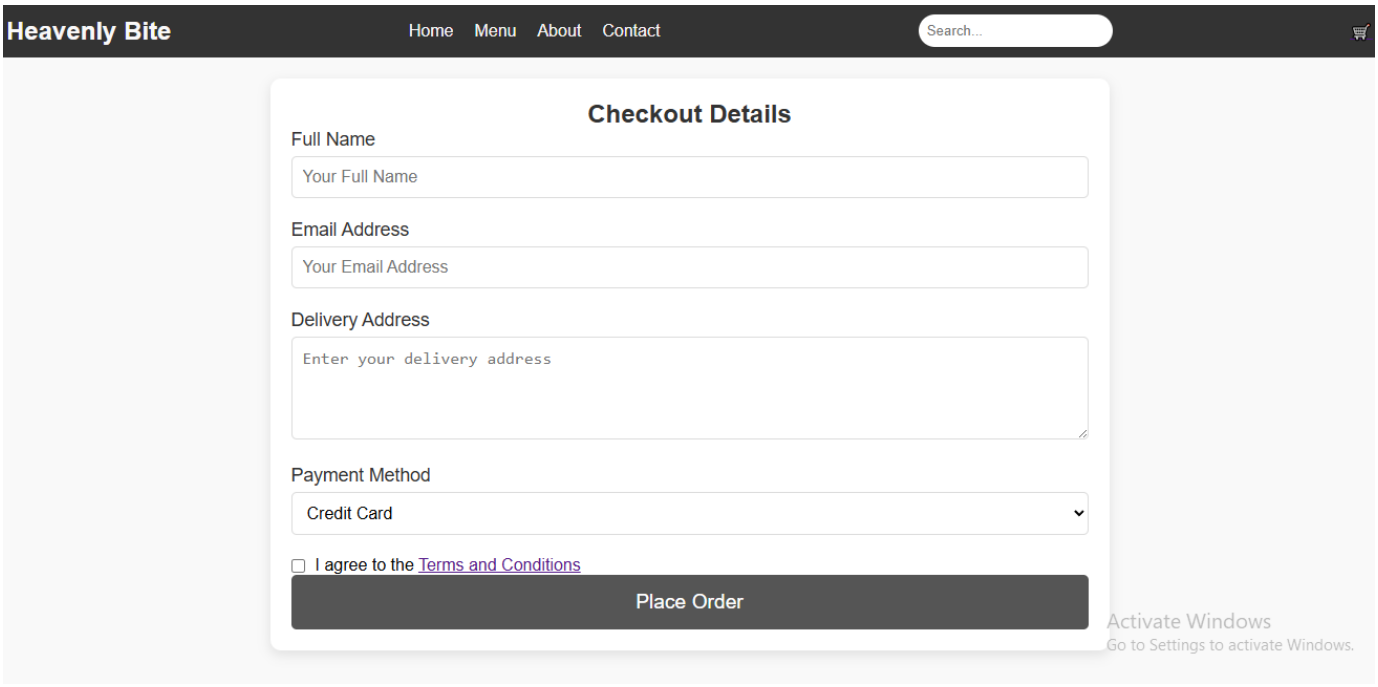

Figure 19:Before the tutor feedback

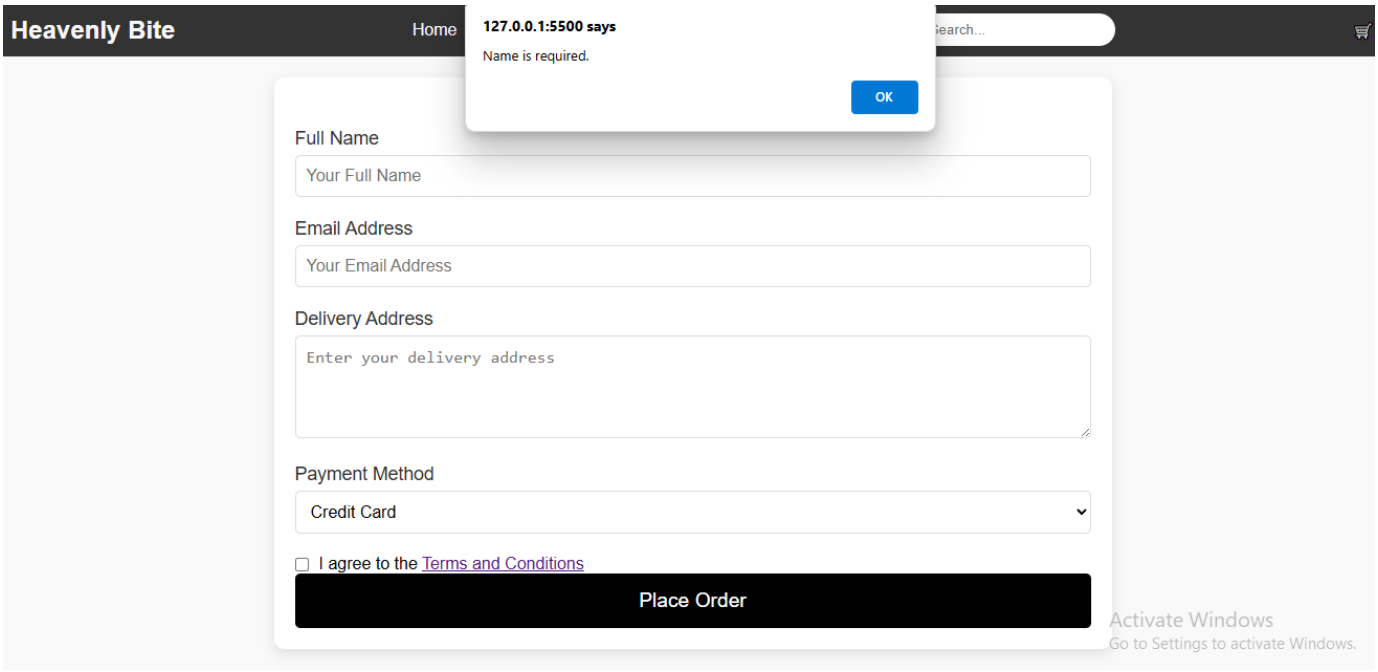

Figure 20:After the tutor feedback

6. References

- W3C HTML Validator. (n.d.). Retrieved from https://validator.w3.org/ WAVE Web Accessibility Tool. (n.d.).
- Retrieved from https://wave.webaim.org/
- axe DevTools. (n.d.). Retrieved from https://www.deque.com/axe/

Appendix: Screenshots

Home page

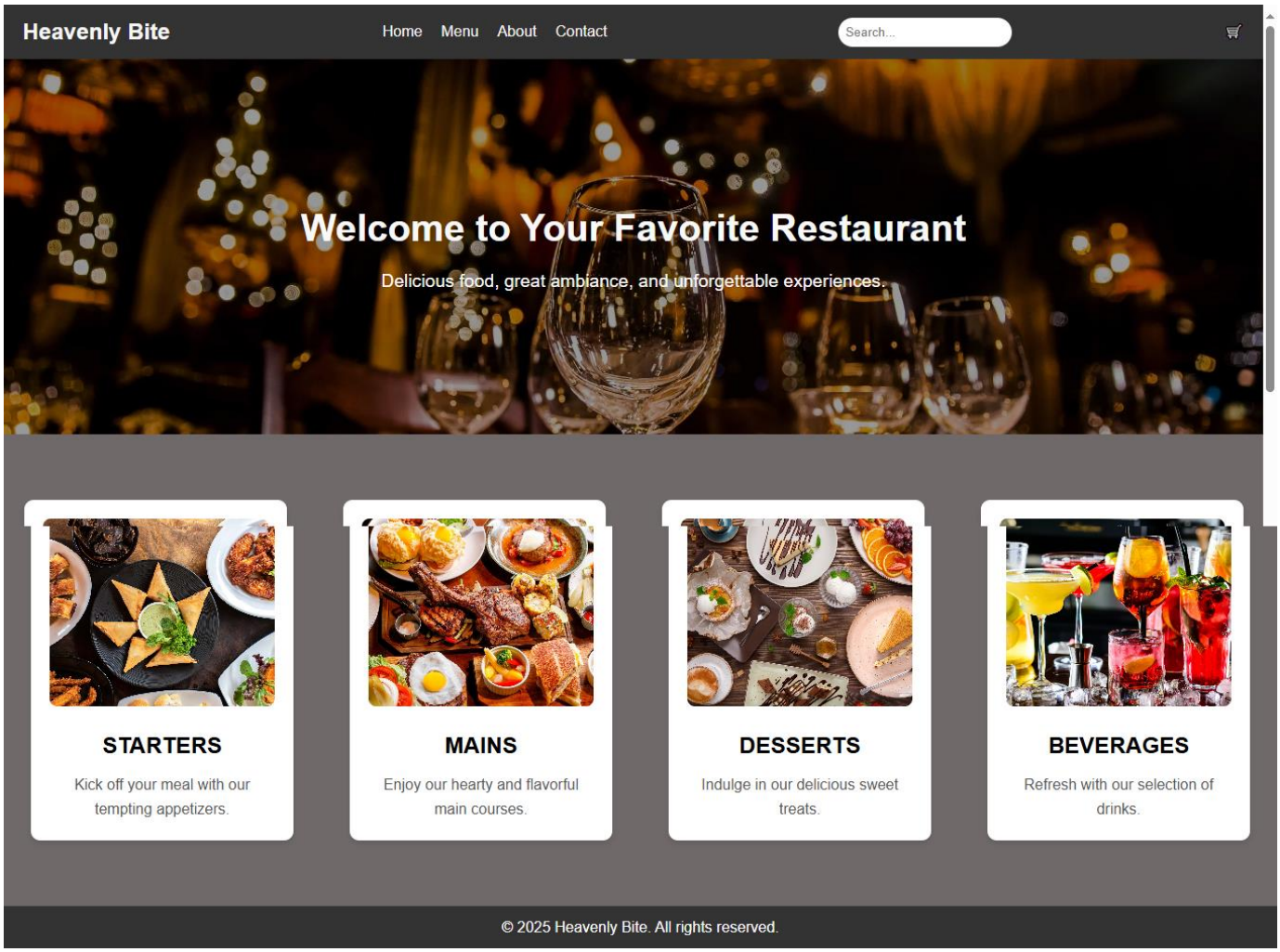

Figure 21:Home page

Menu page with xml

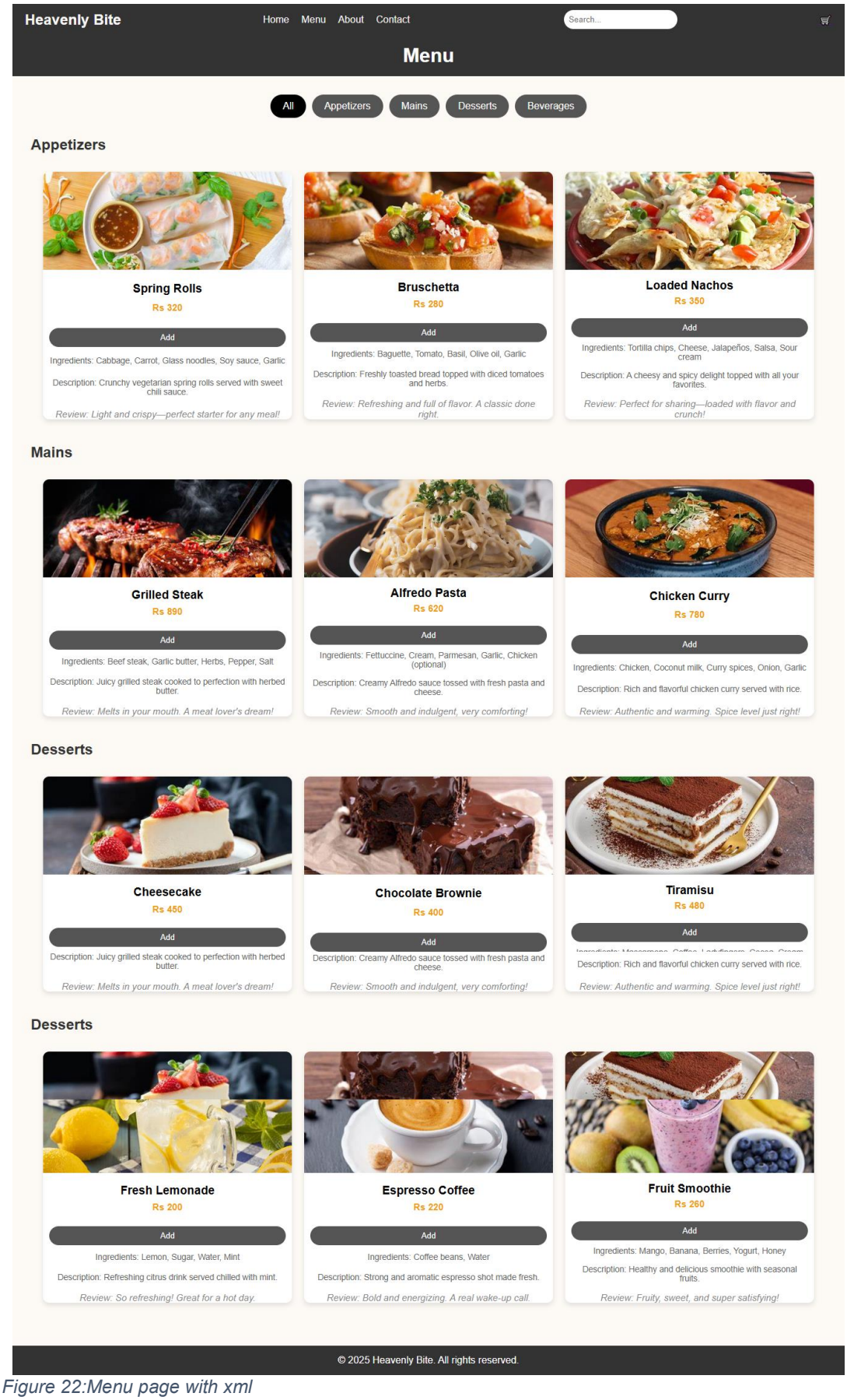

Figure 22:Menu page with xml

Menu page with filtering

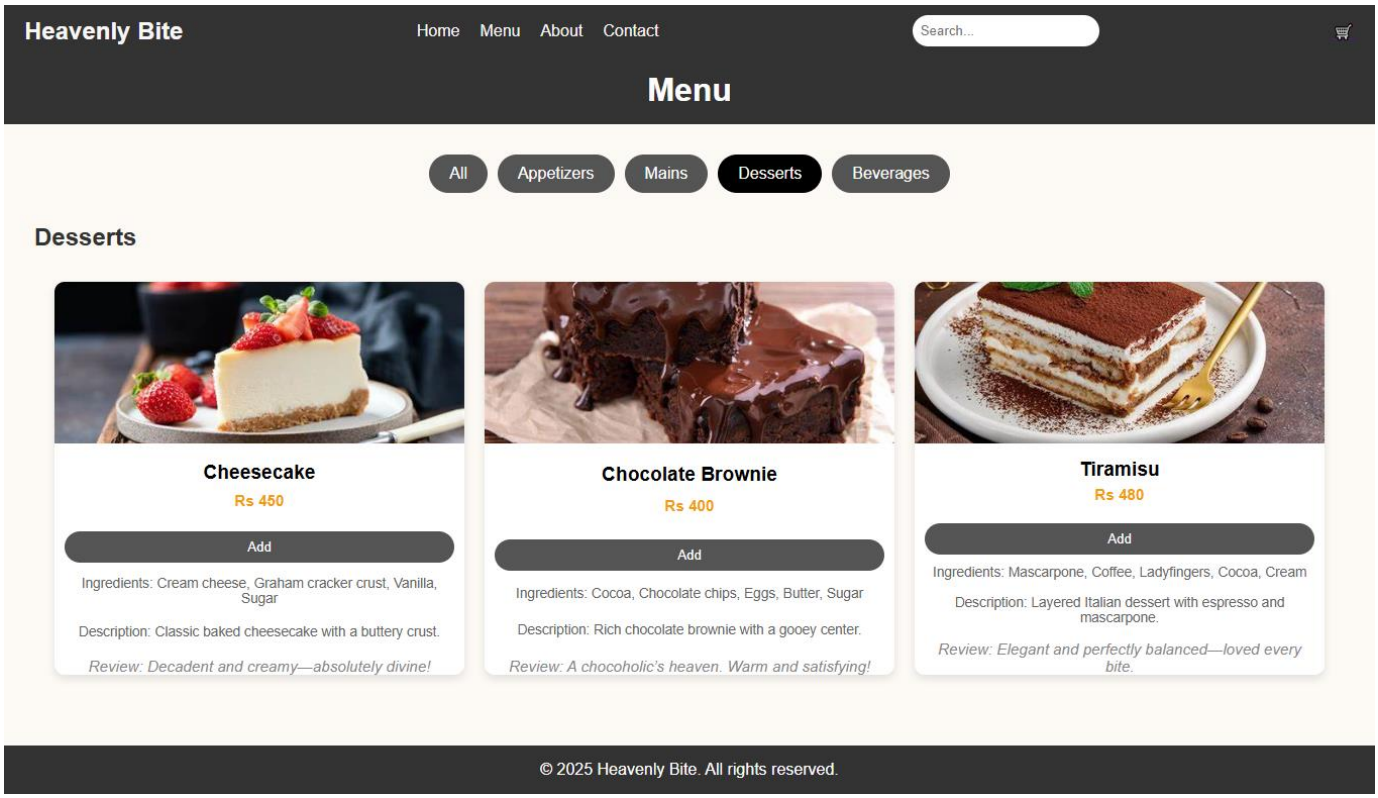

Figure 23:menu page with filtering

Order form with form elements

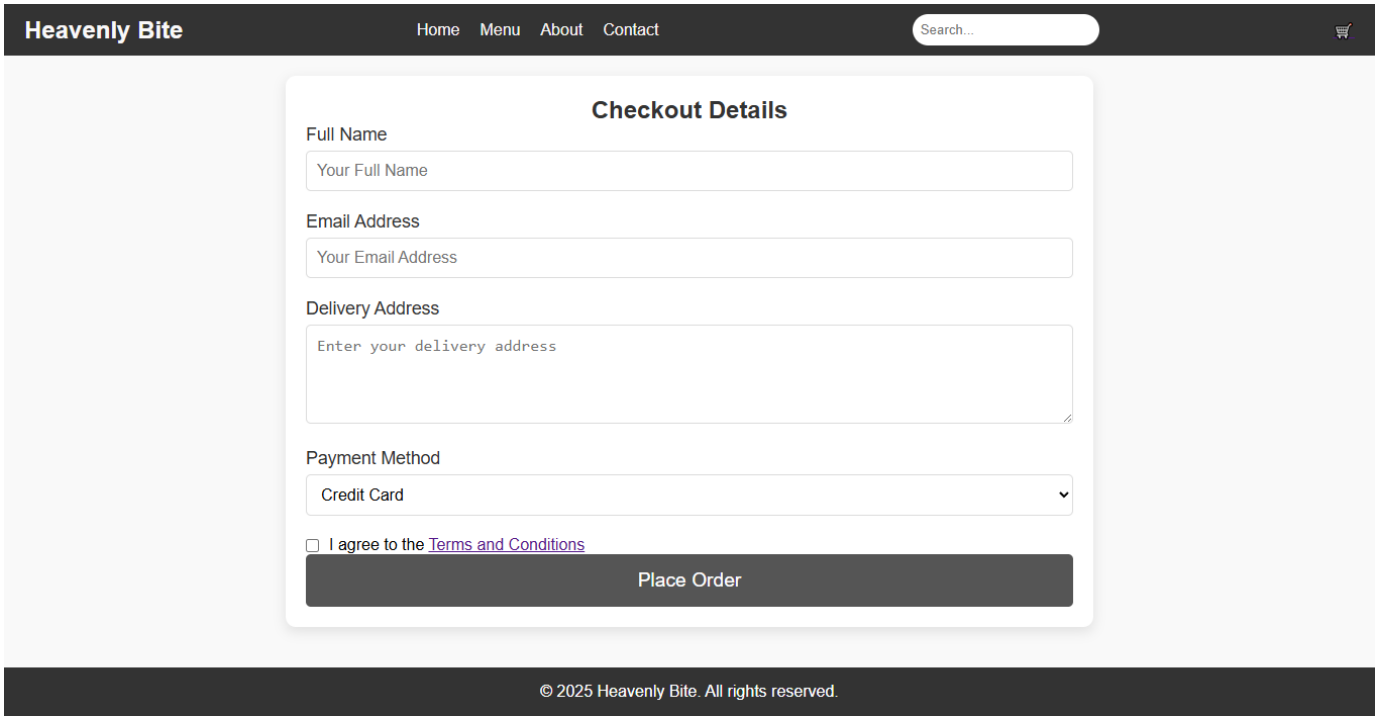

Figure 24:Order form

Cart

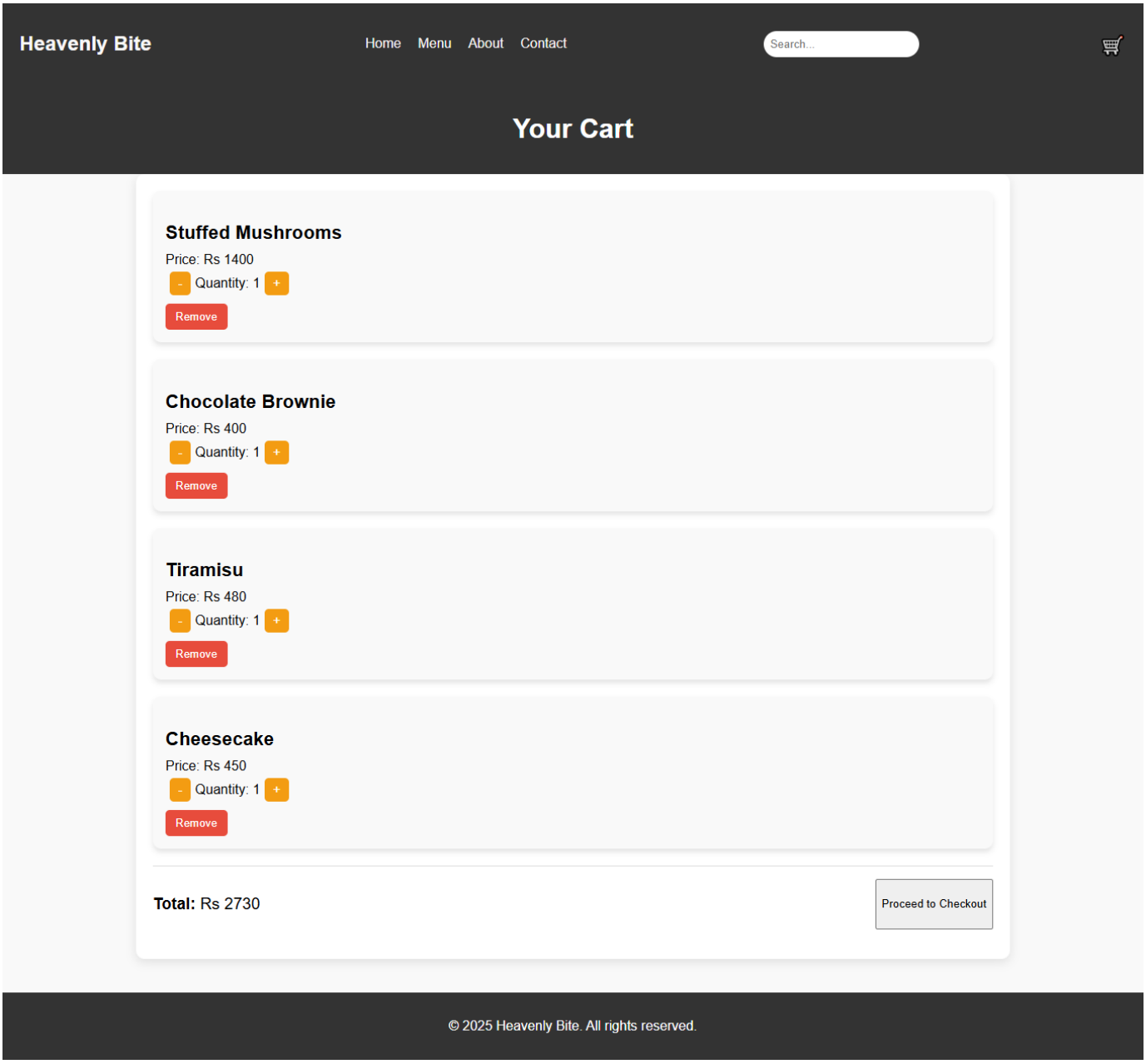

Figure 25:Cart