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

Table Of Figures

1.Introduction

This project involved creating and building a full restaurant website with the key elements of modern web design, including responsive layout, user interaction, and accessibility. The site includes a homepage with an interactive navigation menu, a menu page with JavaScript-based category filtering, and an order form with custom JavaScript validation. Moreover, the project uses XML data to display customer feedback on menu items, where there is the illustration of working with data sources that are in a structured format. The aim of this exercise was to apply UX/UI design principles like web accessibility, consistent layout, color contrast, and intuitive navigation, and ensure the technical quality of code through testing and validation. The result is visually appealing website that meets the coursework's design and functionality requirements.The idea was to create an accessible and user-friendly site for browsing through a menu, filtering products, reading reviews, and ordering.

- The most noteworthy features implemented are:
- A home page with navigational interaction.
- A menu page with filterable foods categorized as Appetizers, Mains, Desserts, and Beverages.
- A working order form incorporating JavaScript validation.
- XML integration to show and display customer reviews dynamically.
- Global CSS file uniform styling.
- Accessibility testing with tools such as WAVE and axe.

This report outlines the development process, the technical implementation, and the user interface and experience decisions, along with a self-reflection and validation evidence.

2.Technical Discussion

Technical incorporation of the restaurant website was implemented using foundation front-end technologies: CSS, JavaScript, HTML, and XML. This part elaborates on how each component of the website was developed, implications of the interactions with JavaScript, and how XML incorporation was accomplished in order to meet the requirements of the assessment.

1.Menu Filtering with JavaScript

The menu page (menu.html) displays a list of foods categorized into Appetizers, Mains and Desserts. Each menu item is prefixed with a data-category attribute in HTML to facilitate category-based filtering.

The filtering logic was done in vanilla JavaScript. When a user clicks a category button (say,

"Appetizers"), JavaScript loops through all items and hides the ones that do not match the clicked category. This makes the menu dynamic and improves user experience.

Purpose of This Feature:

- To allow users to browse menu items by preference easily
- To minimize clutter by showing only the concerned section

Technical Challenges:

- Enabling smooth toggling of items
- Maintaining the layout when items are shown or concealed

```
function filterCategory(category) {
  const categories = document.querySelectorAll('.menu-category');
  categories.forEach(section => {
    section.style.display = (category === 'all' || section.id === category) ? 'block' : 'none';
  });
}
```

Figure 1:filtering

2.Order Form Validation with JavaScript

The form on order.html contains six different types of inputs:

- Text input for name
- Email input for email address
- Textarea for delivery address
- Radio buttons for meal options
- Checkboxes for additional toppings

In order to meet the needs of the coursework, JavaScript-based validation has been employed instead of HTML5 form validation. Three field name, email, and address were selected for mandatory checks. Validation checks that:

- No field is empty
- Email is in valid format

- Address is long enough to count
- Success message is shown via alert() if the form is valid on all fronts.

Reasons for which JavaScript was utilized:

- HTML5 form validation was prohibited as per the specifications
- JavaScript provides more flexibility and customizable messages
- It mimics real-world client-side form validation

```
// Validate Name
if (name.length < 3) {
  document.getElementById("nameError").textContent = "Please enter a valid name (min 3 characters).";
  isValid = false;
}

// Validate Email
const emailPattern = /^[^ ]+@[^ ]+\.[a-z]{2,3}$/;
if (!emailPattern.test(email)) {
  document.getElementById("emailError").textContent = "Please enter a valid email address.";
  isValid = false;
}

// Validate Address
if (address.length === 0) {
  document.getElementById("addressError").textContent = "Address cannot be empty.";
  isValid = false;
}

// Validate Payment
if (payment === "") {
  document.getElementById("paymentError").textContent = "Please select a payment method.";
  isValid = false;
}
```


Figure 2:Validation

3.XML to display reviews

To store structured data about customer reviews, an XML file called reviews.xml was made. Every element has:

- The name of the menu item.
- Ingredients for the item
- Price

- The client's comments

The XML file was loaded using JavaScript's fetch() API. The XML string is transformed into a DOM structure by the DOMParser object, which is subsequently parsed and added as readable content to the HTML.

The Reason for Using XML:

- Both humans and machines can read and understand XML.
- It satisfies the evaluation criteria for incorporating outside data.
- Review data can be stored and shown on the menu page with its help.

```
function loadReviews() {
  const xhr = new XMLHttpRequest();
  xhr.open("GET", "menu_reviews.xml", true);
  xhr.onreadystatechange = function () {
    if (xhr.readyState === 4 && xhr.status === 200) {
      const xml = xhr.responseXML;
      const items = xml.getElementsByTagName("item");

      for (let i = 0; i < items.length; i++) {
        const item = items[i];
        const id = item.getAttribute("id");
        const description = item.getElementsByTagName("description")[0].textContent;
        const ingredients = item.getElementsByTagName("ingredients")[0].textContent;
        const review = item.getElementsByTagName("review")[0].textContent;
```

Figure 3:Display using XML

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

3.1 Navigation Techniques

The navigation bar is placed at the top of each page and includes links to the home page, menu, and order form. This grants access to all main sections on the site. It is themed using hover effects in CSS for greater interactivity. Breadcrumbs are not used since there are very few pages, but the design keeps users informed through explicit page titles and headings.

The website structure includes:

- index.html (Home)
- menu.html
- about.html
- contact.html

The navigation is made to be easily accessible, uniform throughout all pages, and shaped like buttons with hover effects.

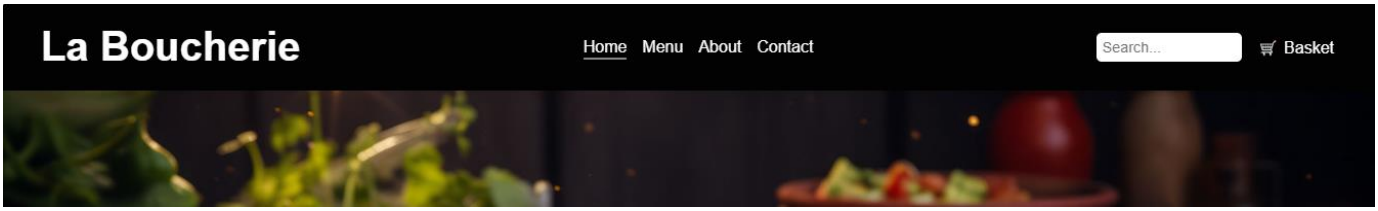

Figure 4:Navigation bar

```css
.nav-container {
    display: flex;
    align-items: center;
    gap: 20px;
}

h1 {
    margin: 0;
    font-size: 2.5rem;
    color: #ffffff;
}

nav ul {
    list-style: none;
    margin: 0;
    padding: 0;
    display: flex;
    gap: 15px;
}

nav ul li a {
    color: #fff;
    text-decoration: none;
    font-size: 1rem;
    transition: color 0.3s ease;
}
```


Figure 5:Navigation bar css

```html
<nav class="nav-container">
    <ul>
        <li><a href="home.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>
</nav>
```

Figure 6:Navigation bar html

3.2 Color balance/selection/consistency

The color scheme has a dark base color for highlights and buttons, but over neutral colors for readability. These colors were utilized to represent the dark and welcoming theme of a restaurant. All elements use consistent colors to provide a cohesive visual experience.

- **Primary**: #333
- **Secondary**: White and light gray backgrounds

```css
.category-link {
    display: block;
    margin-top: 8px;
    padding: 8px 15px;
    background: #333;
    color: #ccc;
    border-radius: 8px;
    text-decoration: none;
    font-weight: bold;
    transition: background 0.3s ease;
    font-size: 1rem;
}

.category-link:hover {
    background: #868581;
    color: #000;
}

footer {
    text-align: center;
    padding: 25px;
    background: #000;
    color: #fff;
}
```

Figure 7:Color balance

3.3 Color Contrast Test

The color palette was checked using tools like the WAVE Accessibility Tool to ensure sufficient contrast between text and background colors. It improves readability, especially for visually impaired users.

e.g: white text on dark buttons and black text on light backgrounds all passed the contrast tests.

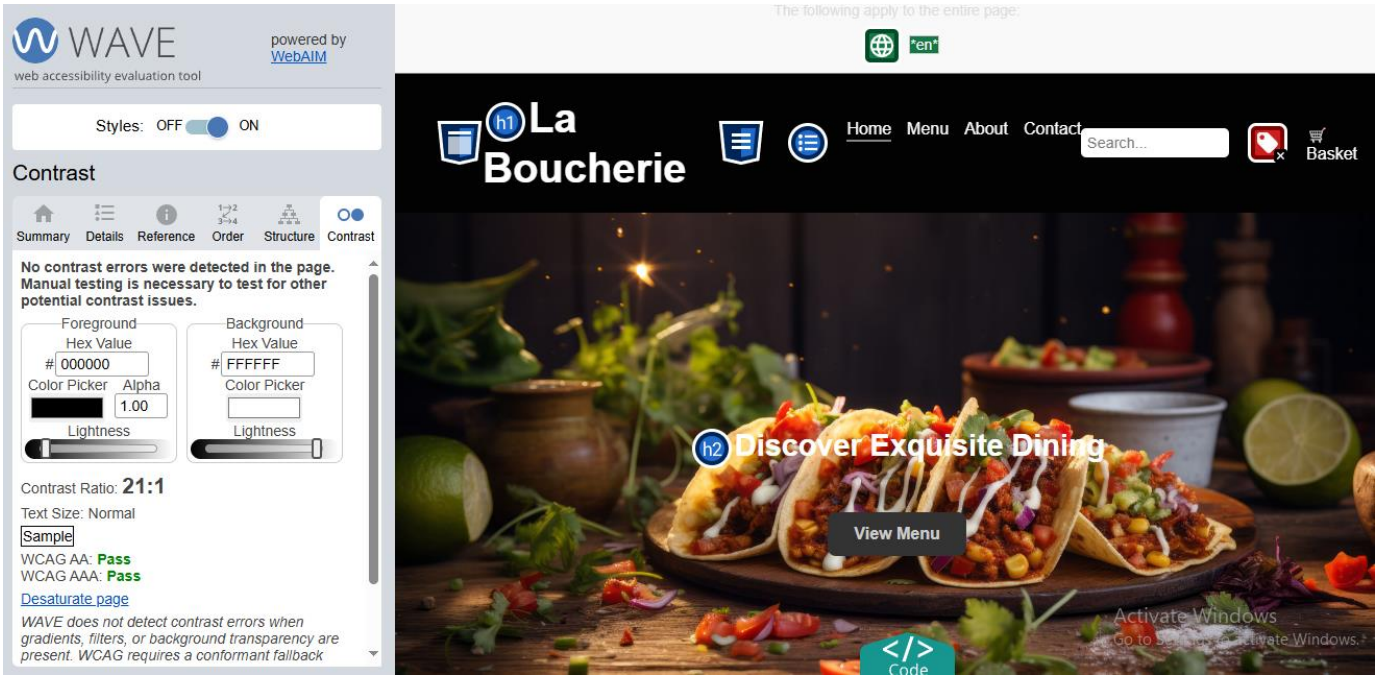

Figure 8:Wave color contrast (Home page)

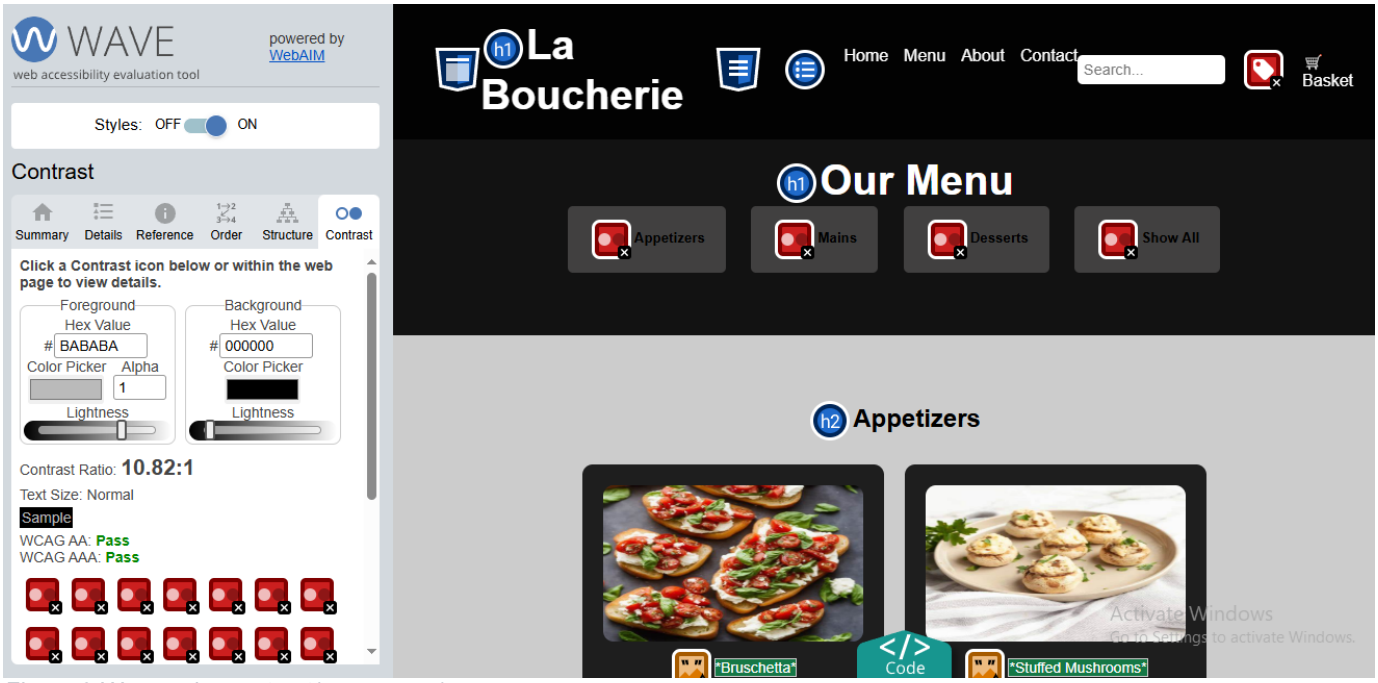

Figure 9:Wave color contrast(menu page)

3.4 Typography / consistency

Fonts such as 'Poppins' or 'Serif Sans' were used for body text, with large bold fonts employed for titles and headings. The font sizes and line heights were adjusted using CSS to maintain readability and hierarchy. Consistency throughout all pages was maintained through the application of reusable classes for headings, paragraphs, and button labels.

- **Heading Font**: 'Poppins', sans-serif;
- **Body Font**: 'Open Sans', sans-serif;
- **Size**: 16px for body, 24–32px for headings

```css
body {
    font-family: 'Poppins', sans-serif;
    margin: 0;
    padding: 0;
    background: #121212;
    color: #ddd;
}

header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 20px 40px;
    background: rgba(0, 0, 0, 0.9);
    position: sticky;
    top: 0;
    z-index: 1000;
}
```

Figure 10:Typography

Typography is defined in a global stylesheet to guarantee consistency.

3.5 Accessibility

Among the accessibility features are:

- Text accessibility, which includes clear headings and paragraph structures;

```html
<section class="menu-section">
    <h2>Discover Exquisite Dining</h2>
    <a href="menu.html" class="menu-button">View Menu</a>
</section>

<section class="about-us">
    <h2>About Us</h2>
    <p>Welcome to La Boucherie, where culinary excellence meets a cozy ambiance. Our chefs craft deli
</section>
```

Figure 11:Text accessibility

- Image accessibility, which includes meaningful alt attributes for every image.

```
<section class="menu-category" id="appetizers">
  <h2>Appetizers</h2>
  <div class="menu-items">
    <div class="menu-item" data-id="bruschetta">
      <img src="appetizer1.jpeg" alt="Bruschetta">
      <h3>Bruschetta</h3>
      <p class="price">Rs. 1200</p>
      <button onclick="addToCart('Bruschetta', 1200)">Add</button>
    </div>
    <div class="menu-item" data-id="Stuffed Mushrooms">
      <img src="appetizer2.jpeg" alt="Stuffed Mushrooms">
      <h3>Stuffed Mushrooms</h3>
      <p class="price">Rs. 1400</p>
      <button onclick="addToCart('Stuffed Mushrooms', 1400)">Add</button>
    </div>
    <div class="menu-item" data-id="Spring Rolls">
      <img src="appetizer3.jpeg" alt="Spring Rolls">
      <h3>Spring Rolls</h3>
      <p class="price">Rs. 1100</p>
      <button onclick="addToCart('Spring Rolls', 1100)">Add</button>
    </div>
  </div>
</section>
```

Figure 12:Image accessibility

- Form Accessibility: Every form input has a associated with it, and visible labels were not substituted with placeholders.Radio buttons and other grouped inputs were tagged with fieldset and legend.

```
<div class="container">
  <h2>Checkout</h2>
  <form id="orderForm" action="order_confirmation.html" method="POST" novalidate>
    <label for="name">Full Name:</label>
    <input type="text" id="name" name="name" />
    <div class="error" id="nameError"></div>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" />
    <div class="error" id="emailError"></div>

    <label for="address">Address:</label>
    <input type="text" id="address" name="address" />
    <div class="error" id="addressError"></div>

    <label for="payment">Payment Method:</label>
    <select id="payment" name="payment">
      <option value="">Select Payment Method</option>
      <option value="credit_card">Credit Card</option>
      <option value="paypal">PayPal</option>
      <option value="cash_on_delivery">Cash on Delivery</option>
    </select>
    <div class="error" id="paymentError"></div>
```

Figure 13:Form accessibility

These methods support users who use assistive technologies or screen readers.

3.6 Accessibility Test

Accessibility testing on the menu page and homepage was done using axe Accessibility Checker and WAVE. Both of them reported very minimal issues, which were primarily minor contrast warnings. All critical accessibility tests were passed.

Home page(index.html)

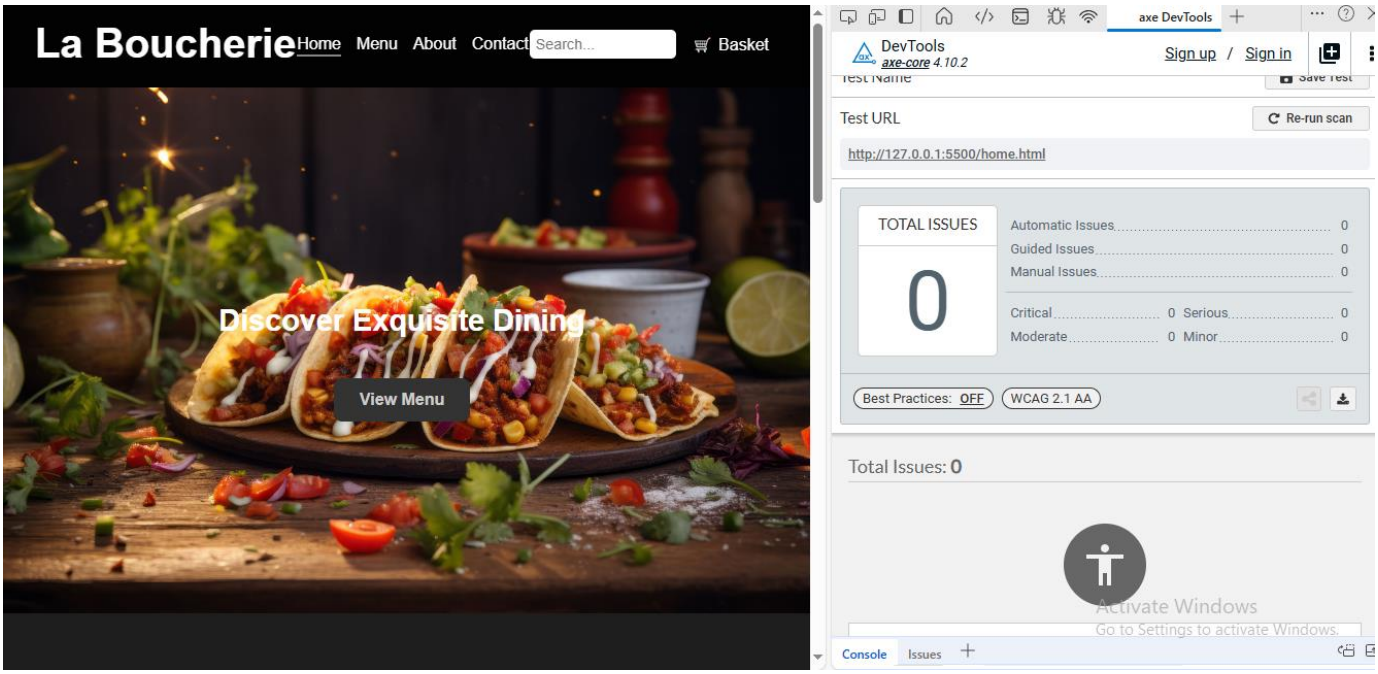

Figure 14:wave accessibility report on index.html

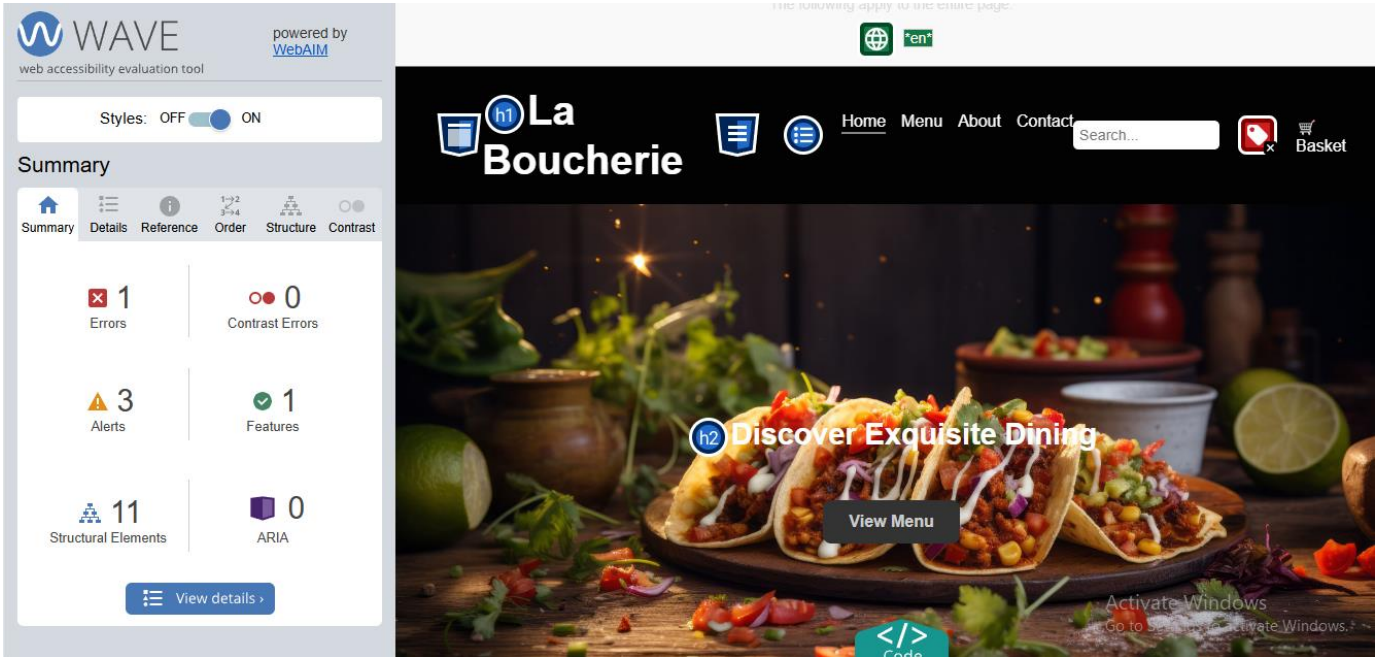

Figure 15:Axe accessibility report on index.html

Menu page(menu.html)

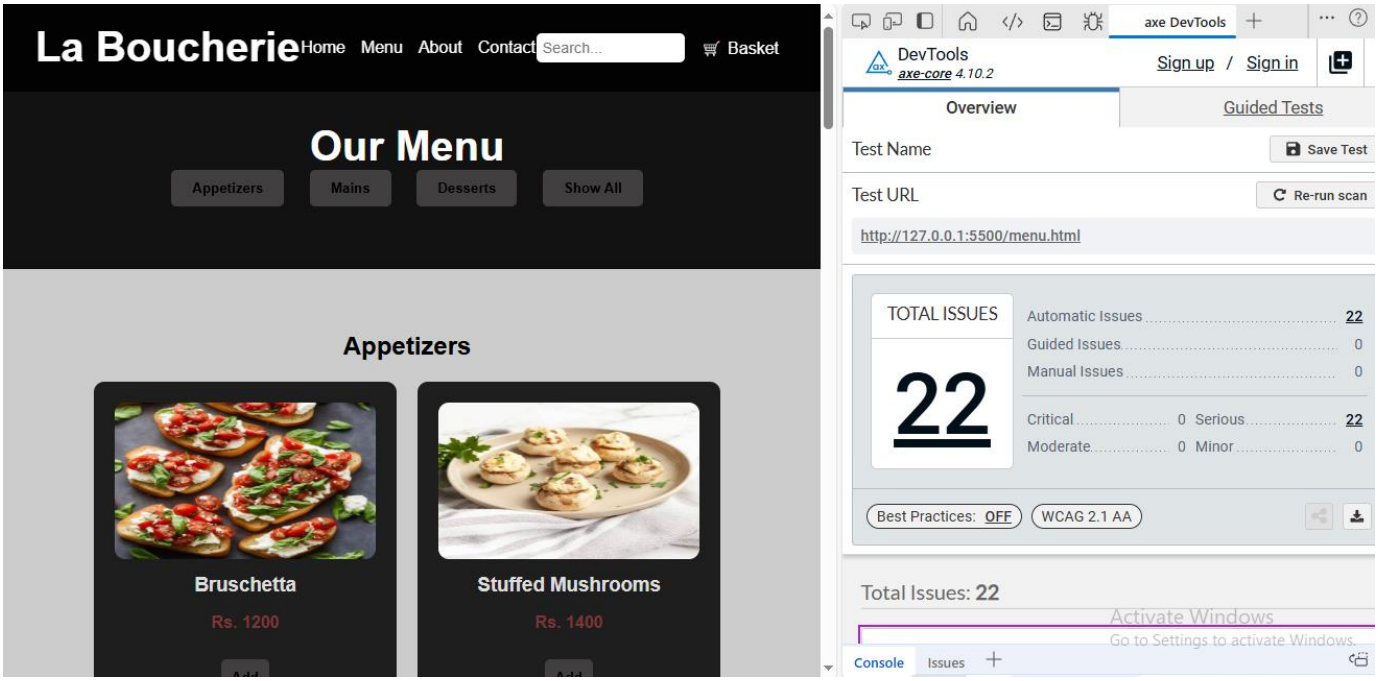

Figure 16:Wave accessibility report on menu.html

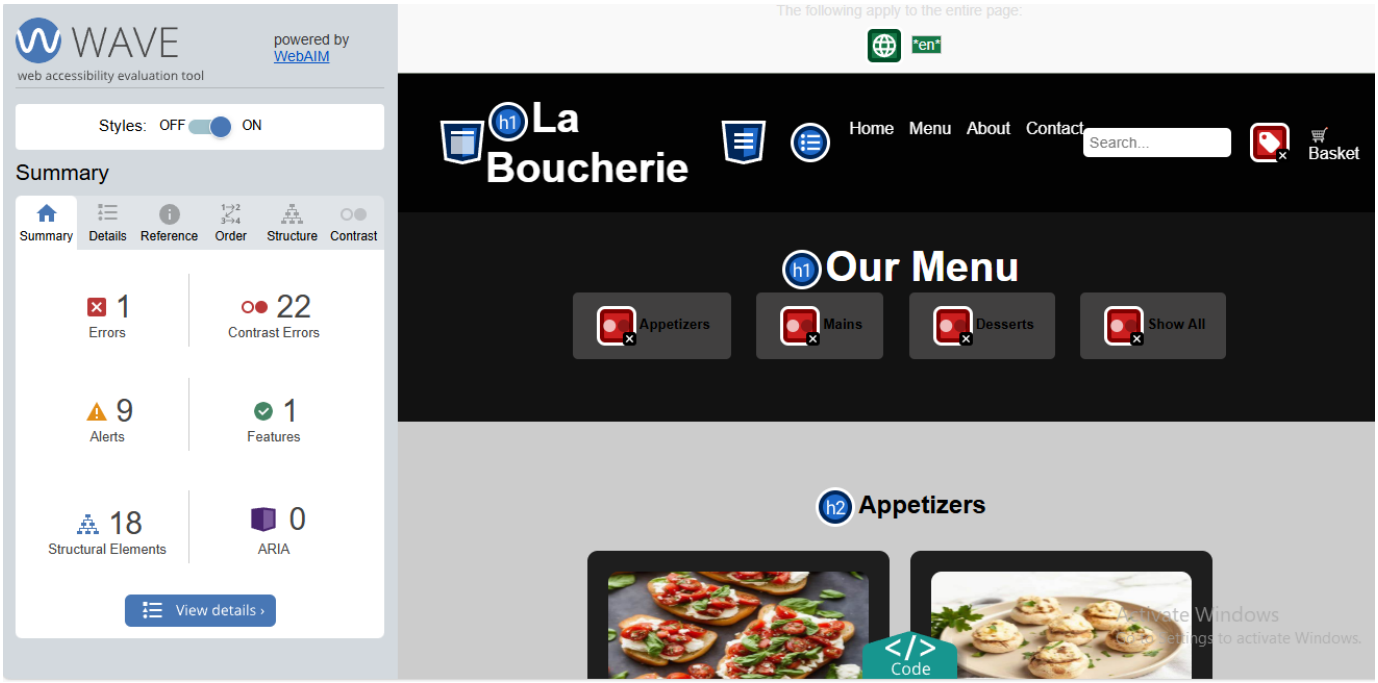

Figure 17:Axe accessibility report on menu.html

4.Validation Reports

To ensure that the website is web development standards compliant, two of the most critical pages index.html (home page) and menu.html (menu page) were validated using the official W3C Markup Validation Service for HTML and the W3C CSS Validation Service for CSS.

Home page(index.html)

The following image shows the report given by the W3C validation service for the home page(index.html)

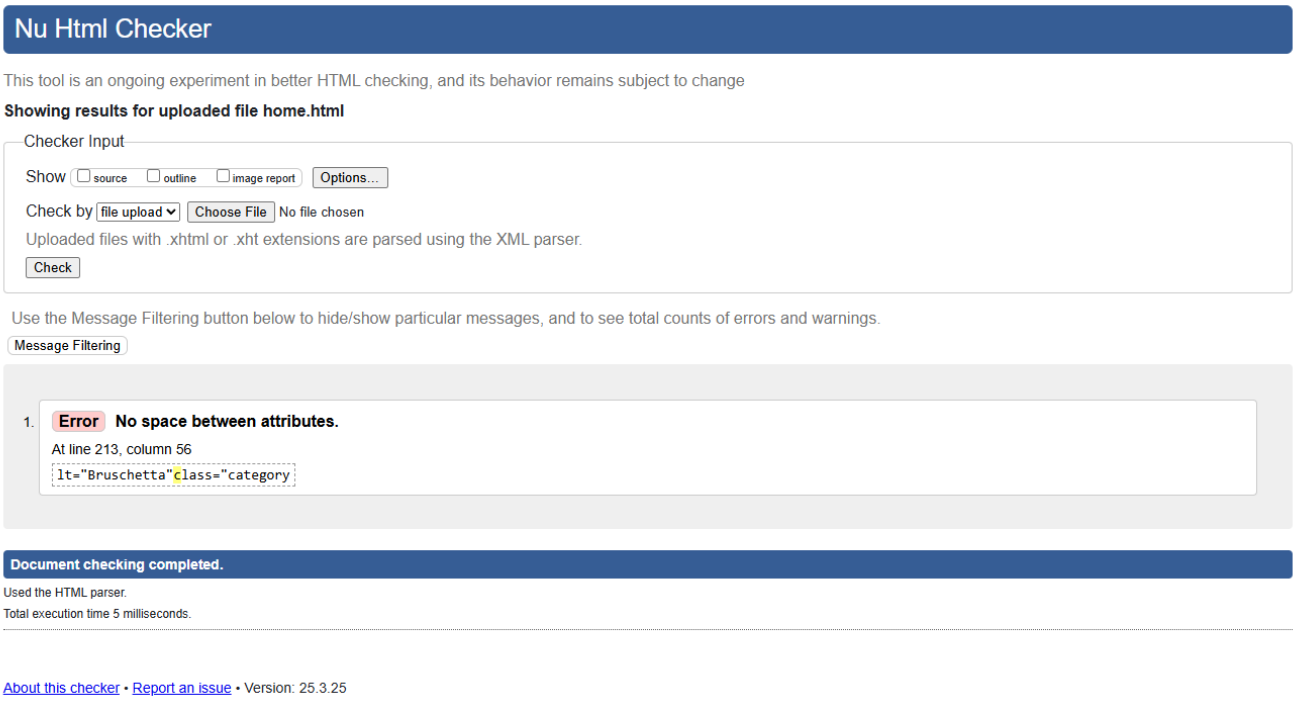

Figure 18:W3C validation(home page)

Menu page(menu.html)

The following image shows the report given by the W3C validation service for the home page(index.html)

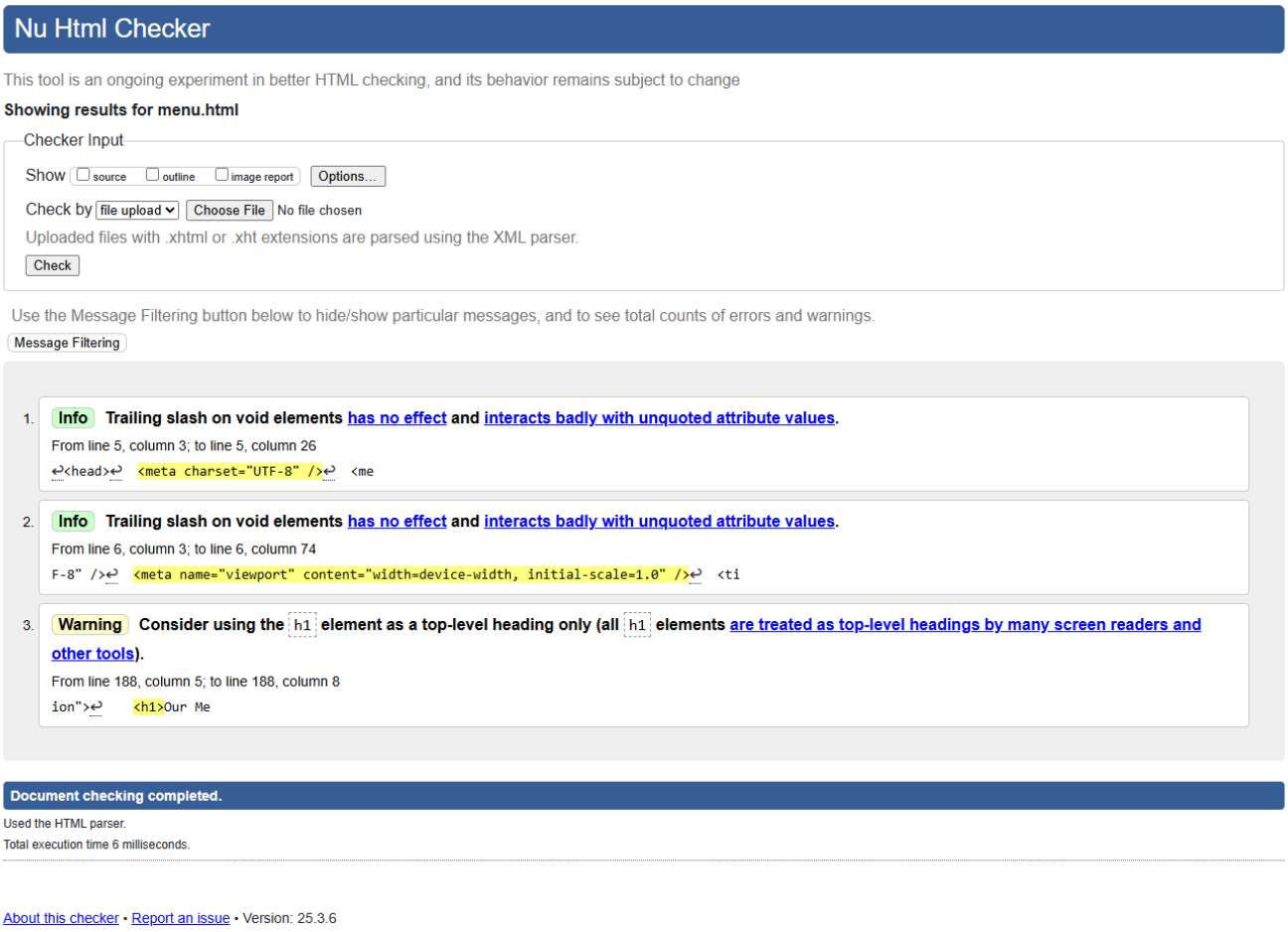

Figure 19:W3C validation for menu page

All pages were passed through the W3C HTML Validator, and its output did not report any critical errors. Minor warnings were reviewed and addressed as appropriate.

This validation provides the assurance that:

- All HTML tags are well-formed and correctly nested
- No deprecated or faulty tags were utilized
- Accessibility and semantic best practices are followed

5.Self-Reflection

Designing this restaurant site was a satisfying but challenging task. One of the most frustrating issues that I faced was using form validation in JavaScript without implementing HTML5 in-built validation, which required improved understanding of event handling and manipulation of the DOM. I also struggled at first with dynamic menu item filtering but, through experience and experimenting with different logic fixes, I managed to create a smooth and responsive filter system. Tutor comments helped me refine the UI design specifically, simplifying the color scheme to enhance readability and improving navigation clarity. These refinements made the site more usable and visually appealing. Overall, this project enhanced my hands-on proficiency in HTML, CSS, JavaScript, and XML, and taught me about the importance of accessibility, consistency in design, and adherence to web standards.

Challenges:

- Implementation of form validation without HTML5 constraints
- Dynamic XML parsing and rendering
- Preserving accessibility at the cost of design appeal

Solutions:

- Specific JS functions for the validations
- DOMParser for XML
- Accessibility tool-based regular checks

Tutor Feedback :

Our tutor told me focus more on the input types of the html forms so I went ahead , found information and updated the code again.

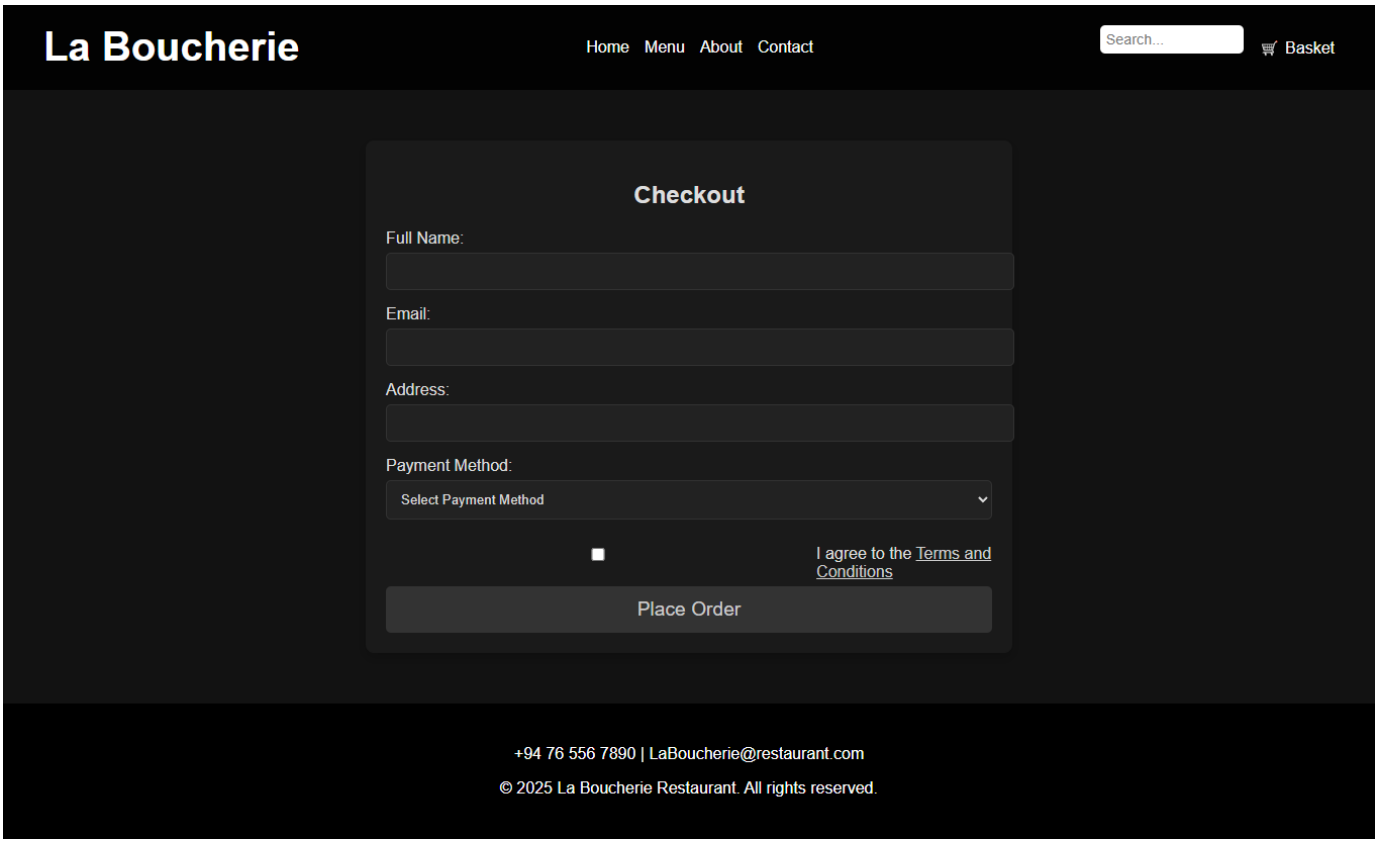

Figure 20:Order page well defined in a user friendly manner

```
<div class="container">
  <h2>Checkout</h2>
  <form id="orderForm" action="order_confirmation.html" method="POST" novalidate>
    <label for="name">Full Name:</label>
    <input type="text" id="name" name="name" />
    <div class="error" id="nameError"></div>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" />
    <div class="error" id="emailError"></div>

    <label for="address">Address:</label>
    <input type="text" id="address" name="address" />
    <div class="error" id="addressError"></div>

    <label for="payment">Payment Method:</label>
    <select id="payment" name="payment">
      <option value="">Select Payment Method</option>
      <option value="credit_card">Credit Card</option>
      <option value="paypal">PayPal</option>
      <option value="cash_on_delivery">Cash on Delivery</option>
    </select>
    <div class="error" id="paymentError"></div>

    <div class="terms">
      <input type="checkbox" id="terms" name="terms" />
      <label for="terms">I agree to the
        <a href="terms.html" style="color: #ccc;">Terms and Conditions</a></label>
    </div>
    <div class="error" id="termsError"></div>
```


Figure 21:Form

6.References

- WAVE Accessibility Tool. (n.d.). *wave.webaim.org*.
- axe DevTools. (n.d.). *https://www.deque.com/axe/*
- W3C Validator. (n.d.). *validator.w3.org*
- W3Schools. (n.d.). *HTML, CSS, JavaScript Tutorials*.

Appendix: Screenshots

Home page

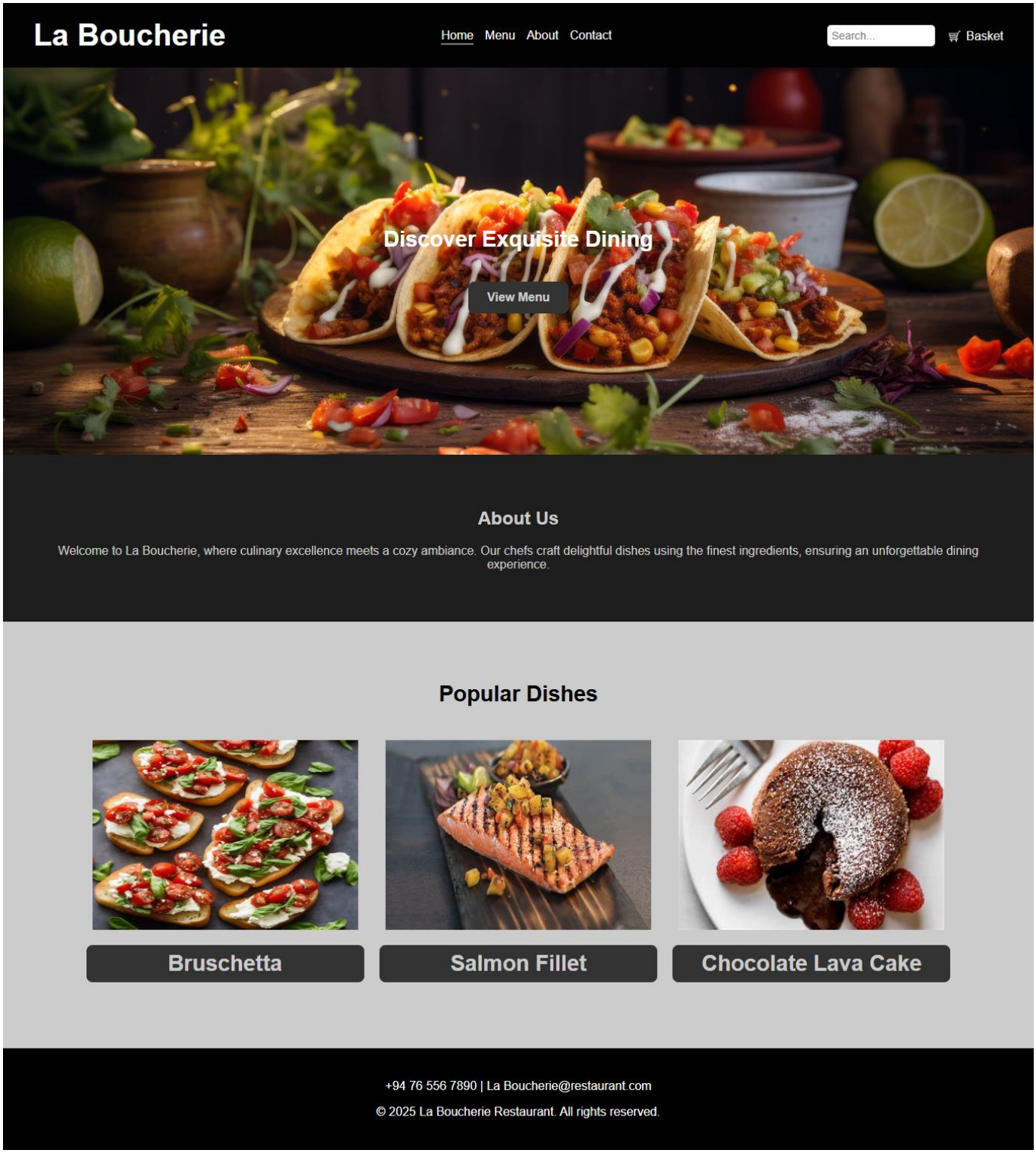

Figure 22:Home page

Menu page with filtering option

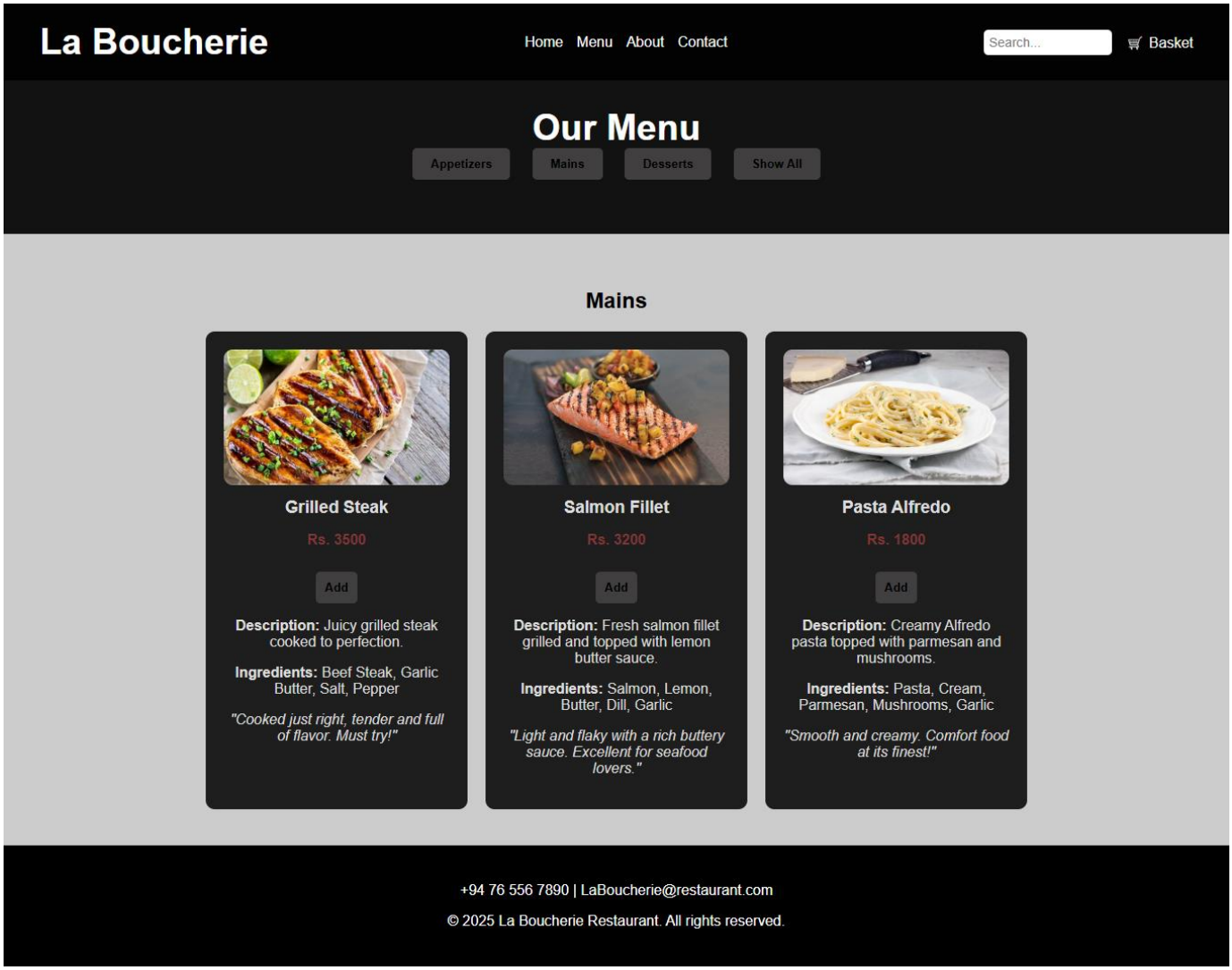

Figure 23:Menu page with filtering

Order page with validation

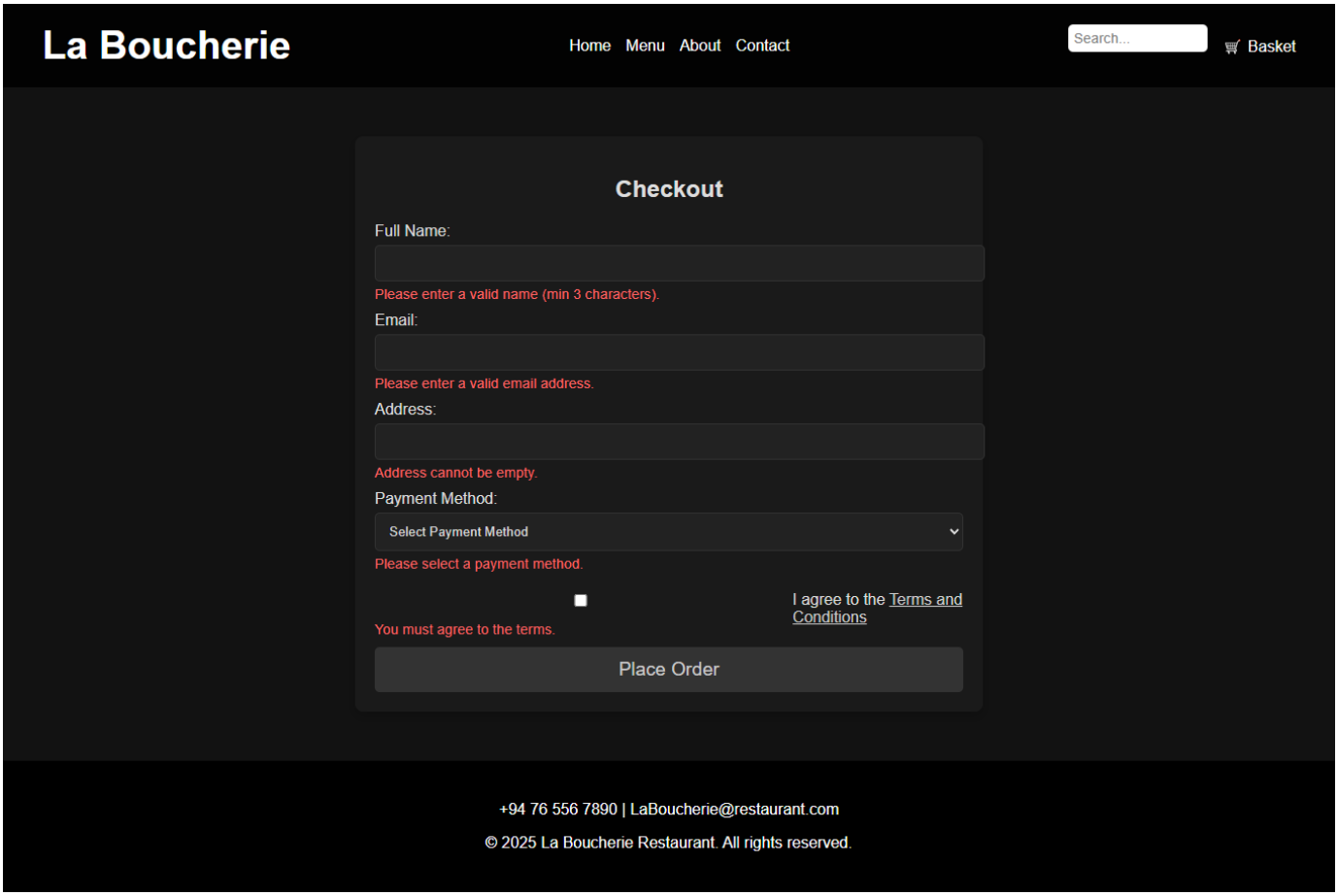

Figure 24:Order page with validation

Menu page with xml integrate

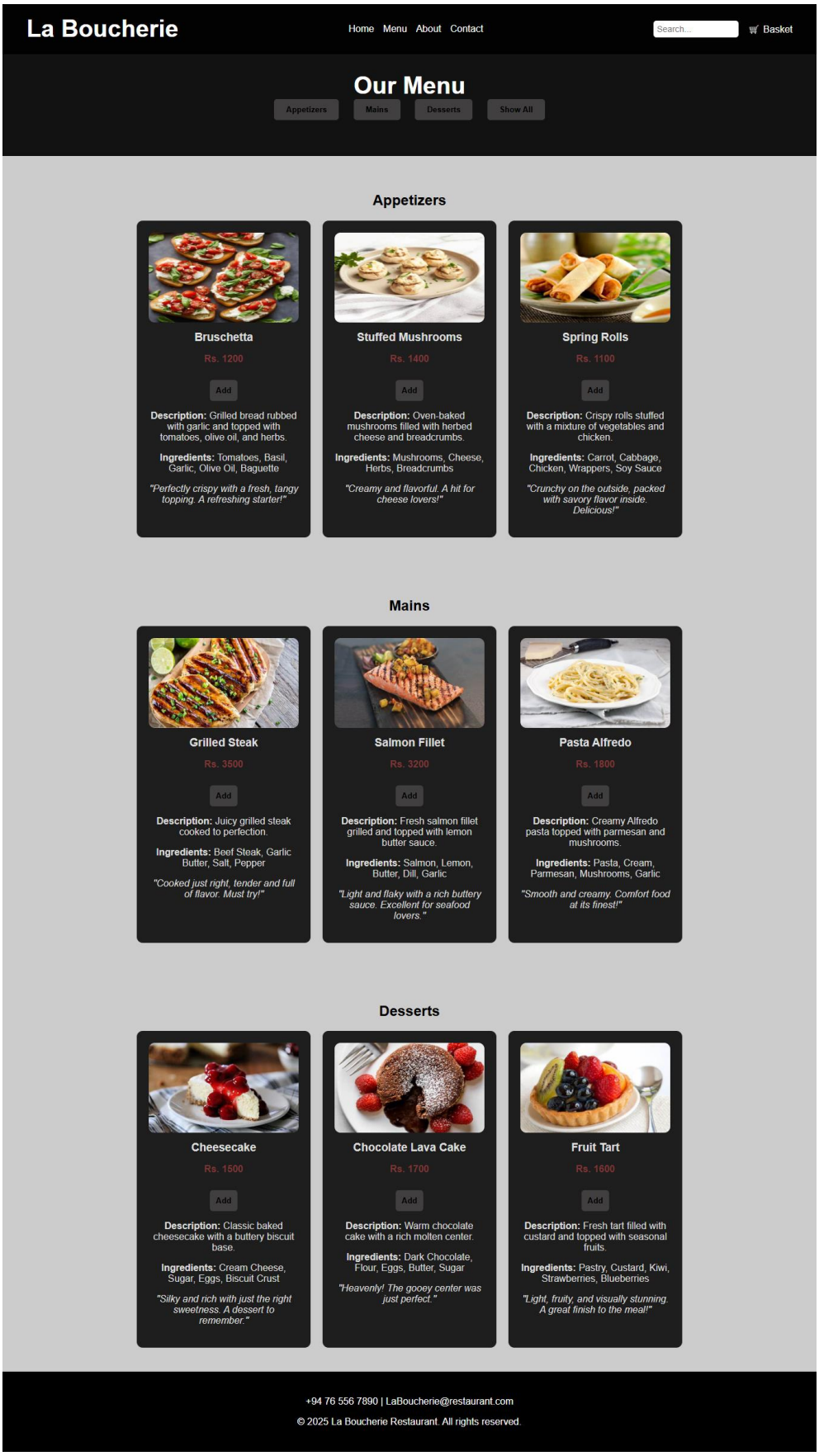

Figure 25:menu page with xml