

Code Explanation: One-Page Mini Restaurant Website (Assignment 3)

Visha Hameed

December 2025

Overview

This project creates a **responsive, single-page website** for a restaurant using HTML, CSS, and JavaScript, focusing on modern web standards and user interaction, as required by Assignment 3.

1 Code Breakdown

1.1 1. index.html (Structure)

This file defines the **content and layout** of the webpage.

- **Structure:** Uses semantic sections (`<header>`, `<section>`, `<footer>`) to organize the page content: Home, Menu, Reservation, Reviews, and Contact.
- **Menu:** The menu is structured hierarchically for three categories (Starters, Main Course, Desserts) and includes 2-3 dishes per category.
- **Forms:** Contains the table reservation form with all required input fields, which are marked as **required** for basic browser validation.
- **Reviews:** Includes three review items, initially prepared for the slideshow.

1.2 2. style.css (Presentation and Responsiveness)

This file controls the **visual appearance and responsiveness** of the website.

- **Layout:**
 - The navigation bar and menu categories utilize **Flexbox** (`display: flex`) for distribution and alignment.
 - The menu items within categories and the reservation form are structured using **CSS Grid** (`display: grid`) for clean element arrangement.
- **Interactivity:** Includes hover effects (`:hover`) on navigation links and menu items for user feedback.
- **Responsiveness:** Achieved using **Media Queries** (`@media (max-width: 768px)`) to adapt the layout for mobile devices, specifically:
 - The header navigation shifts to a column layout.
 - The menu layout changes from three columns to a single column.

1.3 3. script.js (Dynamic Functionality)

This file adds the required **dynamic functionality** using JavaScript and DOM manipulation.

- **Review Slider:**

- The script selects review elements and uses a counter (**currentReview**).
- On the "Next Review" click, it toggles the **hidden** CSS class on elements to hide the current review and show the next one.
- The index wraps using the **modulo operator (%)**:

$$(currentReview + 1) \% reviews.length$$

- **Form Validation:**

- An event listener on the form intercepts submission and prevents default action (**event.preventDefault()**).
- It checks if any required input field's value is empty.
- If validation fails, an **alert()** is shown. If successful, a confirmation **alert()** is displayed, followed by form reset (**reservationForm.reset()**).