RESTAURANT MANAGEMENT SYSTEM

Minor Project Report

Submitted to

SRI PADMAVATI MAHILA VISVAVIDYALAYAM

In Partial fulfilment of the requirement for the

MASTER OF COMPUTER APPLICATIONS

III SEMESTER

By

PANCHAKSHARI DEEPTHI (2024MCA16074)

TATA DURGA BHAVANI (2024MCA16100)

ULASI GNANESWARI (2024MCA16103)

Under the guidance of

G. PRATHYUSHA

Accredited by NAAC with A + Grade ISO 9001 : 2015 Certified

DEPARTMENT OF COMPUTER SCIENCE

SRI PADMAVATI MAHILA VISVAVIDYALAYAM

(Women's University)

Tirupati-517502(A.P), Andhra Pradesh

DECEMBER**, 2025**

DEPARTMENT OF COMPUTER SCIENCE

SRI PADMAVATI MAHILA VISVAVIDYALAYAM

(Women's University)

Tirupati-517502(A.P), Andhra Pradesh, India

Accredited with , A+" Grade by NAAC

CERTIFICATE

This is to certify that the project work entitled "**Restaurant Management System** "is a bonafide record of work carried by PANCHAKSHARI DEEPTHI (2024MCA16074), TATA DURGA BHAVANI (2024MCA16100) and ULASI GNANESWARI (2024MCA16103). In the **Department of Computer Science**, SRI PADMAVATI MAHILA VISVAVIDYALAYAM, Tirupati in partial fulfilment of the requirements of III Semester of **MASTER OF COMPUTER APPLICATIONS**. The content of the Project Report has not been submitted to any other University for the award of any degree.

GUIDE **HEAD OF THE DEPARTMENT**

DECLARATION

We hereby declare that MCA III Semester minor project entitled **"RESTAURANT MANAGEMENT SYSTEM"** was done at **Department of Computer Science, Sri Padmavati Mahila Visvavidyalayam**, Tirupati, in the year 2025- 2026 under the guidance of **G. Prathyusha** in partial fulfilment of requirements of MCA III Semester.

We also declare that this project is our original contribution of the best of our knowledge and belief. We further declare that this work has not been submitted for the award of any other degree of this or any other university/Institution.

Signature of the Students

ACKNOWLEDGEMENT

We are greatly indebted to our guide **G. Prathyusha** for taking keen interest in my project work and providing valuable suggestions in all the possible areas of improvement.

We express our sincere thanks to the teaching staff of the Department of Computer Science for extending support and encouragement to us in all the stages of the project work.

We gratefully acknowledge and express our gratitude to the non-teaching staff of the Computer Science Department who supported us in preparing the project report.

Signature of the Student(s)

INDEX

ABSTRACT

This project presents a fully frontend-based **Restaurant Management System** developed using **HTML, CSS, and JavaScript**. The system provides an interactive and user-friendly interface for managing restaurant activities such as viewing menus, placing orders, reserving tables, and generating sample bills. With responsive layouts and dynamic UI elements, the application ensures accessibility across different devices. JavaScript is used to handle client-side logic including form validation, order calculations, and real-time updates, eliminating the need for backend integration. This project demonstrates how frontend technologies can be utilized to simulate restaurant workflow efficiently while enhancing user experience.

1. INTRODUCTION

1.1. UNIVERSITY PROFILE:

Sri Padmavati Mahila Visvavidyalayam(Women's University) is a public university located in Tirupati, Andhra Pradesh, India. Established in 1983 by Sri N.T.Rama Rao, the Former Chief Minister of Andhra Pradesh. It is dedicated to women's education and empowerment, offering a wide range of undergraduate, postgraduate, and doctoral programs across various disciplines.

Key Features:

-Founding Vision: The university was established to promote higher education for women, enabling them to achieve academic excellence and contribute to society.

-Campus: The university campus is spread over a large area, providing a conducive environment for learning and research. It includes well-equipped classrooms, laboratories, libraries, hostels, and other amenities.

- Programs Offered:

-Undergraduate Programs: Various disciplines such as Arts, Sciences, Engineering, Technology, Commerce, and Education.

-Postgraduate Programs: Specialized programs across disciplines, including Arts, Sciences, Management, and Engineering.

-Doctoral Programs: Research opportunities in multiple fields -Research and Development: The university encourages research and innovation, with several centers dedicated to specific areas of study.

-Affiliations and Accreditations: The university is recognized by the University Grants Commission (UGC) and has various programs accredited by **NAAC** (National Assessment and Accreditation Council) with an „**A+**“ grade.

-Student Life: The university offers a vibrant campus life with numerous cultural, sports, and extracurricular activities aimed at holistic development.

-Community Engagement: The university emphasizes social responsibility, with various outreach programs and initiatives to support local communities.

Sri Padmavati Mahila Visvavidyalayam is a significant institution for women's education in Andhra Pradesh, contributing to the academic and professional growth of women in the region.

2. PROBLEM DEFINITION

2.1. AIM:

The primary aim of the **Restaurant Management System** is to design and develop an interactive and user-friendly web application that simplifies the essential tasks within a restaurant environment. The goal is to automate menu display, order placement, billing calculations, and table reservations using only frontend technologies.

2.2. PROBLEM DEFINITION:

Traditional Restaurant Management System often involves numerous manual tasks which can lead to various operational problems. Manual ordering, paper-based billing, and unorganized table reservations create inefficiencies and errors, which affect customer experience and restaurant productivity.

Identified Problems:

1. **Manual Order Errors:** Handwritten or verbal order-taking can cause mistakes, miscommunication, and slow service.
2. **Time-Consuming Billing Process:** Calculating bills manually is slow and can lead to incorrect totals.
3. **Unorganized Menu Handling:** Paper menus can become outdated or damaged and lack interactivity.
4. **Inefficient Table Reservation:** Managing reservations manually often leads to confusion or double bookings.
5. **Lack of Digital Experience:** Customers expect modern and fast digital interfaces in today's world.
6. **No Centralized Workflow:** Manual processes do not provide a unified system for managing operations

2.2.1. Existing System (Manual System)

Most small restaurants traditionally rely on manual processes for ordering, billing, and reservations. These processes often involve handwritten notes, verbal communication, and paper-based menus. While simple, this approach presents several challenges:

Limitations of the Existing Manual System

1. **High chances of human error:** Miscommunication between waiters and kitchen staff leads to incorrect orders.
2. **Time-consuming billing:** Manual calculation of bills delays service.
3. **Unorganized table reservation process:** Paper logs or verbal bookings can result in double bookings or confusion.
4. **No data tracking or centralized system:** Difficult to maintain records of daily orders, reservations, or income.
5. **Poor customer experience:** Customers expect digital menus and faster service in modern restaurants.
6. **Menu updates are difficult:** Changing printed menus takes time and money.

2.2.2. Proposed System (Frontend-Based RMS)

The proposed system is a digital, interactive, and user-friendly Restaurant Management System built using **HTML, CSS, and JavaScript**. It replaces manual operations with an automated interface where users can browse the menu, place orders, view their bill, and reserve tables.

Features of the Proposed System

- Digital menu display
- Add-to-cart functionality
- Automated billing
- Table reservation form
- Responsive design
- Clean, modern UI
- Dynamic functionality using JavaScript

This system improves the efficiency of restaurant management and offers a better user experience.

2.3. OBJECTIVES:

Main Objective

To develop an interactive, user-friendly, and efficient frontend-based Restaurant Management System using HTML, CSS, and JavaScript to streamline menu display, order handling, billing simulation, and table reservations.

Detailed Objectives

1. To Create a User-Friendly Interface

- Design a clean and attractive interface for easy navigation.
- Provide a smooth user experience using responsive layouts.

2. To Digitize Menu Display

- Display restaurant menu items in a structured and visually appealing way.
- Allow users to easily browse food items and check their prices.

3. To Implement an Order Management System

- Enable customers to add or remove items from their cart.
- Update the order list dynamically using JavaScript.

4. To Automate Billing Calculation

- Automatically calculate the total bill based on selected items.
- Display real-time updates as items are added or removed.

5. To Provide a Table Reservation Feature

- Create a simple reservation form for users.
- Validate user inputs and show reservation confirmation.

6. To Demonstrate Client-Side Functionality Without Backend

- Handle all operations (cart, billing, reservation) using only JavaScript.
- Store temporary data using local arrays or variables.

7. To Ensure Device Compatibility

- Make the system accessible on laptops, tablets, and mobile devices.
- Use responsive design techniques (media queries, flexbox, grid).

8. To Improve Operational Efficiency in Restaurants

- Reduce manual errors in order-taking and billing.
- Speed up service through digital interaction.

9. To Prepare a Foundation for Future Enhancements

- Create a structure that can be later expanded with backend integration.
- Enable easy improvement for real-world restaurant deployment.

3. SYSTEM ANALYSIS

System Analysis involves studying and understanding the existing system, identifying its limitations, and determining the requirements for the proposed system. For a Restaurant Management System, the goal is to examine how restaurant operations currently function and how digital transformation can simplify and enhance those operations.

3.1. Software Requirement Specifications:

Introduction

The software requirements specify the tools, platforms and technologies neede to develop, run, and test the Restaurant Management System. Since this project is entirely frontend-based, the requirements are minimal and rely on widely available, open-source tools.

Purpose:

The operating system provides the base environment required to run web browsers and code editors. Any OS that supports modern browsers can run the project smoothly.

Scope of the Project

In-Scope Features

- Menu browsing
- Adding items to cart
- Generating bill summary
- Removing items from cart
- Table reservation form
- About and contact sections
- Responsive UI for all screens
- Fully frontend operations

Out-of-Scope Features

- No backend server
- No user authentication
- No real payment gateway
- No database storage
- No real-time kitchen system

Functional Requirements

User Requirements

- User should be able to view the menu.
- User should add or remove items from the cart.
- User should see automatic billing updates.
- User should fill out a table reservation form.

Non-Functional Requirements

A. Performance Requirements

- The system should load quickly.
- Operations like adding items must be instantaneous.

B. Usability Requirements

- Easy navigation
- Clear menu layout
- Large, readable text
- Responsive for mobile users

C. Reliability Requirements

- The system should function correctly without errors.
- JavaScript should handle invalid inputs gracefully.

D. Security Requirements

- Basic form validation
- No backend → No data security risks

E. Scalability Requirements

- Can be integrated with backend or database in future.
- Easy to add more menu items.

3.2. System Requirements:

System Requirements define the Hardware and Software specifications needed to develop, run and test the Restaurant Management System. Because this project is entirely frontend – based , the requirements are simple and easily accessible.

- Hardware Requirements
- Software Requirements

3.2.1. Hardware Requirements:

The selection of hardware is very important in the existence and the proper working of any software. In this selection of hardware, the size and capacity requirements are also important.

Minimum Hardware Requirements

Processor : Dual-core (1.6 GHz or above)

Purpose → To run Browser and code editor smoothly.

Ram: 2GB

Purpose→ Sufficient for lightweight frontend development.

Storage: 500MB free space

Purpose→ To store project files , images, and tools.

Display: 1280 x 720 resolution

Purpose→ To view the UI clearly.

Input Devices: Keyboard, Mouse.

Purpose→ For coding and system navigation.

3.2.2. Software Requirements:

The software Requirements Specifications produce at the culmination of the analysis tasks. One of the most difficult tasks is that the selection of the software. Once the system requirement is known by determinant whether a particular software fits the requirements.

1. Operating System

Examples:

- Windows 7, 8, 10 or 11
- Linux (Ubuntu, Fedora)
- macOS

Purpose:

The operating system provides the base environment required to run web browsers and code editors . Any OS that supports modern browsers can run the project smoothly.

2. Web Browser

Examples:

- Google Chrome (Recommended)
- Mozilla Firefox
- Microsoft Edge
- Safari (macOS)

Purpose:

The browser is used to run, test, and view the Restaurant Management System. Since the project is frontend only, the browser executes the HTML, CSS and JavaScript code directly, displaying the full functionality of the system.

3. Code Editor

Examples:

- Visual Studio Code (VS Code)
- Sublime Text
- Notepad++
- Atom

Purpose:

The code editor is needed to write, edit and organize the HTML, CSS and JavaScript files . Editors like VS Code provide features such as syntax highlighting , auto-completion and debugging tools that make development faster and easier.

4. Frontend Technologies

A. HTML5

Purpose:

HTML is used to create the structure and layout of the web pages, such as menus, forms, butons, and content sections.

B. CSS3

Purpose:

CSS is used to design the visual appearance , including colors, styles, fonts, layouts, and responsiveness. It ensures that the system looks clean, modern, and user-friendly.

C. JavaScript

Purpose:

JavaScript handles the functionality and interactivity, such as:

- Adding items to the cart

- Calculating the total bill
- Updating the order list dynamically
- Validating the reservation form
- Displaying confirmation messages

5. (Optional) Live Server Extension (VS Code)

Purpose:

The Live Server extension allows the project to run on a local host and automatically refreshes the page whenever changes are made to the code. This speeds up the testing and development process.

6. Image Editing Software (Optional)

Examples:

- Photoshop
- GIMP
- Canva

7. Version Control Software (Optional)

Example:

- Git
- GitHub Desktop

1.1. Feasibility Study

A feasibility study helps determine whether the project is practical and worthwhile.

3.3.1. Technical Feasibility

- Only basic frontend technologies are required (HTML, CSS, JavaScript).
- No backend or server is required.
- Can run on any device with a web browser.

Conclusion: Technically feasible and easy to develop.

3.3.2. Economic Feasibility

- No cost required for development tools.
- Open-source technologies used.
- Low cost of maintenance.

Conclusion: Economically feasible.

3.3.3. Operational Feasibility

- Simple, intuitive interface.
- Reduces staff workload.
- Enhance Customer satisfaction.

Conclusion: Highly feasible operationally.

3.4. Modelling Approaches:

3.4.1. UML diagrams :

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML comprises two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to or associated with UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software systems, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

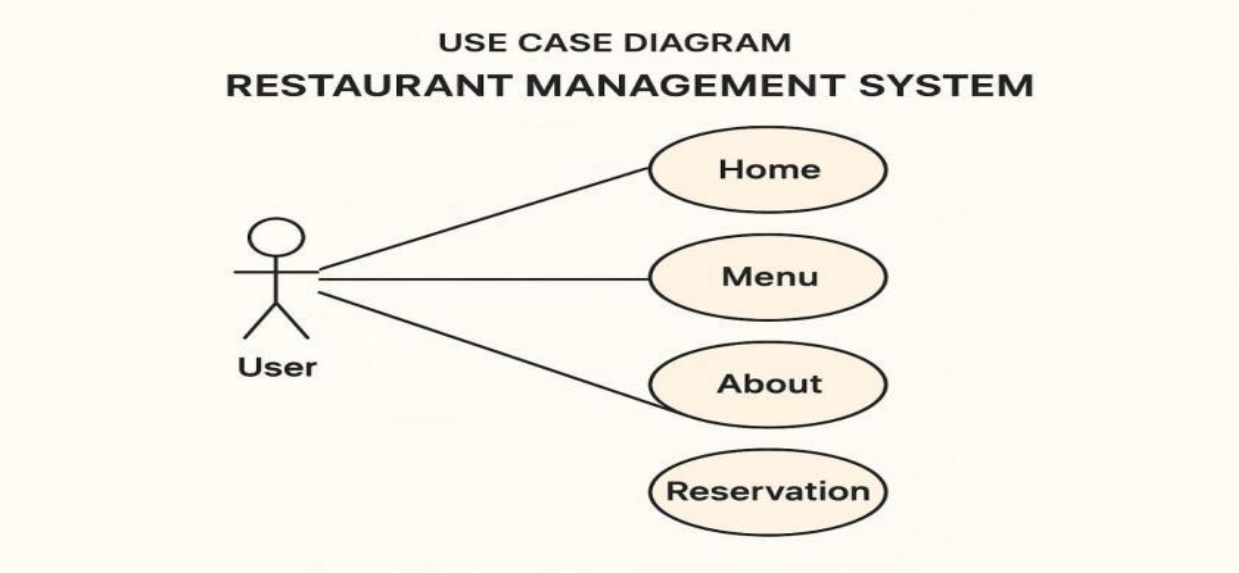
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of the OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

3.4.1.1. Use Case Diagram:

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical

overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



Use Case Diagram for Restaurant Management System

Actors

- **User / Customer** – the person visiting the website.

Use Cases

Home Page

- View homepage
- Browse featured items/offers
- Navigate to other pages (menu, about, reservation)

Menu Page

- View food categories
- View food items and details
- Filter or scroll items

About Page

- View restaurant information

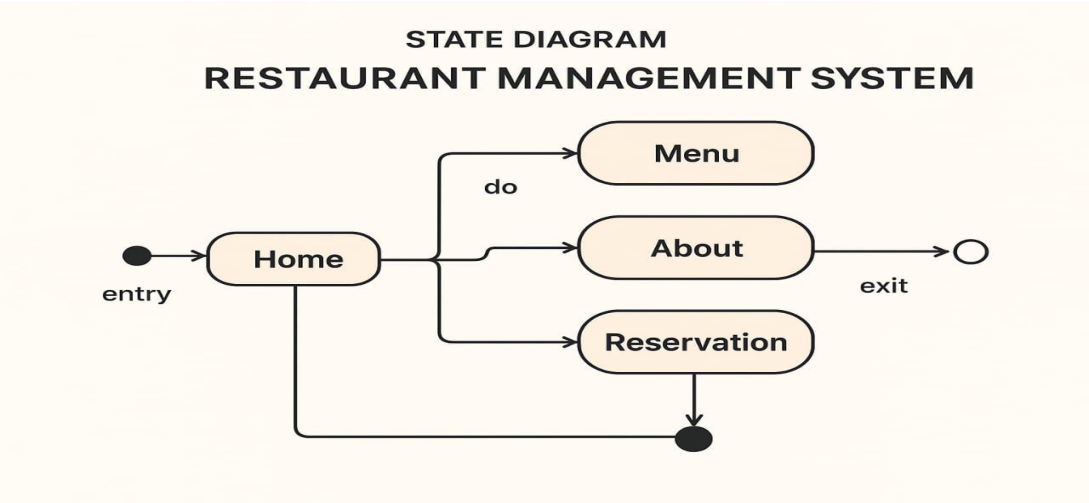
- View contact details, history, mission, etc.

Reservation Page

- Fill reservation form
- Select date, time, number of guests
- Submit reservation request
- Receive confirmation message on UI

3.4.1.2. State Chart Diagram:

A state chart diagram is a graphical representation of the states and transitions of an object or system in response to events, used primarily in software modelling. It shows how an entity progresses through different states, triggered by specific events, and includes elements like states, transitions, events, and actions. Commonly used in UML (Unified Modelling Language), it helps visualize dynamic behaviour, especially in systems with complex states, such as user interfaces or protocol implementations.



State Chart Diagram for Restaurant Management System

1. Start State

- The system loads (browser opens the site).

2. Home Page

User can:

- View introduction

- Click navigation links
- Go to Menu, About, or Reservation

3. Menu Page

User can:

- View list of dishes
- Scroll or filter items
- Navigate back to Home
- Move to About or Reservation

4. About Page

User can:

- Read restaurant details
- Navigate to Home
- Go to Menu or Reservation

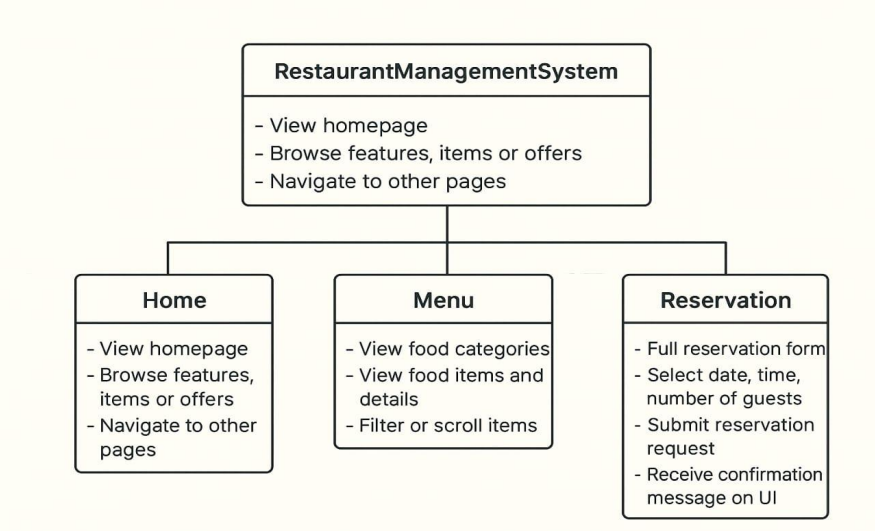
5. Reservation Page

User can:

- Fill reservation form
- Submit reservation
- Upon submission, return to Home or stay for confirmation.

3.4.1.3. Class Diagram:

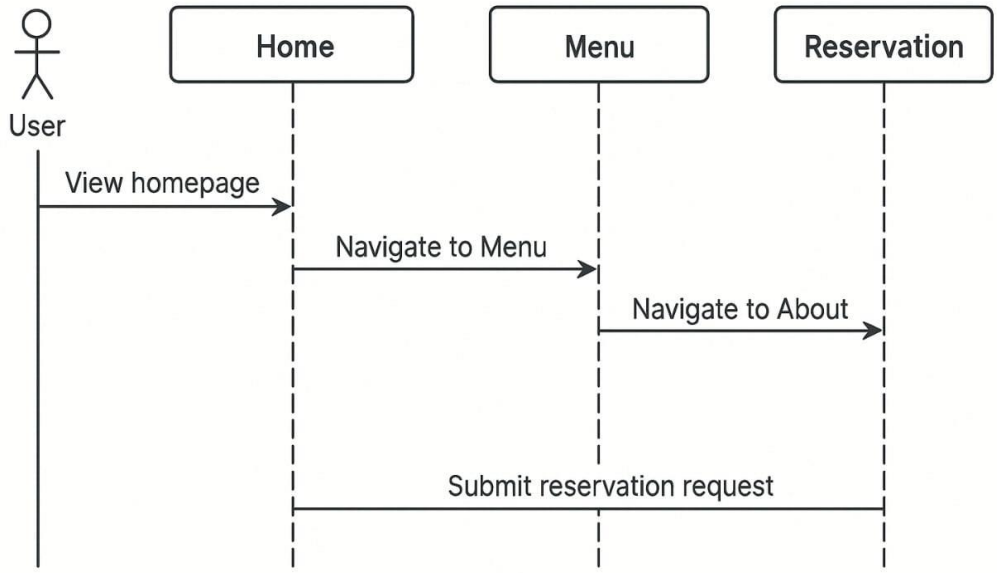
A class diagram in UML is a visual representation of a system's structure. It shows classes, their attributes, methods, and the relationships between objects. Key elements include classes, associations, inheritance, aggregation, and composition. It is used to model the static structure of a system, providing a blueprint for coding and a tool for communication and documentation.



Class Diagram for Restaurant Management System

3.4.1.4. Sequence Diagram :

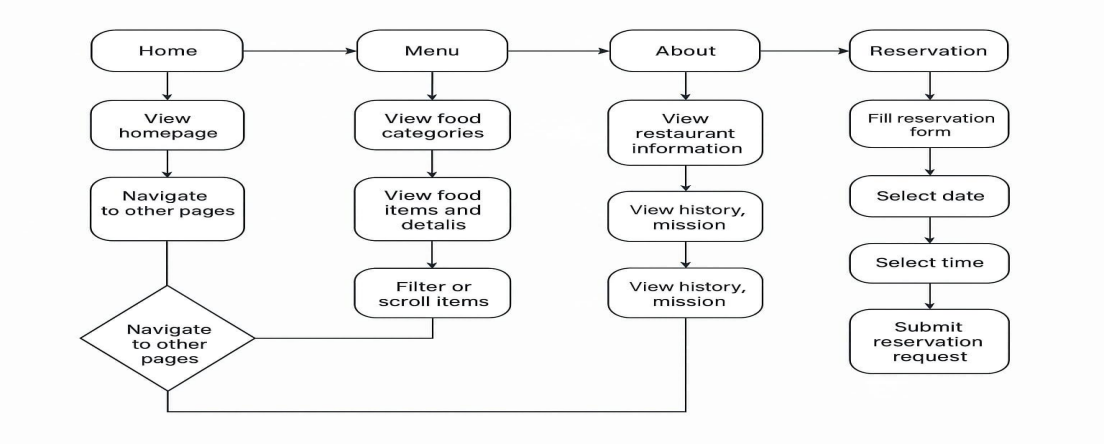
A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



Sequence Diagram for Restaurant Management System

3.4.1.5. Activity Diagram :

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



Activity Diagram for Restaurant Management System

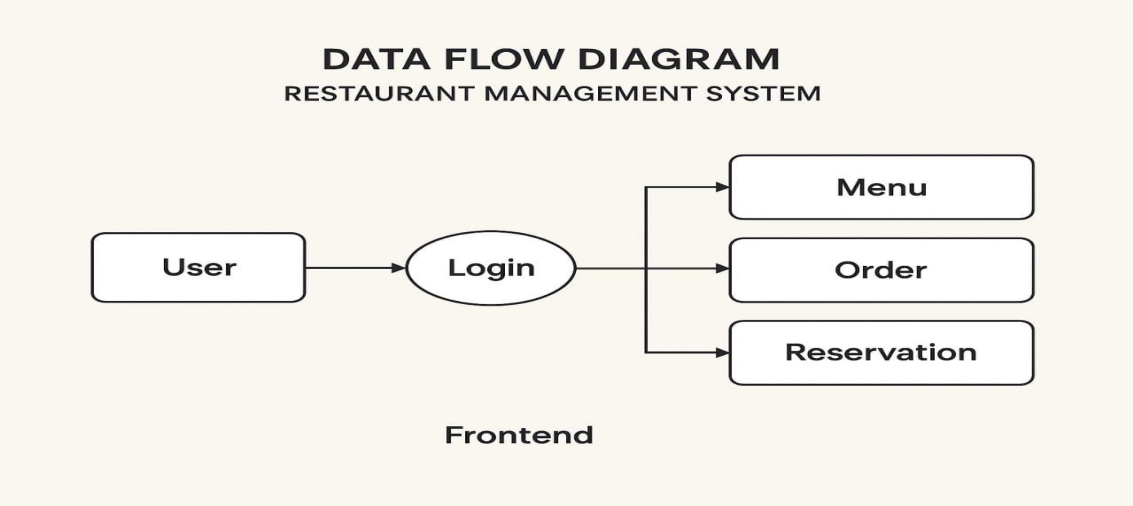
3.4.2. Data flow Diagram:

1. The DFD is also called a bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



Data Flow Diagram for Restaurant Management System

3.4.3. Class-Based Modelling:

A **Class-Based Modeling** for a **Restaurant Management System (Frontend Only)**, focusing on the **Home, Menu, About, and Reservation** pages. I'll provide **both the class diagram structure** and a **detailed explanation of each class and its purpose**.

n frontend-only systems, **class-based modeling** usually represents:

- UI components (pages, forms, widgets)
- Data models used to render content
- Navigation or state management components

Here, we model:

- **HomePage**
- **MenuPage**
- **About Page**
- **Reservation Page**
- **Supporting models**: Menu Item and Reservation
- **Navigation Manager** for handling page transitions

3.4.3.1. CRC Modelling:

CRC (Class-Responsibility-Collaborator) cards for the **Restaurant Management System Frontend with Home, Menu, About, and Reservation, CRC modelling focuses on responsibility of each class and collaborators they interact with.**

1. Homepage

Class: Homepage
Responsibilities:

- Display homepage content (title, intro, featured items)
- Navigate to other pages (Menu, About, Reservation)
- Highlight featured dishes or promotions

Collaborators:

- Navigation Manager (to move between pages)
- Menu Item (for featured items)

2. Menu Page

Class: Menu Page
Responsibilities:

- Display list of menu items
- Filter items by category
- Show details of each menu item

Collaborators:

- Menu Item (to display dishes)
- Navigation Manager (to navigate to other pages)

3. About Page

Class: About Page
Responsibilities:

- Display restaurant information (history, mission, contact info)
- Show opening hours
- Navigate to other pages

Collaborators:

- Navigation Manager (to move between pages)

4. Reservation Page

Class: Reservation Page
Responsibilities:

- Display reservation form
- Accept user input (name, email, phone, date, time, guests)
- Validate form inputs
- Submit reservation and show confirmation

Collaborators:

- Reservation (to store booking data)
- Navigation Manager (to navigate to other pages)

5. Menu Item

Class: Menu Item
Responsibilities:

- Store details of a dish (id, name, description, price, category)
- Provide details for rendering on Menu Page and Homepage

Collaborators:

- Menu Page (displays list of menu items)
- Homepage (for featured items)

6. Reservation

Class: Reservation
Responsibilities:

- Store reservation data (name, email, phone, date, time, guests)
- Confirm reservation and provide feedback to the user

Collaborators:

- Reservation Page (collects and submits reservation data)

7. Navigation Manager

Class: Navigation Manager
Responsibilities:

- Handle frontend page transitions
- Maintain navigation state if SPA-like behaviour

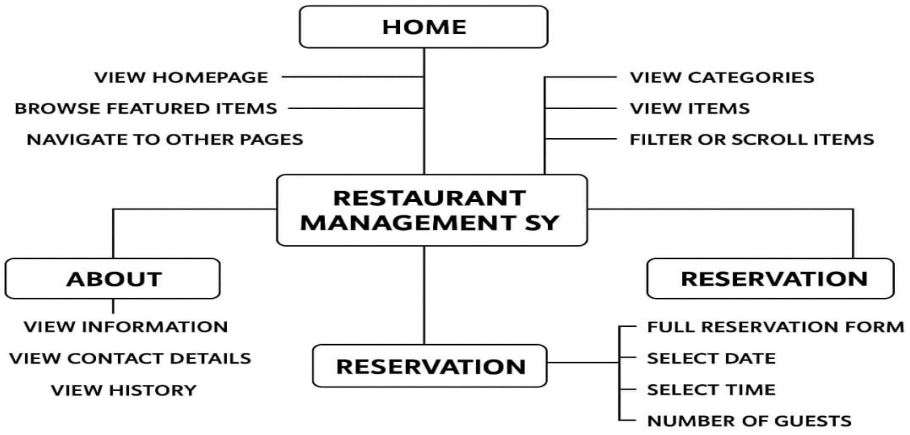
Collaborators:

- Homepage, Menu Page, About Page, Reservation Page (all pages use it to navigate)

3.4.4. Data Models:

3.4.4.1. ER-Diagrams:

For a **Restaurant Management System (Frontend Only)** with **Home, Menu, About, and Reservation**, the **ER (Entity-Relationship) Model** focuses on the **data structures** that the frontend will interact with. Even though frontend mainly displays information, it still deals with entities like **Menu Item** and **Reservation**, which can be modelled.



ER diagrams are used in database design to clarify requirements and structure.

1. Entities and Attributes

1.1. Menu Item

- Represents a dish available in the restaurant.
- **Attributes:**
  - menuItemID (Primary Key)
  - name
  - description
  - price
  - category (e.g., Appetizer, Main Course, Dessert)
  - imageURL (optional, for frontend display)

1.2. Reservation

- Represents a booking made by a user.
- **Attributes:**
  - reservationID (Primary Key)
  - name (user's name)
  - email

- phone
- date
- time
- guests (number of people)
- status (confirmed, pending)

1.3. Pages

Even though Home and About pages do not store data, we can represent the as **non-persistent entities** to clarify the frontend structure.

- **HomePage**
  - title
  - introText
  - featuredItems (links to MenuItem)
- **AboutPage**
  - aboutText
  - contactInfo
  - openingHours
- **MenuPage**
  - displayedItems (links to MenuItem)
- **ReservationPage**
  - reservationForm (collects data for Reservation)

2. Relationships

1. **HomePage → MenuItem**
   - **Relationship:** "features"
   - A homepage may feature **0..n MenuItems**.
2. **MenuPage → MenuItem**
   - **Relationship:** "displays"
   - Menu page displays **1..n MenuItems**.
3. **ReservationPage → Reservation**
   - **Relationship:** "creates"
   - Each reservation page allows the user to create **1 Reservation**.

4. SYSTEM DESIGN

System Design is the process of examining and evaluating a system to understand its components, functionality, and performance. It involves identifying problems, gathering requirements, and determining solutions to improve efficiency, effectiveness, and overall system performance. The goal is to ensure the system meets the needs of its users and operates effectively within its environment.

4.1.Design Principle:

The design principles for a Library Management System (LMS) focus on creating an intuitive, efficient, and scalable system that meets the needs of both library staff and users. Key principles include:

1. **Separation of Concern(SoC):** Each part of the system should have a single responsibility. Frontend handles presentation(Home, Menu, Reservation, Pages).
   Backend handles data storage , business logic and reservation management.

2. **Scalability:** The system should support growth in users, data, and functionality.
   Easily add new pages(Offers, Gallery).
   Backend can handle more menu items and reservations.

3. **Modularity:** Divide the system into self-contained modules.
   Modules like menu management , reservation handling, user authentication(if needed),
   Each frontend page is a separate module/component.

4. **User-Centric Design :** Design the system around user needs and usability.
   Simple navigation, clear calls-to-action(like " Reserve Table" or " View Menu").
   Responsive design for desktop, tablet, and mobile.

5. **Security:** Protect user and business data. Validate reservation forms to prevent malicious input. Secure sensitive information like user contact details.

6. **Feedback & Error Handling :** System should inform users of actions and errors.
   Confirmation messages for successful reservations. Error messages when required fields are missing or invalid.

7. **Simplicity:** Include only essential features in the UI and system. Home page shows only intro, navigation, and featured items.
   Reservation form only asks for necessary details.

5. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

5.1. Types of testing:

5.1.1. Unit Testing :

Unit testing is a software testing technique where individual components of a program (usually functions or methods) are tested in isolation to ensure they perform as expected. The main goal is to identify bugs early, improve code quality, and ensure that each unit works independently before integration. Unit tests are automated, typically written in a framework such as JUnit or pytest, and focus on validating the correctness of small, isolated pieces of functionality. In our project, we have implemented a series of unit tests to verify that each module and function performs correctly in various scenarios. By running these tests regularly, we can ensure the robustness of the individual components and quickly address any issues, thus maintaining high code quality and reliability throughout the development process.

5.1.2. Integration Testing :

Integration testing is a type of software testing where individual software modules are combined and tested as a group to ensure they work together as expected. The goal is to identify issues that may arise from the interaction between different components, such as data flow or communication errors. Integration tests typically focus on verifying the interfaces and the interactions between various parts of the system, ensuring that integrated components function correctly when combined. In our project, we have conducted thorough integration testing to check how different modules interact and exchange data. By simulating real-world workflows and end- to-end use cases, we ensure that all integrated components work seamlessly together, identifying and resolving any issues that could affect system performance or user experience.

5.1.3. System Testing:

System testing is a type of testing where the entire software application is tested as a whole to ensure it meets the specified requirements and performs as intended in a real-world environment. This testing evaluates the complete system's behavior, including its functionality, performance, security, and compatibility, under various conditions. The goal is to validate the system's end-to- end performance and ensure all components work together harmoniously in a fully integrated environment. In our project, we have conducted extensive system testing by simulating realistic user scenarios, evaluating overall functionality, performance, and security across the entire system. Through this comprehensive testing, we confirm that the system meets the business requirements, delivers a smooth user experience, and functions optimally under different conditions and loads.

5.1.4. White Box Testing:

White box testing is a software testing method where the internal structure, design, and implementation of the software are thoroughly tested. This approach requires knowledge of the code itself, as the focus is on verifying the internal logic, control flow, data flow, and overall functioning of the system. White box testing involves techniques like code coverage analysis, path testing, and examining function calls to ensure there are no hidden bugs or vulnerabilities. In our project, we have implemented white box testing by reviewing and testing the internal code for all major functions and modules. By performing tests on individual code paths, ensuring full test coverage, and inspecting the interactions within the codebase, we guarantee that the internal workings of our system are robust, efficient, and error-free.

5.1.5 .Black box Testing:

Black box testing is a software testing technique where the tester focuses on verifying the functionality of the software without any knowledge of its internal workings or code. The primary goal is to validate whether the system meets the specified requirements and behaves as expected from the user's perspective, focusing on input-output behaviour, user interfaces, and overall system performance. In our project, we have performed extensive black box testing by simulating real user interactions and verifying that the system responds correctly to different inputs, handles edge cases, and delivers the expected outputs. This approach ensures that the software meets user expectations, is intuitive, and functions properly across various scenarios, without the need for understanding the underlying code.

5.2. Test Cases:

Below are **frontend-only test cases** specifically for a **Restaurant Website** that includes the pages:

- ✓ **Home**
- ✓ **Menu**
- ✓ **About**
- ✓ **Reservation**

These test cases focus on UI, navigation, validation, responsiveness, and user interaction (no backend logic).

1. HOME PAGE – Test Cases

1. Verify that the home page loads correctly without UI distortion.
2. Verify that the hero image, title, and tagline are visible on page load.
3. Verify that the navigation bar displays Home, Menu, About, Reservation links.
4. Verify that clicking the "Reserve Table" button redirects to the Reservation page.
5. Verify that all home page images load correctly (no broken images).
6. Verify that scrolling is smooth and no content overlaps.
7. Verify that the home page layout adapts properly on mobile, tablet, and desktop.
8. Verify that the navigation bar becomes a hamburger menu on small screens.
9. Verify that the footer is displayed at the bottom of the page.

2. MENU PAGE – Test Cases

1. Verify that the Menu page loads successfully when clicked in the nav bar.
2. Verify that all food items are displayed with name, description, price, and image.
3. Verify that images are clear and not stretched or broken.
4. Verify that if category filters exist (e.g., Starters, Drinks), selecting them updates the displayed items.
5. Verify that menu items do not overlap when scrolling.
6. Verify that hover effects or animations (if any) work correctly on desktop.
7. Verify that menu items stack properly in mobile view.
8. Verify that long descriptions do not overflow outside their container.

3. ABOUT PAGE – Test Cases

1. Verify that the About page loads correctly when clicked in the navigation bar.
2. Verify that the restaurant story/mission text is displayed without formatting issues.
3. Verify that staff or chef images load clearly.
4. Verify that contact details (address, phone, email) are shown.
5. Verify that social media icons appear and animate/highlight on hover.
6. Verify that all content is readable on mobile (no cropped text).

7. Verify that the layout adjusts correctly when resizing the browser window.

4. RESERVATION PAGE – Test Cases

1. Verify that the Reservation page loads successfully.
2. Verify that the reservation form contains Name, Phone, Email, Date, Time, and Number of Guests fields.
3. Verify that the form shows validation errors when submitted with empty fields.
4. Verify that the Name field accepts only valid characters.
5. Verify that the Phone number field validates incorrect phone numbers.
6. Verify that the Email field shows an error for invalid email format.
7. Verify that the date picker opens when clicking the Date field.
8. Verify that the time selector opens for the Time field.
9. Verify that the Number of Guests dropdown selects the correct number.
10. Verify that the Submit button is enabled only when all required fields are filled correctly.
11. Verify that a success or confirmation message appears after submitting (UI only).
12. Verify that the form layout is responsive on mobile devices.

5. NAVIGATION & GENERAL UI – Test Cases

Navigation

1. Verify that each navigation link (Home, Menu, About, Reservation) redirects to the correct page.
2. Verify that the active page is highlighted in the navbar.
3. Verify that the mobile hamburger menu opens and closes correctly.

General UI

1. Verify consistent fonts and styling across all pages.
2. Verify that no text overlaps or spills out of its container.
3. Verify that all buttons are clickable and have hover effects (if designed).
4. Verify that there are no broken links anywhere on the site.
5. Verify that loading and animations appear smooth.

6. IMPLEMENTATION

The **Restaurant Management System (RMS)** is implemented as a **frontend-only web application** using **HTML, CSS, and JavaScript**. This system allows users to browse restaurant information, view menus, learn about the restaurant, and submit table reservations—without requiring any backend server or database. All functionality runs directly in the user's browser.

.

1. Pages and Features

1.1 Home Page

- Serves as the landing page for the website.
- Includes a welcome banner, restaurant highlights, and navigation links to other pages.
- Designed using **HTML** for structure and **CSS** for layout, colors, and fonts.
- May include small JavaScript enhancements like slideshows or animations.

Purpose: Provides an inviting first impression and directs users to important sections of the website.

1.2 Menu Page

- Displays all food and beverage items offered by the restaurant.
- Menu items may be categorized (Starters, Main Course, Desserts, Drinks) with names, descriptions, and prices.
- Built using **HTML** to define the items and **CSS** to create a clean, attractive layout.
- **JavaScript** can be used to filter menu categories or highlight special items dynamically.

Purpose: Helps customers browse and choose dishes easily, improving their online experience.

1.3 About Page

- Provides information about the restaurant's history, mission, vision, chefs, and services.
- Uses **HTML** for text, images, and layout structure.
- **CSS** ensures appealing presentation with consistent formatting, spacing, and colors.

Purpose: Builds trust and informs users about the restaurant's values and offerings.

1.4 Reservation Page

- Contains a reservation form where customers can submit their booking details, including:
    - Name
    - Contact number

    - Date and time of reservation
    - Number of guests
    - Special requests (optional)
- **JavaScript** validates user input, ensuring all required fields are filled and correctly formatted.
- Confirmation messages are displayed after submission.
- **LocalStorage (optional)** can temporarily store reservation data in the browser to persist information across page reloads.

Purpose: Enables customers to book tables conveniently, improving customer experience.

2. Implementation Details

2.1 HTML Structure

- Each page is structured with semantic HTML elements:
    - <header> for navigation
    - <section> for content areas (Home, Menu, About, Reservation)
    - <footer> for contact information
- Forms and tables are used for reservations and menu listings.

2.2 CSS Styling

- CSS is used to:
    - Style navigation bars, buttons, and links
    - Arrange menu items in grid or card layouts
    - Add hover effects and animations for better UX
    - Make the design responsive for different screen sizes

2.3 JavaScript Functionality

- Validates reservation form inputs.
- Dynamically updates UI elements, such as highlighting menu items or displaying confirmation messages.
- Optionally stores reservation details in **localStorage**, allowing temporary persistence without a backend.
- Handles interactive elements like dropdown menus or slideshows on the Home or Menu pages.

7. CONCLUSION

The frontend-only Restaurant Management System successfully demonstrates how a complete restaurant website can be built using **HTML, CSS, and JavaScript** without relying on a backend server or database. The system includes essential pages such as **Home, Menu, About, and Reservation**, providing users with a clear interface to explore the restaurant, browse dishes, learn about the establishment, and submit table reservations.

By using **JavaScript** for interactivity and optional **localStorage** for temporary data handling, the system offers a functional and responsive experience entirely within the browser. While it cannot support multi-user access, permanent storage, or online ordering, this project effectively showcases the capabilities of frontend web development and serves as a strong foundation for learning and extending into more advanced full-stack applications in the future.

8. BIBILOGRAHY

Books:

- "A Journey with Web Technology" – Dilkeshwar Pandey, Upendra Kumar Tiwari, Rajesh Kumar Pathak, First Edition, ACME LEARNING PVT LTD, 2010□
- "Software Testing Concepts and Practices" – K. Mustafa, R.A. Khan, Narosa Publishing House, 2007□
- "The Unified Modeling Language User Guide" – Grady Booch, James Rumbaugh, Ivar Jacobson, Sai Printopack PVT LTD, Ninth Impression, 2009□
- "Software Engineering – A Practitioner"s Approach" – Roger S. Pressman, Seventh Edition, Tata McGraw-Hill International Edition, 2010□

Reference Links:

- W3Schools (https://www.w3schools.com/whatis/whatis_html.asp)□

  A comprehensive and beginner-friendly platform for learning web technologies like HTML, CSS, and JavaScript.

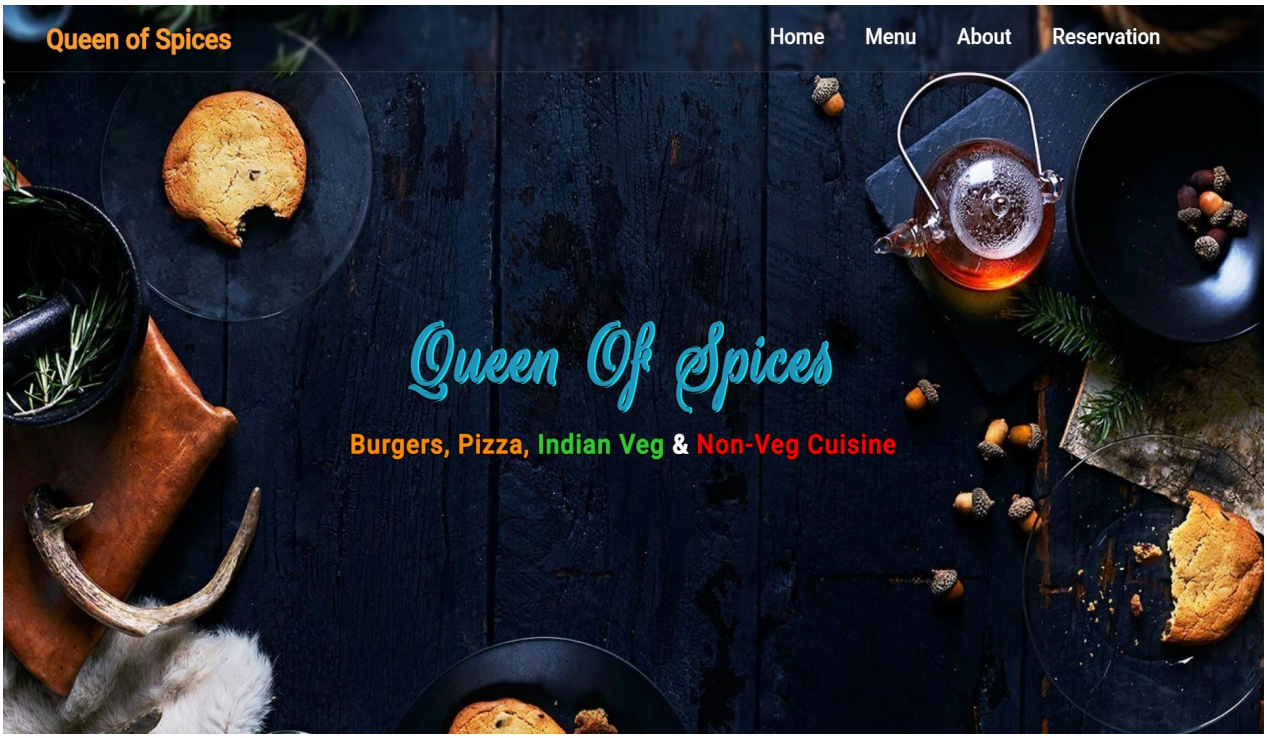
- GeeksforGeeks (https://www.geeksforgeeks.org/100-days-of-web-development/)□

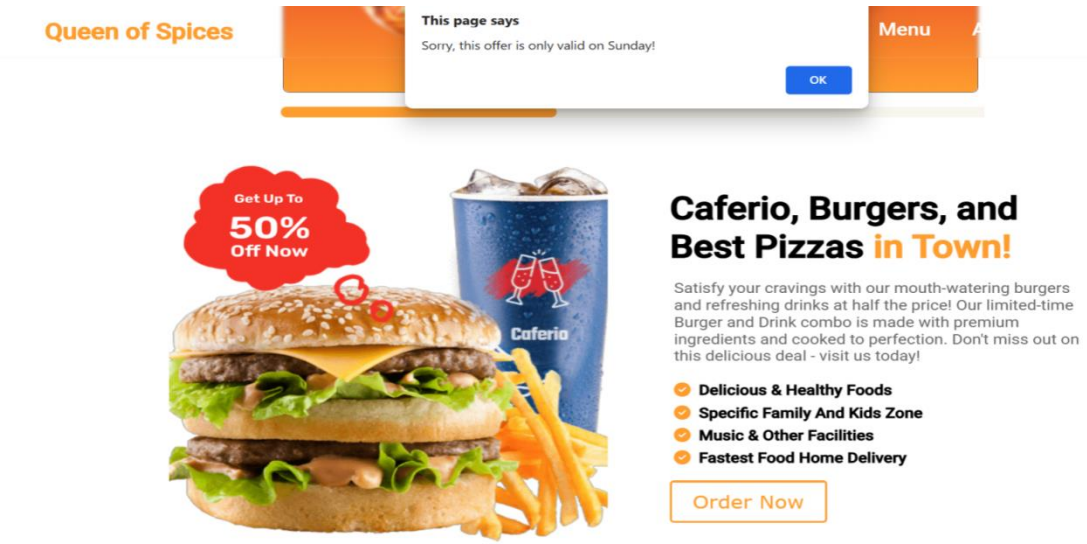
  Covers tutorials and articles on web development, JavaScript, and other programming languages.

9. APPENDIX

APPENDIX A: Screens

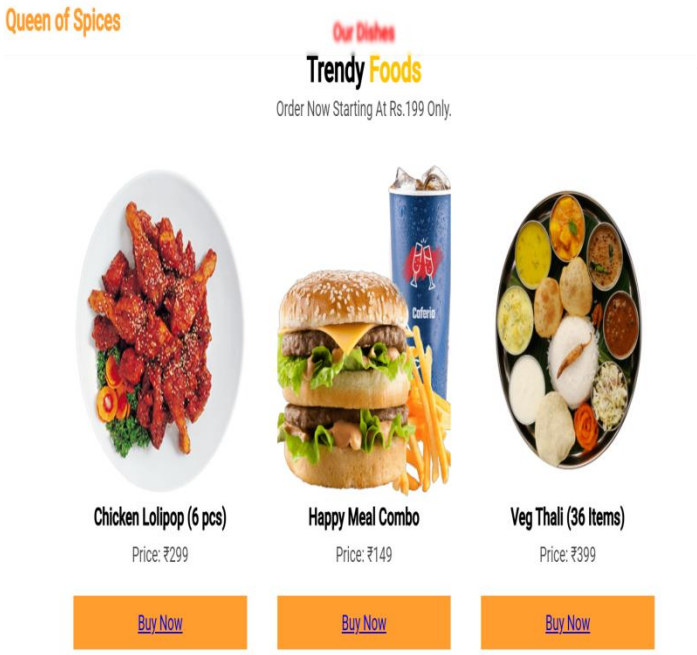
HOME PAGE:

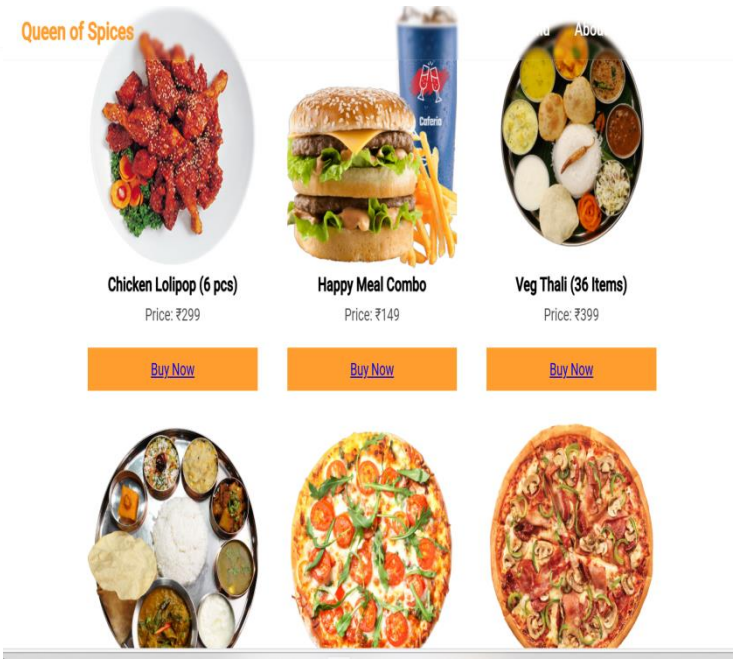


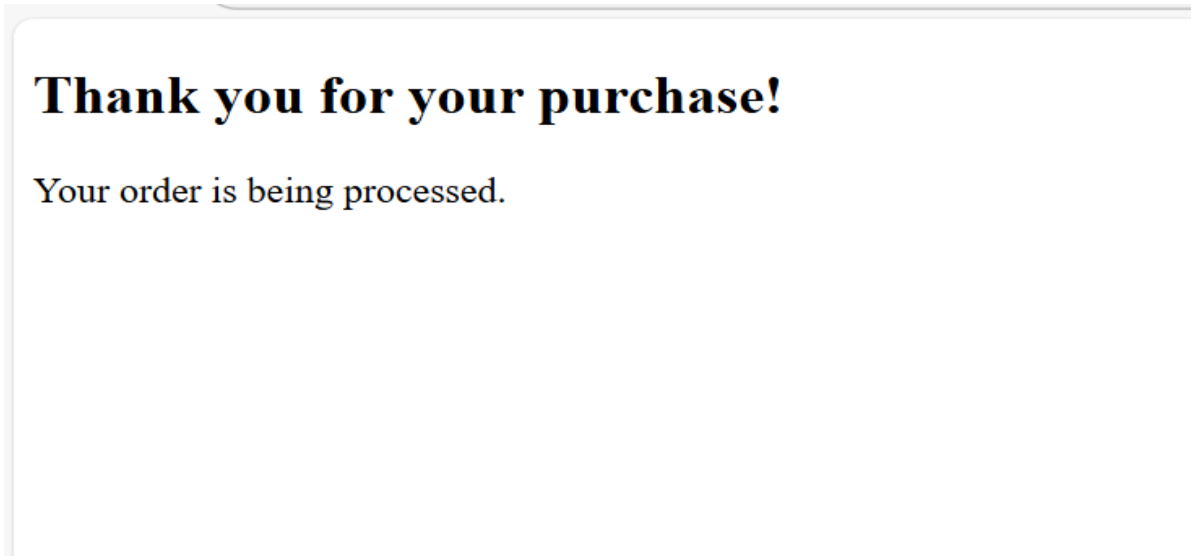


SPECIAL OFFER:

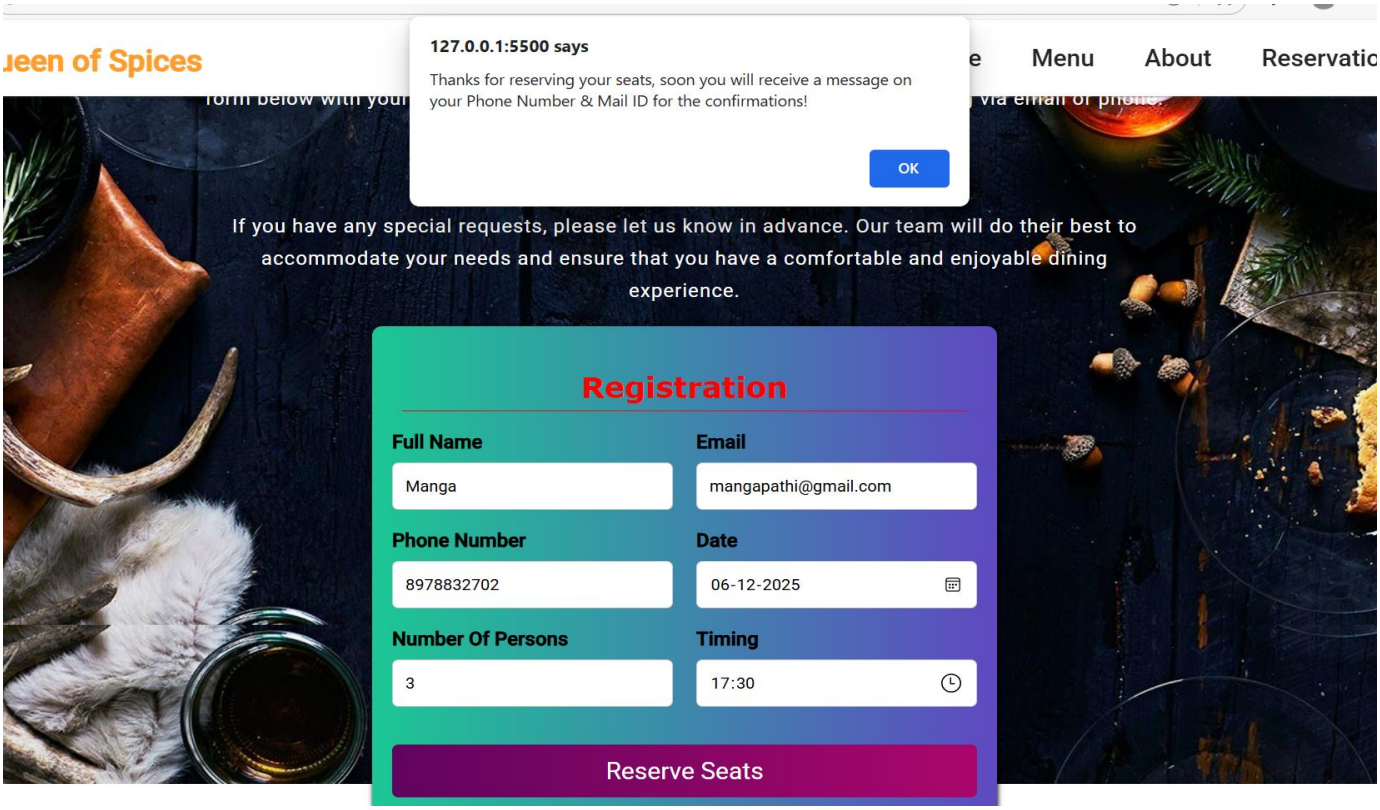
MENU:







RESERVATION:



ABOUT:

TESTIMONIALS:

Our Customer Reviews

Jeff Bezos
CEO @Amazon

★★★★★

"This place is great! Atmosphere is chill and cool but the staff is also friendly. They know what they're doing and what they're talking about, and you can tell making the customers happy is their main priority."

Steve Ballmer
CEO @Microsoft

★★★★★

"It's a great experience. The ambiance is very welcoming and charming. Amazing wines, food and service. Staff are extremely knowledgeable and make great recommendations"

Sundar Pichai
CEO @Google

★★★★★

"The Food we had enjoyed at the time of dinner. It was really delicious taste with great quality, everything had unique taste which we had ordered, nice arrangement and services from the staff while eating."

Company

About Us
Our Services
Privacy Policy
Trademark

Get Help

FAQ
Delivery
Returns
Order Status
Payment Options

Follow Us

APPENDIX B: Source Code

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Queen of Spices | Burgers, Pizza, Indian Veg & Non-veg Cuisine</title>

    <!-- favicon -->
    <link rel="shortcut icon" href="./favicon.svg" type="image/svg+xml">

    <!-- external css -->
    <link rel="stylesheet" href="./css/style.css">

    <!-- javascript file -->
    <script src=".js/script.js" defer></script>

    <!-- Font Awesome link -->
    <script src="https://kit.fontawesome.com/5289663089.js" crossorigin="anonymous"></script>
</head>

<body>

    <!-- Header Section -->
    <section class="header-img">
        <header class="header-main">
            <nav class="nav-main">
                <h1 class="main-title"><a href="#">Queen of Spices</a></h1>
                <div class="openMenu"><i class="fa-solid fa-bars" style="color: #ff9d2e;"></i></div>
                <ul class="nav-ul">
                    <li><a href="#">Home</a></li>
                    <li><a href="#menu">Menu</a></li>
                    <li><a href="about.html">About</a></li>
                    <li><a href="reservation.html">Reservation</a></li>
                    <li><div class="closeMenu"><i class="fa-solid fa-xmark"></i></div></li>
                </ul>
            </nav>
        </header>
```

```
      <div class="header-context">
         <h1>Queen of Spices</h1>
         <h3 style="text-shadow: -1px -1px 0 black, 1px -1px 0 black, -1px 1px 0 black, 1px 1px 0
black;"><span style="color: #FF8C00;">Burgers, Pizza, </span><span style="color: #33c933;">Indian
Veg</span><span style="color: white;"> & </span><span style="color: #FF0000;">Non-veg
Cuisine</span></h3>
      </div>
   </section>

   <!-- Promo Cards -->
   <section id="promo-card">
      <div class="promo-container">
         <div class="promo-card">
            <svg xmlns="http://www.w3.org/2000/svg" width="60" height="60" fill="none"
               xmlns:v="https://vecta.io/nano">
               <g clip-path="url(#A)" fill="#ff9d2d">
                  <path
                     d="M43.033.002L42.563 0c-7.896 0-15.555 1.546-22.767 4.597-6.965 2.946-13.22
7.163-18.592 12.535l-.043.044c-1.548 1. 7 2.7 0 0 1 2.698-2.698c.048 0 .087.039.087.087v.964c0
.343.143.676.392.912a1.26 1.26 0 0 0 .931.343l1.43-.076z" />
               </g>
               <defs>
                  <clipPath>
                     <path fill="#fff" d="M0 0h60v60H0z" />
                  </clipPath>
               </defs>
            </svg>
            <h3 class="h3-card-title">Maxican Pizza</h3>
            <p class="card-text">Satisfy your craving for both pizza and Mexican food with our delicious
creation.
            </p>
            <img src="./images/Veg_pizza.png" width="200" height="200" alt="Maxican Pizza"
class="promo-img">
         </div>

         <div class="promo-card">
            <svg xmlns="http://www.w3.org/2000/svg" width="60" height="60" fill="none"
               xmlns:v="https://vecta.io/nano">
               <g clip-path="url(#A)">
                  <path
                     d="M56..042-1.27l.246-.25c.331.117.62.268.938.437.694.367 1.479.783
2.898.991.443.065.881.097 1.306.097.569 0 1.117-.058 1.63-.169v1.937h-8.431a.88.88 0 1 0 0
1.761h9.954c.819 0 1.484.666 1.484 1.484a11.43 11.43 0 0 1-11.412 11.412h0z"
```

```
                fill="#ff9d2d" />
            </g>
            <defs>
              <clipPath>
                <path fill="#fff" d="M0 0h60v60H0z" />
              </clipPath>
            </defs>
          </svg>
          <h3 class="h3-card-title">Burger Kingo</h3>
          <p class="card-text">Get ready to tackle our big burgers, stacked high with delicious
toppings.</p>
          <img src="./images/cta-banner.png" width="200" height="200" alt="Burger Kingo"
class="promo-img">
        </div>

        <div class="promo-card">
          <svg xmlns="http://www.w3.org/2000/svg" width="60" height="60" fill="none"
            xmlns:v="https://vecta.io/nano">
            <g clip-path="url(#A)" fill="#ff9d2d">
              <path
                d="M30.057 32.375c-5.536 0-10.04 4.504-10.04 10.04s4.504 10.04 10.04 10.04 10.04-
4.504 10.04-10.04-4.908 1.336 4.152 2.042 6.488 2.042l.364-.007.173-.007.152-.009a11.41 11.41 0 0 0
1.582-.206 11.25 11.25 0 0 0 4.218-1.813 11.31 11.31 0 0 0 4.091-5.229.24.24 0 0 1 .222-
.158h10.23a.23.23 0 0 1 .232.277h0z" />
            </g>
            <defs>
              <clipPath>
                <path fill="#fff" d="M0 0h60v60H0z" />
              </clipPath>
            </defs>
          </svg>
          <h3 class="h3-card-title">French Fry</h3>
          <p class="card-text">Our crispy golden French fries are the perfect side to any meal.</p>
          <img src="./images/French_Fries.png" width="200" height="200" alt="French Fry"
class="promo-img">
        </div>

        <div class="promo-card">
          <svg xmlns="http://www.w3.org/2000/svg" width="60" height="60" fill="none"
            xmlns:v="https://vecta.io/nano">
            <g clip-path="url(#A)" fill="#ff9d2d">
              <path
                d="M14.837 40.062c0 3.106 2.528 5.634 5.634 5.634s5.634-2.528 5.634-5.634a5.64
5.64 0 0 0-5.634-5.634c-3.106 H37.39v-4.299c0-4.187-.591-8.624-1.915-14.388l-.141-.613a62.6 62.6 0
```

```
0 1-1.503-10.774h1.08a.88.88 0 0 0 .879-.879.88.88 0 0 0-.879-.879h-1.146l-.019-1.463v-
1.387h12.942v1.387z" />
                </g>
                <defs>
                    <clipPath>
                        <path fill="#fff" d="M0 0h60v60H0z" />
                    </clipPath>
                </defs>
            </svg>
            <h3 class="h3-card-title">Soft Drinks</h3>
            <p class="card-text">Quench your thirst with our refreshing selection of soft drinks.</p>
            <img src="./images/Soft_DRinks.png" width="200" height="200" alt="Soft Drinks"
class="promo-img">
        </div>

        <div class="promo-card">
            <svg xmlns="http://www.w3.org/2000/svg" width="60" height="60" fill="none"
                xmlns:v="https://vecta.io/nano">
                <g clip-path="url(#A)" fill="#ff9d2d">
                    <path
                        d="M43.033.002L42.563 0c-7.896 0-15.555 1.546-22.767 4.597-6.965 2.946-13.22
7.163-18.592  1 2.698-2.698c.048 0 .087.039.087.087v.964c0 .343.143.676.392.912a1.26 1.26 0 0 0
.931.343l1.43-.076z" />
                </g>
                <defs>
                    <clipPath>
                        <path fill="#fff" d="M0 0h60v60H0z" />
                    </clipPath>
                </defs>
            </svg>
            <h3 class="h3-card-title">Extra Large Pizza</h3>
            <p class="card-text">Feed the whole family with our delicious and generously sized large
pizzas.</p>
            <img src="./images/chicken_pizza.png" width="200" height="200" alt="Maxican Pizza"
                class="promo-img">
        </div>
    </div>
</section>

<!-- About -->
<section>
    <div class="about-container">
        <div class="about-banner">
            <img src="./images/about-banner.png" alt="Burger with Drinks" class="about-img1">
```

```
            <img src="./images/sale-shape-red.png" alt="get up to 50% off now" class="about-img2">
        </div>
        <div class="about-content">
            <h2 class="h2-about-title">
                Caferio, Burgers, and Best Pizzas
                <span class="span">in Town!</span>
            </h2>

            <p class="section-text">
                Satisfy your cravings with our mouth-watering burgers and refreshing drinks at half the price! Our limited-time
                Burger and Drink combo is made with premium ingredients and cooked to perfection. Don't miss out on this delicious deal - visit us today!
            </p>
            <ul class="about-ul">
                <li>
                    <i class="fa-solid fa-circle-check" style="color: #ff9d2e;"></i>
                    <span class="about-text">Delicious & Healthy Foods</span>
                </li>
                <li>
                    <i class="fa-solid fa-circle-check ticks" style="color: #ff9d2e;"></i>
                    <span class="about-text">Specific Family And Kids Zone</span>
                </li>
                <li>
                    <i class="fa-solid fa-circle-check ticks" style="color: #ff9d2e;"></i>
                    <span class="about-text">Music & Other Facilities</span>
                </li>
                <li>
                    <i class="fa-solid fa-circle-check ticks" style="color: #ff9d2e;"></i>
                    <span class="about-text">Fastest Food Home Delivery</span>
                </li>
            </ul>

            <button type="button" class="btn-order-now" onclick="alert('Sorry, this offer is only valid on Sunday!')">Order Now</button>
        </div>
    </div>
</section>

<!-- Menu Items -->
<section id="menu" style="padding-top: 48px;">
    <div class="menu-header">
        <h4 class="subhead">Our Dishes</h4>
        <h1 class="Heading">Trendy <span style="color:  rgb(255, 196, 0);">Foods</span></h1>
```

```
        <p class="Discription">Order Now Starting At Rs.199 Only.</p>
    </div>

    <!-- All Dishes -->
    <div class="allitems">
        <div class="Items">
            <img class="dishpic" src="./images/food-menu-3.png" alt="">
            <h3 class="Dish">Chicken Lolipop (6 pcs)</h3>
            <p class="price-tag">Price: ₹299</p>
            <br>
            <button type="button" class="btn" onclick="show_alert()"><a href="cart.html">Buy
Now</a></button>
        </div>

        <div class="Items">
            <img class="dishpic" src="./images/about-banner.png" alt="">
            <h3 class="Dish">Happy Meal Combo</h3>
            <p class="price-tag">Price: ₹149</p>
            <br>
            <button type="button" class="btn" onclick="show_alert()"><a href="cart.html">Buy
Now</a></button>
        </div>

        <div class="Items">
            <img class="dishpic" src="./images/R.png" alt="">
            <h3 class="Dish">Veg Thali (36 Items)</h3>
            <p class="price-tag">Price: ₹399</p>
            <br>
            <button type="button" class="btn" onclick="show_alert()"><a href="cart.html">Buy
Now</a></button>
        </div>


        <div class="Items">
            <img class="dishpic" src="./images/chickenthali.png" alt="">
            <h3 class="Dish">Non-Veg Thali</h3>
            <p class="price-tag">Price: ₹499</p>
            <br>
            <button type="button" class="btn" onclick="show_alert()"><a href="cart.html">Buy
Now</a></button>
        </div>

        <div class="Items">
            <img class="dishpic" src="./images/Veg_pizza.png" alt="">
```

```
            <h3 class="Dish">Veg Pizza</h3>
            <p class="price-tag">Price: ₹199</p>
            <br>
            <button type="button" class="btn" onclick="show_alert()"><a href="cart.html">Buy
Now</a></button>
        </div>

        <div class="Items">
            <img class="dishpic" src="./images/chicken_pizza.png" alt="">
            <h3 class="Dish">Roasted Chicken Pizza</h3>
            <p class="price-tag">Price: ₹249</p>
            <br>
            <button type="button" class="btn" onclick="show_alert()"><a href="cart.html">Buy
Now</a></button>
        </div>

        <div class="Items">
            <img class="dishpic" src="./images/OIP (1).jpg" alt="">
            <h3 class="Dish">Paneer 65</h3>
            <p class="price-tag">Price: ₹229</p>
            <br>
            <button type="button" class="btn" onclick="show_alert()"><a href="cart.html">Buy
Now</a></button>
        </div>

        <div class="Items">
            <img class="dishpic" src="./images/chinese_noodles.png" alt="">
            <h3 class="Dish">Chinese Noodles</h3>
            <p class="price-tag">Price: ₹249</p>
            <br>
            <button type="button" class="btn" onclick="show_alert()"><a href="cart.html">Buy
Now</a></button>
        </div>

        <div class="Items">
            <img class="dishpic" src="./images/Chicken-Biryani.jpg" alt="">
            <h3 class="Dish">Chicken Biryani</h3>
            <p class="price-tag">Price: ₹299</p>
            <br>
            <button type="button" class="btn"><a href="cart.html">Buy Now</a></button>
        </div>

        <div class="Items">
            <img class="dishpic" src="./images/chicken-butter-masala.jpg" alt="">
```

```
            <h3 class="Dish">Butter Chicken Masala</h3>
            <p class="price-tag">Price: ₹279</p>
            <br>
            <button type="button" class="btn" onclick="show_alert()"><a href="cart.html">Buy
Now</a></button>
        </div>

        <div class="Items">
            <img class="dishpic" src="./images/OIP.jpg" alt="">
            <h3 class="Dish">Veg Biryani</h3>
            <p class="price-tag">Price: ₹199</p>
            <br>
            <button type="button" class="btn"><a href="cart.html">Buy Now</a></button>
        </div>

        <div class="Items">
            <img class="dishpic" src="./images/butter-naan.jpg" alt="">
            <h3 class="Dish">Butter Naan</h3>
            <p class="price-tag">Price: ₹80</p>
            <br>
            <button type="button" class="btn" onclick="show_alert()"><a href="cart.html">Buy
Now</a></button>
        </div>
      </div>
   </section>

   <!-- Customer Review -->
   <section>
      <div class="testimonials">
         <div class="inner">
            <h1>Our Customer Reviews</h1>
            <div class="inner-border"></div>

            <div class="row">
               <div class="col">
                  <div class="testimonial">
                     <img src="./images/Jeff_Bezos.png" alt="Jeff_Bezos">
                     <div class="name">Jeff Bezos <br> CEO @Amazon
                     </div>
                        <hr>
                     <div class="star">
                        <i class="fa-sharp fa-solid fa-star"></i>
                        <i class="fa-sharp fa-solid fa-star"></i>
                        <i class="fa-sharp fa-solid fa-star"></i>
```

```
                    <i class="fa-sharp fa-solid fa-star"></i>
                    <i class="fa-sharp fa-solid fa-star"></i>
                </div>
                <p>"This place is great! Atmosphere is chill and cool but the staff is also friendly. They
know
                    what they’re doing and what they’re talking about, and you can tell making the
customers
                    happy is their main priority."</p>
            </div>
        </div>

        <div class="col">
            <div class="testimonial">
                <img src="./images/Steve_Ballmer.jpeg" alt="Steve_Ballmer">
                <div class="name">Steve Ballmer <br> CEO @Microsoft
                </div>
                    <hr>
                <div class="star">
                    <i class="fa-sharp fa-solid fa-star"></i>
                    <i class="fa-sharp fa-solid fa-star"></i>
                    <i class="fa-sharp fa-solid fa-star"></i>
                    <i class="fa-sharp fa-solid fa-star"></i>
                    <i class="fa-sharp fa-solid fa-star"></i>
                </div>
                <p>"It’s a great experience. The ambiance is very welcoming and charming. Amazing
wines, food
                    and service. Staff are extremely knowledgeable and make great
recommendations"</p>
            </div>
        </div>

        <div class="col">
            <div class="testimonial">
                <img src="./images/google_ceo.jpeg" alt="google_ceo">
                <div class="name">Sundar Pichai<br>CEO @google
                </div>
                    <hr>
                <div class="star">
                    <i class="fa-sharp fa-solid fa-star"></i>
                    <i class="fa-sharp fa-solid fa-star"></i>
                    <i class="fa-sharp fa-solid fa-star"></i>
                    <i class="fa-sharp fa-solid fa-star"></i>
                    <i class="fa-sharp fa-solid fa-star"></i>
                </div>
```

```
                    <p>"The Food we had enjoyed at the time of dinner. It was really delicious taste with
great
                        quality, everything had unique taste which we had ordered, nice arrangement and
services
                        from the staff while eating."</p>
                </div>
            </div>

        </div>
    </div>
</section>

<!-- Footer -->
<footer>
    <div class="footer-row">
        <div class="footer-col">
            <h4 class="footer-title">company</h4>
            <ul>
                <li><a href="about.html">about us</a></li>
                <li><a href="#">our services</a></li>
                <li><a href="#">privacy policy</a></li>
                <li><a href="#">trademark</a></li>
            </ul>
        </div>
        <div class="footer-col">
            <h4 class="footer-title">get help</h4>
            <ul>
                <li><a href="#">FAQ</a></li>
                <li><a href="#">delivery</a></li>
                <li><a href="#">returns</a></li>
                <li><a href="#">order status</a></li>
                <li><a href="#">payment options</a></li>
            </ul>
        </div>
        <div class="footer-col">
            <h4 class="footer-title">follow us</h4>
            <div class="social-links">
                <a href="#" title="FaceBook"><i class="fab fa-facebook-f"></i></a>
                <a href="#" title="Twitter"><i class="fab fa-twitter"></i></a>
                <a href="#" title="Instagram"><i class="fab fa-instagram"></i></a>
                <a href="#" title="LinkedIn"><i class="fab fa-linkedin-in"></i></a>
            </div>
        </div>
```

```
        </div>


    </footer>

</body>
</html>
```

Reservation.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Book A Table | Queen of Spices - Burgers, Pizza, Indian Veg & Non-veg Cuisine</title>

    <!-- external css -->
    <link rel="stylesheet" href="./css/style.css">
    <link rel="stylesheet" href="./css/reservation.css">

    <!-- favicon -->
    <link rel="shortcut icon" href="./favicon.svg" type="image/svg+xml">

    <!-- Font Awesome link -->
    <script src="https://kit.fontawesome.com/5289663089.js" crossorigin="anonymous"></script>

    <!-- javascript code -->
    <script src="/js/script.js" defer></script>
    <script src="/js/reservation.js" defer></script>
</head>

<body>
    <!-- Header Section -->
    <section class="header-img">
        <header class="header-main">
            <nav class="nav-main">
                <h1 class="main-title"><a href="#">Queen of Spices</a></h1>
                <div class="openMenu"><i class="fa-solid fa-bars" style="color: #ff9d2e;"></i></div>
                <ul class="nav-ul">
                    <li><a href="index.html">Home</a></li>
                    <li><a href="index.html#menu">Menu</a></li>
                    <li><a href="about.html">About</a></li>
```

```
            <li><a href="#">Reservation</a></li>
            <div class="closeMenu"><i class="fa-solid fa-xmark"></i></div>
          </ul>
        </nav>
      </header>
      <div class="container">
        <h2>Book A <span>Table</span></h2>
        <div class="reservation-details">
          <p>Welcome to our restaurant! To ensure that you have the best dining experience, we highly
recommend
            making
            a
            reservation. Our reservation system is quick and easy - simply fill out the form below with
your
            preferred
            date and time, and we will confirm your booking via email or phone.</p>
          <h2>Special Requests</h2>
          <p>If you have any special requests, please let us know in advance. Our team will do their best
to
            accommodate your needs and ensure that you have a comfortable and enjoyable dining
experience.</p>
        </div>
        <div class="table-container">
          <form name="reservation-form">
            <h2>Registration</h2>
            <div class="content">
              <div class="input-box">
                <label for="name">Full Name</label>
                <input type="text" placeholder="Enter full name" name="name" required>
              </div>
              <div class="input-box">
                <label for="email">Email</label>
                <input type="email" placeholder="Enter your email" name="email" required>
              </div>
              <div class="input-box">
                <label for="phone">Phone Number</label>
                <input type="tel" placeholder="Enter phone number" name="phone" pattern="[0-
9]{10}" required>
              </div>
              <div class="input-box">
                <label for="date">Date</label>
                <input type="date" placeholder="Enter Date" name="date" required>
              </div>
              <div class="input-box">
```

```
                    <label for="no-of-persons">Number Of Persons</label>
                    <input type="number" placeholder="Number Of Persons" min="1" max="10"
name="no-of-persons"
                        required>
                </div>
                <div class="input-box">
                    <label for="time">Timing</label>
                    <input type="time" placeholder="Enter phone number" name="time" required>
                </div>
            </div>
            <div class="button-container">
                <button type="submit" value="Reserve Seats" name="Reserve Seats">Reserve
Seats</button>
            </div>
        </form>
    </div>
  </div>
 </section>
</body>
</html>
```

Reservation.js

```
// Book A Table Form
const form = document.querySelector('form');
const submitButton = document.querySelector('input[type="submit"]');

form.addEventListener('submit', (event) => {
  event.preventDefault();

  if (form.checkValidity()) {
    alert("Thanks for reserving your seats, soon you will receive a message on your Phone Number &
Mail ID for the confirmations!");
    form.reset();
  } else {
    alert("Please fill out all required fields!");
  }
});

'use strict';

// NavBar Scroll Toggle
window.addEventListener("scroll", () => {
  var header = document.querySelector(".nav-main");
```

```
    header.classList.toggle("active", window.scrollY > 20);
});

// HumBurger Menu
const mainMenu = document.querySelector(".nav-ul");
const openMenu = document.querySelector(".openMenu");
const closeMenu = document.querySelector(".closeMenu");
const menu_items = document.querySelectorAll(".nav-ul li a");

menu_items.forEach((item) => {
    if (window.innerWidth < 800) {
        item.addEventListener("click", close);
    }
});
```

Script.js

```
openMenu.addEventListener("click", show);
closeMenu.addEventListener("click", close);

function show()
{
    mainMenu.style.display = "flex";
    mainMenu.style.top = "0";
}

function close()
{
    mainMenu.style.top = "-100vh";
}

// Alert message for Menu Items
function show_alert(){
    alert("No Feature Added!!!");
}
```

About.css

```
/* Importing Font Style from Google Fonts */
@import
url('https://fonts.googleapis.com/css2?family=Roboto:wght@400;500&family=Rubik:wght@400;500;60
0;700&family=Shadows+Into+Light&display=swap');
@import url('https://fonts.googleapis.com/css2?family=Rouge+Script&display=swap');
```

```
* {
    padding: 0;
    margin: 0;
    box-sizing: border-box;
    font-family: Roboto, 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
}
body {
    background-color: white;
}
html {
    scroll-behavior: smooth;
}

/* Custom Scroll Bar */
::-webkit-scrollbar {
    width: 8px;
}
::-webkit-scrollbar-track {
    background-color: #F9F6F0;
}
::-webkit-scrollbar-thumb {
    background-color: #FF9D2E;
    border-radius: 4rem;
}

/* About Us Page */
section.abs-section {
    margin: 0;
}
.abs-section {
    position: relative;
    height: 100vh;
    background-image: url("../images/hero-bg.jpg");
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
}
.abs-title {
    font-family: cursive, sans-serif;
    font-size: 48px;
    font-weight: bold;
    text-align: center;
    margin: 0;
    color: #FF9D2E;
```

```
    padding: 80px 0;
    text-shadow: -1px -1px 0 white, 1px -1px 0 white, -1px 1px 0 white, 1px 1px 0 white;
    padding-right: 20%;
    margin-bottom: -10px;
}
.abs-image img {
    position: absolute;
    top: 12%;
    right: 0;
    width: 40%;
}
.abs-section p {
    color: #F9F6F0;
    margin-right: 45%;
    margin-left: 4%;
    font-family: Roboto, sans-serif;
    font-size: 20px;
    padding: 6px;
    margin-bottom: 32px;
}
.abs-section p span a {
    text-decoration: none;
    color: #FF9D2E;
}
@media screen and (max-width: 1000px) {
    .abs-section {
        height: auto;
        padding: 80px 16px 0;
        display: flex;
        justify-content: center;
        align-items: center;
        flex-direction: column;
    }
    .abs-title {
        text-align: center;
        padding: 8px 12px;
        font-size: 40px;
    }
    .abs-image img{
        position: relative;
        width: 100%;
    }
    .abs-image, .abs-title {
        width: 64%;
```

```
        padding: 16px;
    }
    .abs-section p {
        width: 100%;
        margin: 0;
        padding: 0;
    }
    .abs-section p:not(:last-child) {
        margin-bottom: 24px;
    }
    .abs-section p:last-child {
        padding-bottom: 40px;
    }
}
```