|  |  |
| --- | --- |
| **RESTAURANT WEBSITE SYSTEM** | **PROJECT TITLE** |

A Mini Project Report

submitted by

**RIZVANA K A (MES24MCA-2045)**

to the APJ Abdul Kalam Technological University

in partial fulfilment of the requirements for the award of the Degree

of

Master of Computer Applications



**Department of Computer Applications**

MES College of Engineering

Kuttippuram, Malappuram – 679582

October, 2025

**Declaration**

I undersigned hereby declare that the project report **RESTAURANT WEBSITE SYSTEM** submitted for partial fulfilment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by me under supervision of **NOWSHAD C V**, Assistant Professor **HYDERALI K**, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

RIZVANA K A (MES24MCA-2045)

10-10-2025

DEPARTMENT OF COMPUTER APPLICATIONS

MES COLLEGE OF ENGINEERING, KUTTIPPURAM



**CERTIFICATE**

This is to certify that the report entitled **RESTAURANT WEBSITE SYSTEM** is a bonafide record of the Mini Project work during the year 2025-26 carried out by **RIZVANA K A (MES24MCA-2045)** submitted to the APJ Abdul Kalam Technological University, in partial fulfilment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor Head of The Department

**Acknowledgment**

I endeavor stands incomplete without dedicating my gratitude to a few people who have contributed towards the successful completion of my project. I pay my gratitude to the Almighty for His invisible help and blessings for the fulfillment of this work. At the outset, I express my heartfelt thanks to my Head of the Department, Prof. **HYDERALI K**, for permitting me to carry out this project work. I take this opportunity to express my profound gratitude to my project guide, **NOWSHAD C V**, for his valuable support, guidance, and encouragement throughout the course of this project. I also extend my sincere thanks to my project coordinator, **C M SULAIKHA**, for her timely advice, constant monitoring, and strict schedules that helped me complete the project on time. I am equally grateful to all my teaching and non-teaching staff of the department for their encouragement, guidance, and wholehearted support during this project. Last but not least, I am deeply indebted to my family and friends who stood by me with their constant encouragement, motivation, and precious help in completing this project successfully.

RIZVANA K A (MES24MCA-2045)

**Abstract**

The Restaurant Website System is an online platform designed to enhance restaurant services and customer engagement through a digital interface. This web-based system allows users to log in via email or phone number and access various features including menu browsing, food details, today's specials, and signature dishes

The project is motivated by the increasing demand for digital transformation in the hospitality industry, aiming to reduce the inconvenience customers face with traditional, manual processes like phone calls for reservations and orders. The system offers a fully functional restaurant website that facilitates online booking, ordering, and real-time customer interaction.

The core objective is to provide a centralized digital solution where customers can log in, view food menus with images, place orders, reserve seats for specific dates, and provide feedback on their dining experience. Additional features include photo galleries of the restaurant's ambiance and meals, as well as "About Us" and contact pages.

The project development uses front-end technologies such as HTML, CSS, and JavaScript for creating an interactive user interface. The backend is powered by Python, using frameworks like Django, while the database is managed through MySQL to store customer data, booking records, menu items, and reviews. The application is developed in Pycharm, and is compatible with Windows.

In conclusion, the Restaurant Website System provides a modern digital solution that streamlines restaurant operations and elevates the dining experience for customers.

**Table of Contents**

[List of Figures viii](#_Toc211267407)

[List of Tables ix](#_Toc211267408)

[Chapter 1. Introduction 1](#_Toc211267409)

[1.1 Motivation 1](#_Toc211267410)

[1.2 Objectives 2](#_Toc211267411)

[1.3 Contributions 2](#_Toc211267412)

[1.4 Report Organization 3](#_Toc211267413)

[Chapter 2. System Study 4](#_Toc211267414)

[2.1 Existing System 4](#_Toc211267415)

[2.2 Proposed System 4](#_Toc211267416)

[2.3 Functionalities of Proposed System 5](#_Toc211267417)

[Chapter 3. Methodology 7](#_Toc211267418)

[3.1 Introduction 7](#_Toc211267419)

[3.2 Software Tools 8](#_Toc211267420)

[3.2.1 Python 8](#_Toc211267421)

[3.2.2 Django 8](#_Toc211267422)

[3.2.3 MySQL 9](#_Toc211267423)

[3.3 Module Description 9](#_Toc211267424)

[3.3.1 User Module 9](#_Toc211267425)

[3.3.2 Admin Module 9](#_Toc211267426)

[3.4 User Story 10](#_Toc211267427)

[3.5 Product Backlog 11](#_Toc211267428)

[3.6 Project Plan 12](#_Toc211267429)

[3.7 Sprint Backlog 13](#_Toc211267430)

[3.8 Database Design 14](#_Toc211267431)

[Chapter 4. Results and Discussions 17](#_Toc211267432)

[4.1 Results 17](#_Toc211267433)

[Chapter 5. Conclusion 20](#_Toc211267434)

[References ……………………………………………………………………………………21](#_Toc211267435)

[Appendix…. 22](#_Toc211267436)

[Appendix A Data Flow Diagram 22](#_Toc211267437)

[Appendix B ER Diagram 25](#_Toc211267438)

[Appendix C Source Code 26](#_Toc211267439)

[Appendix D Screenshot 32](#_Toc211267440)

List of Figures

[**Figure 4.1:** Login Page [4] 18](#_Toc180355235)

[**Figure 4.2:** Registration Page[4] 18](#_Toc180355236)

[**Figure 4.3:** Available Tables[4] 19](#_Toc180355237)

[**Figure 0.1:** DFD Level 0 22](#_Toc180355233)

[**Figure 0.2:** DFD Level 1 23](#_Toc180355233)

[**Figure 0.3:** DFD Level 2 24](#_Toc180355233)

[**Figure 0.4:** ER Diagram 25](#_Toc180355233)

List of Tables

[**Table 3.2.0:** List the software tools or languages used fot the project development 8](#_Toc180355238)

[**Table 3.4.1:** User Story 10](#_Toc180355239)

[**Table 3.5.1:** Product Backlog 11](#_Toc180355239)

[**Table 3.6.1:** Project Plan 12](#_Toc180355239)

[**Table 3.7.1:** Sprint Backlog 13](#_Toc180355239)

[**Table 3.8.1:** Database Design 15](#_Toc180355239)

# Introduction

The Restaurant Website System is a web-based platform developed to digitalize and simplify restaurant operations while enhancing the overall customer experience. In traditional restaurant setups, customers often face difficulties when booking tables, browsing menus, or placing food orders through manual methods such as phone calls or walk-ins. These approaches are time-consuming, error-prone, and lack efficiency. To overcome these challenges, the proposed system provides an integrated online solution where customers can log in, explore digital menus with images, place food orders, book tables for specific dates, and share feedback about their dining experience. The system also enables restaurant administrators to efficiently manage menus, reservations, and customer reviews through a centralized dashboard. Developed using technologies like HTML, CSS, JavaScript, and Django with MySQL, this project ensures better accessibility, accuracy, and operational efficiency, thereby bridging the gap between restaurants and their customers.

## Motivation

The motivation behind developing the Restaurant Website System arises from the growing need for digital solutions that simplify restaurant operations and enhance customer satisfaction. In many restaurants, customers still depend on manual methods such as phone calls or in-person visits to place orders or reserve tables, which often lead to delays, miscommunication, and booking errors. With the increasing use of technology in everyday life, people expect faster and more convenient ways to access services. This project aims to provide a modern, user-friendly platform that allows customers to easily browse menus, make reservations, and place online orders, all in one place. By reducing manual workload and minimizing errors, the system improves the overall efficiency of restaurant management while offering customers a seamless and satisfying dining experience.

## Objectives

The main objectives of the Restaurant Website System are:

* To provide secure customer registration and login through email, phone number, or username.
* To allow customers to browse menus with food details, daily specials, and signature dishes.
* To enable seamless online ordering and secure table reservations.
* To incorporate a customer feedback and review system for continuous improvement.

To support restaurant owners with centralized management of bookings, customer data, and reviews.

## Contributions

The contributions of this project compared to traditional/manual systems are:

* Developed a **centralized web-based platform** that integrates menu browsing, online ordering, and table reservations into one system.
* Designed a **user-friendly interface** that enhances customer experience through interactive menus and real-time updates.
* Implemented a **customer feedback and review feature** to improve restaurant services and build customer trust.
* Created an **admin dashboard** for managing food items, reservations, orders, and user reviews efficiently.
* Ensured **data accuracy and security** by using a structured database system (MySQL) and modern web technologies like Django, HTML, CSS, and JavaScript.

## Report Organization

The project report is divided into five chapters:

* **Chapter 2 – System Study**: Explains the existing system, its limitations, and the proposed system functionalities.
* **Chapter 3 – Methodology**: Describes the project methodology, including module descriptions, user stories, sprint details, and database design.
* **Chapter 4 – Results and Discussions**: Provides outputs of the implemented system and discussion on how objectives were met.
* **Chapter 5 – Conclusion**: Summarizes the work done, contributions, and scope for future enhancements.

# System Study

The traditional restaurant system often relies on manual processes for reservations, menu browsing, and customer feedback, which can lead to errors, delays, and poor customer experience. Customers face inconvenience when ordering food or booking tables through phone calls, and restaurants struggle to showcase their offerings effectively. To solve these issues, the proposed Restaurant Website System provides a digital platform that simplifies ordering, booking, and interaction, making the entire process faster, more transparent, and user-friendly.

## Existing System

Most restaurants still use manual methods like phone calls and paper records for reservations, orders, and feedback. These systems often cause booking mistakes, slow service, and poor customer experience. Customers can’t view menus or specials online, and restaurants struggle to collect reviews or show their ambiance. To solve these problems, a digital system is needed that combines ordering, booking, and feedback into one easy-to-use website.

## Proposed System

* Provides a web-based platform for restaurants to manage orders, reservations, and customer feedback.
* Allows secure user login via email or phone number.
* Enables customers to place online orders and reserve tables for specific dates.
* Includes a feedback system for reviewing food quality and staff service.
* Admin panel manages users, menu items, bookings, and reviews.

**Benefits of the Proposed System**:

* **Technological**: Uses secure and scalable web technologies.
* **Functional**: Combines ordering, booking, and feedback in one platform.
* **Non-Functional**: Improves speed, accuracy, and customer satisfaction.

## Functionalities of Proposed System

1. **User Registration and Login**

* Users can sign up and log in using email or phone number.
* Secure authentication for customers and admins.

1. **Menu Browsing**

* Customers can view food items with images.
* Includes daily specials and signature dishes.

**3. Online Food Ordering**

* Users can select items, add to cart, and place orders.
* Order details are stored and managed in the database.

**4. Table Reservation**

* Customers can book seats for specific dates and times.
* Booking records are updated in real time.

**5. Feedback and Review System**

* Users can submit feedback on food and staff service.
* Admins can view and respond to reviews.

**6. Admin Dashboard**

* Admin can manage users, menu items, bookings, and feedback.
* Controls system content and monitors activity

# Methodology

The development of the Restaurant Website System requires a structured software methodology to ensure smooth and systematic execution of the project. A methodology serves as a blueprint, guiding each phase of development—such as requirement analysis, design, coding, testing, and deployment. It helps reduce errors and ensures that key user requirements like customer login, menu browsing, online ordering, table reservation, and feedback collection are properly implemented.

By following a defined methodology like Agile, the development team can manage the complexity of integrating multiple modules and user roles, improve collaboration, and deliver the system within the planned timeline. It also enhances the overall quality, reliability, and usability of the application, ensuring that the system meets its primary goal of improving customer experience, streamlining restaurant operations, and enabling digital transformation in the hospitality sector.

## Introduction

The **Restaurant Website System** follows the **Agile Software Development Methodology**, which focuses on flexibility, collaboration, and delivering functional parts of the system in short, iterative cycles called sprints. This approach allows developers to continuously gather feedback, refine requirements, and improve the system at every stage. Each sprint aims to develop a specific module, such as the user login, menu browsing, or reservation feature, and deliver a working version that can be tested and reviewed before moving to the next stage.

Using Agile helped ensure that the project stayed adaptive to changes in requirements and feedback from the guide. It encouraged better communication, reduced development risks, and maintained consistent progress. As a result, the system was developed efficiently, meeting both functional and user expectations while maintaining high quality and performance.

## Software Tools

The tools and technologies used in the development of this project are listed in **Table 3.2**.**0**

**Table 3.2.0:** List the software tools or languages used for the project development

|  |  |
| --- | --- |
| Operating System | Windows 11 |
| Front End | JavaScript, HTML, CSS |
| Back End | Python (Django) |
| Framework | Django |
| Database | MySQL |
| IDE | PyCharm |
| Version Control | Git |

### Python

For the development of the Restaurant Website System, Python was chosen as the backend language because it is easy to learn, fast to develop, and well-suited for web applications. Python frameworks like Django help manage user login, menu display, online ordering, table booking, and feedback collection with less code and more security. Python also works smoothly with databases like MySQL and Firebase, making it ideal for handling customer data and restaurant operations. Its flexibility and strong community support make it a reliable choice for building and maintaining this system.

### Django

For my Restaurant Website System, I chose Django as the backend framework because it is easy to use, secure, and helps build websites quickly. Django has built-in features like login system, admin panel, and database support, which are perfect for handling customer orders, table bookings, and feedback. I used Python as the backend language because it works well with Django and is simple to understand. For the frontend, I used HTML, CSS, and JavaScript to design the pages and make them look attractive and interactive. I used MySQL to store user data, orders, and reviews.

### MySQL

I choose MySQL as the database for my Restaurant Website System because it is fast, reliable, and easy to use. MySQL helps store and manage important data like customer details, menu items, orders, table bookings, and feedback. It works well with Django and supports secure data handling, which is important for restaurant operations. MySQL also allows smooth connections between the backend and frontend, making the system more efficient and user-friendly.

## Module Description

A module is a functional unit of the system that performs specific operations. The proposed Restaurant Website System is divided into two major modules: User Module and Admin Module. Each module works together to ensure smooth and efficient operation of the restaurant system.

### User Module

The User Module is designed for customers who use the restaurant website. It allows users to register and log in securely using their email or phone number. Once logged in, they can browse the digital menu, view food items with images and prices, place online orders, and book tables by selecting the date. Customers can also give feedback about their experience, which helps improve service quality. This module makes the system easy to use and provides a smooth experience for restaurant visitors.

### Admin Module

The Admin Module is used by the restaurant staff to manage the whole system. It allows the admin to log in securely and control different parts of the website. The admin can add, edit, or remove food items from the menu, check and manage customer orders, confirm or cancel table bookings, and read customer feedback. This module helps the restaurant run smoothly by keeping everything organized and up to date.

## User Story

User stories describe how different users interact with the system to achieve specific goals. They help guide the development process by focusing on user needs and expected outcomes. The user stories for this project are listed in **Table 3.4.1** below.

**Table 3.4.1**: User story

|  |  |  |  |
| --- | --- | --- | --- |
| User Story ID | As a type of User | I want to | So that I can |
| 1 | Admin | Login | Access the system securely with correct username and password |
| 2 | Admin | Manage Menu | |  |  | | --- | --- | |  | Add, edit, or delete food items, categories, and prices | |
| 3 | Admin | Manage Reservations | Approve, reject, or update table bookings |
| 4 | Admin | Generate Reports | View sales, reservations, and customer feedback for decision making |
| 5 | Customer | Register and log in | Access the system and use its features |
| 6 | Customer | Browse the menu | |  |  | | --- | --- | |  | View available food items with images and prices | |
| 7 | Customer | Place an order | |  |  | | --- | --- | |  | Order food online and get it delivered or ready for pickup | |
| 8 | Customer | Book a table | Reserve a seat at the restaurant for a specific date and time |
| 9 | Customer | Give feedback | Share my experience and help improve the restaurant’s service |

## Product Backlog

The product backlog lists all the major tasks required to build the Restaurant Website System. Each item includes its priority, estimated time, and current status. The detailed backlog is shown in **Table 3.5.1** below.

**Table 3.5.1**: Product Backlog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Priority** | **Estimate (Hours)** | **Status** |
| 1 | User Registration & Login | High | 6 | Completed |
| 2 | Profile Management | High | 5 | Completed |
| 3 | Menu Browsing & Display | High | 8 | Completed |
| 4 | Specials & Signature Dishes | High | 6 | Completed |
| 5 | Search & Filter Menu Items | High | 6 | Completed |
| 6 | Cart and Order Placement | High | 7 | Completed |
| 7 | Table Booking System | High | 6 | Completed |
| 8 | Feedback Submission | Medium | 4 | Completed |
| 9 | Admin Login | High | 3 | Completed |
| 10 | Menu Management (Admin) | High | 6 | Completed |
| 11 | Reservation Management (Admin) | High | 5 | Completed |
| 12 | Report Generation (Admin) | Medium | 5 | Completed |

## Project Plan

A project plan outlines the timeline and progress of tasks based on user stories. It helps track development across sprints and ensures that each feature is completed on time. The plan includes task names, start and end dates, duration, and current status. The detailed project plan for this system is shown in **Table 3.6.1** below.

**Table 3.6.1**: Project Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User Story ID** | **Task Name** | **Start Date** | **End Date** | **Days** | **Status** |
| 1,2,3,45 | Sprint 1 | 01/09/25 | 14/09/25 | 14 | Complete |
| 01/09/25 | 14/09/25 | 14 | Complete |
| 01/09/25 | 14/09/25 | 14 | Complete |
| 6,7,8,9 | Sprint 2 | 15/09/25 | 28/09/25 | 14 | Complete |
| 15/09/25 | 28/09/25 | 14 | Complete |
| 15/09/25 | 28/09/25 | 14 | Complete |
| 29/09/25 | 12/10/25 | 14 | Complete |
| 29/09/25 | 12/10/25 | 14 | Complete |
| 29/09/25 | 12/10/25 | 14 | Complete |

## Sprint Backlog

The sprint backlog outlines the tasks planned for each sprint, along with estimated hours and daily progress. It helps the development team manage time, track completion, and stay aligned with agile goals. Each sprint includes specific backlog items linked to user stories. The detailed sprint backlog for this project is shown in **Table 3.7.1** below.

**Table 3.7.1**: Sprint Backlog

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Backlog Item** | **Completion Date** | **Original Estimation (Hours)** | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** | **Day 6** | **Day 7** | **Day 8** | **Day 9** | **Day 10** |
| **SPRINT 1** | | | | | | | | | | | | |
| Project Setup & DB Creation | 20/07/2026 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Admin Login | 01/09/2025 | 3 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Menu Management | 01/09/2025 | 6 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Reservation Management | 01/09/2025 | 5 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| **SPRINT 2** | | | | | | | | | | | | |
| User Registration/Login | 15/09/2025 | 6 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Profile Management | 15/09/2025 | 5 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Backlog Item** | **Completion Date** | **Original Estimation (Hours)** | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** | **Day 6** | **Day 7** | **Day 8** | **Day 9** | **Day 10** |
| **SPRINT 2** | | | | | | | | | | | | |
| Menu Browsing & Display | 15/09/2025 | 6 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Specials & Signature Dishes | 15/09/2025 | 6 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Search & Filter Menu Items | 15/09/2025 | 6 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Table Booking System | 29/09/2025 | 6 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Feedback Submission | 29/09/2025 | 4 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

## Database Design

The database design defines the structure and relationships of tables used in the Restaurant Website System. It supports secure user authentication, menu management, order placement, table booking, and feedback collection. This schema ensures efficient data handling and scalability across both Admin and User module. The detailed schema is presented in **Table 3.8.1** below.

**Table 3.8.1**: Database Design

|  |  |  |
| --- | --- | --- |
| **Collection** | **Attributes** | **Purpose** |
| **users** | * username * password | Stores details of registered users including customers and admins |
| **menu\_items** | * itemId (PK) * item\_name * category * price | Stores menu items with pricing and category details |
| **orders** | * orderId (PK) * userId (FK * itemId (FK * quantity * order\_date * status | Stores food orders placed by users |
| **reservations** | * reservationId (PK) * userId (FK) * date * status | Stores table booking details made by users |
| **feedbacks** | * feedbackId (PK) * userId (FK) * message * submitted\_on | Stores feedback submitted by user |
| **Collection** | * **Attributes** | **Purpose** |
| **admin\_reports** | * reportId (PK) * generated\_by (FK → users) * report\_type * generated\_date * details | Stores system-generated reports for admin review |

# Results and Discussions

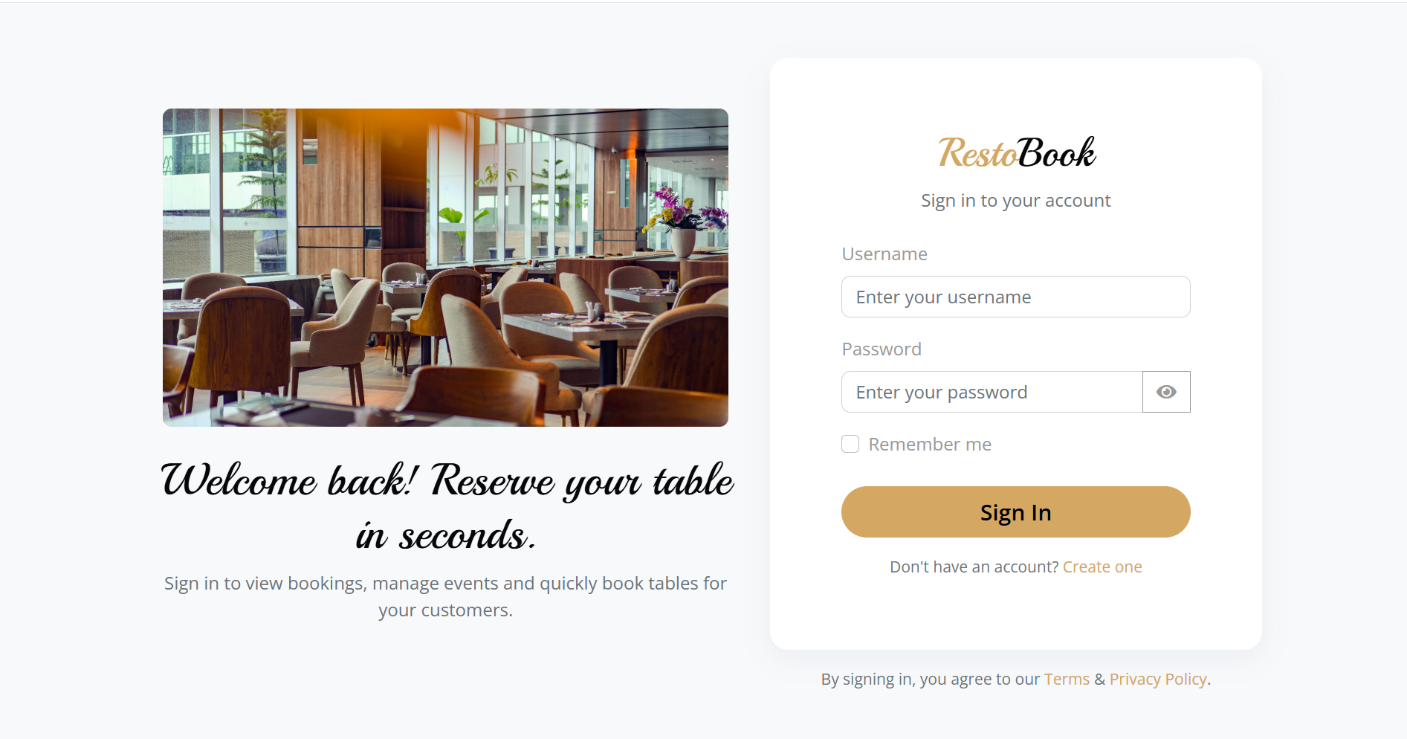
The Restaurant Website System was successfully developed with two main modules: **User Module** and **Admin Module**. Each module plays a vital role in streamlining restaurant operations and enhancing the customer experience. The system allows users to browse the menu, place orders, reserve tables, and submit feedback. The admin module enables restaurant staff to manage menu items, view reservations, track orders, and analyze customer feedback.

This chapter presents the results of the implemented system, supported by screenshots of key interfaces. The discussion highlights how each form contributes to the overall functionality and usability of the system. Forms with valid sample data are shown to demonstrate real-world usage. Basic forms such as login and registration are excluded to focus on more critical features.

## Results

This section presents the main working pages of the Restaurant Website System. Screenshots of selected forms are included to demonstrate how each module functions in practice. Only forms with valid sample data are shown to reflect real usage. Less important forms such as login and registration are excluded.

Each figure is numbered and includes a caption that describes its purpose. A brief explanation is provided for each form to show how it works within the system.



**Figure 4.1:** Login page[4]

The login page allows users and admins to sign in using their username and password. It provides secure access to features like table booking, order placement, and menu management.

A screenshot of a login form

AI-generated content may be incorrect.

**Figure 4.2:** Registration page[4]

**Figure 4.2** This form allows users to enter their contact details and location to request restaurant services. It includes fields like name, phone number, address, and pin code, followed by a “Submit Now” button to confirm the request.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 4.2:** Available Tables [4]

**Figure 4.3** This page allows the admin to manage restaurant tables by adding, editing, or deleting table details like name, number, and capacity. It helps organize seating arrangements efficiently for reservations.

# Conclusion

The Restaurant Website System was developed to provide a digital solution for managing restaurant operations, focusing on table reservations, menu browsing, order placement, and customer feedback. The system consists of two main modules—User and Admin—that work together to streamline interactions between customers and restaurant staff. Key features such as dynamic menu management, real-time order tracking, and reservation scheduling were successfully implemented, resulting in a user-friendly and efficient platform.

The project achieved its core objectives by reducing manual errors, improving service coordination, and enhancing the overall dining experience. It demonstrates the potential of web-based systems in transforming traditional restaurant workflows into structured, automated processes. However, certain advanced functionalities like payment gateway integration and multi-role access control could not be implemented due to time limitations and resource constraints. These features can be added in future versions by expanding the system architecture and integrating third-party APIs.

In conclusion, the developed system lays a strong foundation for scalable restaurant management and offers valuable scope for future enhancements. It reflects the importance of combining user-centric design with practical functionality to meet real-world needs.

# References

|  |  |
| --- | --- |
| [1] | Bootstrap, “Bootstrap Documentation,” [Online]. Available: https://getbootstrap.com/docs/. . [Accessed: 07 October 2025]. |
| [2] | GeeksforGeeks, “Introduction to Web Development,” [Online]. Available: https://www.geeksforgeeks.org/web-development/. . [Accessed: 08 October 2025]. |
| [3] | FreeCodeCamp, “Responsive Web Design Principles,” [Online]. Available: https://www.freecodecamp.org/news/responsive-web-design-principles/. . [Accessed: 08 October 2025]. |
| [4] | MDN Web Docs, “HTML: HyperText Markup Language,” Mozilla Foundation. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTML. . [Accessed: 08 October 2025]. |

[5] A. Sharma, *Effective UI/UX Design for Web Applications*, TechPress Publications, 2022.

Appendix

1. Data Flow Diagram

**LEVEL 0**

A diagram of a restaurant

AI-generated content may be incorrect.

**Figure 0.1** DFD Level 0

**LEVEL 1**

A diagram of a company

AI-generated content may be incorrect.

**Figure 0.2** DFD Level 1

**LEVEL 2**

A diagram of a user

AI-generated content may be incorrect.

**Figure 0.3** DFD Level 2

1. ER Diagram

A diagram of a diagram

AI-generated content may be incorrect.

**Figure 0.4** ER Diagram

1. Source Code

Views.py

from django.contrib.auth import authenticate, login, logout  
from django.contrib.auth.decorators import login\_required  
from django.contrib.auth.hashers import make\_password  
from django.http import HttpResponse  
from django.contrib.auth.models import User,Group  
from django.shortcuts import render, get\_object\_or\_404, redirect  
from django.contrib import messages  
from datetime import datetime  
  
  
from myapp.models import \*  
  
  
def login\_page(request):  
 return render(request,'login.html')  
  
def login\_post(request):  
 username = request.POST['username']  
 password = request.POST['password']  
 *# print(request.POST)* user = authenticate(request, username=username, password=password)  
 if user is not None:  
 *# print(user,'kkkkkkkk')* if user.groups.filter(name='admin').exists():  
 *# print('llll')* login(request, user)  
 messages.success(request,'logined Successfully..!!')  
 return redirect('/myapp/admin\_home/')  
 elif user.groups.filter(name='user').exists():  
 login(request,user)  
 messages.success(request,'logined Successfully..!!')  
  
 return redirect('/myapp/user\_home/')  
 else:  
 return redirect('/myapp/')  
 else:  
 messages.warning(request, 'invalid username or password')  
 return redirect('/myapp/')  
  
@login\_required(login\_url='/myapp/')  
def admin\_add\_table(request):  
 return render(request,'admins/add table.html')  
  
  
@login\_required(login\_url='/myapp/')  
  
def add\_table\_post(request):  
 tablename=request.POST['tablename']  
 capacity=request.POST['capacity']  
  
 obj=Table\_table()  
 obj.tablename=tablename  
 obj.capacity=capacity  
 obj.save()  
 return redirect('/myapp/admin\_view\_table')  
  
  
@login\_required(login\_url='/myapp/')  
  
def user\_reg(request):  
 return render(request,'user/user\_reg\_index.html')  
def user\_reg\_post(request):  
 name=request.POST['name']  
 email=request.POST['email']  
 phone=request.POST['phone']  
 address=request.POST['address']  
 pin=request.POST['pin']  
 post=request.POST['post']  
 username=request.POST['username']  
 password=request.POST['password']  
  
 user=User.objects.create(username=username, password=make\_password(password),first\_name=name,email=email)  
 user.groups.add(Group.objects.get(name='user'))  
  
  
 obj=User\_table()  
 obj.name=name  
 obj.email=email  
 obj.address=address  
 obj.pin=pin  
 obj.post=post  
 obj.phone=phone  
 obj.LOGIN=user  
 obj.save()  
  
 return redirect('/myapp/')  
@login\_required(login\_url='/myapp/')  
  
def user\_view\_food\_rating(request):  
 a=Review\_table.objects.all()  
 return render(request,'user/view\_food\_ratings.html',{'data':a})  
  
def user\_add\_food\_rating(request,id):  
 request.session['rateid']=id  
 return render(request,'user/add\_food\_rating.html')  
  
def food\_rate\_post(request):  
 rating=request.POST['rating']  
 review=request.POST['review']  
  
  
 obj=Review\_table()  
 obj.rating=rating  
 obj.review=review  
 obj.date=datetime.today()  
 obj.USER=User\_table.objects.get(LOGIN\_id=request.user.id)  
 obj.FOOD=Food\_Items\_table.objects.get(id=request.session['rateid'])  
 obj.save()  
 return redirect('/myapp/user\_view\_food\_rating/')  
@login\_required(login\_url='/myapp/')  
  
def raz\_pay(request, amount, id):  
 from django.contrib import messages  
 import razorpay  
  
 try:  
 amount = float(amount)  
 except ValueError:  
 messages.error(request, "Invalid amount.")  
 return redirect('myapp/user\_view\_cart2/#block')  
  
 razorpay\_api\_key = "rzp\_test\_MJOAVy77oMVaYv"  
 razorpay\_secret\_key = "MvUZ03MPzLq3lkvMneYECQsk"  
 client = razorpay.Client(auth=(razorpay\_api\_key, razorpay\_secret\_key))  
  
 amount\_paise = int(round(amount \* 100))  
  
 order\_data = {  
 'amount': amount\_paise,  
 'currency': 'INR',  
 'receipt': f'order\_rcptid\_{id}',  
 'payment\_capture': '1',  
 }  
 try:  
 razorpay\_order = client.order.create(data=order\_data)  
 except Exception:  
 messages.error(request, "Failed to create Razorpay order. Try again.")  
 return redirect('/myapp/user\_view\_cart2/#block')  
  
 context = {  
 'razorpay\_api\_key': razorpay\_api\_key,  
 'amount': amount\_paise,  
 'currency': 'INR',  
 'order\_id': razorpay\_order['id'],  
 'display\_amount': amount,  
 'order\_main\_id': id,  
 }  
 return render(request, 'user/pp.html', context)  
  
  
from django.views.decorators.csrf import csrf\_exempt  
from django.http import JsonResponse  
import hmac, hashlib  
  
@login\_required(login\_url='/myapp/')  
  
def user\_logout(request):  
 logout(request)  
 return redirect('/myapp/')

1. Screenshot

Available Foods

A screenshot of a computer

AI-generated content may be incorrect.

Feedback

A screenshot of a computer

AI-generated content may be incorrect.

Orders

A screenshot of a website

AI-generated content may be incorrect.

Table Bookings

A screenshot of a phone

AI-generated content may be incorrect.

Payment

A white background with a blue and white background

AI-generated content may be incorrect.