

ACKNOWLEDGEMENT

I sincerely express my gratitude to everyone who contributed to the successful development of my restaurant management system Eat Ease for my semester mid-defense. I am especially thankful to Abhishek Koirala sir and Rishav Acharya sir for their invaluable guidance, support, and encouragement, which played a crucial role in shaping this project. My heartfelt appreciation extends to my mentors and instructors for their insightful feedback and continuous support. I am also grateful to my colleagues for their knowledge sharing and technical discussions, which enriched my understanding. A special acknowledgment goes to the open-source community and the developers of Django and React, whose frameworks formed the foundation of this system. Lastly, this project was inspired by real-world challenges in restaurant management, and I hope it effectively streamlines restaurant operations, enhances user experience, and improves booking efficiency.

ABSTRACT

Eat Ease is an online platform designed to streamline restaurant management and enhance the dining experience through an efficient seat booking system. Customers can browse restaurants, check seat availability, book or cancel reservations, and manage their profiles, while restaurant owners can handle bookings, update seat statuses, and manage reservations. The platform defines three distinct roles: Admin, Customer, and Owner, each with specific permissions to ensure smooth operations. Admins have overall control, ensuring the system runs efficiently. Eat ease is built using Django for the backend and React for the frontend, providing a secure and user-friendly interface for seamless restaurant booking and management.

Keywords: *Restaurant management, Seat booking, Admin control, Customer reservations.*

TABLE OF CONTENTS

ACKNOWLEDGEMENT	i
ABSTRACT.....	ii
TABLE OF CONTENTS	iii
LIST OF FIGURES	v
LIST OF TABLES	vi
LIST OF ABBREVIATIONS.....	vii
CHAPTER 1: INTRODUCTION.....	1
Background.....	1
1.2 Problem Statement.....	1
1.4 Scope and Limitation.....	2
1.5 Methodology	3
CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW	5
2.1 Background Study.....	5
2.2 Literature Review.....	5
CHAPTER 3: SYSTEM ANALYSIS AND DESIGN.....	6
3.1 System Analysis.....	6
3.1.1 Requirement Analysis.....	6
3.1.2 Feasibility Analysis.....	7
3.1.3 Schedule Feasibility.....	8
3.2 System Design.....	9
3.2.1 ER Diagram.....	9
3.2.2 Class Diagram.....	10

3.2.3 Sequence Diagram.....	12
3.2.4 DFD Diagram.....	13
3.2.4 Activity Diagram.....	15
CHAPTER 4: IMPLEMENTATION AND TESTING.....	16
CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATION.....	18
4.1 Conclusion.....	18
4.2 Learner’s Outcome.....	18
4.3 Future Recommendation.....	18
REFERENCES.....	19
APPENDIX.....	20

LIST OF FIGURES

Figure 1 Iterative Waterfall Model	3
Figure 3.1 Use Case Diagram	7
Figure 3.2 ER Diagram	9
Figure 3.3 Class Diagram	10
Figure 3.4 Sequence Diagram.....	12
Figure 3.5 DFD Diagram	13
Figure 3.6 Activity Diagram	15

LIST OF TABLES

Table 1 Gantt Chart.....	Error! Bookmark not defined.
--------------------------	-------------------------------------

LIST OF ABBREVIATIONS

CRUD:	Create, Read, Update, Delete
Django:	A high-level Python web framework
ER:	Entity Relationship
MYSQL:	My Structured Query Language
React JS:	JavaScript Library for Website Frontend
TailwindCSS:	Open Source CSS framework

CHAPTER 1: INTRODUCTION

Background

Managing restaurant reservations efficiently has always been a challenge, often leading to miscommunication, overbookings, and customer dissatisfaction. Traditional reservation methods, such as phone calls or in-person bookings, can be time-consuming and prone to errors. Additionally, restaurant owners often struggle with optimizing seat occupancy, while customers face difficulties in securing their desired bookings. To address these inefficiencies, Eat Ease provides a simple, digital solution that simplifies restaurant seat reservations, enhances customer experience, and improves operational efficiency for restaurant owners.

1.2 Problem Statement

The restaurant industry lacks a centralized, user-friendly system that allows customers to simply browse, book, and manage their reservations while enabling restaurant owners to efficiently handle bookings and seat availability. Many establishments still rely on outdated methods, leading to overbooking, long wait times, and customer dissatisfaction. Additionally, restaurant owners struggle to keep track of available seats in real-time, making it difficult to optimize their seating arrangements. Without an integrated digital solution, both customers and restaurant owners face challenges in managing reservations effectively. Eat Ease aims to bridge this gap by providing a platform that ensures smooth, hassle-free restaurant seat booking and management.

1.3 Objectives

The objectives of this project are:

- To develop a digital platform that simplifies restaurant seat reservations for customers.
- To enable restaurant owners to efficiently manage bookings, cancellations, and seat availability.
- To provide an intuitive and user-friendly interface for both customers and restaurant staff.

1.4 SCOPE AND LIMITATION

The Eat Ease Restaurant Management System is designed to enhance the booking experience for both customers and restaurant owners. The key features include:

Scopes:

- **Restaurant Browsing:** Customers will be able to explore various restaurants, check details, and view available seating options.
- **Seat Reservation and Booking Management:** Users will conveniently book and cancel seats, while restaurant owners will manage bookings, update seat availability, and handle reservations efficiently.
- **User-Friendly Experience:** A simple and intuitive interface will ensure that customers can easily navigate through restaurant listings, book seats, and manage their reservations.
- **Booking Status Updates:** Restaurant owners will be able to change the status of reservations (e.g., confirmed, canceled, pending) to ensure smooth management.

Limitations:

- **No Food Ordering Feature:** The system is focused only on seat reservations and does not include food ordering or delivery functionalities.
- **No Automated Payment Processing:** Payments for reservations, if required, must be handled externally, as the platform does not integrate an online payment system.
- **Single Owner Management Per Restaurant:** Each restaurant is managed by a single registered owner, limiting collaborative management for multiple stakeholders within the same restaurant.

1.5 METHODOLOGY

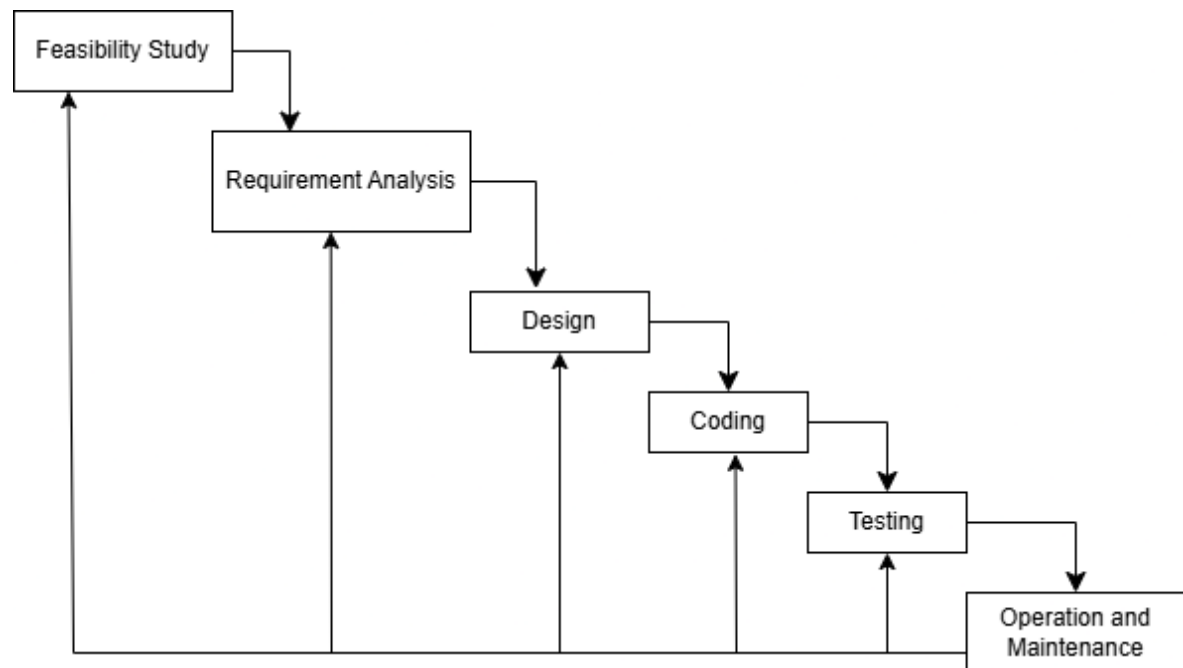


Figure 1.1: Iterative Waterfall Model

For the development of Eat Ease, the Iterative Waterfall Model was followed. This approach maintains a structured development process while allowing necessary refinements at each stage. Unlike the traditional waterfall model, it includes feedback loops, enabling modifications before progressing further. This ensures a well-structured system for the mid-defense presentation and future final defense.

Phases of the Iterative Waterfall Model

1) Feasibility Study

This phase determines whether the project is practical and achievable. It evaluates factors such as technical feasibility, cost, resources, and potential risks. The outcome of this phase is a feasibility report, which helps in deciding whether the project should proceed.

2) Requirement Analysis

In this phase, functional and non-functional requirements of the system are gathered and documented. Functional requirements define the system's features and operations, while non-functional requirements focus on performance, security, and usability. Stakeholders collaborate to ensure all necessary aspects are considered before moving to the design phase.

3) System Design

A structured plan is created for the system's architecture. It involves breaking down the system into smaller components (modules) and defining their structure, data flow, and interactions. This phase includes both high-level design (overall system structure) and detailed design (internal logic, database schema, and process flows), serving as a blueprint for development.

4) Coding and Testing of the Modules

The system design is implemented by writing actual code. Each module is developed separately and tested through unit testing to ensure it functions correctly. This phase focuses on writing efficient, error-free code before integrating different modules.

5) Integration and System Testing

After individual modules are built, they are integrated to form a complete system. Integration testing ensures that different components work together seamlessly. Following this, system testing verifies that the system meets the initial requirements and performs as expected before deployment.

6) Deployment and Maintenance

Once testing is complete, the system is deployed for real-world use. During this phase, bug fixes, updates, and optimizations are carried out based on user feedback. Maintenance ensures that the system remains functional, secure, and efficient over time, allowing for future enhancements as needed.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

Nepal's restaurant industry has experienced significant growth in digital adoption, driven by increasing internet penetration and the demand for efficient service delivery. This shift has led to the increase of various restaurant management systems (RMS) focusing to the local market. These systems aim to streamline operations, enhance customer experience, and provide data-driven insights for business growth.

Despite the availability of these solutions, challenges presents, such as the need for more localized features, affordability for small to medium enterprises. Addressing these gaps presents an opportunity for innovative solutions like EAT EASE to make a significant impact.

2.2 Literature Review

1. Restronp

Restronp is a Nepal-made all-in-one system designed specifically for local restaurants [3]. It offers features like real-time order tracking, staff management, and detailed reporting. The system is cloud-based, allowing access from multiple devices, and supports integration with various payment methods. Its affordability and localized support make it a popular choice among Nepali restaurateurs.

2. Atithi

Atithi provides comfortable hospitality solutions, focusing on enhancing customer engagement [4]. Key features include digital menus with ordering functionality, online table reservations, customer feedback systems, and mobile-responsive websites. Atithi's user-friendly interface and customizable options shows to the diverse needs of Nepali restaurants.

These systems highlight the growing trend of digitization in Nepal's restaurant industry. However, there remains a need for solutions that offer deeper localization, affordability, and integration with systems and people. EAT EASE can position itself uniquely by addressing these specific needs, providing a needy solution for the Nepali market.

CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

System analysis examines the current reservation process to identify inefficiencies like double bookings and no-shows. It reveals gaps in existing solutions—such as high costs excluding small restaurants—and defines key requirements for improvement. This structured approach ensures our Restaurant Seat Booking System delivers real-time updates, automation, and scalability to benefit both diners and restaurants.

3.1.1 Requirement Analysis

Requirement analysis identifies the main features needed for EAT EASE to work effectively and meet user needs. Key features include user accounts, restaurant browsing, order management, real-time tracking, secure payment, and feedback.

i) Functional Requirement

- **User Accounts & Authentication:** Users can register and log in using email, phone, or social accounts. They can manage profiles, addresses, and recover forgotten passwords securely.
- **Restaurant browse:** Users can browse restaurants by location.
- **Real-Time Seat Tracking:** Users can track their seats in real-time
- **Notifications:** Restaurants can offer discounts and users receive notifications about bookings and updates.

ii) Non-Functional Requirement

- **Scalability:** The system can smoothly handle more users, restaurants, and bookings as it grows.
- **Reliability:** Ensures high availability with minimal downtime for users, restaurants, and admins.
- **Security:** Protects user data with strong authentication and authorization to prevent unauthorized access.
- **Performance Efficiency:** Fast system responses through optimized database queries and caching for smooth browsing and booking.

Use Case Diagram

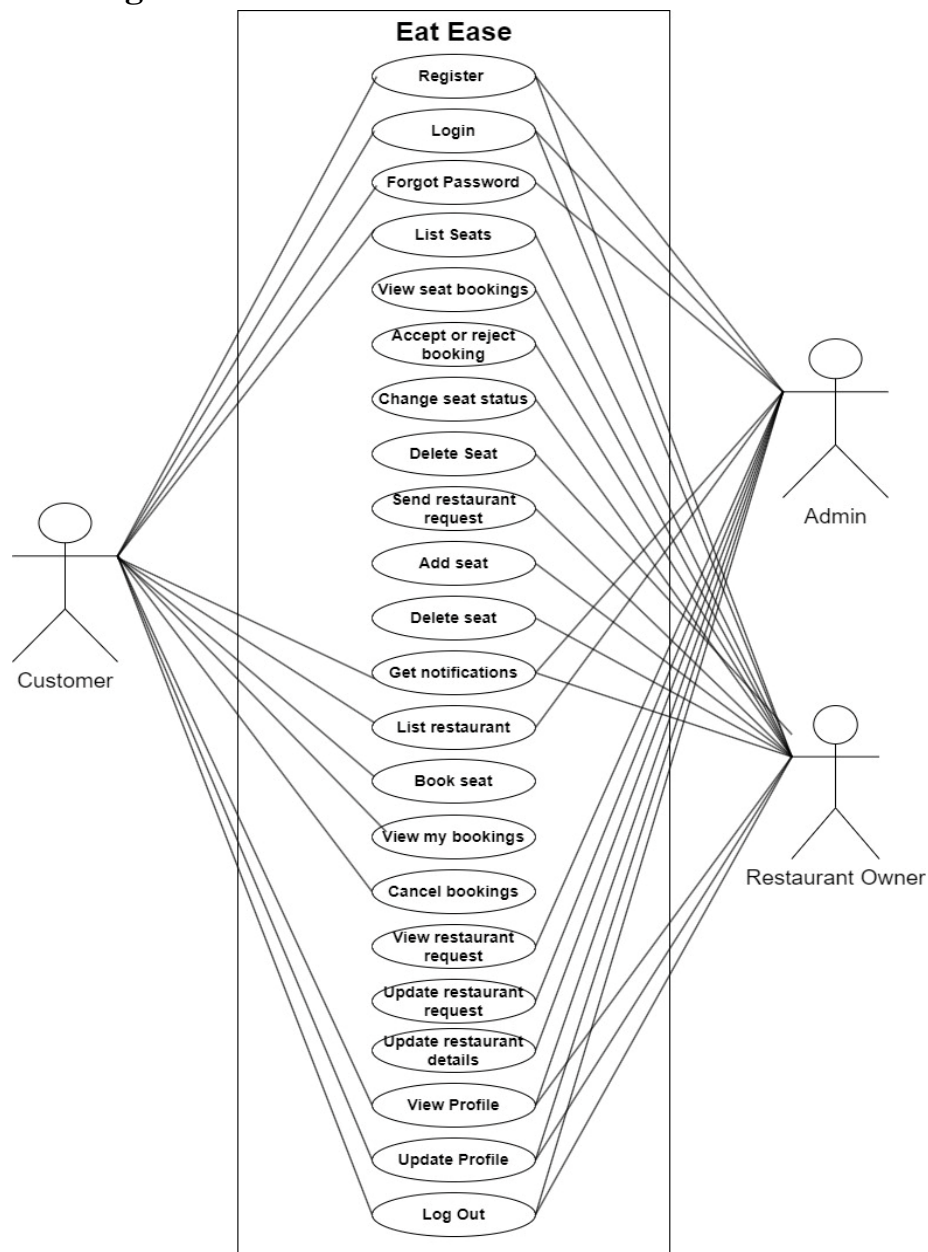


Figure 3.1: Use Case Diagram

3.1.2 Feasibility Analysis

i) Technical Feasibility

The system was technically feasible as it utilized widely used technologies. Django provided a scalable backend, while React ensured a seamless and interactive user experience. The use of Tailwind CSS enhanced frontend design, and MySQL was used for the database without compromising performance.

ii) Operational Feasibility

The platform was designed for both customers and restaurant owners, ensuring smooth functionality. Customers could easily book and manage reservations, while owners had control over seat availability and booking status. With a structured workflow, the system enhanced efficiency and improved restaurant operations.

iii) Economic Feasibility

The project was cost-effective, as it leveraged open-source technologies such as Django, React, and Tailwind CSS. This minimized development expenses while ensuring a high-quality system. Maintenance costs were also reduced due to the efficient architecture.

3.1.3 Schedule Feasibility

The system has been designed to be develop in phases, starting with core functionalities like restaurant listing and seat booking. The project timeline shows how to ensure timely completion without compromising quality. Below is the Gantt chart that outlines the respective timelines.

Restaurant Seat Booking System																
TIME TASKS	JANUARY 2025				FEBRUARY 2025				MARCH 2025				APRIL 2025			
	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 1	WEEK 2	WEEK 3	WEEK 4
Feasibility Study and Planning																
Requirement Analysis																
System Design																
Coding and Testing																
Operation and Maintenance																
Documentation																

Table 1: Gantt Chart

3.2 System Design

3.2.1 ER Diagram

An ER Diagram visually models a database's structure using entities (tables), attributes (fields), and relationships (connections). It shows how data links together through one-to-one, one-to-many, or many-to-many relationships, serving as a blueprint for efficient database design.

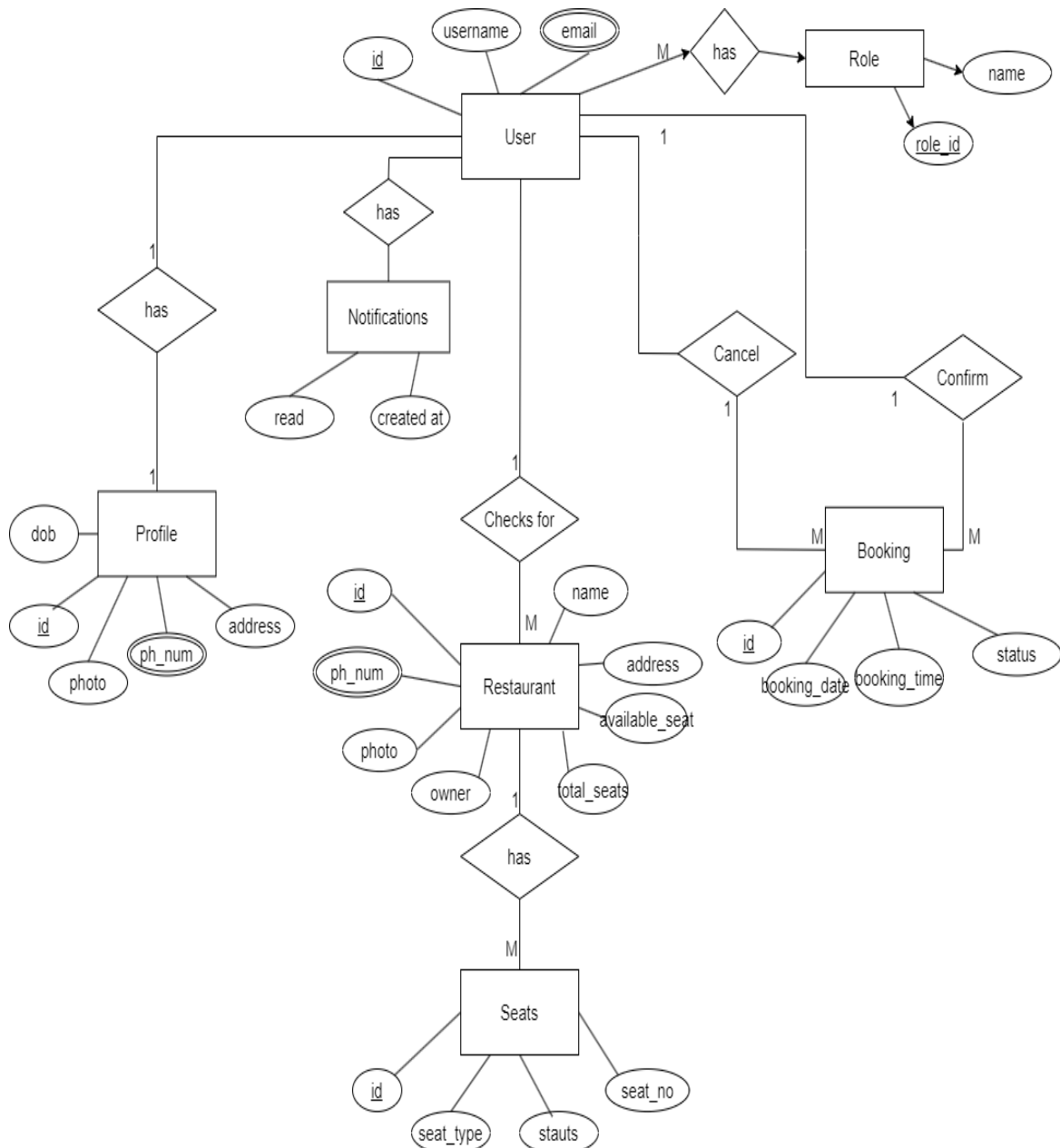


Figure 3.2: ER Diagram

3.2.2 Class Diagram

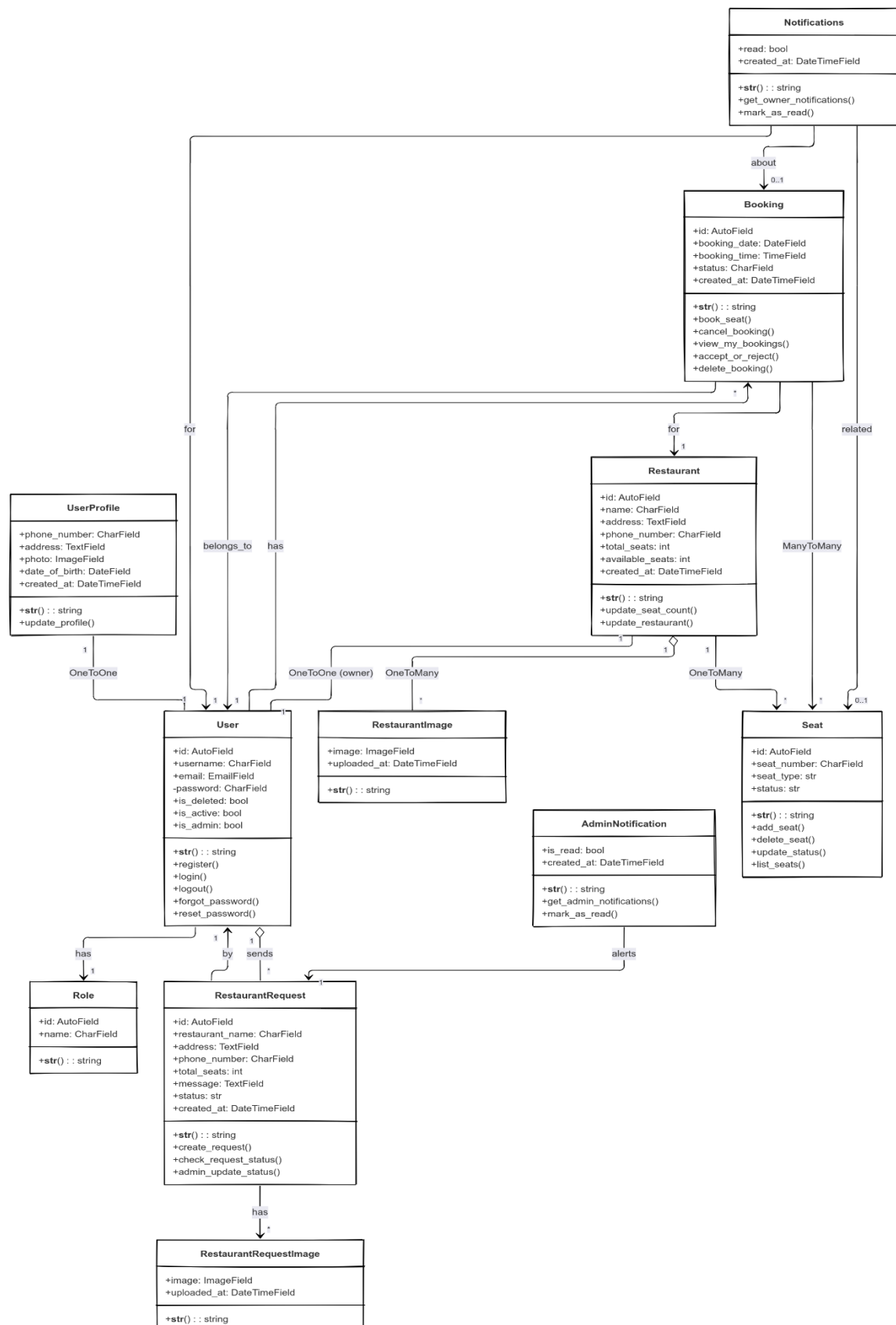


Figure 3.3 : Class Diagram

The class diagram for EatEase represents the structural design of the system, showing how different components interact. It defines key entities such as User, Restaurant, Booking, UserProfile, Seat and illustrates their relationships.

Key Relationships in the System

- **User & Role (1-to-1 Association):**
Each user is assigned a specific role (Admin, Customer, or Restaurant Owner). A role defines the user's permissions in the system.
- **User & UserProfile (1-to-1 Association):**
Each user has a profile that stores personal details like phone number, address, photo, and date of birth. This helps in personalizing user experience.
- **Restaurant Owner & Restaurant (1-to-Many Association):**
A restaurant owner can own multiple restaurants, but each restaurant is associated with only one owner.
- **User & Booking (1-to-Many Association):**
A user can book multiple tables at different restaurants, but each booking is linked to a single user.
- **Restaurant & Booking (1-to-Many Association):**
A restaurant can have multiple bookings, but each booking belongs to a specific restaurant.
- **Booking & Seat (Many-to-Many Association):**
Each booking can include multiple seats, and each seat can be booked by different users at different times.

This class diagram provides a structured overview of how users, restaurants, bookings, and seats interact in the **EatEase** system. It ensures clarity in managing restaurant seat availability, user roles, and the overall booking process

3.2.3 Sequence Diagram

A sequence diagram represents the flow of interactions between different components in a system over time.

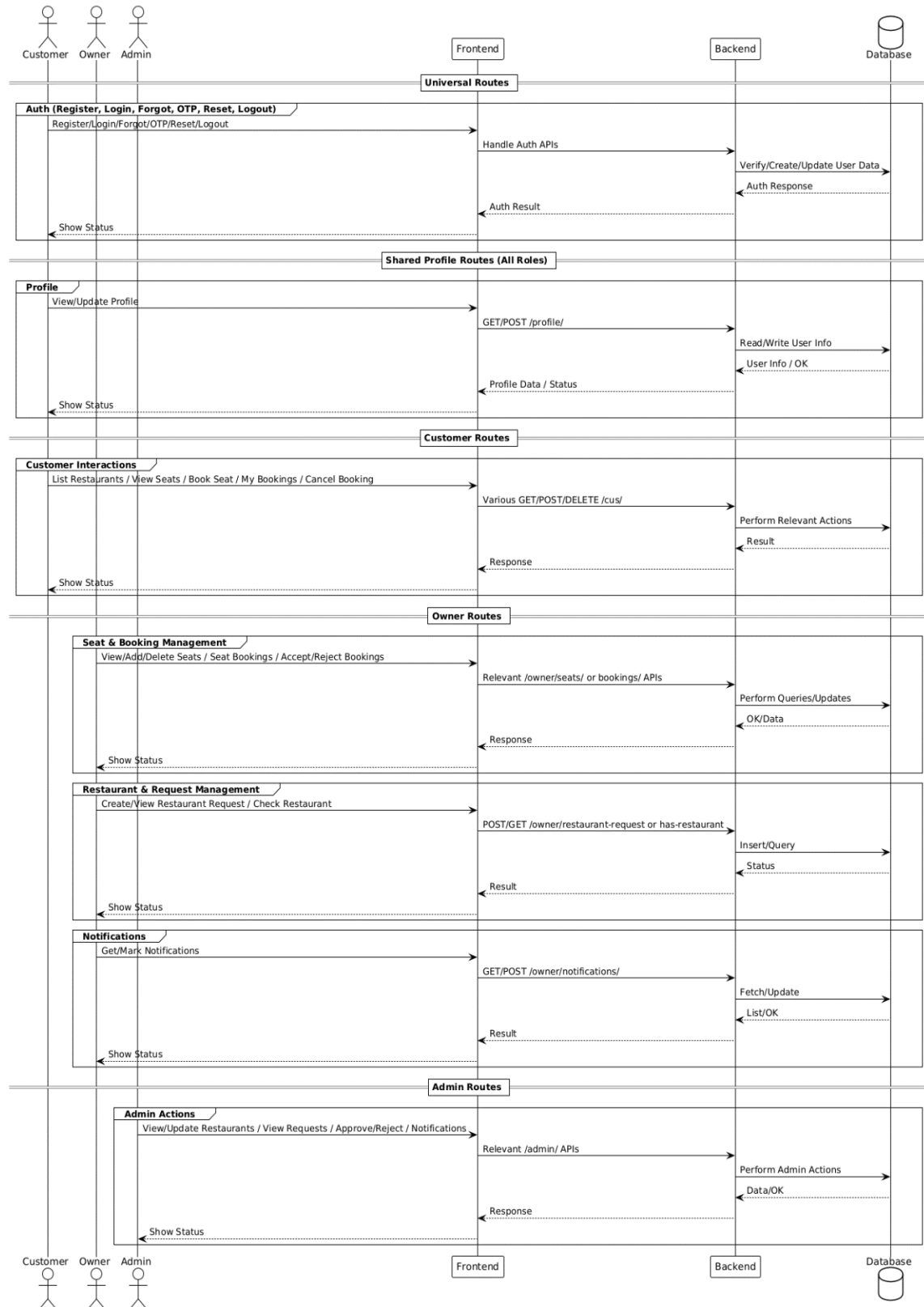


Figure 3.4: Sequence Diagram

3.2.4 Data Flow Diagram

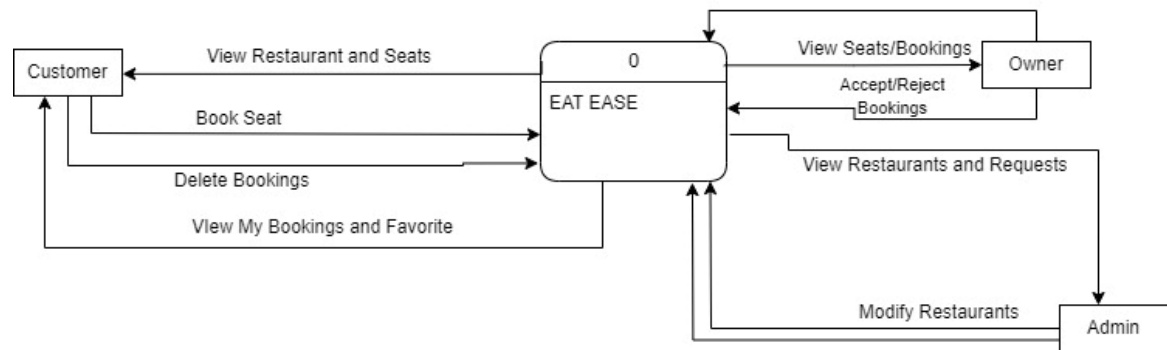


Figure 3.5: Context Level Diagram

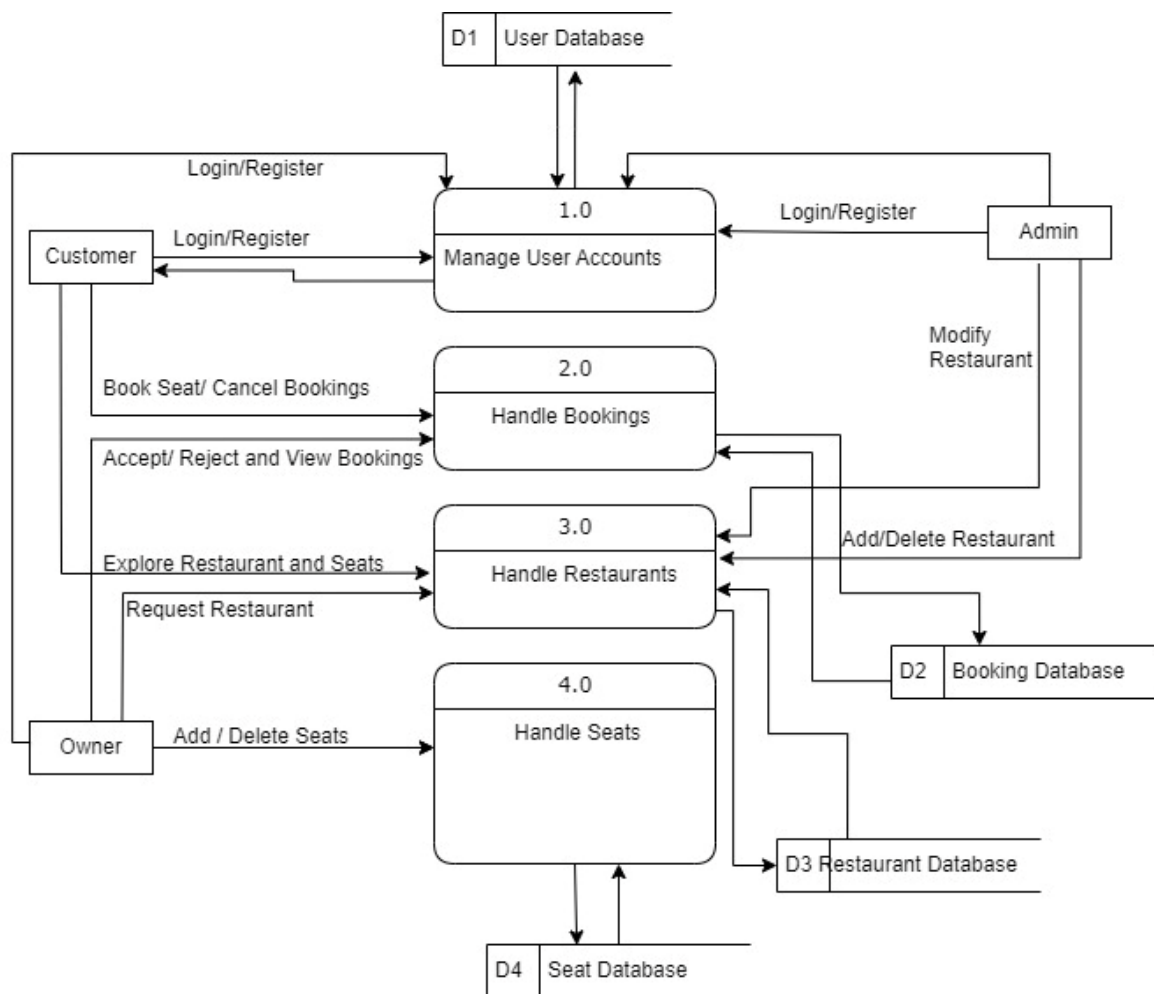


Figure 3.6: Level 0 DFD

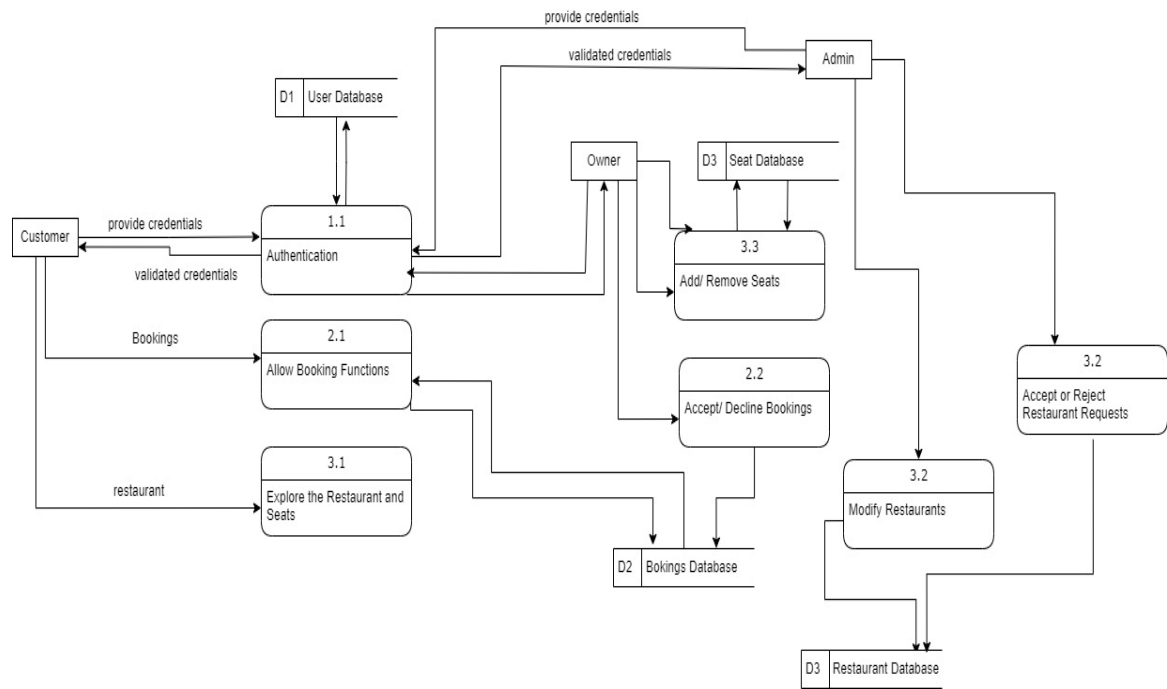


Figure 3.7: Level 1 DFD

3.2.5 Activity Diagram

An activity diagram visualizes the step-by-step workflow of a process, showing decisions, parallel executions, and transitions between activities. It helps in analyzing business logic and improving process efficiency. These diagrams are useful for modeling dynamic aspects of a system.

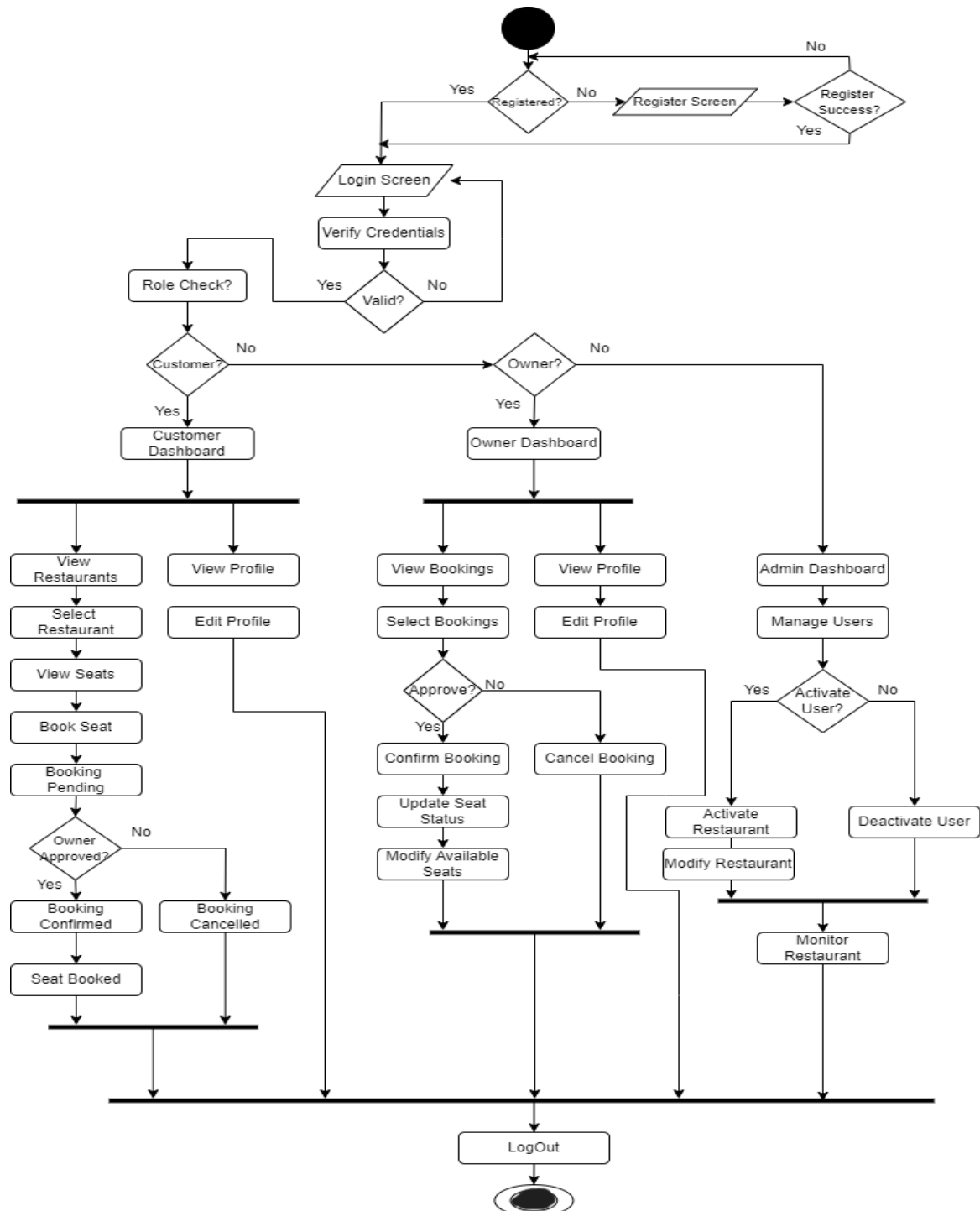


Figure 3.8: Activity Diagram

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

The implementation phase focused on building EatEase based on the structured system design created earlier. Various tools and technologies were utilized to ensure efficient development, testing, and deployment.

4.1.1 Tools Used

i. CASE Tools

- **Draw.io:**
Used for designing UML diagrams such as Use Case, Class, Sequence, Activity, and Deployment diagrams. These diagrams helped in visualizing the system architecture and workflow.
- **Git&GitHub:**
Version control and collaboration were managed through Git and GitHub. Git allowed efficient tracking of changes, while GitHub provided a platform for remote collaboration, backup, and code review.
- **Postman:**
Used for API testing of the Django backend. Postman ensured that APIs for restaurant browsing, booking, user authentication, and seat management responded correctly and securely before frontend integration.

ii. IDEs

- **Visual Studio Code:**
Used for frontend development with React. Lightweight and highly customizable, VS Code enabled faster development of responsive user interfaces with Tailwind CSS integration.
- **MySQLWorkbench:**
Utilized for database design, query management, and visualizing data relationships. It supported schema optimization and helped manage EatEase's database tables like Users, Restaurants, Bookings, and Seats efficiently.

iii. Programming Languages

- **Python:**

Backend development was done using Python with Django and Django REST Framework. Python provided robust server-side logic, secure authentication, and API handling with seamless integration with MySQL.

- **JavaScript(ReactJS):**

Frontend development used ReactJS for building dynamic, responsive, and component-driven interfaces, ensuring a smooth user experience for restaurant browsing, booking, and profile management.

iv. Database Platform

- **MySQL:**

All structured data related to users, restaurants, bookings, and seat management was stored in a MySQL database. Django simplified database operations by allowing database queries through Python models, improving development efficiency and security.

CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATION

5.1 Conclusion

Eat Ease simplifies restaurant management by streamlining seat reservations and improving coordination between customers and restaurant owners. Customers can easily find restaurants, book seats, and modify reservations, while owners can efficiently manage bookings and seat availability. By providing a well-structured system, Eat Ease reduces operational challenges, enhances the dining experience, and ensures smooth management for both users and restaurant owners.

5.2 Learner's Outcome

Developing this project helped me strengthen my skills in web application development, particularly in building a strong backend with Django and an interactive frontend with React and Tailwind CSS with MySQL database. I learned how to implement role-based access control, optimize database interactions, and ensure smooth user experience through responsive design. Additionally, tackling real-world challenges such as handling concurrent bookings, managing restaurant data, and improving user authentication provided valuable problem-solving experience.

5.3 Future Recommendation

To further improve the system, the following enhancements can be considered:

- **Flexible Seating Options:** Allow restaurant owners to define different seating arrangements, such as private dining areas or shared tables.
- **User Feedback & Ratings:** Enable customers to rate restaurants and leave reviews to help others make informed choices.
- **Enhanced Search & Filtering:** Implement filters based on cuisine, location, and available time slots for easier seat selection.
- **Mobile App Development:** Create a mobile application to make the system more accessible for customers and restaurant owners.
- **Automated Reminders:** Send notifications for upcoming reservations and status updates to improve communication.

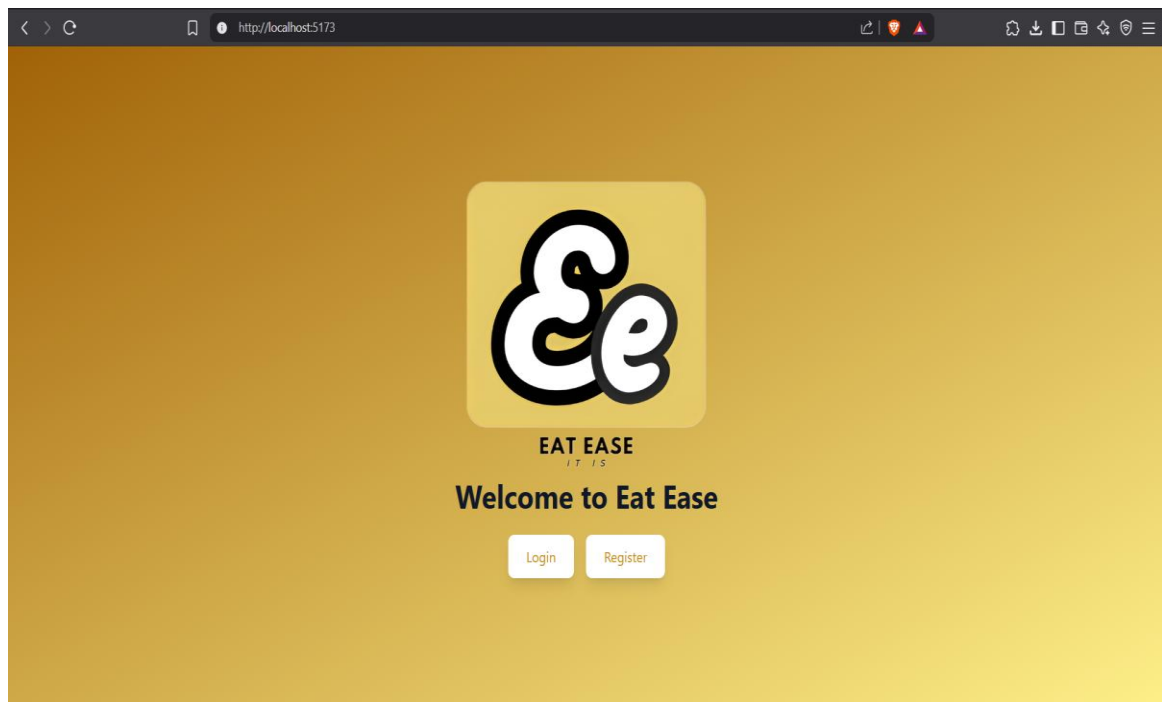
These improvements will enhance the overall efficiency of Eat Ease, providing a better experience for users and restaurant owners alike.

REFERENCES

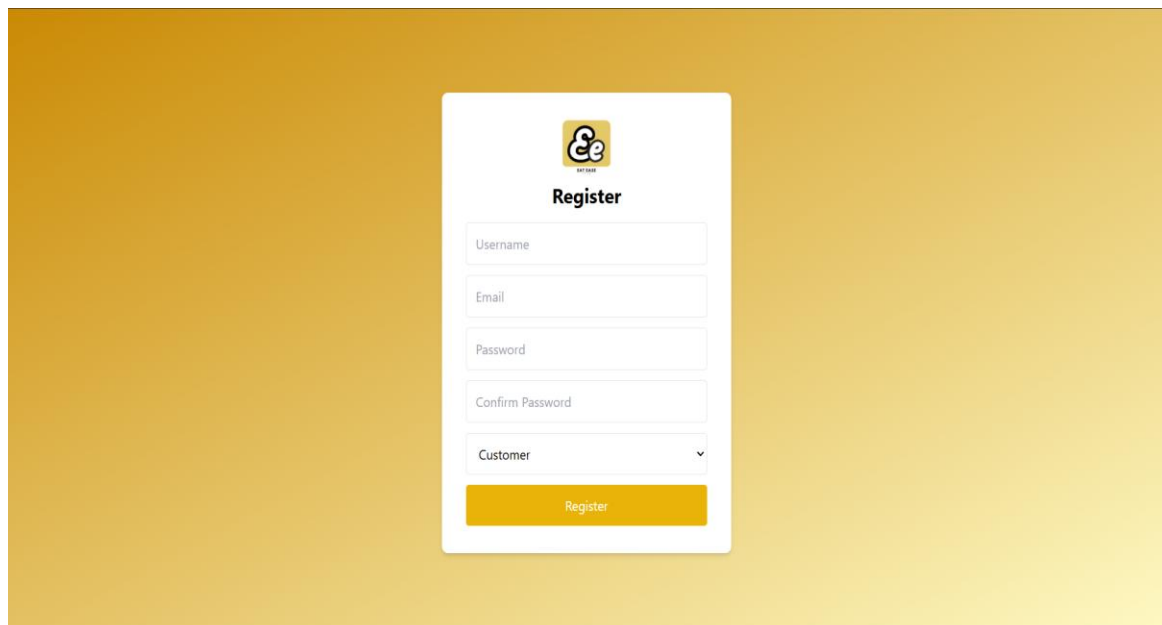
- [1] N. Patel, "Feasibility Analysis in Software Development: A Complete Guide," *HubSpot*, [Online]. Available: <https://blog.hubspot.com/feasibility-analysis>. [Accessed 5 March 2025].
- [2] W3Schools, "React.js Tutorial," *W3Schools*, 2024. [Online]. Available: <https://www.w3schools.com/react/>. [Accessed 20 December 2024].
- [3] Restronp, "All-in-One POS System for Nepali Restaurants," *Restronp*, [Online]. Available: <https://www.restronp.com/>. [Accessed 10 April 2025].
- [4] Atithi, "Hospitality Solutions for Restaurants," *Atithi Nepal*, [Online]. Available: <https://www.atithi.com.np/>. [Accessed 10 April 2025].

APPENDIX

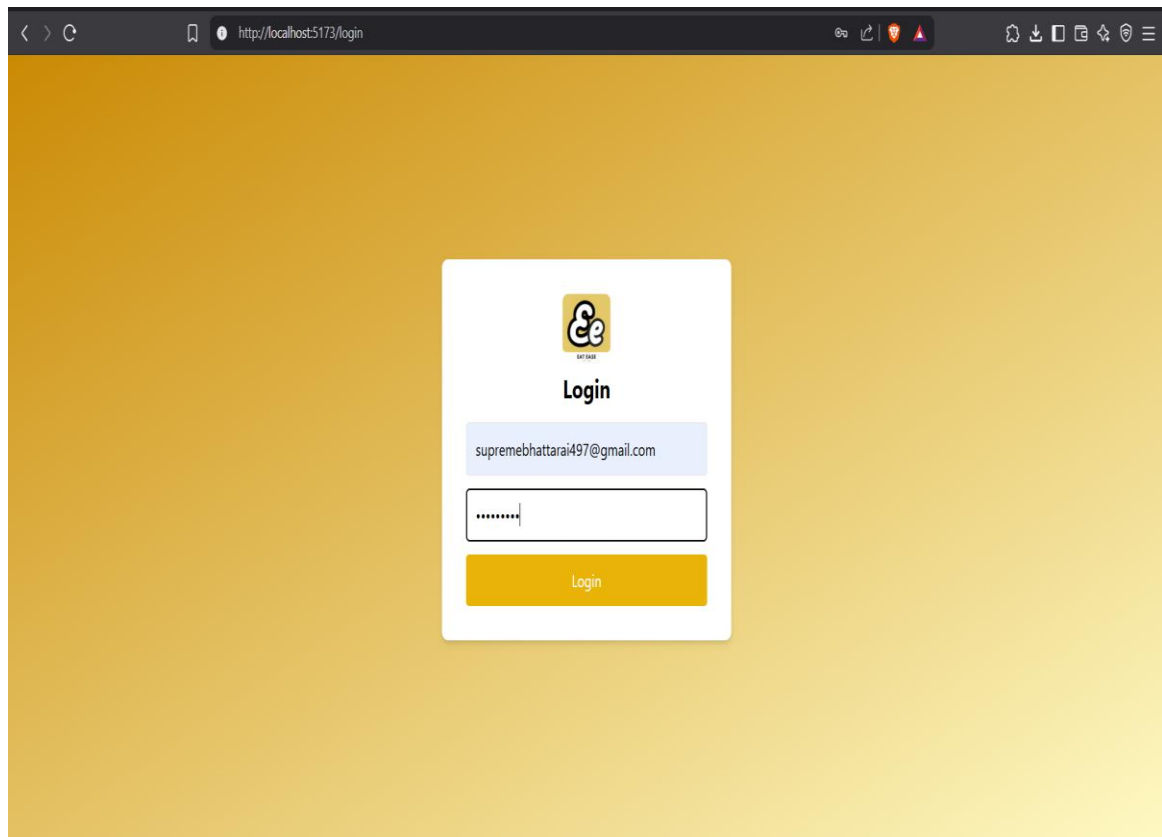
Snapshots:



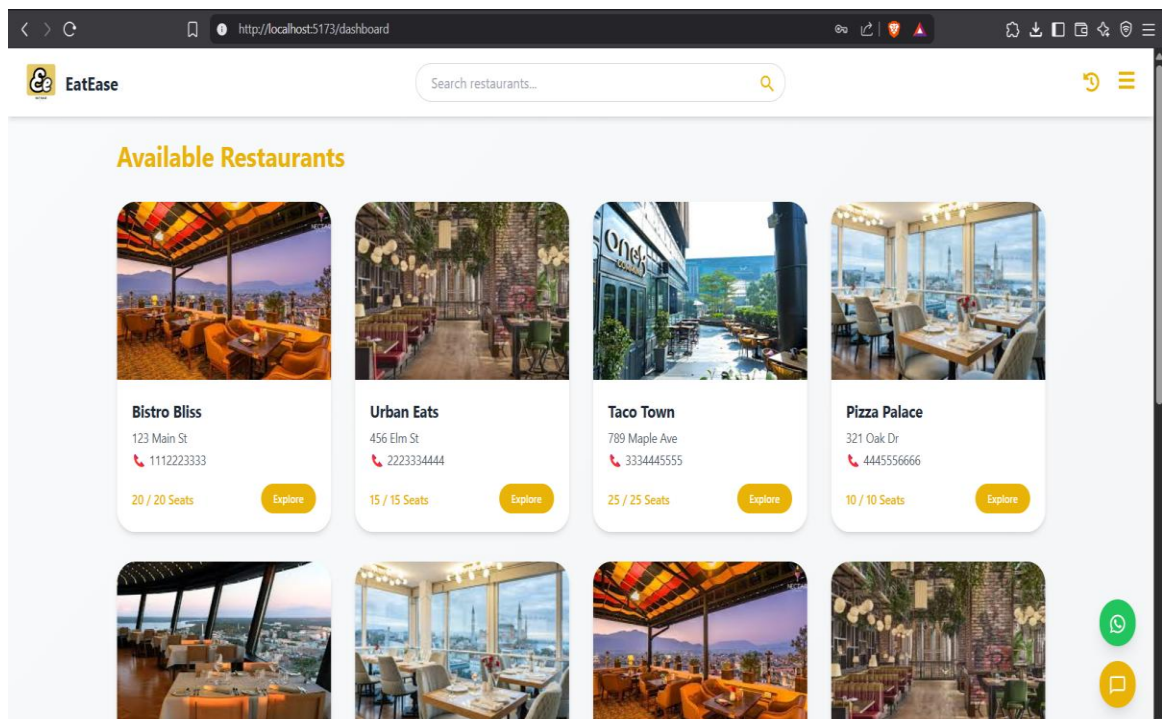
Snapshot 1: Opening Page



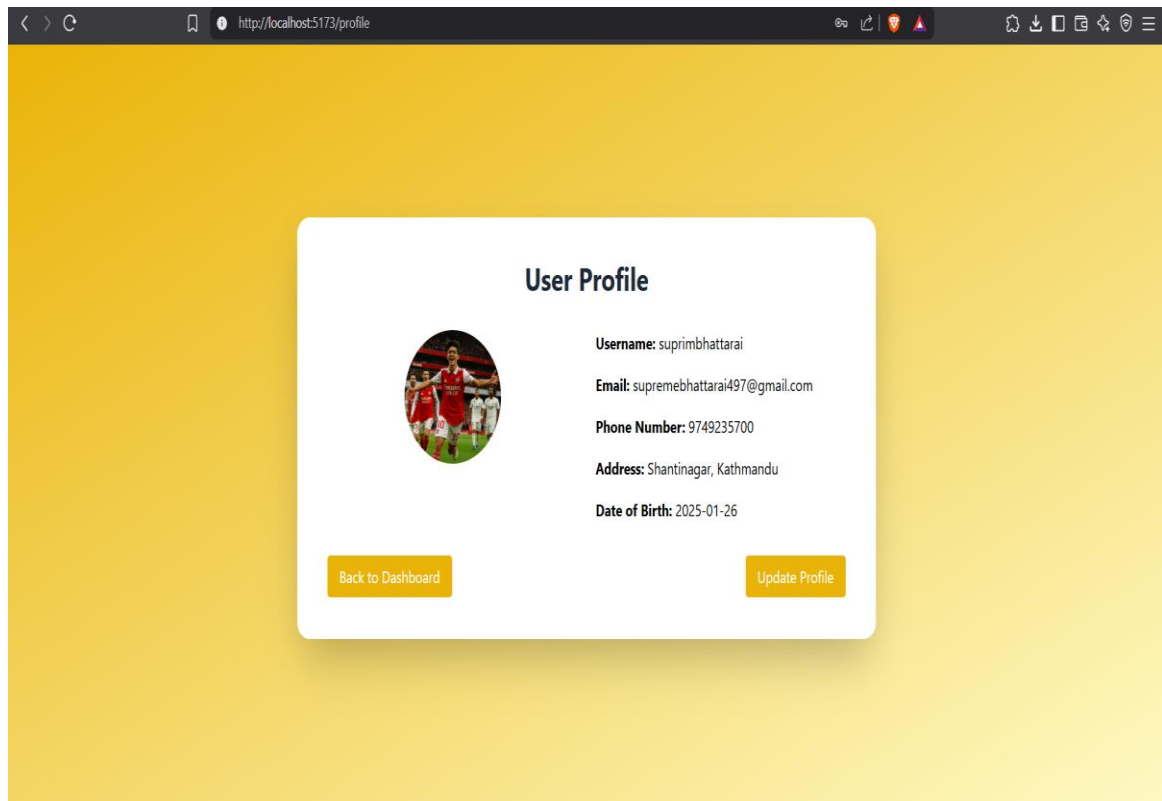
Snapshot 2: Register Page



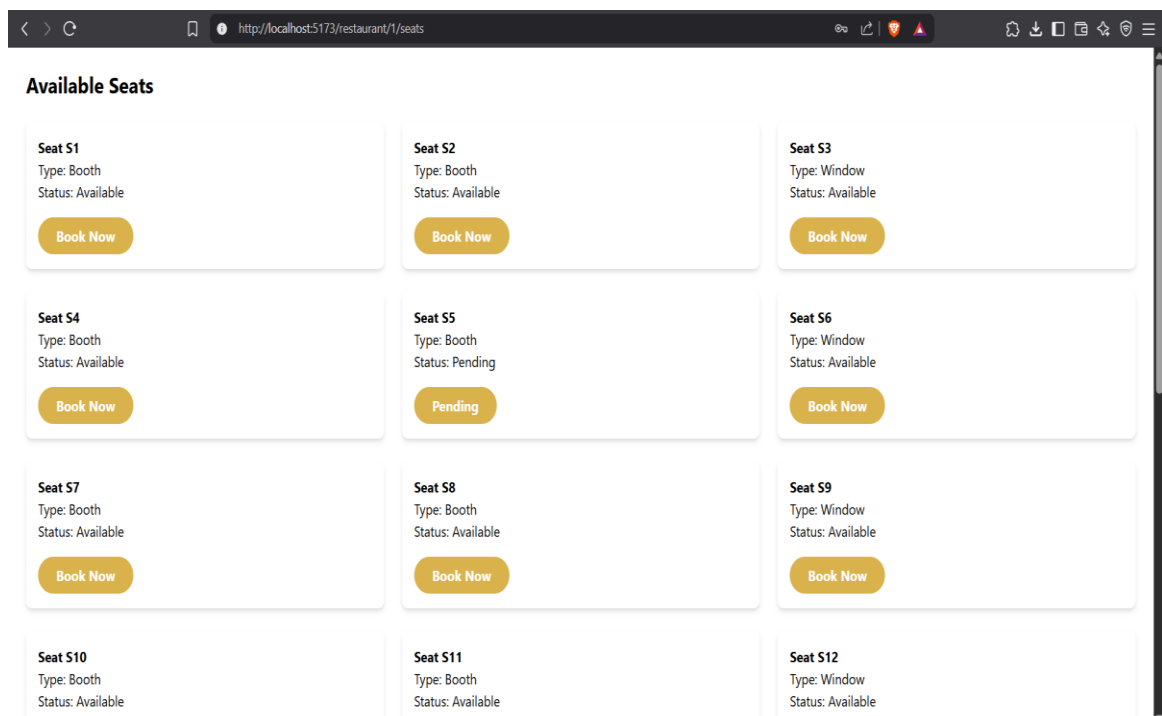
Snapshot 3: Login Page



Snapshot 4: Customer Dashboard



Snapshot 5: User Profile Page



Snapshot 6: Restaurant Seat Display Page