National University
of computer and emerging sciences

Final Project Report

Database Systems

Food Ordering Web App

Bushra Rais (22k-5151)

Dania ishaq (22k-4811)

BS(SE)5A

Department of Computer Science BS(SE)

FAST-NUCES KARACHI

ACKNOWLEDGEMENT

The successful presentation of the DBMS Mini Project would be incomplete without acknowledging the individuals whose constant guidance and support crowned my efforts with success.I would like to extend my gratitude to the MANAGEMENT of **FAST NUCES KARACHI**, for providing all the necessary facilities to successfully complete and present the DBMS Mini Project. I am deeply thankful to **Miss Ravia Ejaz**, Lab Instructor, for her invaluable guidance and assistance throughout the practical implementation of the project.

I also express my heartfelt thanks to **Miss Alina Arshad**, Theory Instructor, for her continuous support, encouragement, and insightful explanations that enhanced my understanding of the concepts.

Lastly, I extend my gratitude to all the teaching and non-teaching staff of the Department of Computer Science for their cooperation and assistance during the project.This acknowledgment stands as a testament to the collaborative effort and mentorship that contributed to the successful completion of my project.

ABSTRACT

The **Online Food Ordering Web Application** is a comprehensive platform designed to streamline the food ordering and delivery process by connecting customers, restaurants, and administrators. It enables users to browse restaurants, place orders, and make secure payments, while allowing restaurants to manage menus, process orders, and track inventory efficiently. Administrators oversee platform operations, ensuring smooth functionality, managing accounts, and maintaining service quality. This project aims to provide a user-friendly and scalable solution that enhances the convenience and efficiency of the food delivery ecosystem.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 OVERVIEW

The Online Food Ordering Web Application is designed to simplify and enhance the food delivery process by connecting customers, restaurants, and administrators through a single platform. The system provides users with a seamless experience for browsing restaurant menus, placing and tracking orders, and making payments. Restaurants can efficiently manage their menus, process orders, and track inventory, while administrators ensure smooth platform operations and maintain service quality. This project leverages modern web technologies and database management systems to deliver a reliable, scalable, and user-friendly solution for the food delivery industry.

1.2 PROBLEM STATEMENT

The primary goal of the Online Food Ordering Web Application is to provide a unified and efficient platform for customers, restaurants, and administrators. It addresses the challenges of manual food ordering processes by offering an intuitive interface that ensures smooth interactions, better inventory management, and enhanced service delivery. The project aims to eliminate inefficiencies in food ordering while improving user satisfaction and operational effectiveness.

1.3 DATABASE MANAGEMENT SYSTEM

A Database Management System (DBMS) is a crucial component of this project, enabling the storage, retrieval, and manipulation of data systematically. Oracle Database, known for its robust performance and security features, has been utilized for managing all database operations. SQL (Structured Query Language) is used to handle queries related to user accounts, restaurant menus, orders, and feedback. The DBMS provides essential functionalities such as concurrency, data integrity, and backup management, ensuring the smooth operation of the application. Oracle's advanced features and support for transactional operations make it an ideal choice for this project.

1.4 SQL

SQL is a standard language for storing, manipulating and retrieving data in databases. Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control.

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.[13]Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different database systems without adjustments.

1.5 HTML / CSS / Thymeleaf

The front-end of the web application is designed using HTML, CSS, and Thymeleaf.

- HTML provides the structural foundation of the web pages, enabling content presentation in a user-friendly manner.
- CSS is used to style the application, ensuring a visually appealing and responsive user interface.
- Thymeleaf, a Java-based template engine, facilitates the seamless integration of dynamic content within HTML. It works efficiently with Spring Boot, enabling server-side rendering and enhancing the overall user experience.

1.6 JAVA CONNECTIONS WITH SPRING BOOT AND JPA

The backend of the application is developed using Java Spring Boot, a powerful framework for creating enterprise-level applications. Java Persistence API (JPA) is utilized for database connectivity and object-relational mapping. JPA simplifies the interaction between Java objects and database tables, allowing developers to perform CRUD (Create, Read, Update, Delete) operations with minimal boilerplate code.

Spring Boot's integration with JPA ensures efficient database operations while promoting scalability and maintainability. The framework's robust architecture and built-in support for dependency injection enhance the development process, making it the ideal choice for this project.

By combining Oracle for the database, Spring Boot for backend processing, and Thymeleaf for frontend rendering, the Online Food Ordering Web Application delivers a cohesive and efficient solution that meets the demands of modern web applications.

Chapter 2

REQUIREMENTS SPECIFICATION

A computerized way of handling food ordering and restaurant management through an integrated platform is efficient, organized, and time-saving compared to traditional manual methods. This is achieved through a database-driven web application, and its requirements are outlined in this chapter.

2.1 OVERALL DESCRIPTION

A robust, scalable, and secure database-driven web application is required to efficiently manage users, restaurants, and orders. The system must be user-friendly, maintainable, and capable of handling large-scale operations with ease.

2.2 SPECIFIC REQUIREMENTS

2.2.1 SOFTWARE REQUIREMENTS

- **IDE:** IntelliJ IDEA (ultimate version),
- **Web Browser:** Google Chrome
- **Database Support:** Oracle Database
    - Oracle SQL Developer for database management
- **Backend Framework:** Spring Boot
    - Includes integration with Java Persistence API (JPA) for database connectivity
- **Frontend Technologies:**
    - **HTML5** for structure
    - **CSS** for styling and responsive design
    - **Thymeleaf** for dynamic rendering and seamless integration with Spring Boot
- **Operating System:** Windows 10

- **JDK:** Java Development Kit (JDK 11 or later)
- **Server Deployment:** Embedded Tomcat (Spring Boot default server)

2.2.2 HARDWARE REQUIREMENTS

- **Processor:** Intel Core i3 or above
- **RAM:** 4 GB or more
- **Hard Disk Space:** 5 GB or more
- **Monitor:** VGA or HD resolution (1280x720 or higher)
- **Keyboard and Mouse:** Standard input devices

2.2.3 TECHNOLOGY STACK

- HTML5: Used to structure the content and layout of the web pages, providing a robust base for user interactions.

- CSS: Adds style and responsiveness to the user interface, ensuring a consistent and visually appealing design.

- Thymeleaf: A server-side Java templating engine that allows dynamic rendering of HTML, seamlessly integrating with Spring Boot for better design and functionality.

- SQL: The query language used to manage the Oracle database, enabling data manipulation such as insertion, updates, and retrieval for users, restaurants, and orders.

- Spring Boot: A powerful Java-based framework for developing the backend. It simplifies the configuration and development process, offering a production-ready environment.

- Java Persistence API (JPA): Provides an abstraction layer for database interactions, allowing developers to perform CRUD operations with ease and efficiency.

Chapter 3

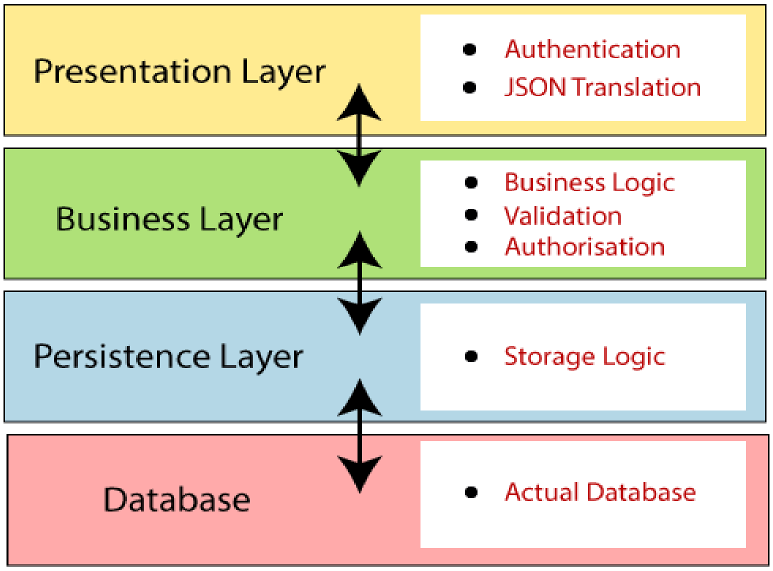
3. DETAILED DESIGN

3.1 SYSTEM DESIGN

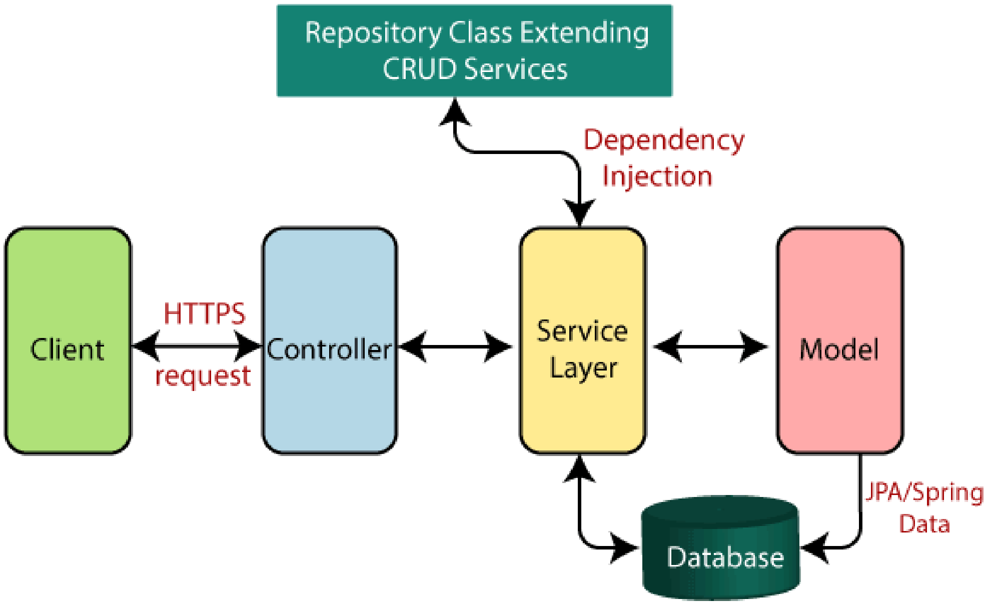
Spring Boot Architecture

Spring Boot is a module of the Spring Framework that simplifies the development of stand-alone, production-grade applications. It follows a layered architecture, with each layer handling specific tasks. The primary layers in a Spring Boot application are:

1. **Presentation Layer:**
   This layer handles HTTP requests and responses. It interacts with the user interface, processes input data, and communicates with the business layer. In the Online Food Ordering System, this could involve handling user requests such as placing an order, viewing available restaurants, or managing their profile.
2. **Business Layer:**
   The business layer contains the logic that processes the data, applies rules, and ensures the proper functioning of the system. It includes service classes that manage tasks such as order processing, user management, and restaurant operations. It communicates with the persistence layer to retrieve or store data.
3. **Persistence Layer:**
   The persistence layer handles the storage and retrieval of data. It interacts with the database using JPA (Java Persistence API) or Spring Data JPA, converting business objects into database rows and vice versa. For example, it would handle storing user orders, restaurant information, and payment details.
4. **Database Layer:**
   The database layer performs CRUD operations (Create, Retrieve, Update, Delete) on the data stored in the database. In this system, data is stored in an Oracle database. CRUD operations ensure that the system can handle operations like placing an order, updating the menu, or viewing past orders.



Spring Boot flow architecture



3.2 ENTITY RELATIONSHIP DIAGRAM

An entity-relationship (E-R) model is a graphical representation used to define and describe the important data processes within a business area. It presents a business data schema through boxes (entities) connected by lines (relationships) that express associations and dependencies between entities. Entities are characterized by additional properties called attributes, which include identifiers known as primary keys. These diagrams may also be referred to as entity-attribute-relationship diagrams. Typically, an E-R model is implemented as a relational database, where each row in a table represents an entity instance and each field represents an attribute. Relationships between entities are implemented using foreign keys. The E-R model can be represented at different levels of abstraction, such as conceptual, logical, and physical, though it is not well-suited for semi-structured or unstructured data. Cardinality notations in an E-R diagram define the attributes of the relationships, indicating if an entity is optional in the relationship.

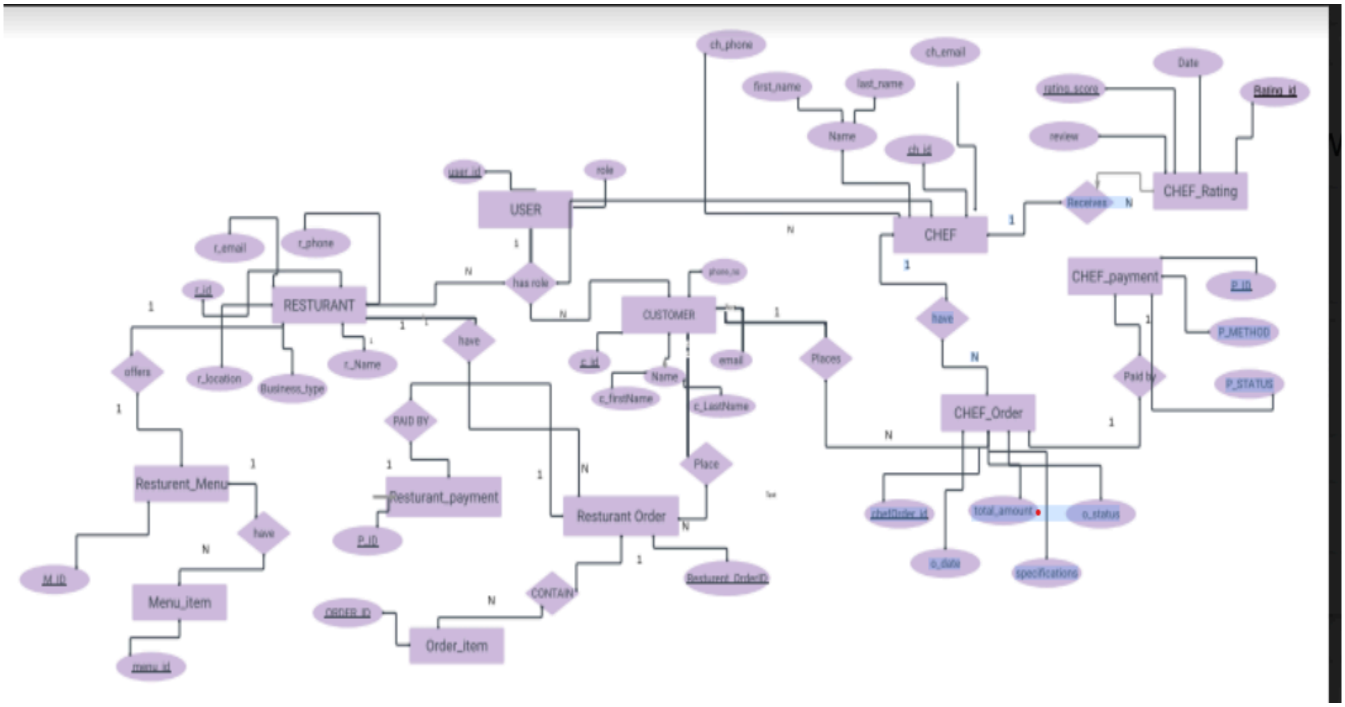
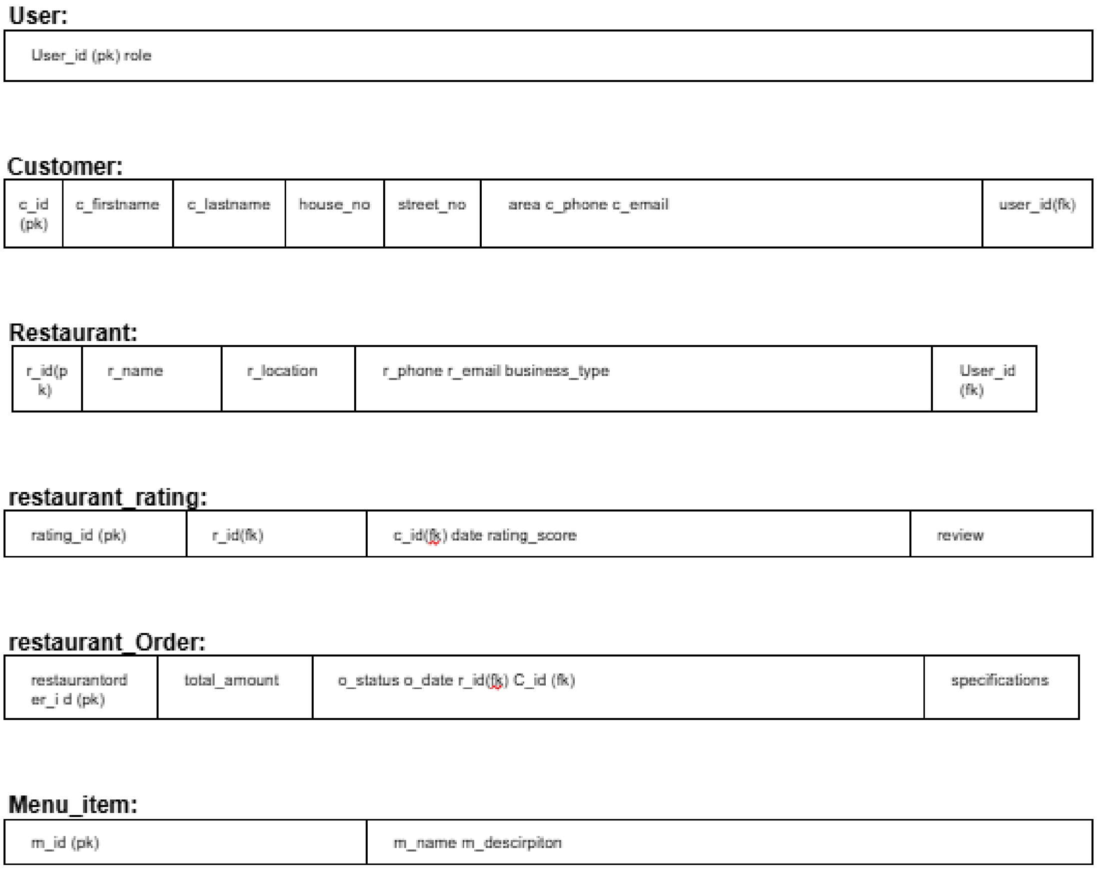


Fig : detailed ERD REPRESENTATION OF FOOD ORDERING APP

3.3 RELATIONAL SCHEMA

The term "schema" refers to the organization of data as a blueprint of how the database is constructed. The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database. A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute. A schema diagram helps organize values in the database. The following diagram shows the schema diagram for the database.



User:

User_id (pk) role

Customer:

c_id (pk)	c_firstname	c_lastname	house_no	street_no	area c_phone c_email	user_id(fk)

Restaurant:

r_id(pk)	r_name	r_location	r_phone r_email business_type	User_id (fk)

restaurant_rating:

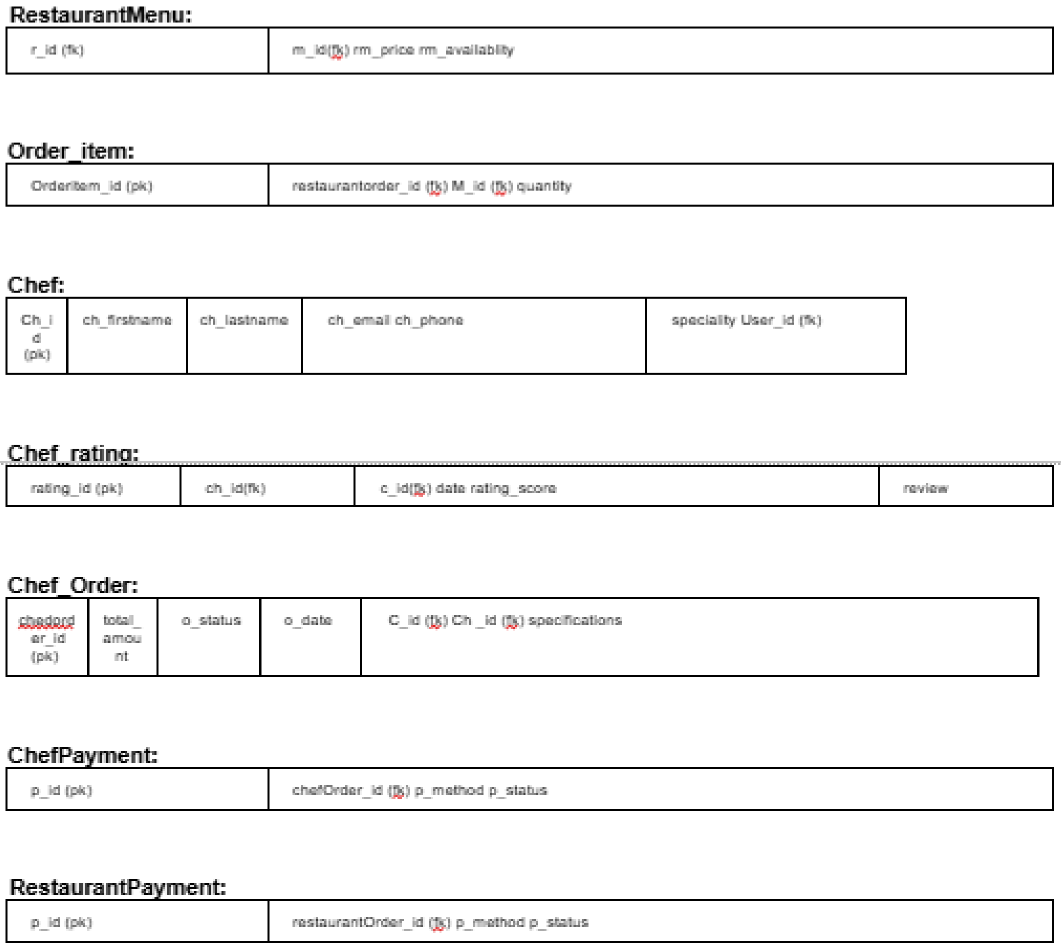
rating_id (pk)	r_id(fk)	c_id(fk) date rating_score	review

restaurant_Order:

restaurantorder_i d (pk)	total_amount	o_status o_date r_id(fk) C_id (fk)	specifications

Menu_item:

m_id (pk)	m_name m_descirpiton



RestaurantMenu:

r_id (fk)	m_id(fk) rm_price rm_availablity

Order_item:

Orderitem_id (pk)	restaurantorder_id (fk) M_id (fk) quantity

Chef:

Ch_id (pk)	ch_firstname	ch_lastname	ch_email ch_phone	speciality User_id (fk)

Chef_rating:

rating_id (pk)	ch_id(fk)	c_id(fk) date rating_score	review

Chef_Order:

chedorder_id (pk)	total_amount	o_status	o_date	C_id (fk) Ch _id (fk) specifications

ChefPayment:

p_id (pk)	chefOrder_id (fk) p_method p_status

RestaurantPayment:

p_id (pk)	restaurantOrder_id (fk) p_method p_status

Above is the final schema for the online food ordering system

3.4 DESCRIPTION OF TABLES

1. **User Table:**
   - user_id (PK): A unique identifier for each user in the system.
   - role: The role assigned to the user, such as customer, restaurant owner, or chef.
2. **Customer Table:**
   - c_id (PK): A unique identifier for each customer.
   - c_firstname: The first name of the customer.
   - c_lastname: The last name of the customer.
   - house_no: The house number of the customer's address.
   - street_no: The street number of the customer's address.
   - area: The area of the customer's location.
   - c_phone: The customer's phone number.
   - c_email: The customer's email address.
   - user_id (FK): A foreign key linking to the User table, identifying the user related to the customer.
3. **Restaurant Table:**
   - r_id (PK): A unique identifier for each restaurant.
   - r_name: The name of the restaurant.
   - r_location: The location of the restaurant.
   - r_phone: The restaurant's contact phone number.
   - r_email: The restaurant's contact email address.
   - business_type: The type of business, such as a fast food restaurant, fine dining, etc.
   - user_id (FK): A foreign key linking to the User table, identifying the user associated with the restaurant.
4. **Restaurant Rating Table:**
   - rating_id (PK): A unique identifier for each rating.
   - r_id (FK): A foreign key linking to the Restaurant table, identifying the rated restaurant.
   - c_id (FK): A foreign key linking to the Customer table, identifying the customer who provided the rating.
   - date: The date when the rating was given.
   - rating_score: The score given by the customer for the restaurant.
   - review: The written review provided by the customer for the restaurant.
5. **Restaurant Order Table:**
   - restaurantorder_id (PK): A unique identifier for each restaurant order.
   - total_amount: The total amount of the order.

- o_status: The status of the order (e.g., pending, completed, cancelled).
- o_date: The date when the order was placed.
- r_id (FK): A foreign key linking to the Restaurant table, identifying the restaurant from which the order was placed.
- c_id (FK): A foreign key linking to the Customer table, identifying the customer who placed the order.
- specifications: Any additional specifications or requests for the order.

6. **Menu Item Table:**
   - m_id (PK): A unique identifier for each menu item.
   - m_name: The name of the menu item.
   - m_description: A description of the menu item.
7. **Restaurant Menu Table:**
   - r_id (FK): A foreign key linking to the Restaurant table, identifying the restaurant offering the menu item.
   - m_id (FK): A foreign key linking to the Menu Item table, identifying the menu item.
   - rm_price: The price of the menu item in the restaurant.
   - rm_availability: A field indicating whether the menu item is available in the restaurant.
8. **Order Item Table:**
   - orderitem_id (PK): A unique identifier for each order item.
   - restaurantorder_id (FK): A foreign key linking to the Restaurant Order table, identifying the order to which this item belongs.
   - m_id (FK): A foreign key linking to the Menu Item table, identifying the menu item in the order.
   - quantity: The quantity of the menu item ordered.
9. **Chef Table:**
   - ch_id (PK): A unique identifier for each chef.
   - ch_firstname: The first name of the chef.
   - ch_lastname: The last name of the chef.
   - ch_email: The email address of the chef.
   - ch_phone: The phone number of the chef.
   - speciality: The area of expertise or specialty of the chef.
   - user_id (FK): A foreign key linking to the User table, identifying the user associated with the chef.
10. **Chef Rating Table:**
    - rating_id (PK): A unique identifier for each chef rating.
    - ch_id (FK): A foreign key linking to the Chef table, identifying the rated chef.
    - c_id (FK): A foreign key linking to the Customer table, identifying the customer who gave the rating.

- date: The date when the rating was given.
- rating_score: The score given by the customer for the chef.
- review: The written review provided by the customer for the chef.

11. **Chef Order Table:**
    - cheforder_id (PK): A unique identifier for each chef order.
    - total_amount: The total amount of the chef's order.
    - o_status: The status of the order (e.g., pending, completed, cancelled).
    - o_date: The date when the chef order was placed.
    - c_id (FK): A foreign key linking to the Customer table, identifying the customer who placed the chef order.
    - ch_id (FK): A foreign key linking to the Chef table, identifying the chef assigned to the order.
    - specifications: Any additional specifications or requests for the chef order.
12. **Chef Payment Table:**
    - p_id (PK): A unique identifier for each chef payment.
    - chefOrder_id (FK): A foreign key linking to the Chef Order table, identifying the payment related to a chef order.
    - p_method: The method of payment (e.g., cash, credit card).
    - p_status: The status of the payment (e.g., pending, completed).
13. **Restaurant Payment Table:**
    - p_id (PK): A unique identifier for each restaurant payment.
    - restaurantOrder_id (FK): A foreign key linking to the Restaurant Order table, identifying the payment related to a restaurant order.
    - p_method: The method of payment (e.g., cash, credit card).
    - p_status: The status of the payment (e.g., pending, completed).

Each of these tables helps manage the different entities and relationships involved in the food ordering system, including customer orders, restaurant information, menu items, chef assignments, and payments. The foreign key constraints ensure the integrity of the relationships between these entities.

Chapter 4

IMPLEMENTATION

4.1 MODULES AND THEIR ROLES

Module 1: Account Type Selection

- Role: Facilitates the user's choice of account type (Personal or Business).
- Key Features:
  - Display an initial page where users select their account type.
  - Direct users to the appropriate login or sign-up process based on their selection.

Module 2: Personal Account Functions

Submodule 2.1: Personal Account Login/Sign-Up

- Role: Allows users to register or log into their personal accounts.
- Sign-Up Workflow:
  - Collect user details such as name, email, phone number, password, etc.
  - Validate the information and create a user account in the database.
- Login Workflow:
  - Validate user credentials against stored data.
  - Grant access to the personal account dashboard on successful authentication.

Submodule 2.2: Placing an Order

- Role: Enables personal account users to place orders.
- Key Features:
  - Restaurant Option:
    - Display a list of available restaurants.
    - Provide an interface to select food items, confirm orders, and submit ratings/reviews.

    - Chef Option:
        - Provide a text box for users to describe custom food requests.
        - Allow users to rate and review chefs based on their experience.

Module 3: Business Account Functions

Submodule 3.1: Business Account Login/Sign-Up

- Role: Supports registration and login for business accounts (Restaurant or Chef).
- Sign-Up Workflow:
    - Display a profile creation page for entering business details (e.g., restaurant name, chef specialization, location).
    - Validate and store the information to create a business profile.
- Login Workflow:
    - Validate business credentials and grant access to the relevant dashboard (Restaurant or Chef).

Submodule 3.2: Restaurant-Specific Functions

- Role: Provides restaurants with tools to manage orders and menus.
- Key Features:
    - Manage Orders:
        - Display pending orders with options to Accept or Reject.
    - Menu Management:
        - Enable restaurants to mark food items as Available/Unavailable.
        - Allow price updates for listed items.

Submodule 3.3: Chef-Specific Functions

- Role: Offers chefs functionality to manage custom orders.
- Key Features:
    - Manage Orders:
        - List pending orders with options to Accept or Reject.
        - Allow chefs to review customer requests and respond accordingly.

Chapter 5

TESTING

5.1 SOFTWARE TESTING

Software testing is the process of ensuring that the software meets its requirements in terms of functionality, quality, and security. It is conducted to identify errors, verify correctness, and evaluate software quality. Testing can be initiated as soon as executable components are available, even if they are incomplete.

For this project, testing focused on verifying:

- Correctness of user account creation and login for both Personal and Business accounts.
- Accuracy of order placement and management.
- Validation of ratings, reviews, and menu availability functionality.

5.2 MODULE TESTING AND INTEGRATION

Module Testing

Each module, such as account management, order placement, and menu handling, was tested individually to identify errors and ensure functionality.

- Personal Account Testing:
    - Verified the login and sign-up process.
    - Tested order placement for both restaurant and chef options, including ratings and reviews.
- Business Account Testing:
    - For restaurants:
        - Tested menu management for marking items as available/unavailable and updating prices.
        - Validated order acceptance/rejection processes.
    - For chefs:

        - Verified the handling of custom order requests and acceptance/rejection workflows.

Integration Testing

Once modules were verified, integration testing ensured that they worked cohesively. Specific tests included:

- Seamless communication between the user and restaurant/chef modules for order management.
- Correct handling of user reviews and ratings across the platform.
- Validation of the flow between module functionalities, such as transitioning from order placement to order management.

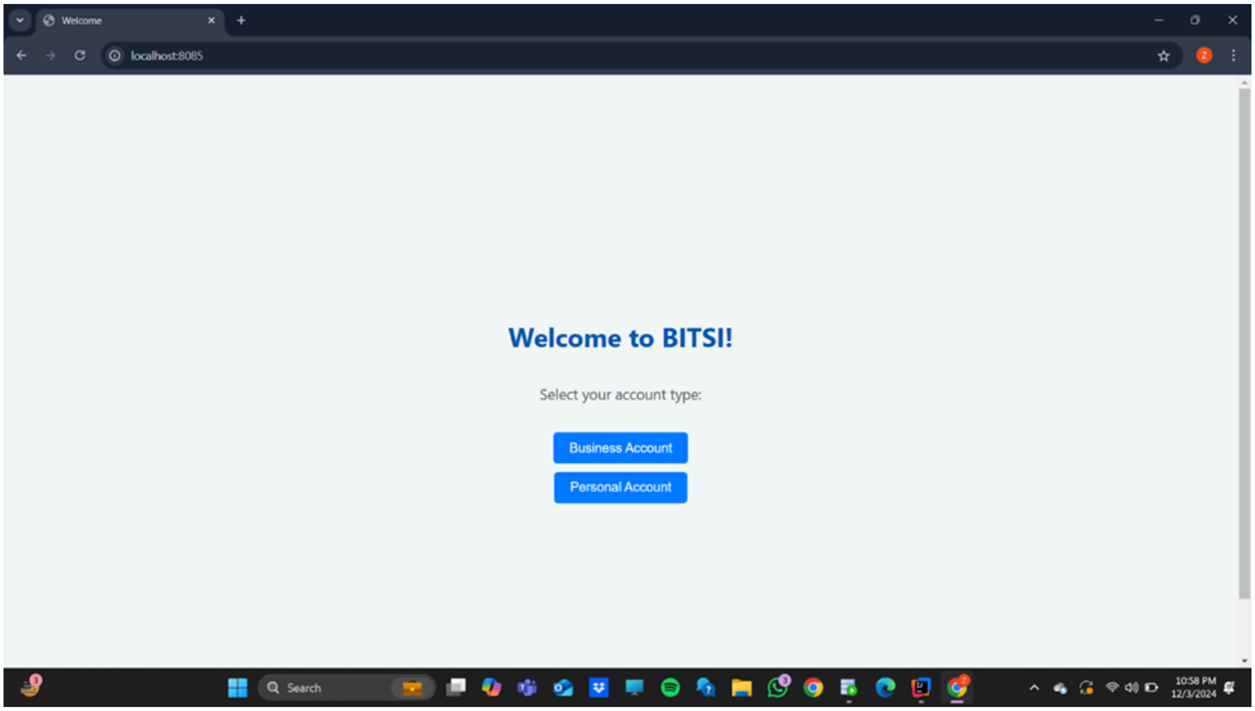
5.3 LIMITATIONS

1. Menu and Payment Feature:
    - Due to the one-to-many relationship between the Restaurant and Menu entities, a composite key is required.
    - Spring Boot does not directly support composite keys, causing issues in managing menu items and payments.
    - This limitation has delayed the implementation of menu updates and payment functionalities.
2. Scalability Challenges:
    - Current testing has been limited to basic functionalities. Scalability and performance tests are pending to ensure smooth operation with multiple concurrent users.

Chapter 6

SNAPSHOTS

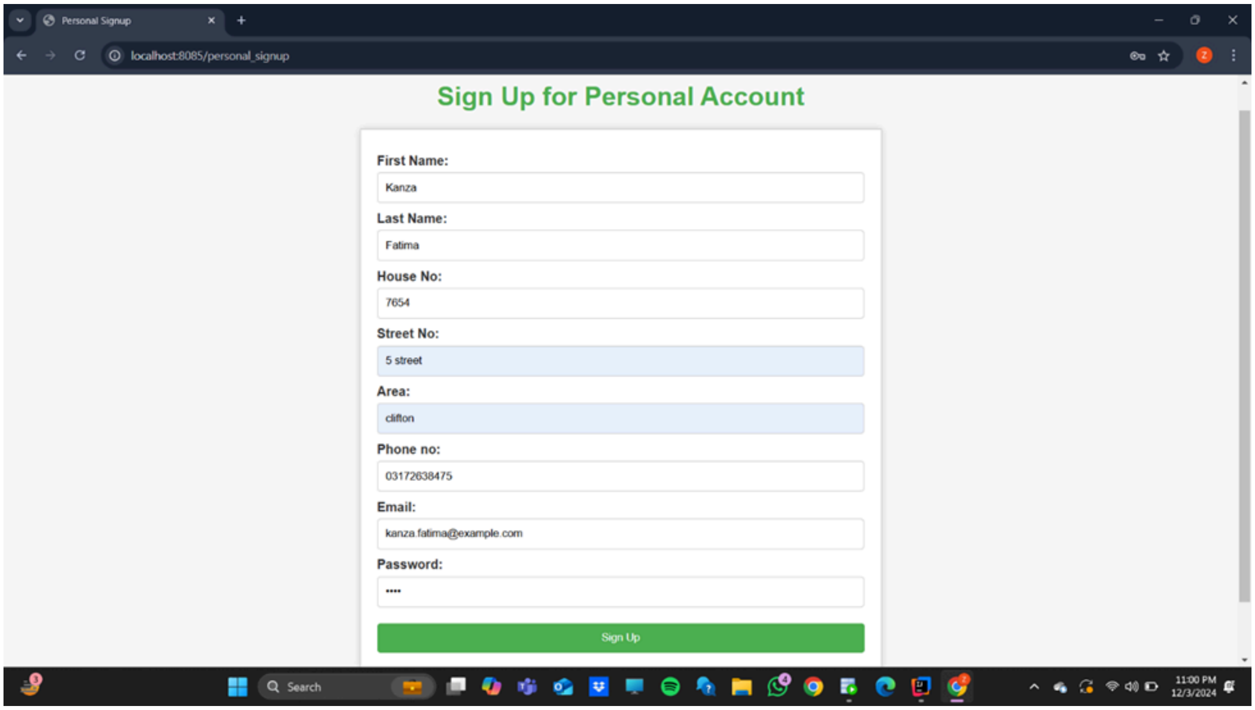
6.1 WELCOME PAGE



Welcome to BITSI!

Select your account type:

6.2 SIGN UP



Sign Up for Personal Account

First Name: Kanza

Last Name: Fatima

House No: 7654

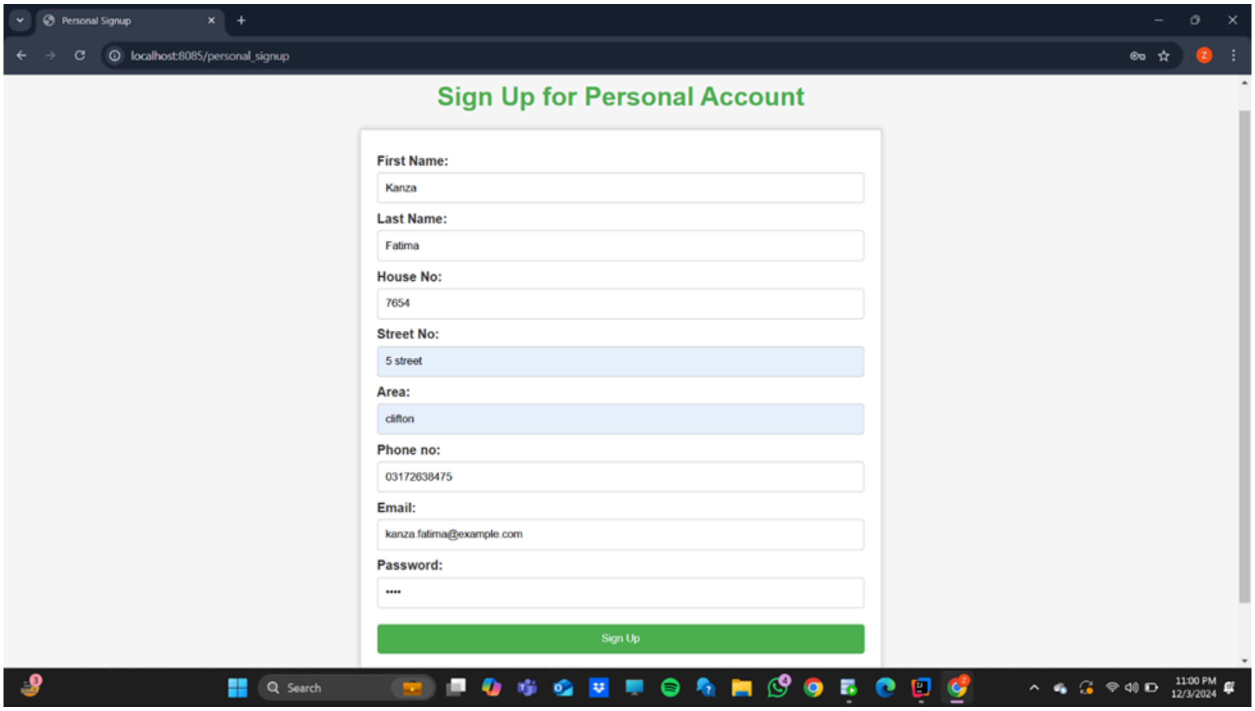
Street No: 5 street

Area: clifton

Phone no: 03172638475

Email: kanza.fatima@example.com

Password: ••••



Sign Up for Personal Account

First Name: Kanza

Last Name: Fatima

House No: 7654

Street No: 5 street

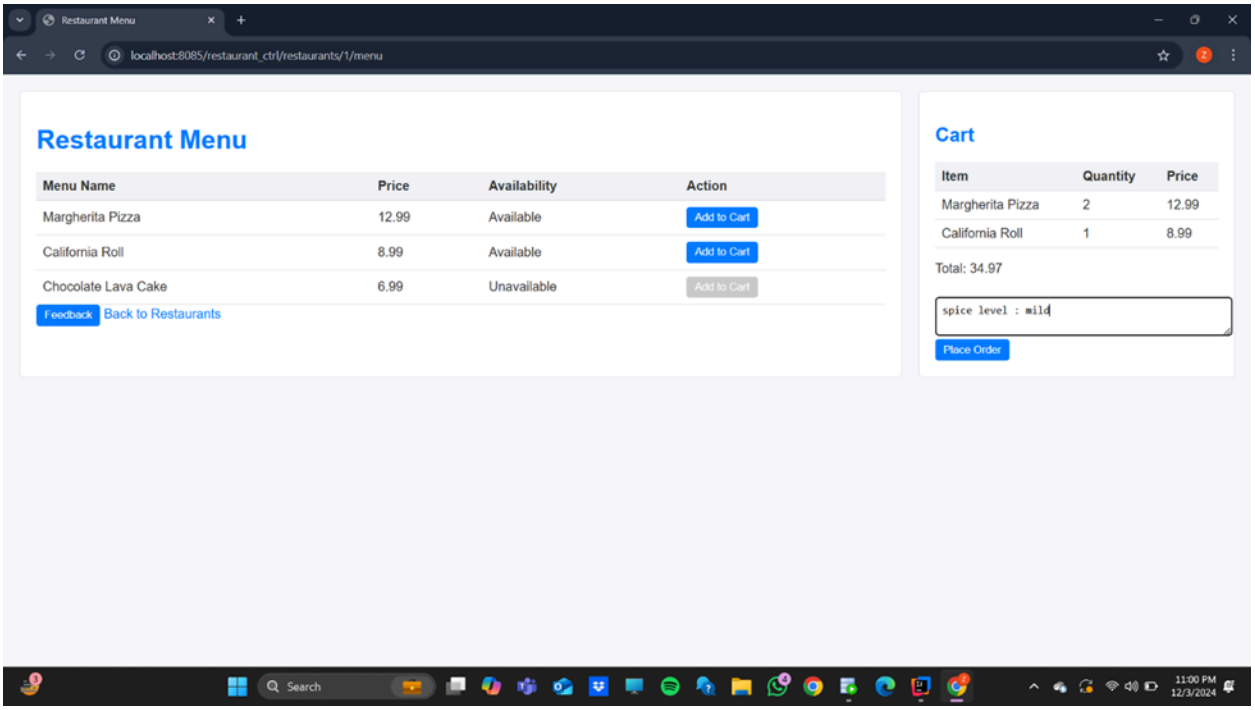
Area: clifton

Phone no: 03172638475

Email: kanza.fatima@example.com

Password: ••••

6.3 RESTAURANT MENU PAGES



Restaurant Menu

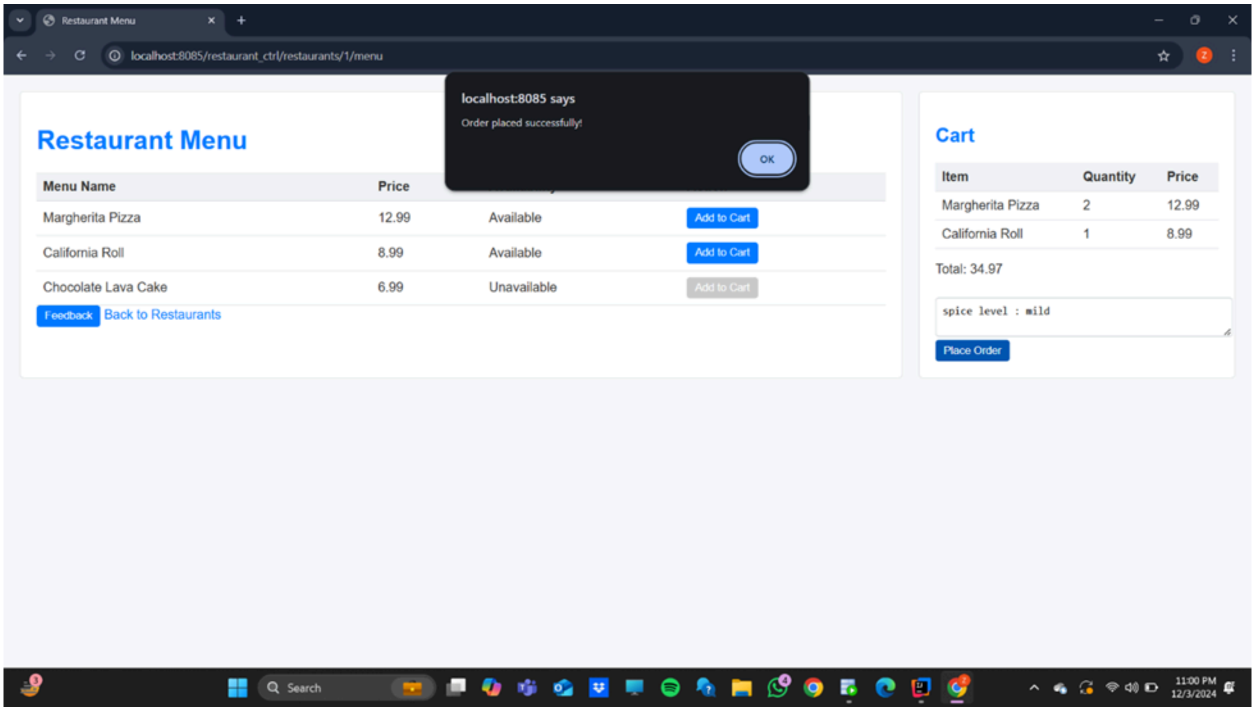
Menu Name	Price	Availability	Action
Margherita Pizza	12.99	Available	Add to Cart
California Roll	8.99	Available	Add to Cart
Chocolate Lava Cake	6.99	Unavailable	Add to Cart

Cart

Item	Quantity	Price
Margherita Pizza	2	12.99
California Roll	1	8.99

Total: 34.97

```
spice level : mild
```



localhost:8085 says

Order placed successfully!

Restaurant Menu

Menu Name	Price		
Margherita Pizza	12.99	Available	Add to Cart
California Roll	8.99	Available	Add to Cart
Chocolate Lava Cake	6.99	Unavailable	Add to Cart

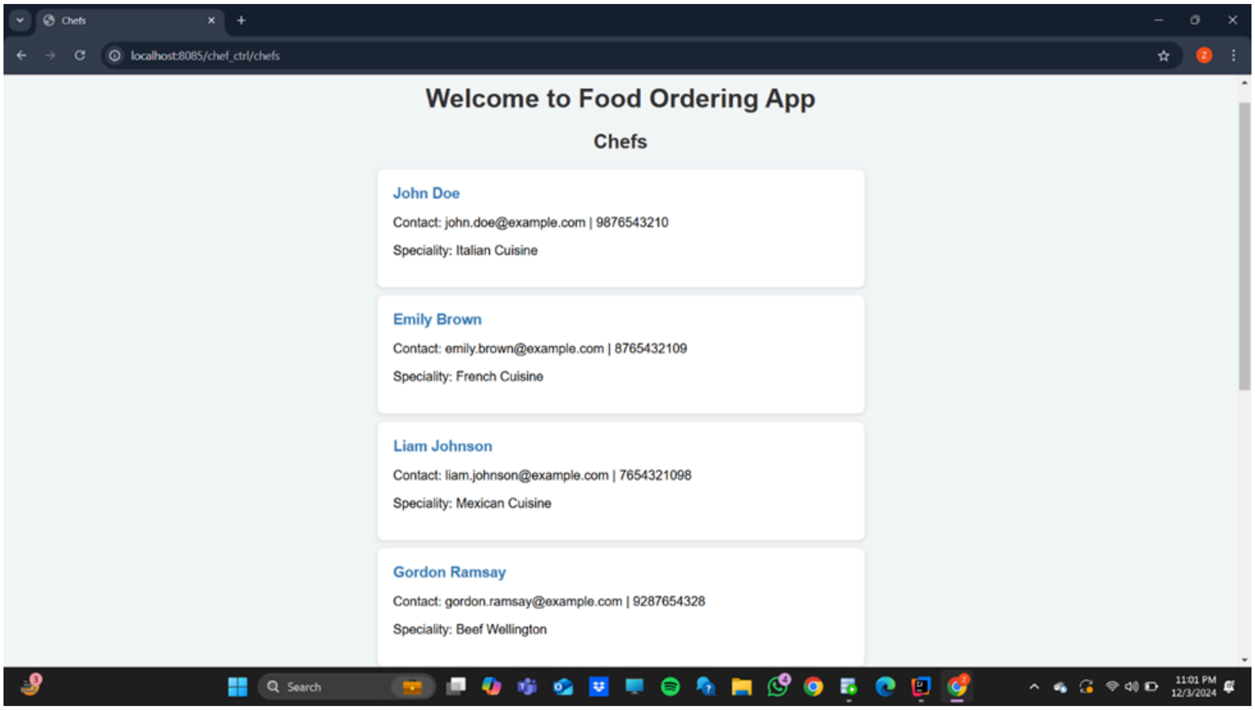
Cart

Item	Quantity	Price
Margherita Pizza	2	12.99
California Roll	1	8.99

Total: 34.97

```
spice level : mild
```

6.4 CHEFS PAGES



Welcome to Food Ordering App

Chefs

John Doe

Contact: john.doe@example.com | 9876543210

Speciality: Italian Cuisine

Emily Brown

Contact: emily.brown@example.com | 8765432109

Speciality: French Cuisine

Liam Johnson

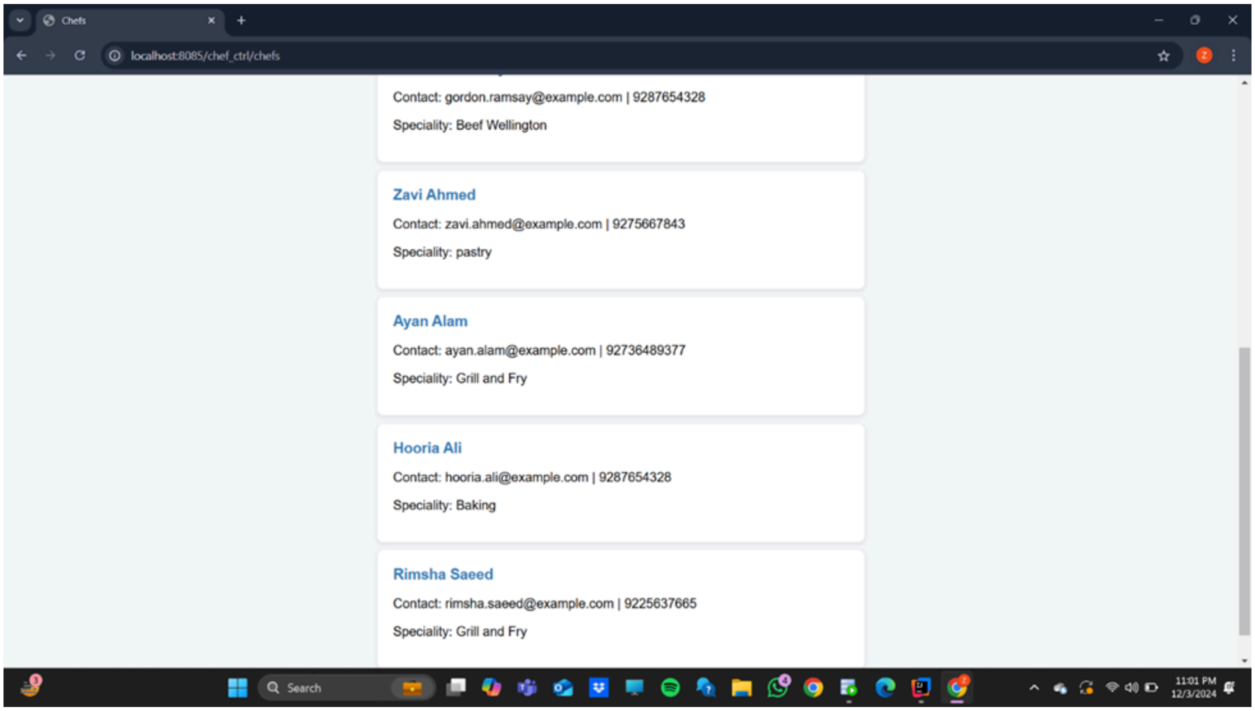
Contact: liam.johnson@example.com | 7654321098

Speciality: Mexican Cuisine

Gordon Ramsay

Contact: gordon.ramsay@example.com | 9287654328

Speciality: Beef Wellington



Contact: gordon.ramsay@example.com | 9287654328

Speciality: Beef Wellington

Zavi Ahmed

Contact: zavi.ahmed@example.com | 9275667843

Speciality: pastry

Ayan Alam

Contact: ayan.alam@example.com | 92736489377

Speciality: Grill and Fry

Hooria Ali

Contact: hooria.ali@example.com | 9287654328

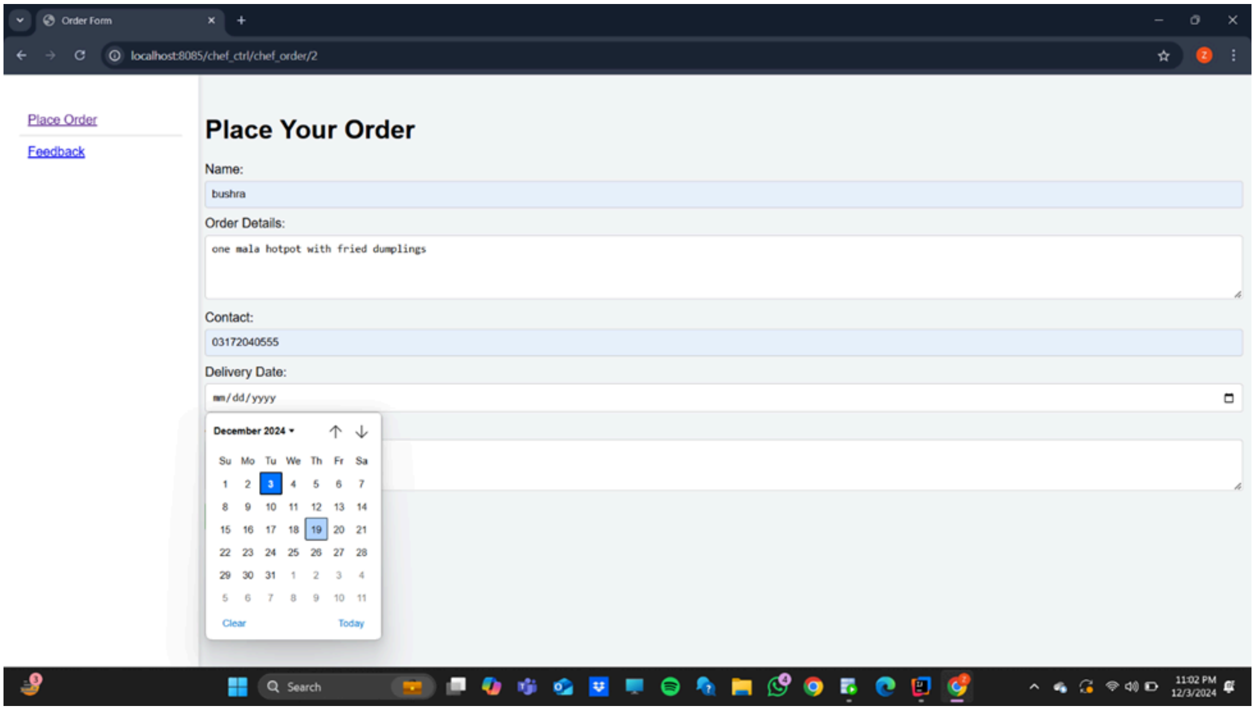
Speciality: Baking

Rimsha Saeed

Contact: rimsha.saeed@example.com | 9225637665

Speciality: Grill and Fry

6.5 PLACE ORDER



Place Your Order

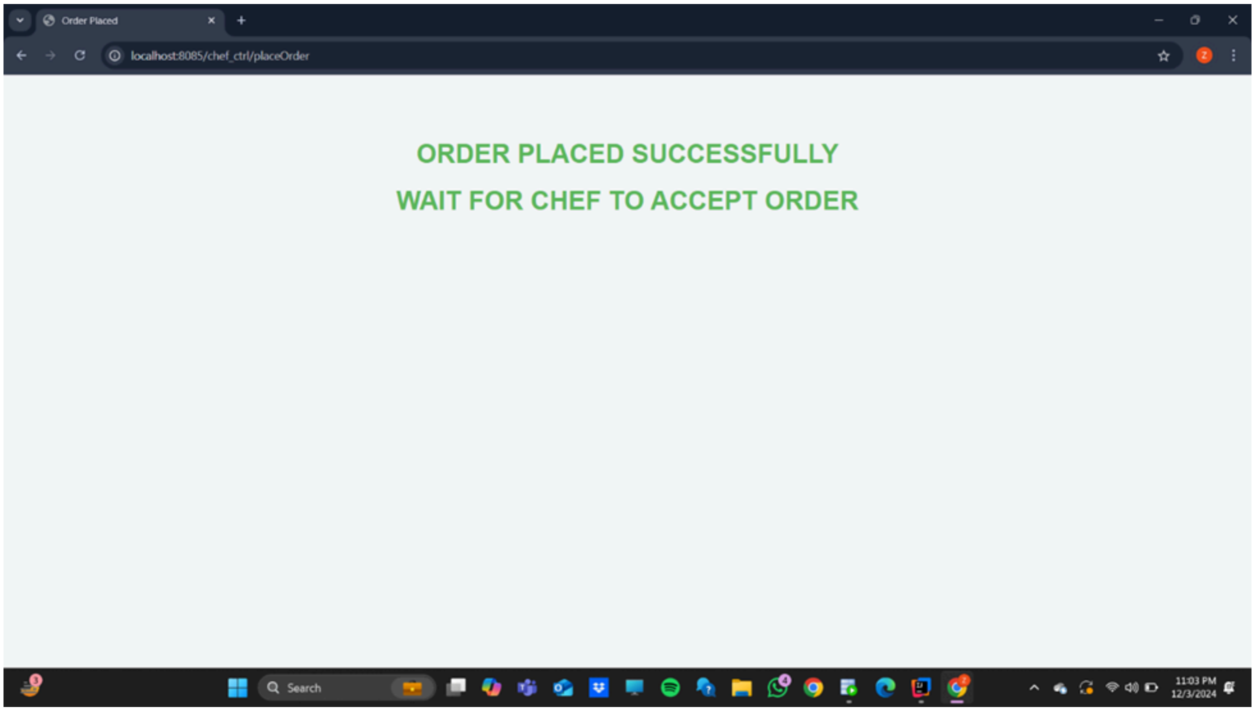
Name: bushra

Order Details:

```
one mala hotpot with fried dumplings
```

Contact: 03172040555

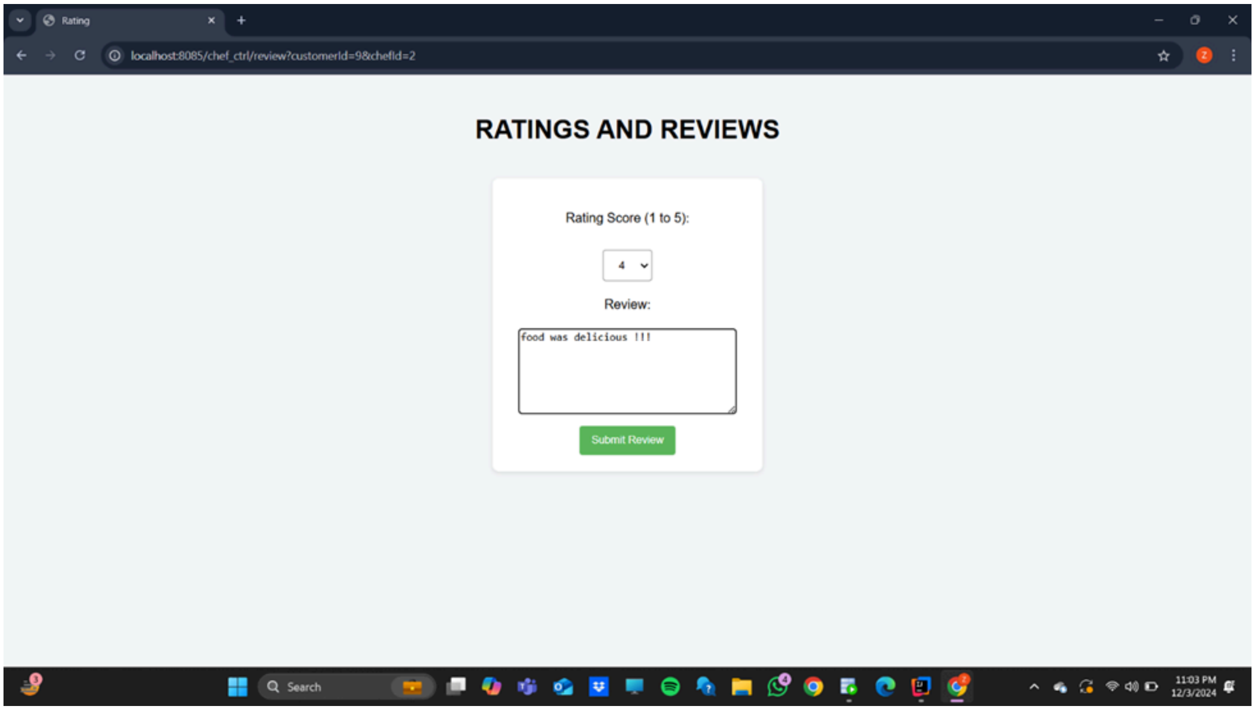
Delivery Date: mm/dd/yyyy



ORDER PLACED SUCCESSFULLY

WAIT FOR CHEF TO ACCEPT ORDER

6.6 RATING AND REVIEW PAGES

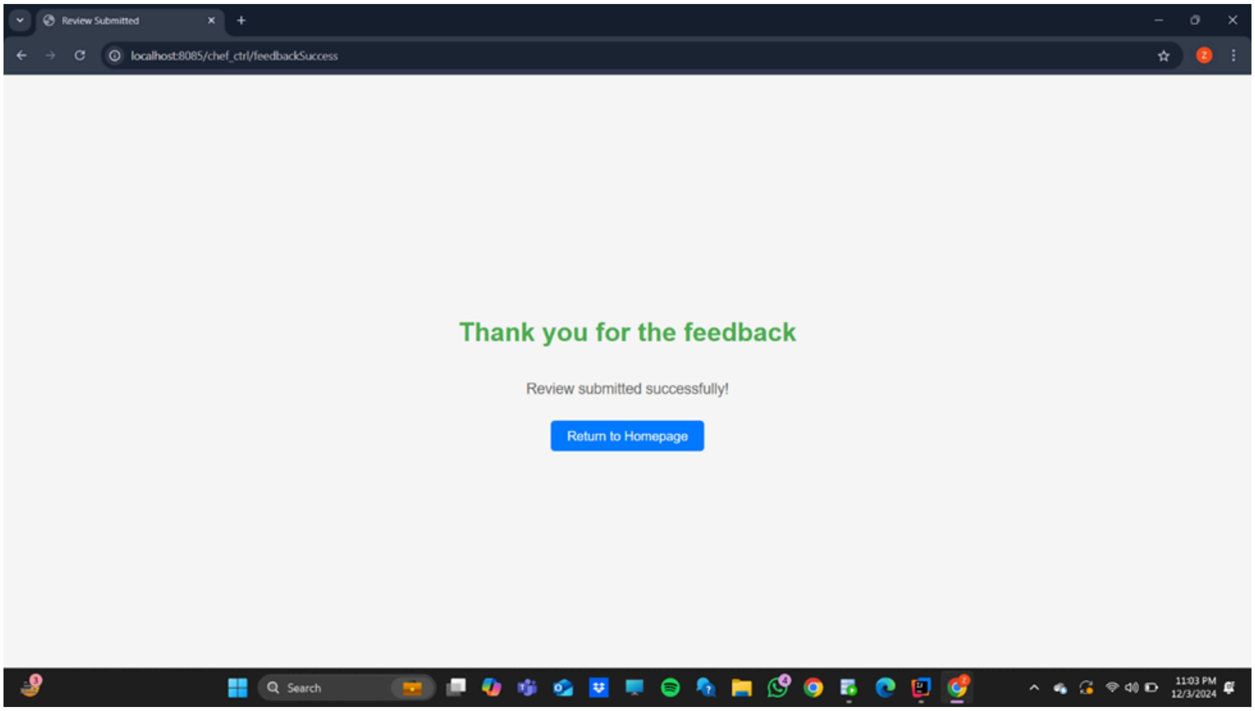


RATINGS AND REVIEWS

Rating Score (1 to 5): 4

Review:

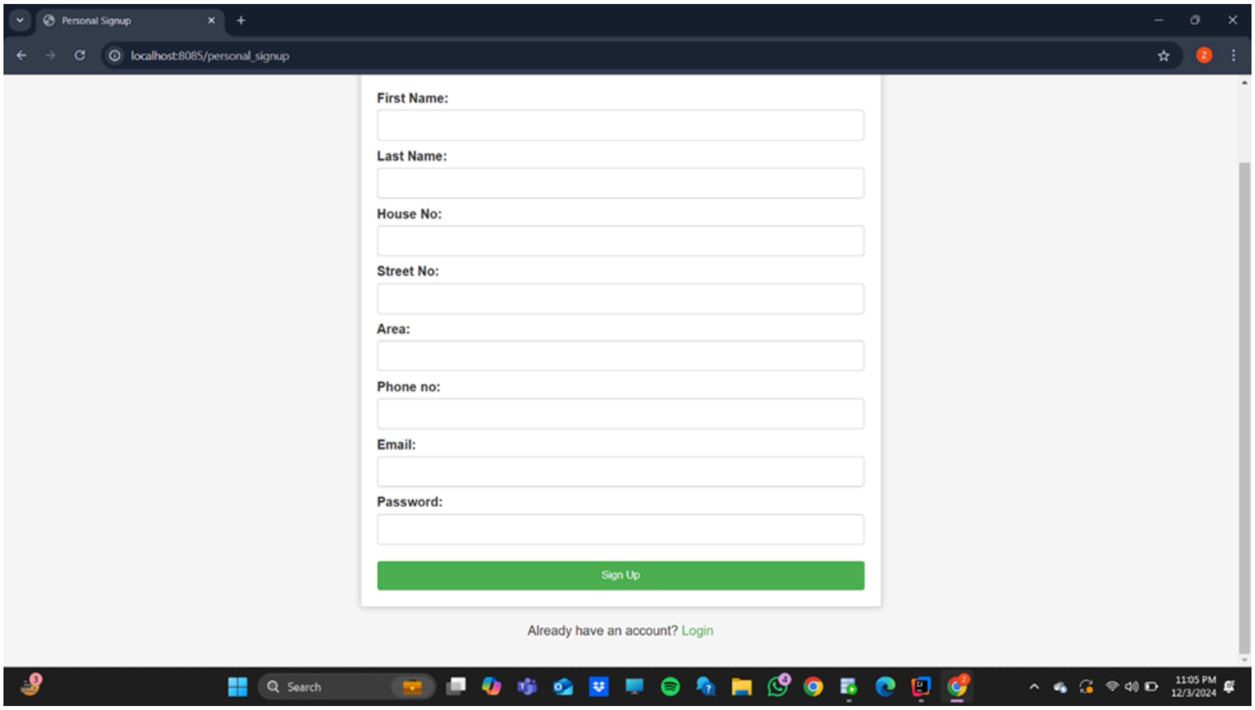
```
food was delicious !!!
```



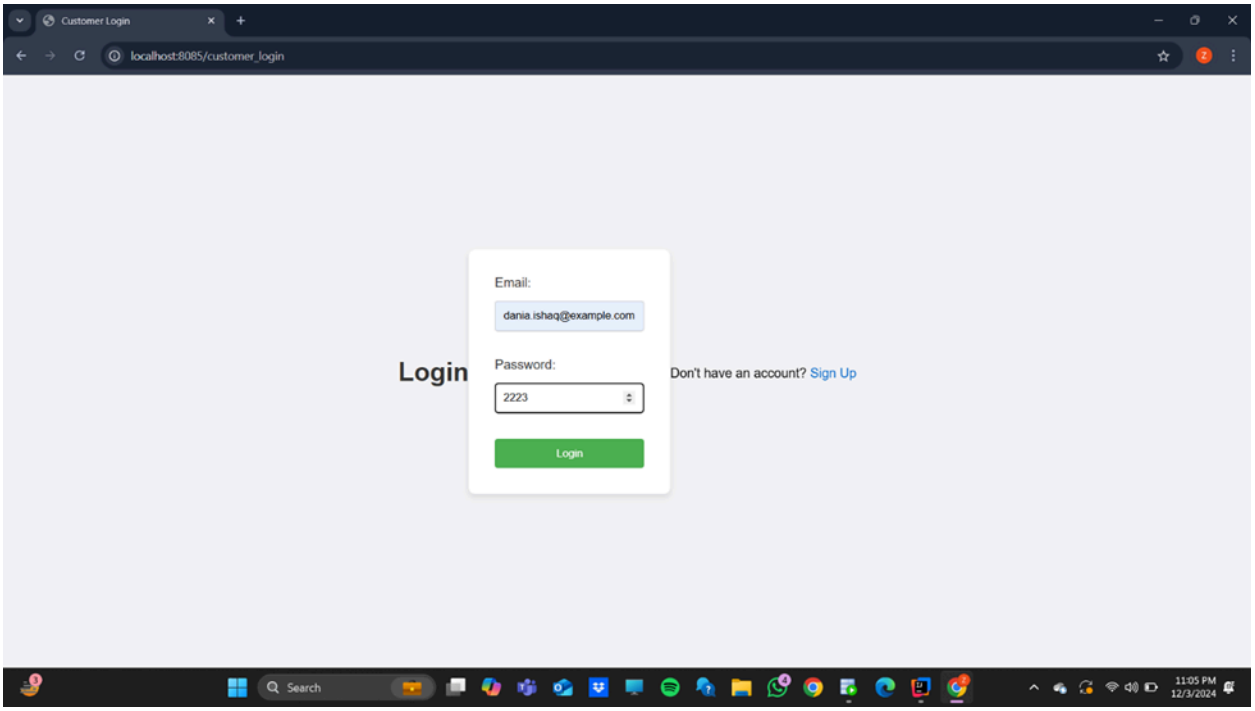
Thank you for the feedback

Review submitted successfully!

6.7 SIGNING UP



Already have an account? Login



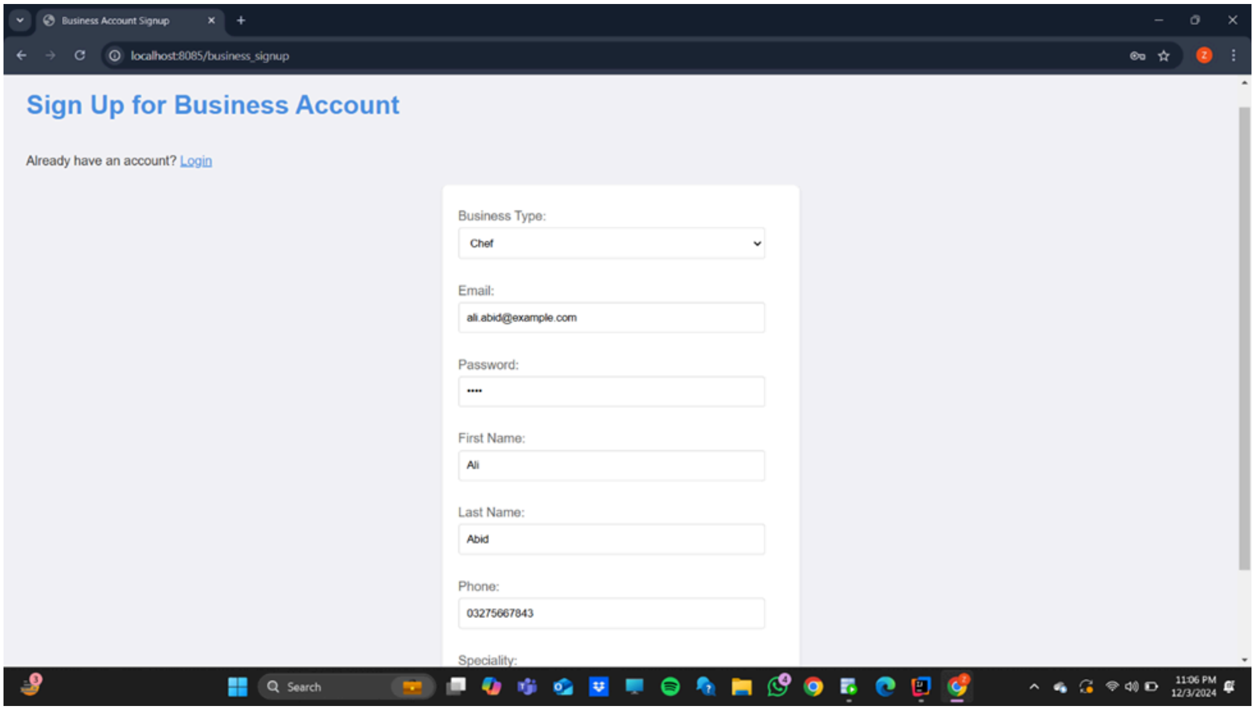
Email: dania.ishaq@example.com

Password: 2223

Login

Don't have an account? Sign Up

6.8 BUSINESS ACCOUNTS FEATURE PAGES



Sign Up for Business Account

Already have an account? Login

Business Type: Chef

Email: ali.abid@example.com

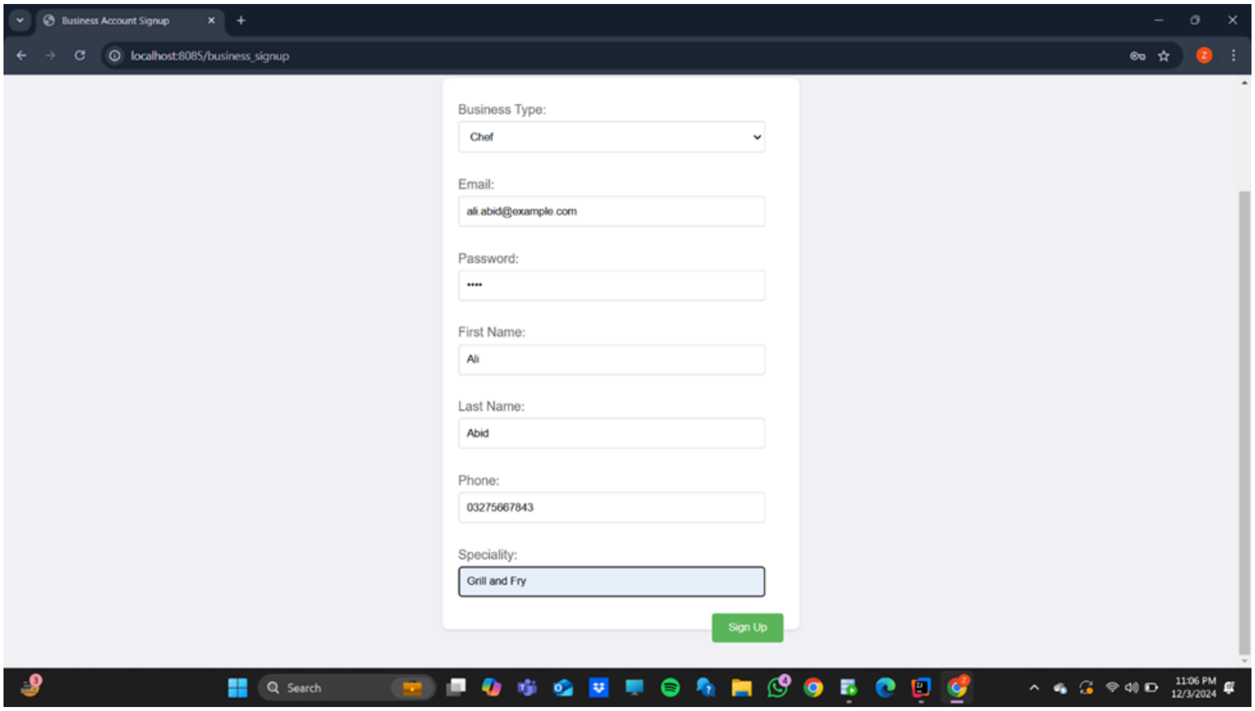
Password: ••••

First Name: Ali

Last Name: Abid

Phone: 03275667843

Speciality:



Business Type: Chef

Email: ali.abid@example.com

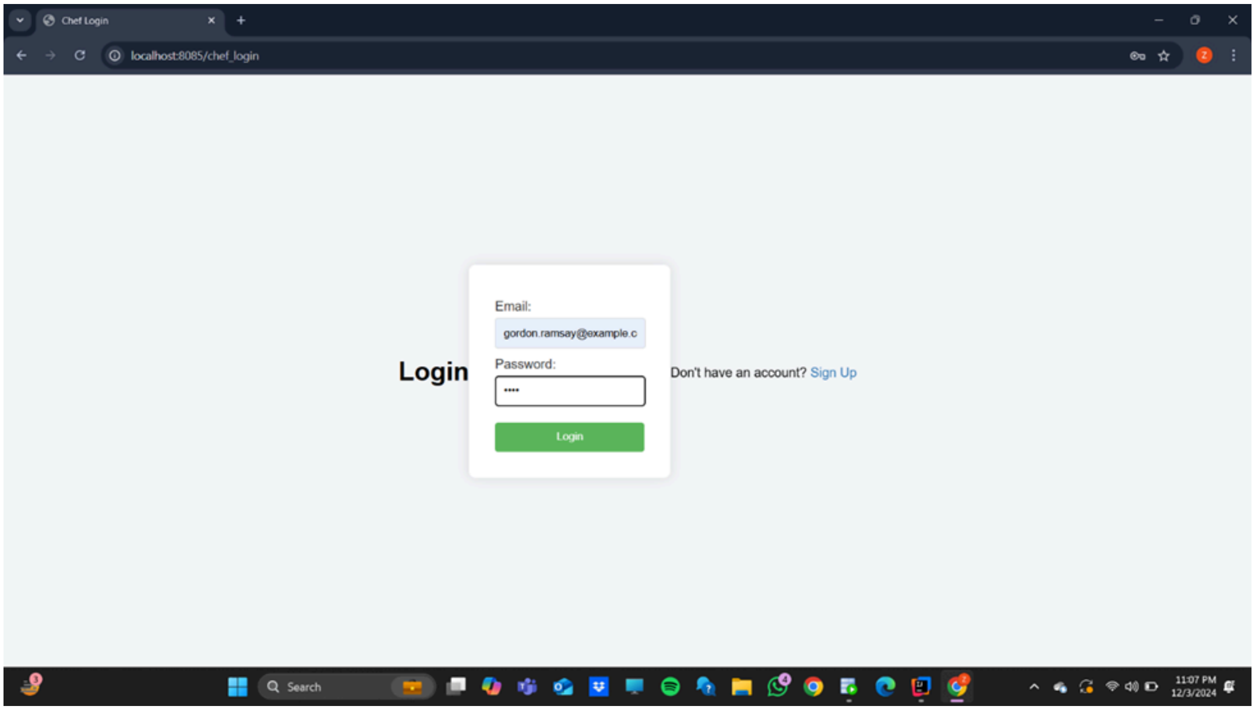
Password: ••••

First Name: Ali

Last Name: Abid

Phone: 03275667843

Speciality: Grill and Fry



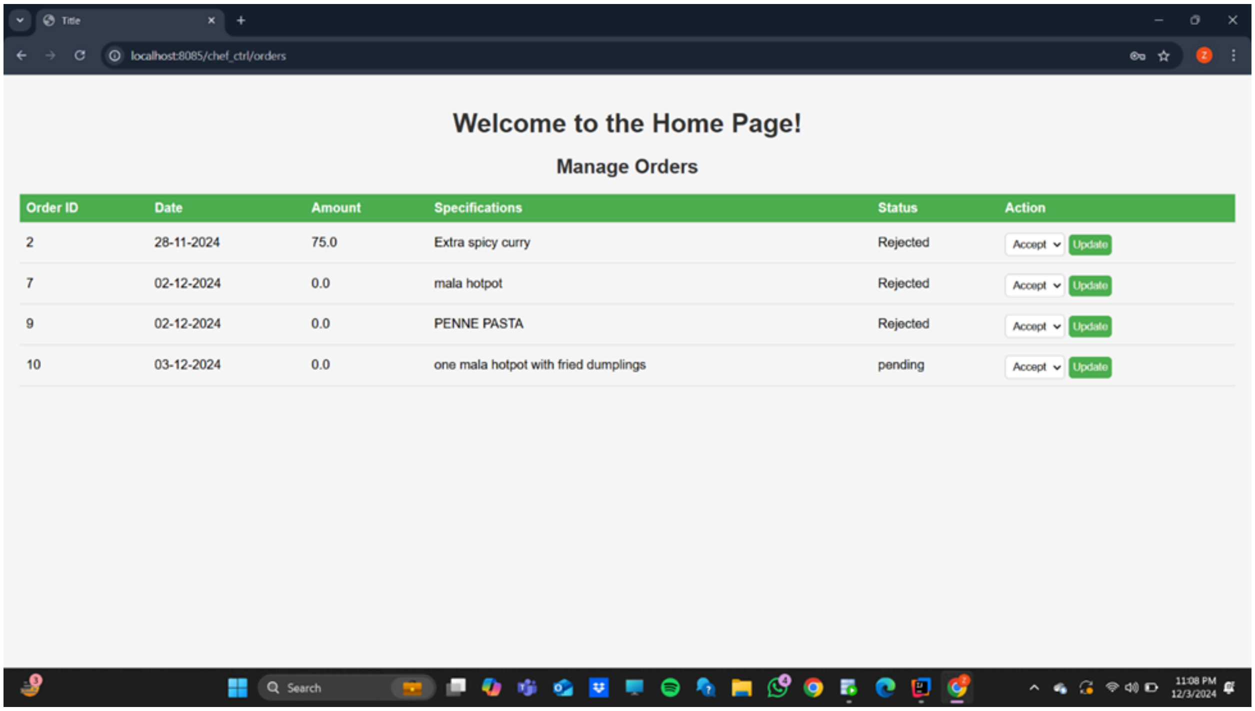
Login

Email: gordon.ramsay@example.c

Password: ••••

Login

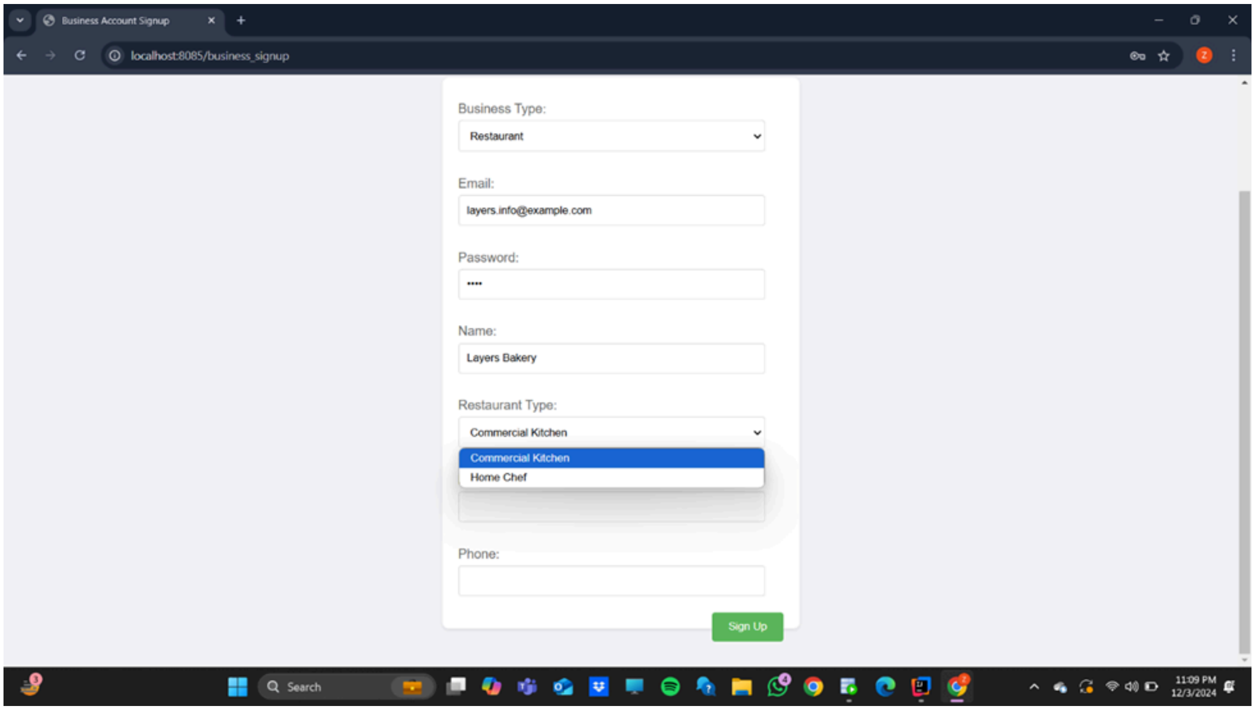
Don't have an account? Sign Up



Welcome to the Home Page!

Manage Orders

Order ID	Date	Amount	Specifications	Status	Action
2	28-11-2024	75.0	Extra spicy curry	Rejected	Accept ⌄ Update
7	02-12-2024	0.0	mala hotpot	Rejected	Accept ⌄ Update
9	02-12-2024	0.0	PENNE PASTA	Rejected	Accept ⌄ Update
10	03-12-2024	0.0	one mala hotpot with fried dumplings	pending	Accept ⌄ Update



Business Type:
Restaurant

Email:
layers.info@example.com

Password:
••••

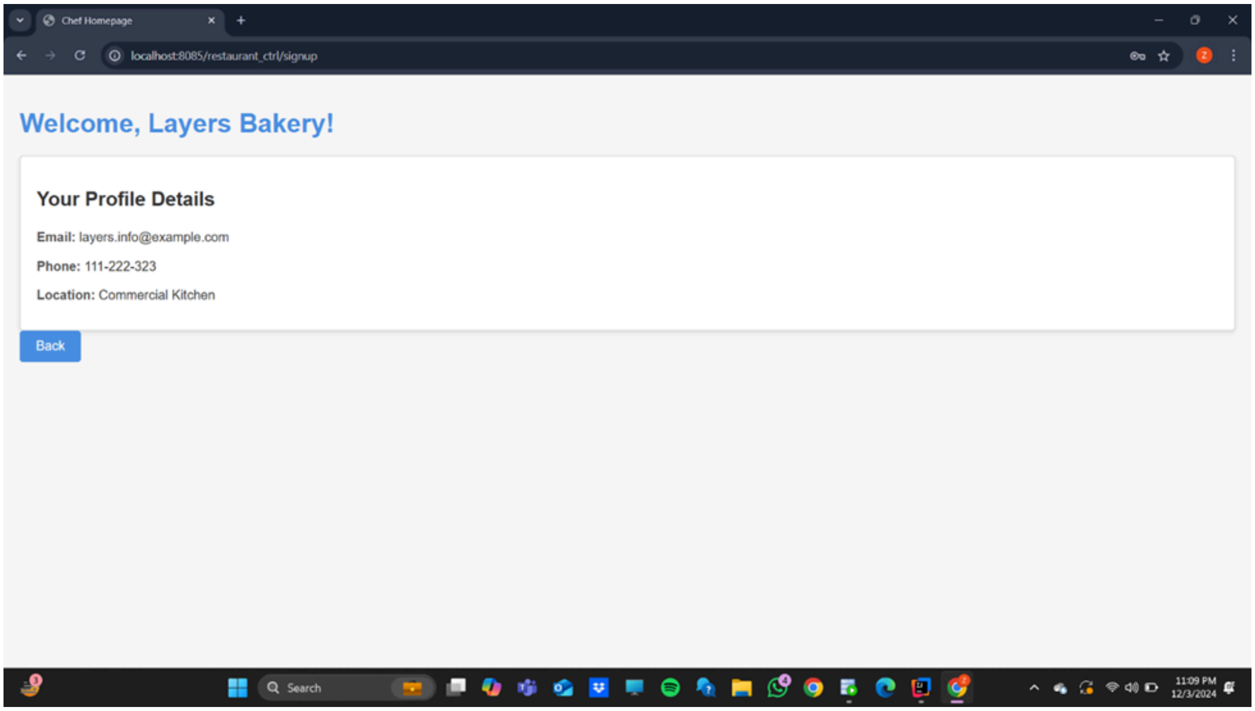
Name:
Layers Bakery

Restaurant Type:
Commercial Kitchen

- Commercial Kitchen
- Home Chef

Phone:

Sign Up



Welcome, Layers Bakery!

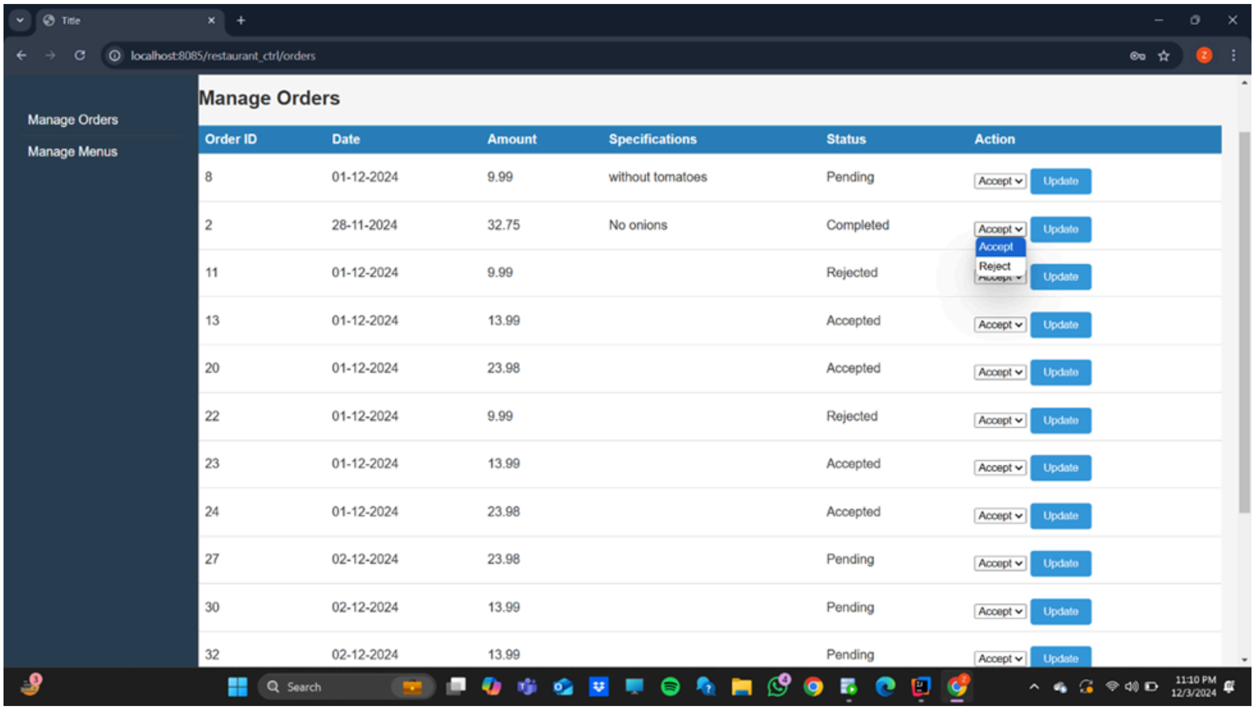
Your Profile Details

Email: layers.info@example.com

Phone: 111-222-323

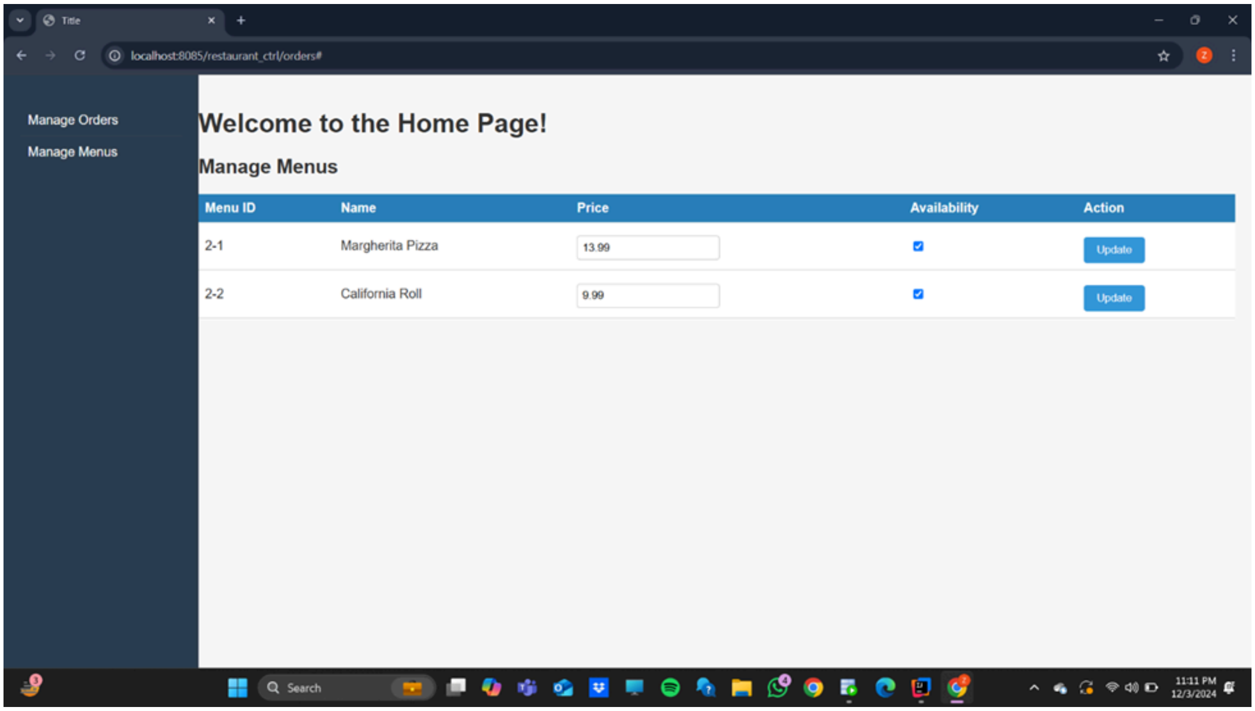
Location: Commercial Kitchen

Back



Manage Orders

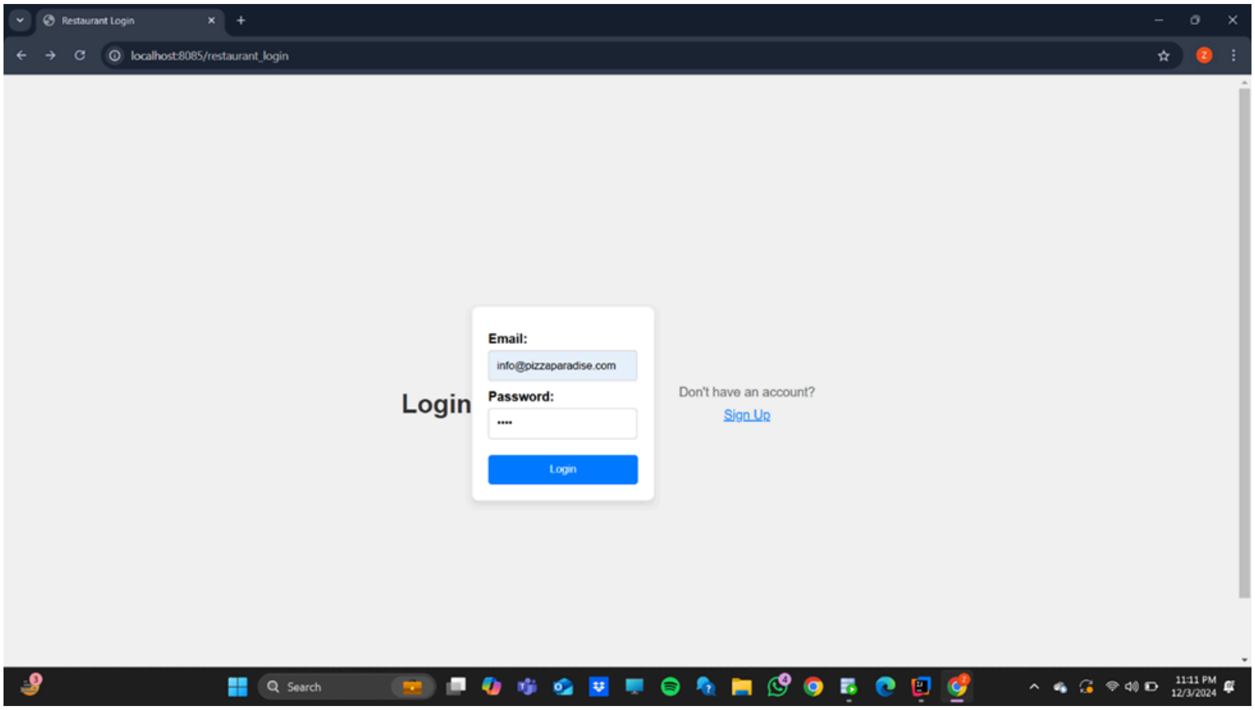
Order ID	Date	Amount	Specifications	Status	Action
8	01-12-2024	9.99	without tomatoes	Pending	Accept ▾ Update
2	28-11-2024	32.75	No onions	Completed	Accept ▾ Update
11	01-12-2024	9.99		Rejected	Accept ▾ Update
13	01-12-2024	13.99		Accepted	Accept ▾ Update
20	01-12-2024	23.98		Accepted	Accept ▾ Update
22	01-12-2024	9.99		Rejected	Accept ▾ Update
23	01-12-2024	13.99		Accepted	Accept ▾ Update
24	01-12-2024	23.98		Accepted	Accept ▾ Update
27	02-12-2024	23.98		Pending	Accept ▾ Update
30	02-12-2024	13.99		Pending	Accept ▾ Update
32	02-12-2024	13.99		Pending	Accept ▾ Update



Welcome to the Home Page!

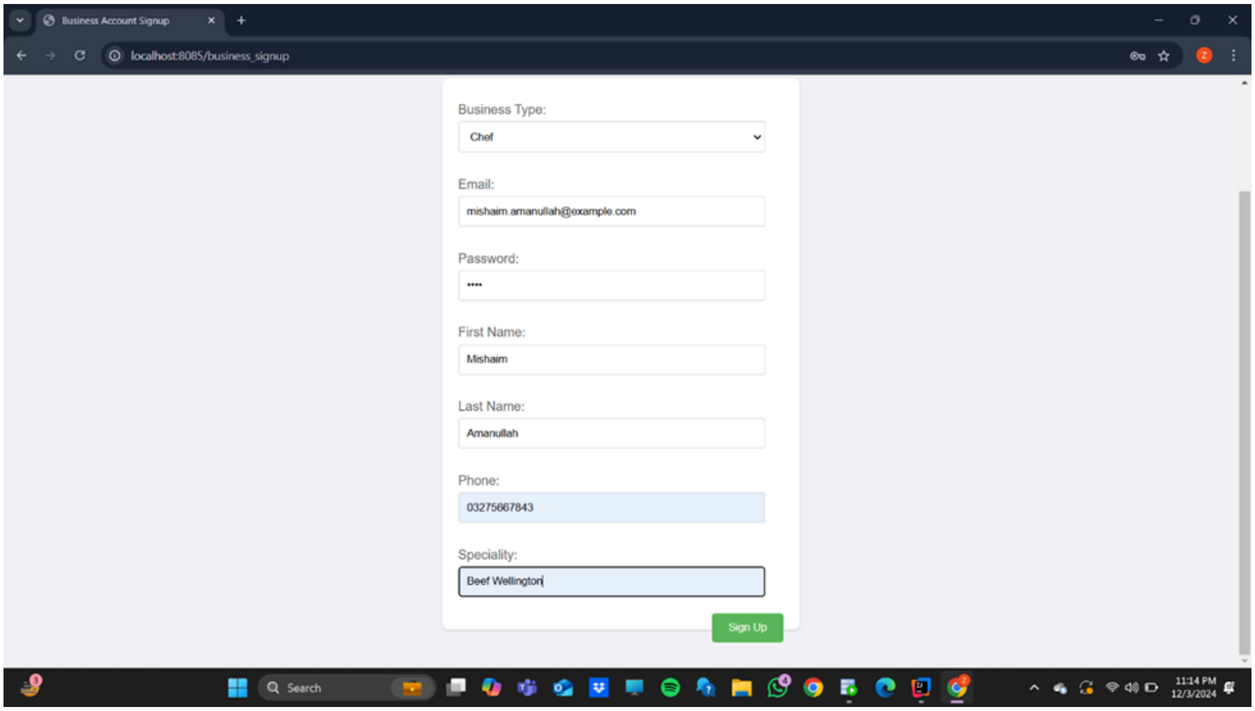
Manage Menus

Menu ID	Name	Price	Availability	Action
2-1	Margherita Pizza	13.99	☑	Update
2-2	California Roll	9.99	☑	Update



Email: info@pizzaparadise.com

Password: ••••



Business Type: Chef

Email: mishaim.amanullah@example.com

Password: ••••

First Name: Mishaim

Last Name: Amanullah

Phone: 03275667843

Speciality: Beef Wellington

Chapter 7

Conclusion

The Online Food Ordering System has been successfully developed and tested to meet the project objectives. It provides a user-friendly platform that simplifies the process of ordering food, managing restaurant menus, and handling custom chef requests. The system facilitates seamless interactions between users, restaurants, and chefs, offering options for both personal and business accounts.

The project achieved its primary goals, including:

- Centralized Database: Ensures efficient storage and retrieval of user, restaurant, chef, and order-related information.
- Simplified Order Management: Allows users to place and track orders easily while enabling restaurants and chefs to manage orders effectively.
- Customizable Options: Provides the flexibility for users to browse menus or request personalized dishes from chefs.
- Scalable Design: The modular structure supports future enhancements, such as improved payment integration and expanded functionalities.
- User Reviews and Ratings: Empowers users to provide feedback, improving service quality and fostering trust.

Although the menu and payment features faced limitations due to a composite key issue in Spring Boot, the project demonstrates robust functionality in other areas, making it a valuable solution for both individual customers and businesses.

With its efficient design and user-focused features, this system is poised to streamline the food ordering process and enhance user satisfaction. Future enhancements could address current limitations, integrate more advanced features, and scale the platform for broader adoption.

CHAPTER 8

Future Enhancements

Future upgrades to the Online Food Ordering System aim to address current limitations and expand its functionality, including:

- Improved Menu Management:
    - Resolving the composite key issue in the restaurant-menu relationship to enable seamless menu updates and payment integration.
    - Adding advanced filtering options for users to search menus by cuisine, dietary preferences, and pricing.
- Enhanced User Interfaces:
    - Developing more intuitive and interactive interfaces for both personal and business accounts.
    - Implementing live order tracking for users and real-time notifications for restaurants and chefs.
- Payment Integration:
    - Introducing secure and flexible payment gateways with options for card payments, e-wallets, and direct bank transfers.
- Advanced Analytics and Reporting:
    - Providing restaurants and chefs with insights on customer preferences, order trends, and performance metrics.
    - Allowing users to view detailed reports on their orders, spending, and feedback history.

These enhancements will strengthen the platform's functionality, provide better experiences for all users, and position it as a comprehensive solution for online food ordering and management.