



Islamic University of Technology (IUT)

Organization of Islamic Cooperation (OIC)

Department of Technical and Vocational Education (TVE)

Project Name:

Foodie Haven

Submitted By

Khadiza-Tul-Kobra: 220032402

Shiha Khatun: 220032404

Ahmad Babajo: 220032407

Submitted To

Tanjila Alam Sathi

Lecturer, Department of Computer Science and Engineering (CSE)

Islamic University of Technology (IUT)

DECLARATION OF AUTHORSHIP

This is to certify that the report titled Food Delivery Platform is the result of independent study, analysis, system design, and development conducted by **Khadiza-Tul-Kobra** (ID: 220032402), **Shiha Khatun** (ID: 220032404), and **Ahmad Babajo** (ID: 220032404). The project was completed under the guidance of **Tanjila Alam Sathi**, Lecturer, Department of Computer Science and Engineering, Islamic University of Technology (IUT), Dhaka, Bangladesh.

It is further affirmed that this report, in whole or in part, has not been previously submitted to any institution or organization for the purpose of obtaining a degree, diploma, or any other academic award.

Khadiza-Tul-Kobra

Student No: 220032402

Shiha Khatun

Student No: 220032404

Ahmad Babajo

Student No: 220032407

Supervisor

Tanjila Alam Sathi, Lecturer
Department of Computer
Science and Engineering,
Islamic University of
Technology

ACKNOWLEDGMENT

We would like to thank all the people who helped us in completing our Food Delivery Platform Project.

First, we are very thankful to our supervisor Tanjila Alam Sathi, Lecturer, Department of Computer Science and Engineering, Islamic University of Technology (IUT), Dhaka, Bangladesh for her kind guidance, support, and useful advice. Her patience and encouragement helped us to overcome many challenges during the project.

ABSTRACT

The Food Delivery Platform helps users, restaurant owners, and delivery personnel manage food orders efficiently. It provides customers with an easy way to user registration and login, browse restaurant and food, view menus, place orders and payment.

The Food Delivery Platform is an application developed using Tailwind CSS, React Native, Node.js, Express.js, TypeScript, MySQL database. Since it is platform-independent, it can be used on almost any device with internet access. Furthermore, the system is adaptable to the unique requirements of various user groups or eateries.

TABLE OF CONTENT

DECLARATION OF AUTHORSHIP.....	ii
ACKNOWLEDGMENT.....	iii
ABSTRACT.....	iv
TABLE OF CONTENT	v
LIST OF FIGURES	vii
Chapter 1 Introduction and Motivation	1
1.1 Project Feature	1
1.2 Objective	1
1.3 Target Audience	1
Chapter 2 Requirement analysis.....	3
2.1 Project Description.....	3
2.2 Admin panel	3
2.3 User panel.....	4
Chapter 3 Tools, Technology and system design.....	5
3.1 Tools and Technologies	5
3.1.1 Frontend.....	5
3.1.2 Backend	5
3.2 Database	6
3.3 System Design.....	8
3.3.1 Project Timeline:	10
Chapter 4 Design and implementation	11
4.1 Design.....	11
4.1.1 Use case diagram:	11
4.1.2 ER diagram:	12
4.1.3 Sequence Diagram.....	13
4.1.4 Activity diagram	14
4.2 Implementation	16
4.2.1 Registration page	16
4.2.2 User Verification page.....	16
4.2.3 Admin and user login page:.....	17
4.2.4 Home page:.....	18
4.2.5 Browsing food	19
4.2.6 Add Filters (category, Price)	19

4.2.7	Search bar functionality	19
4.3	Payment System:	21
4.4	Order place:	22
4.4.1	Order History Page Layout:	22
4.5	Review and Rating:	23
4.6	Admin Part	24
4.6.1	Create Restaurant	25
4.6.2	Add Form Validation	25
4.6.3	Add Food	25
4.6.4	Connect Form to Backend	26
4.6.5	Order management	26
4.7	GitLab	27
Chapter 5 CHALLENGES		29
Chapter 6 CONCLUSION		30

LIST OF FIGURES

Figure 1:Database Design	7
Figure 2:Food information.....	7
Figure 3:Restaurant Information.....	8
Figure 4:Project Timeline	10
Figure 5: Use case diagram.....	11
Figure 6: ER diagram.....	12
Figure 7:Sequence diagram	13
Figure 8:Activity Diagram.....	15
Figure 9: Registration page	16
Figure 10: Verification page	17
Figure 11: Admin login.....	18
Figure 12: Home page.....	19
Figure 13: Browsing food	20
Figure 14: Place food order.....	20
Figure 15: Cash on Delivery	21
Figure 16: Card payment	22
Figure 17:User show order list.....	23
Figure 18: Order confirmation.....	23
Figure 19: Review and Rating	24
Figure 21:Create Restaurant	25
Figure 22:Add food.....	26
Figure 23:View Order	27
Figure 24: Manage order.....	27

Chapter 1 Introduction and Motivation

In today's fast-paced world, delivery services for food have become an integral part of many people's everyday lives. "Foodie Haven" is a project dedicated to simplifying and improving the food delivery experience by providing a user-friendly, successful, and dependable platform. The website aims to connect consumers with a variety of eateries, providing quick service, a simple ordering process, and efficient order tracking. Foodie Haven aims to stand out as a simple and trustworthy solution regarding food delivery by addressing typical concerns identified in existing services.

1.1 Project Feature

- Restaurant and menu browsing
- User account creation and order management
- Search and filter options for restaurants and food items
- Responsive design for optimal mobile and desktop experience
- Recommendation System

1.2 Objective

The Foodie Haven project's major goal is to create and build a highly efficient meal delivery website with a focus on user pleasure.

- To design an intuitive UI with a smooth UX for better user experience.
- To offer diverse restaurants and cuisines with fast and reliable delivery.

1.3 Target Audience

Foodie Haven is intended to serve a diverse set of users, including:

1. Busy people who need quick and easy food options.
2. Students frequently seek economical and convenient food delivery.

3. Office workers are looking for efficient lunch and dinner options.
4. Foodies who appreciate trying new cuisines without the inconvenience of dining out.

By recognizing and focusing on the needs of these groups, Foodie Haven hopes to deliver a platform that continuously meets and surpasses client expectations.

Chapter 2 Requirement analysis

2.1 Project Description

The Food Delivery Platform project aims to deliver a seamless and efficient system for managing online food orders. It features several integrated modules that simplify the overall process of browsing menus, placing orders, and managing deliveries, delivering, and managing food for customers, restaurants, and delivery staff. These modules help ensure smooth communication, quick order processing, and improved overall efficiency for all users involved.

- **User Account Creation and Order Management:** Users can register, log in, place orders, and manage their orders easily.
- **Restaurant and Menu Browsing:** Users can explore different restaurants and view available food items.
- **Search Option for Food Items:** A quick search feature helps users find their favorite dishes easily.
- **Review and Rating:** Users can give their experience by providing review and rating.

The Foodie Haven website is built to serve two main types of users: customers and admin. Each group has specific functionalities designed to make the food delivery process simple, efficient, and user-friendly.

2.2 Admin panel

- **Restaurant Management:** Admins can add restaurants to the system by entering basic details like the restaurant name and other necessary information.
- **Food Item Management:** Admins can manage food items by adding new ones, updating existing details, or removing them when necessary.

- **Order Management:** Admins can monitor all customer orders, change their statuses (such as preparing, out for delivery, or delivered), and handle any issues that arise during the process.

2.3 User panel

- **Registration:** Users can easily create an account by providing basic information such as name, email address, password, confirm password.
- **Login:** Registered users can securely access their accounts by entering their login credentials.
- **Food Browsing:** Users can explore a wide selection of restaurants and food items.
- **Placing Orders:** Users can add selected items to their cart, review their order, and confirm delivery details before finalizing the purchase.
- **Browsing and Searching Food:** A user-friendly browsing system helps users find their desired meals quickly and efficiently through filters and search options.
- **Payment Options:** Two payment options are available in the system: Cash on Delivery and Card Payment, allowing users to choose their preferred method during checkout.
- **Review and Rating:** After receiving their orders, users can leave reviews and rate their experiences to help maintain service quality and assist future customers.

Chapter 3 Tools, Technology and system design

3.1 Tools and Technologies

To build an efficient, scalable, and modern food delivery platform, Foodie Haven utilizes a carefully selected technology stack for the frontend, backend, and database.

3.1.1 Frontend

- Tailwind CSS: A utility-first CSS framework that helps create a responsive, visually appealing and easily customizable user interface with less coding effort.
- React Native: A popular JavaScript framework for building interactive and high-performance mobile-friendly frontends, ensuring a smooth user experience across different devices.

3.1.2 Backend

Node.js:

Node.js is a free, cross-platform JavaScript runtime that allows developers to create server-side applications. It uses an event-driven approach with non-blocking I/O, making it fast and efficient for handling many tasks at once, ensuring high efficiency and scalability for handling multiple requests simultaneously.

- Scalability: Can handle high-traffic applications easily.
- Fast Performance: Non-blocking I/O ensures quick server responses.
- JavaScript on Server: Same language can be used for client and server.
- Large Ecosystem: NPM (Node Package Manager) allows easy integration of libraries.

Express.js

Express.js is a minimal and flexible web framework for Node.js that streamlines backend development. It offers powerful features for routing, server management, and integrating middleware efficiently.

- **Simplified Backend Development:** Makes routing and HTTP handling easier.
- **Middleware Support:** Allows adding custom functions efficiently.
- **Flexible & Lightweight:** Suitable for building any type of web application.
- **Robust APIs:** Simplifies the development of RESTful APIs.

TypeScript

TypeScript is a JavaScript extension with static typing that ensures type safety and helps produce more reliable code, helping developers write more reliable and maintainable code. It compiles to JavaScript and works in any browser.

Benefits:

- **Type Safety:** Reduces type-related errors.
- **Better Maintainability:** Easier to maintain in large projects.
- **Improved Developer Experience:** Supports autocompletion, IntelliSense, and easier debugging.
- **Compatibility:** Compiles to JavaScript for browser compatibility.

3.2 Database

MongoDB is a document-oriented NoSQL database designed to handle both structured and unstructured data efficiently. Unlike relational databases with rigid schemas, it uses a flexible schema model and stores data in a JSON-like format called BSON (Binary JSON). This adaptability is especially useful for dynamic applications like Foodie Haven, where data structures may change as new features are introduced. For example, a user document may contain a username, email, password hash, delivery address, and order history. Similarly, a restaurant document may include the restaurant name, location, menu items operating hours, and user ratings. By using MongoDB, each of these entities can be stored as separate documents within collections, providing a highly organized yet flexible data structure.

The screenshot shows the MongoDB Atlas interface for a cluster named 'Foodie haven'. The left sidebar contains navigation options: Overview, DATABASE, Clusters, SERVICES, Atlas Search, Stream Processing, Triggers, Migration, Data Federation, SECURITY, Quickstart, Backup, and Database Access. The 'Data Services' tab is active, displaying a table of collections for a database named 'test'.

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
carts	3	581B	194B	36KB	1	36KB	36KB
foods	36	311.36KB	8.65KB	552KB	1	36KB	36KB
orders	46	13.44KB	300B	36KB	1	36KB	36KB
otps	0	0B	0B	32KB	2	72KB	36KB
restaurants	9	115.09KB	12.79KB	324KB	1	36KB	36KB
reviews	12	1.92KB	165B	36KB	2	72KB	36KB
tempusers	17	1.77KB	107B	36KB	1	36KB	36KB
users	2	447B	224B	36KB	1	36KB	36KB

Figure 1:Database Design

- In the Foodie Haven platform, MongoDB serves as the backbone for storing and managing all critical information. This includes user profiles, restaurant details, menu items, orders, and ratings/reviews. Each of these data types can have unique attributes.

The screenshot shows the MongoDB Atlas interface for a cluster named 'Foodie haven'. The left sidebar contains navigation options: Overview, DATABASE, Clusters, SERVICES, Atlas Search, Stream Processing, Triggers, Migration, Data Federation, SECURITY, Quickstart, Backup, and Database Access. The 'Data Services' tab is active, displaying a table of collections for a database named 'test'. The 'foods' collection is selected, and the 'Find' tab is active, showing a query result for a food item.

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
carts	3	581B	194B	36KB	1	36KB	36KB
foods	36	311.36KB	8.65KB	552KB	1	36KB	36KB
orders	46	13.44KB	300B	36KB	1	36KB	36KB
otps	0	0B	0B	32KB	2	72KB	36KB
restaurants	9	115.09KB	12.79KB	324KB	1	36KB	36KB
reviews	12	1.92KB	165B	36KB	2	72KB	36KB
tempusers	17	1.77KB	107B	36KB	1	36KB	36KB
users	2	447B	224B	36KB	1	36KB	36KB

```

{
  "_id": ObjectId("680d3374cf2dc74ebd3ded40"),
  "name": "Basmati kacchi",
  "price": 300,
  "description": "abcd",
  "category": "fast food",
  "imageUrl": "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAQABAAQ/2wCEAAAGBxMTEHUSEXIT...",
  "rating": null,
  "stock": 11,
  "discount": 6,
  "createdAt": 2025-04-26T19:26:44.108+00:00
}

```

Figure 2:Food information

One of the main advantages of MongoDB is its scalability. As the number of users, restaurants, and orders increases, MongoDB can scale horizontally through sharing. Sharing allows the database to distribute data across multiple servers, ensuring consistent.

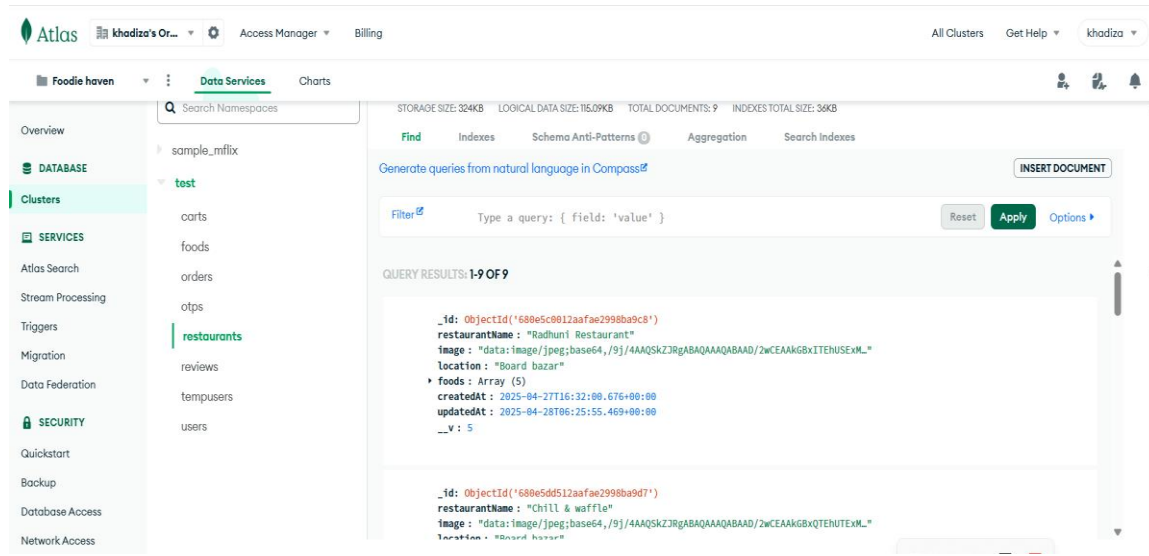


Figure 3: Restaurant Information

- Another key feature is high performance and quick data retrieval. MongoDB supports indexing on multiple fields, which allows fast searches on critical attributes such as restaurant names, user IDs, or order statuses. This is crucial for Foodie Haven, as users expect quick responses when browsing restaurants, filtering menu items, or checking order statuses. Additionally, MongoDB supports aggregation pipelines, which allow complex data processing directly within the database. This feature can be leveraged to generate insights, such as the most popular dishes, peak ordering time

3.3 System Design

The overall system architecture follows a three-tier model where the frontend, backend, and database components work together to deliver a seamless food delivery experience.

- The proposed food delivery application is designed to build a smooth connection between customers, restaurants, and delivery partners. It allows customers to browse restaurants, view menus, and place food orders directly through a user-

friendly interface. Once an order is placed, the restaurant receives a notification, prepares the food, and updates the status in real time. Delivery partners can then accept the order request, pick it up from the restaurant, and deliver it to the customer using navigation support and live tracking features. Customers are able to make secure payments through multiple options, including digital wallets, cards, and cash on delivery, while also receiving real-time updates about their order.

- On the restaurant side, the system helps manage menus, track orders, and monitor performance. For delivery partners, it provides an efficient system to manage assigned deliveries and track earnings. The admin has full control over the platform, ensuring data security, smooth communication, and report generation for overall monitoring.
- The goal of the system is to reduce the time and effort involved in traditional food ordering, provide reliable and fast delivery services, and create a digital platform that benefits all stakeholders. With features like real-time tracking, AI-powered food recommendations, and secure payment integration, the system ensures convenience for customers, growth opportunities for restaurants, and a stable income source for delivery partners.

3.3.1 Project Timeline:

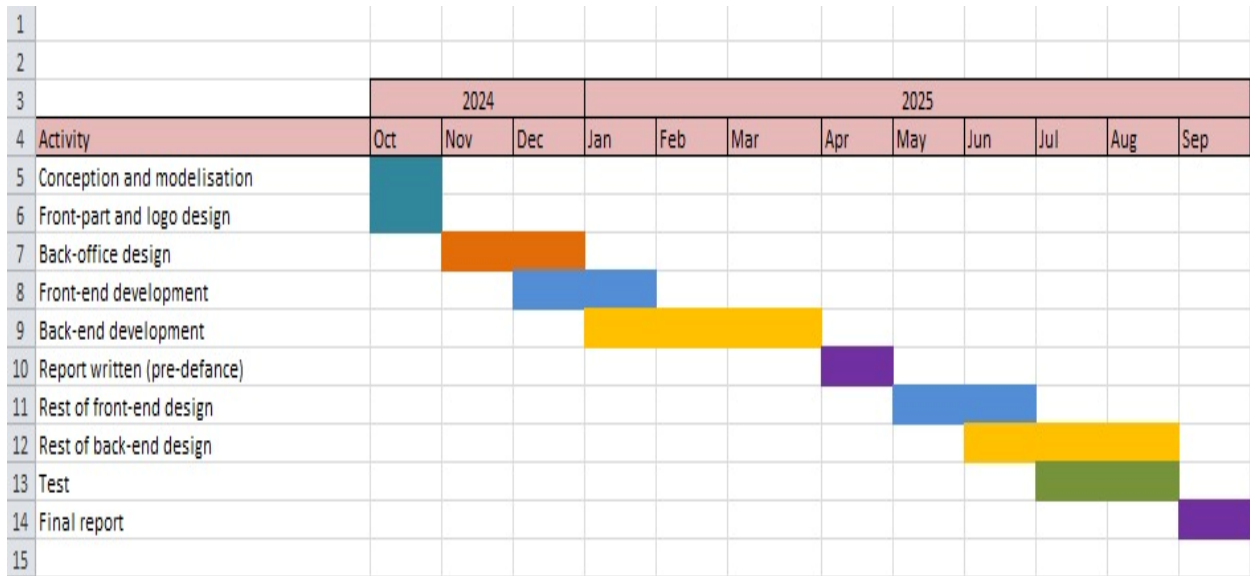


Figure 4:Project Timeline

Chapter 4 Design and implementation

4.1 Design

4.1.1 Use case diagram:

The use case diagram illustrates the functional interactions between two actors: User and Admin. A user can register, log in, and view or update their profile information, search restaurants, view menus, submit food orders, verify payments, track order status, provide feedback, and view available discounts or deals

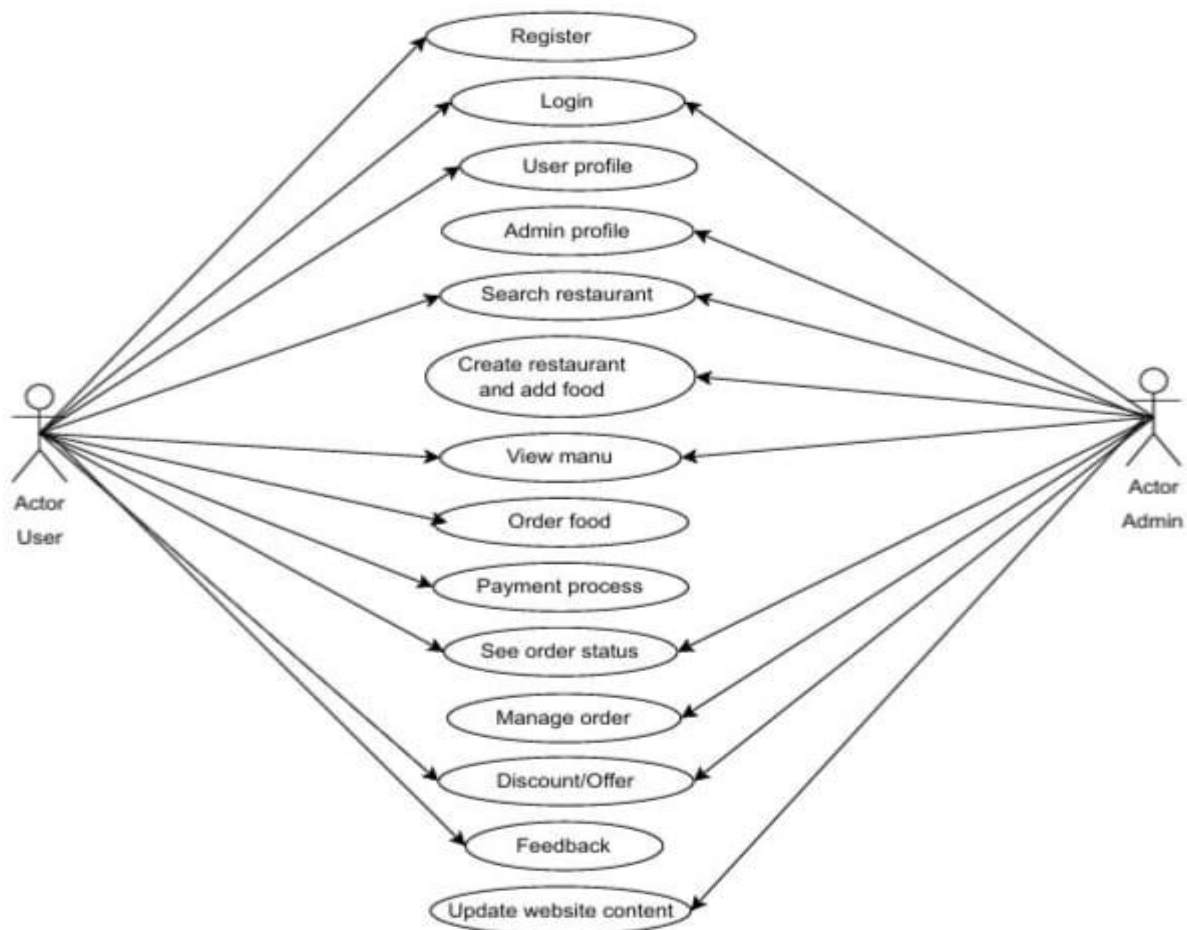


Figure 5: Use case diagram

The website's administrator, on the other hand, can log in to maintain their profile, build restaurants, add food products, manage all orders, change the material on the website, and handle promotions or coupons. This graphic helps to visualize the numerous steps taken by various actors inside the system, making it easier to grasp the meal ordering platform's expectations.

4.1.2 ER diagram:

The ER (Entity Relationship) diagram depicts the data structure of the meal ordering system. It contains entities such as User, Admin, Restaurant, Menu, Order, Order Item, Payment, Feedback, and Discount. Connectors are used to display relationships.

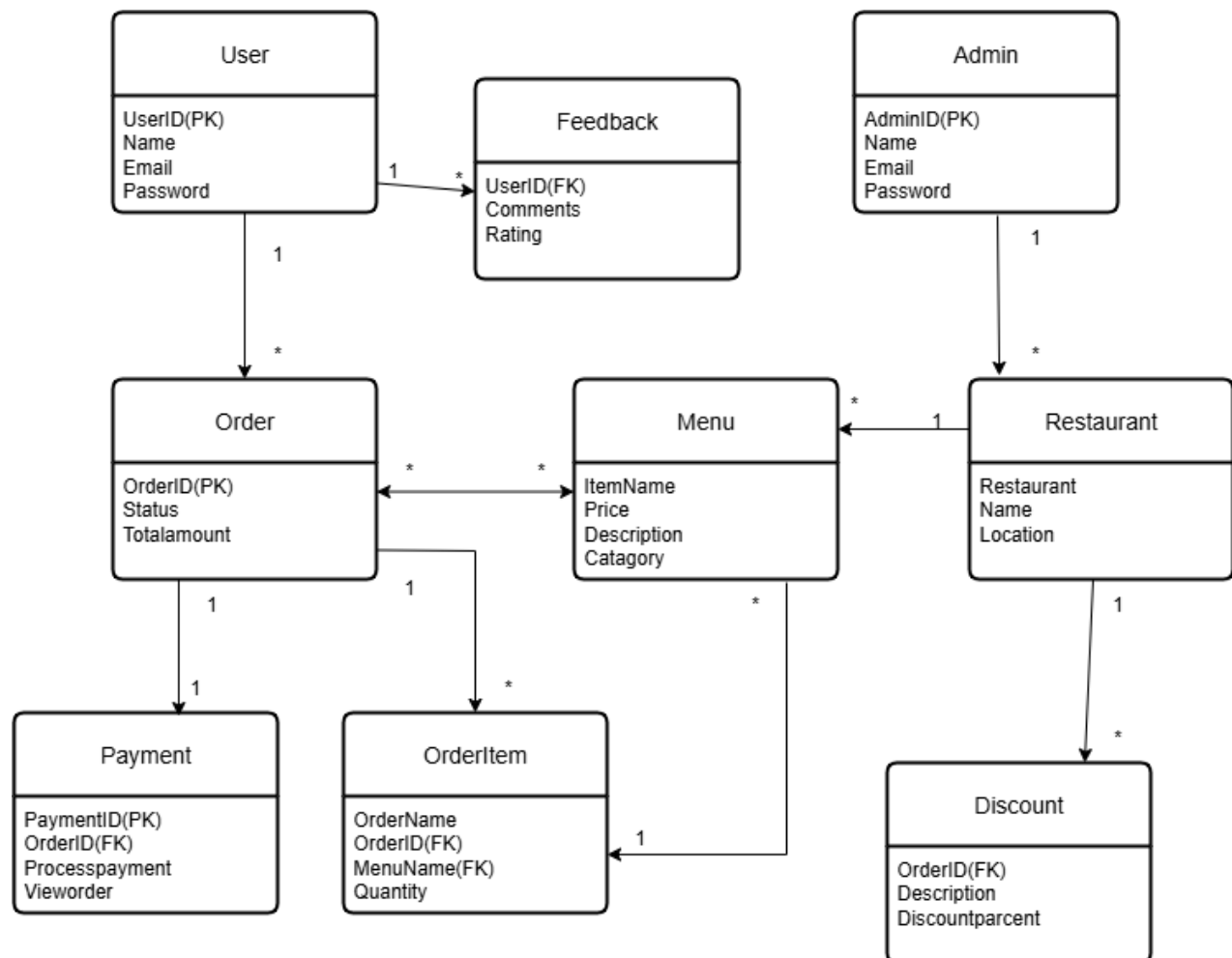


Figure 6: ER diagram

For example, one user can make several orders (one-to-many), but each order may include multiple choices in the menu (many-to-many via Order Item). Each order requires a single payment (one-to-one), yet a restaurant may have a large number of menu items. This ER diagram is important for constructing a well-structured database to support the operations outlined in the use case diagram.

4.1.3 Sequence Diagram

The sequence diagram illustrates the interaction between the user, frontend system, backend server, database, and admin in the Foodie Haven platform.

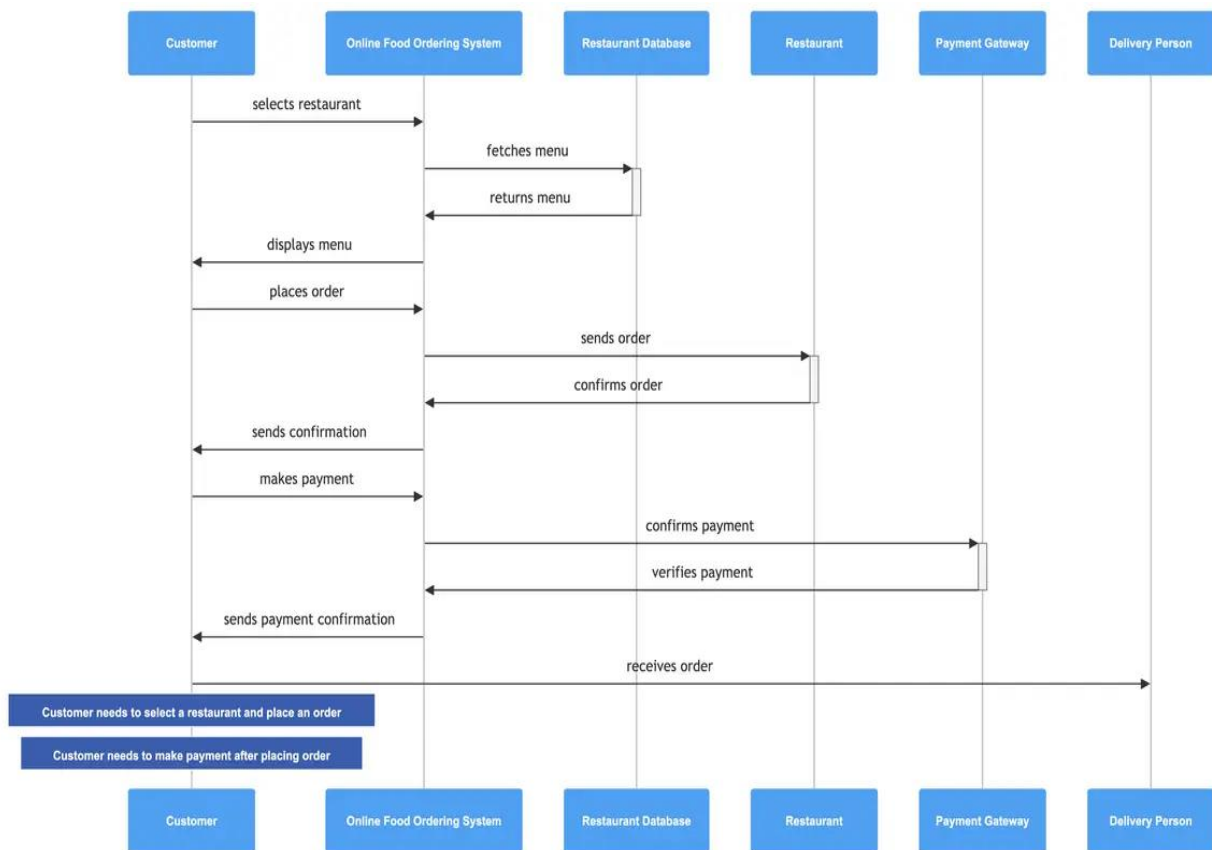


Figure 7:Sequence diagram

The process begins when the user registers or logs in, and the request is forwarded to the backend server for validation with the database. Once authenticated, the user can browse restaurants and menus, which are retrieved from the database and displayed on the frontend.

When the user places an order and selects a payment method, the backend processes the request, stores the order details in the database, and returns a confirmation message to the user. Finally, the admin can view and manage orders by updating their status, which is reflected back to the user in real time. This diagram clearly demonstrates how different components of the system communicate to ensure smooth order placement, management, and tracking.

4.1.4 Activity diagram

The Activity Diagram of Foodie Haven illustrates the step-by-step workflow, food delivery platform for both users and administrators. After successful authentication, the user can browse through available restaurants and food menus, select items, and add them to the cart. Once the Demo or screenshots of the application in development (Both Admin and User). The Activity Diagram for the Admin Part shows how the administrator interacts with the food delivery system to perform different management tasks. The process starts when the admin logs into the system using valid credentials. After a successful login, the admin enters the dashboard, where several options are available to manage system operations.

From the dashboard, the admin can choose to manage users, restaurants, or orders. In the Manage Users section, the admin can view all registered users, update their details, or deactivate suspicious accounts. In the Manage Restaurants section, the admin can add new restaurants, update existing information, or remove inactive ones. Through the Manage Orders option, the admin can track order progress, check payment details, and resolve delivery issues.

Additionally, the admin can monitor Reviews and Ratings submitted by users after receiving their food orders. This helps maintain the quality of service and identify areas for improvement. The admin can also generate various reports to analyze performance metrics, including sales data and customer satisfaction levels.

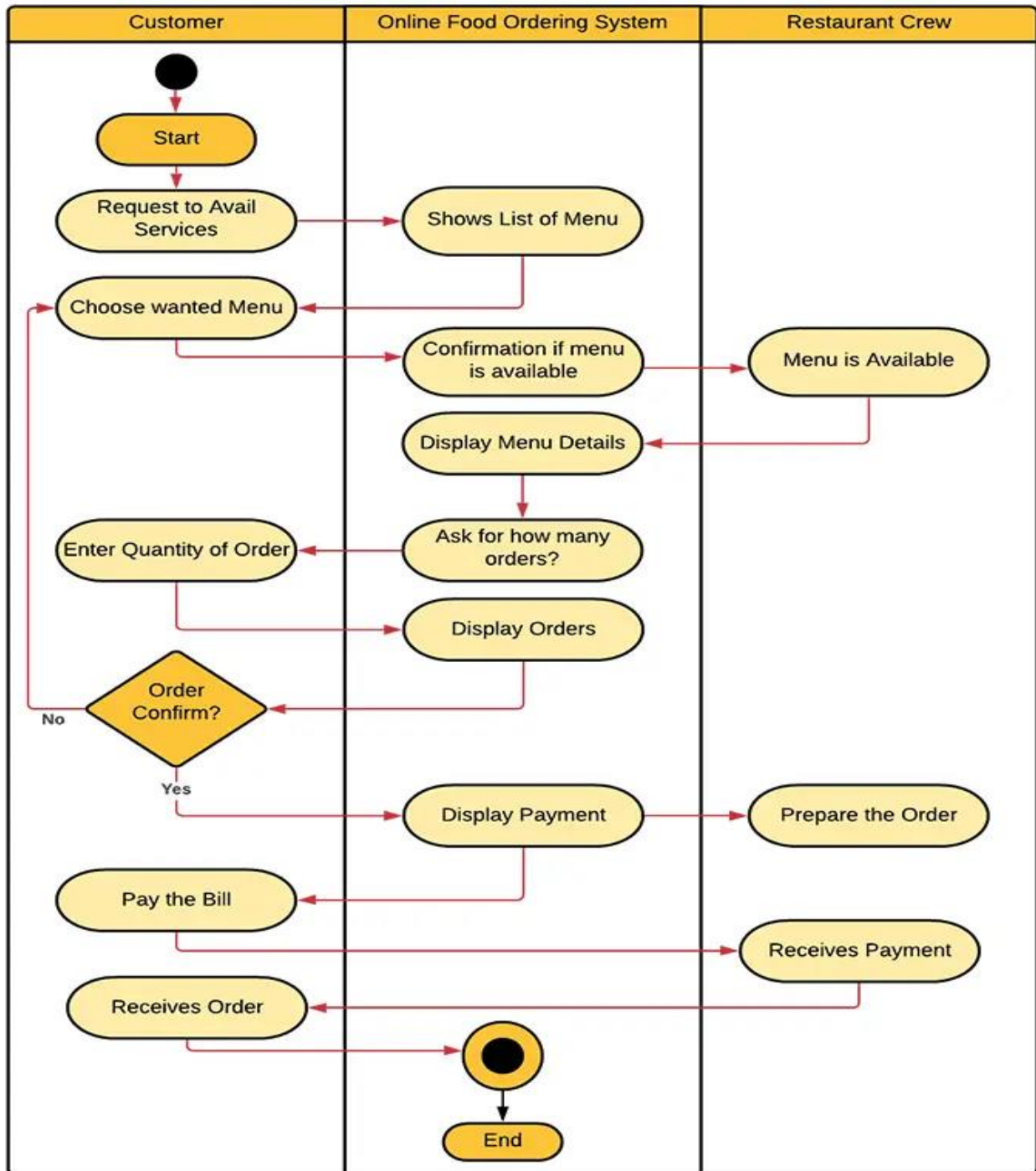


Figure 8:Activity Diagram

After successful authentication, the user can browse through available restaurants and food menus, select items, and add them to the cart. Once the Demo or screenshots of the application in development (Both Admin and User)

4.2 Implementation

4.2.1 Registration page

In the registration process, A user or admin completes a form with details including name, email, username, password, and confirm password. The frontend then sends this information to the backend using a POST /api/register request.

In the login process, the user or admin enters their email and password on the login page. The backend checks if the account exists and verifies the password by comparing it with the hashed one in the database.

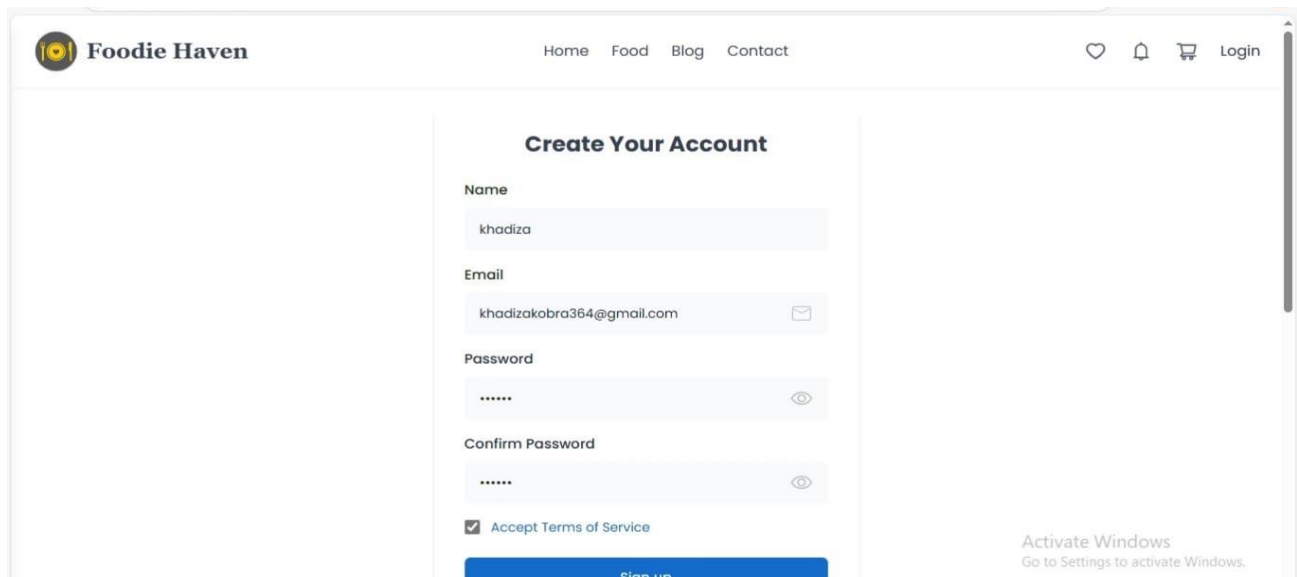
The image shows a web browser window displaying the 'Foodie Haven' registration page. The header includes the site logo, navigation links (Home, Food, Blog, Contact), and user icons (heart, bell, cart, login). The main content area is titled 'Create Your Account' and contains a registration form. The form fields are: Name (with the value 'khadiza'), Email (with the value 'khadizakobra364@gmail.com'), Password (masked with dots), and Confirm Password (also masked with dots). There is a checkbox for 'Accept Terms of Service' which is checked. A blue 'Sign up' button is at the bottom of the form. On the right side of the page, there is a 'Windows' watermark that says 'Activate Windows Go to Settings to activate Windows.'

Figure 9: Registration page

4.2.2 User Verification page

The Verification Page is designed to ensure that newly registered accounts are authentic and secure before granting full access to the system. The Verification Page is designed to ensure that newly registered accounts, whether for users or administrators, are authentic and secure. This page acts as a checkpoint before granting full access to the system, helping to prevent unauthorized entries and

potential security breaches. Typically, the verification process may include email confirmation, OTP (One-Time Password) verification, or admin approval. By implementing this step, the system maintains a secure environment while confirming the legitimacy of each account.

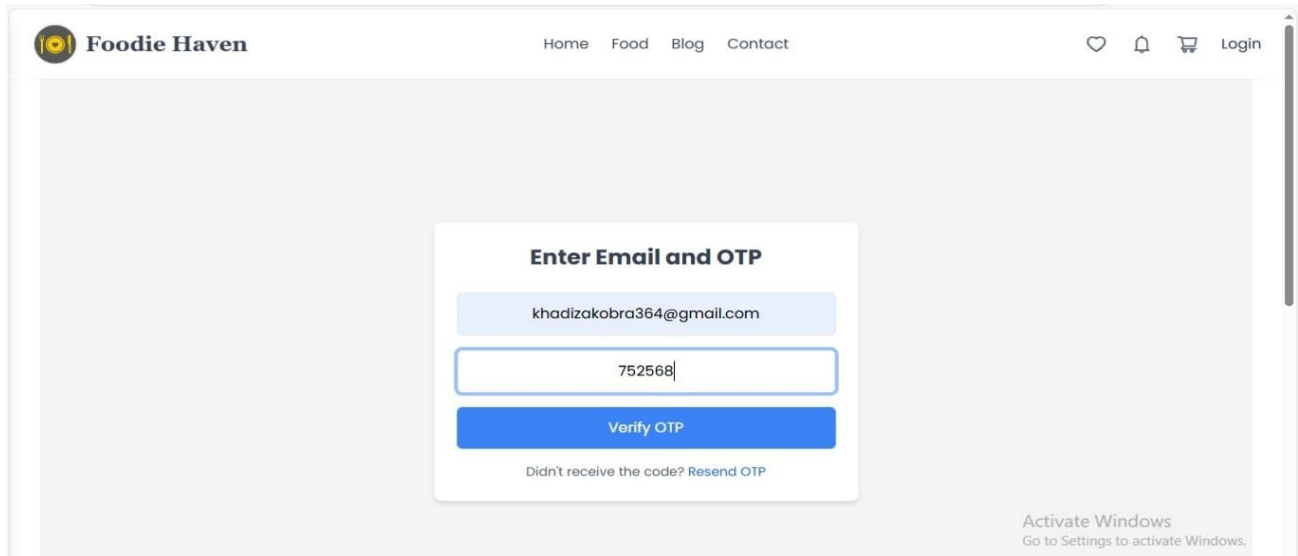


Figure 10: Verification page

4.2.3 Admin and user login page:

The Login Page enables both users and administrators to securely access the Foodie Haven platform. Users and admins enter their registered email and password, which are sent to the backend server for validation against the stored database records.

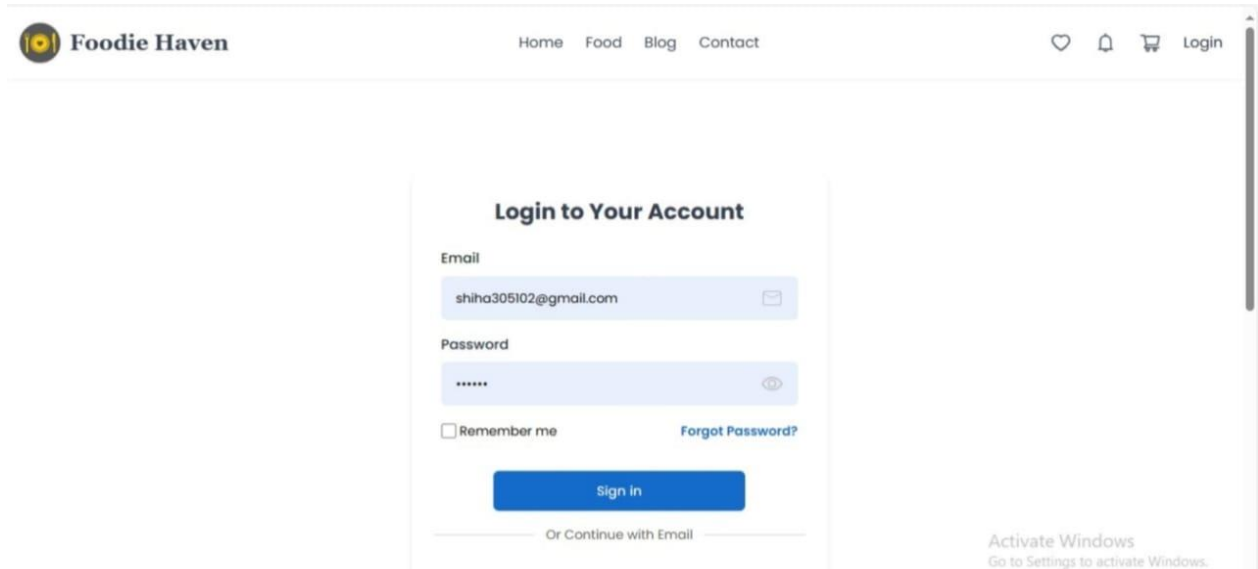


Figure 11: Admin login

If the login information is correct, the system grants access to the appropriate dashboard based on the account type. Regular users are redirected to the Home Page, where they can browse restaurants and place orders, while administrators are directed to the Admin Dashboard for managing restaurants, food items, and orders. In case of incorrect credentials, the system displays an error message, ensuring secure authentication.

4.2.4 Home page:

The Home Page is the main entry point of the Foodie Haven platform after login. On this page, users can see a list of available restaurants, along with their names, logos, and basic information such as location or cuisine type. By selecting a restaurant, users can explore its menu and place an order.

In addition to restaurants, the home page also includes a search bar and filters so that users can quickly find their favorite meals. Featured restaurants, popular dishes, and special offers may also be displayed to enhance user experience. Navigation options to the cart, order history, and profile are easily accessible, making the home page a central hub for browsing and ordering food.

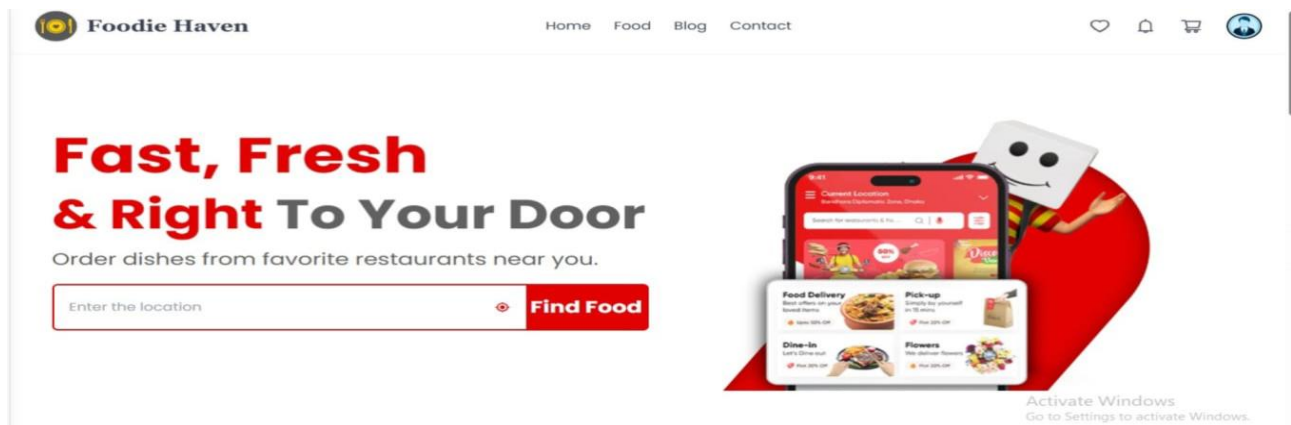


Figure 12: Home page

4.2.5 Browsing food

Users will be able to view all restaurants and explore all available food items along with their prices. They can freely browse and place orders based on their preferences. Description: We used CSS to create a simple layout for the food page. It showed where the food items, search bar, and filters would go. This design helped me follow a clear plan while building the page.

4.2.6 Add Filters (category, Price)

We added filters so users can choose food type (food, dessert), and price range. This helped users find their Favorite items faster.

4.2.7 Search bar functionality

We built a search bar where users can type a food name and see result. This made the food page faster and easier to use.

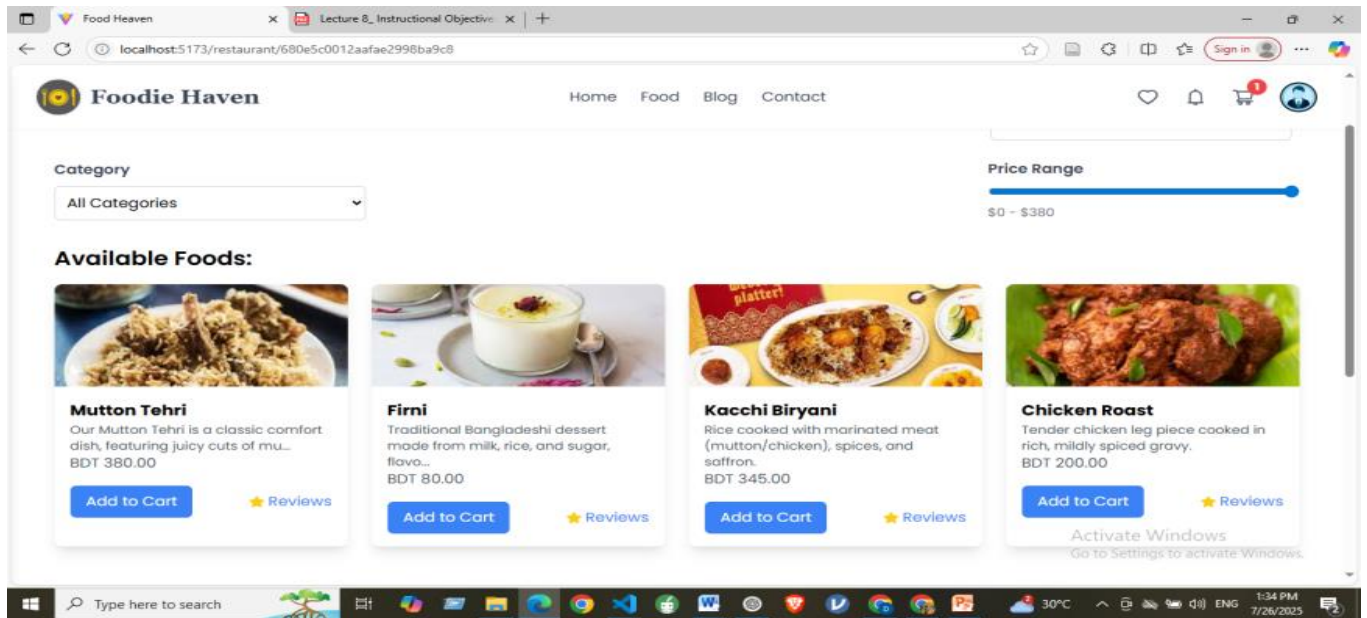


Figure 13: Browsing food

We displayed food like Mutton Tehri, Firni etc., with name, price, short description, and an “Add to Cart” button Connected UI with backend APIs to fetch food items dynamically.

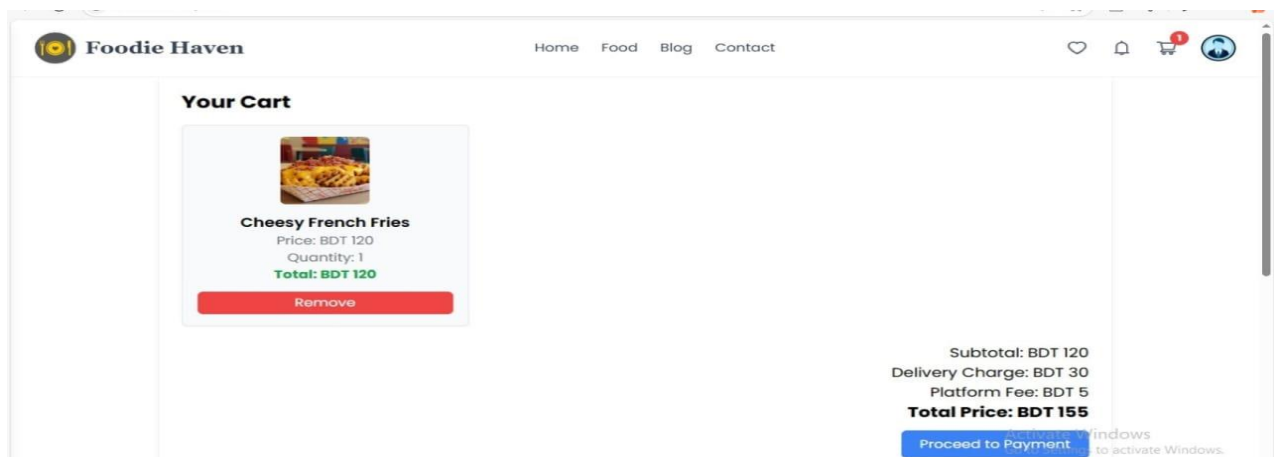
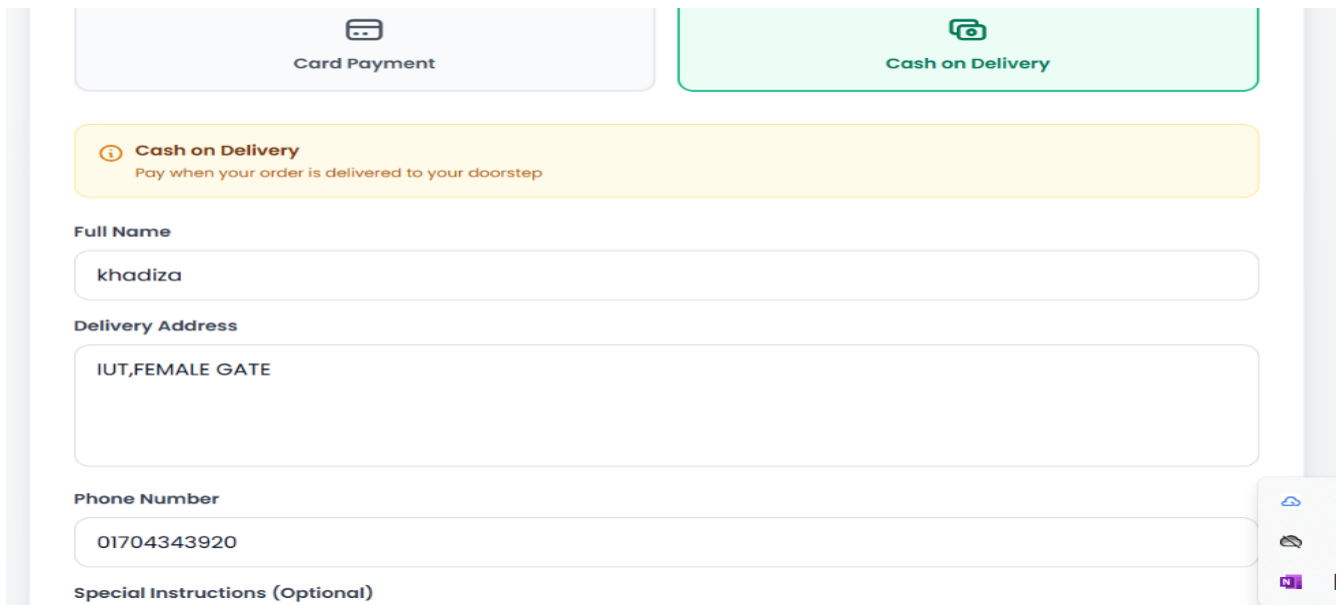


Figure 14: Place food order

4.3 Payment System:

The Food Delivery Platform has two ways to pay. Users can pay online with a debit or credit card, or they can pay in cash when the food is delivered. These options make it simple and convenient for everyone to order food.



The screenshot displays a payment interface for a food delivery platform. At the top, there are two buttons: 'Card Payment' (grey) and 'Cash on Delivery' (green, selected). Below the 'Cash on Delivery' button, a yellow banner reads 'Cash on Delivery' with an information icon and the text 'Pay when your order is delivered to your doorstep'. The form contains the following fields: 'Full Name' with the value 'khadiza', 'Delivery Address' with the value 'IUT,FEMALE GATE', and 'Phone Number' with the value '01704343920'. A 'Special Instructions (Optional)' field is present at the bottom but is empty. On the right side, a vertical toolbar contains icons for cloud storage, chat, and a menu.

Figure 15: Cash on Delivery

For those who prefer not to pay online, the platform also offers a **Cash on Delivery (COD)** option. In this method, users pay the delivery person directly in cash when their food arrives. This option is especially useful for customers who may not have a card or prefer traditional payment methods. By offering both online and cash payments, the platform ensures flexibility and convenience for all users.

The screenshot shows the 'Foodie Haven' website header with navigation links: Home, Food, Blog, and Contact. The main content area is a card payment form. It includes a 'Card Number' field with the value '5555 4444 4444 4444', an 'Expiry Date' field with '06 / 26', and a 'CVC' field with '123'. Below these is a 'Contact Information' section with a 'Phone Number' field containing '01704343920' and a 'Billing Address' field containing 'IUT, Female hall'. At the bottom is an 'Additional Message (Optional)' field with the text 'please give good food'.

Figure 16: Card payment

The platform allows users to pay online using debit or credit cards. When placing an order, users can securely enter their card details, and the payment is processed through a safe payment gateway. This method is fast and convenient, as users can complete the transaction instantly without handling cash. It also provides digital receipts for better tracking of expenses.

4.4 Order place:

Show users a clear list of their past orders with item names, quantity, price, and payment details. Created a clean design to show past orders. We made a simple page layout where users can see their previous orders. Each order shows the Order ID, status (like Completed), total price, and payment status (Paid).

4.4.1 Order History Page Layout:

Created a clean design to show past order. We made a simple page layout where users can see their previous orders. Each order shows the Order ID, status (like Completed), total price, and payment status (Paid).

Order ID	Status	Total	Payment	Items
#0qVyuEeQ	Completed	\$335	Paid	Firni Qty: 1 \$80 Write Review
#1ZAv5Pna	Completed	\$725	Paid	Mutton Kacchi - Half Qty: 1 \$220 Write Review
#07it7Sbt	Completed	\$415	Paid	Kacchi Biryani Qty: 2 \$345 Write Review

Figure 17:User show order list

After completing the payment, the user will see a **“Payment Successful”** message on the screen. This confirmation lets the user know that their order has been placed successfully and the payment was received.

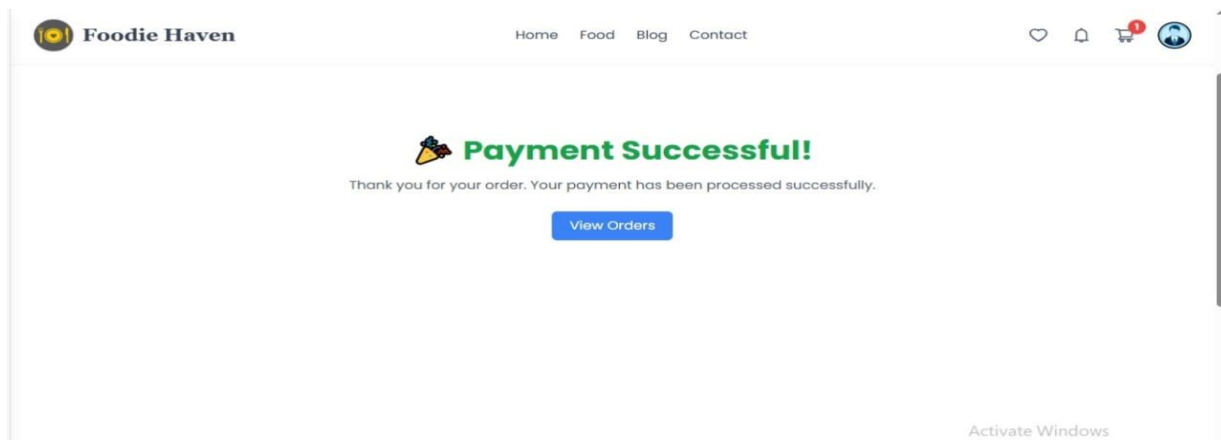


Figure 18: Order confirmation

4.5 Review and Rating:

The Review and Rating feature allows users to share their feedback after receiving their food orders. This helps other users make better choices and lets restaurants improve their services

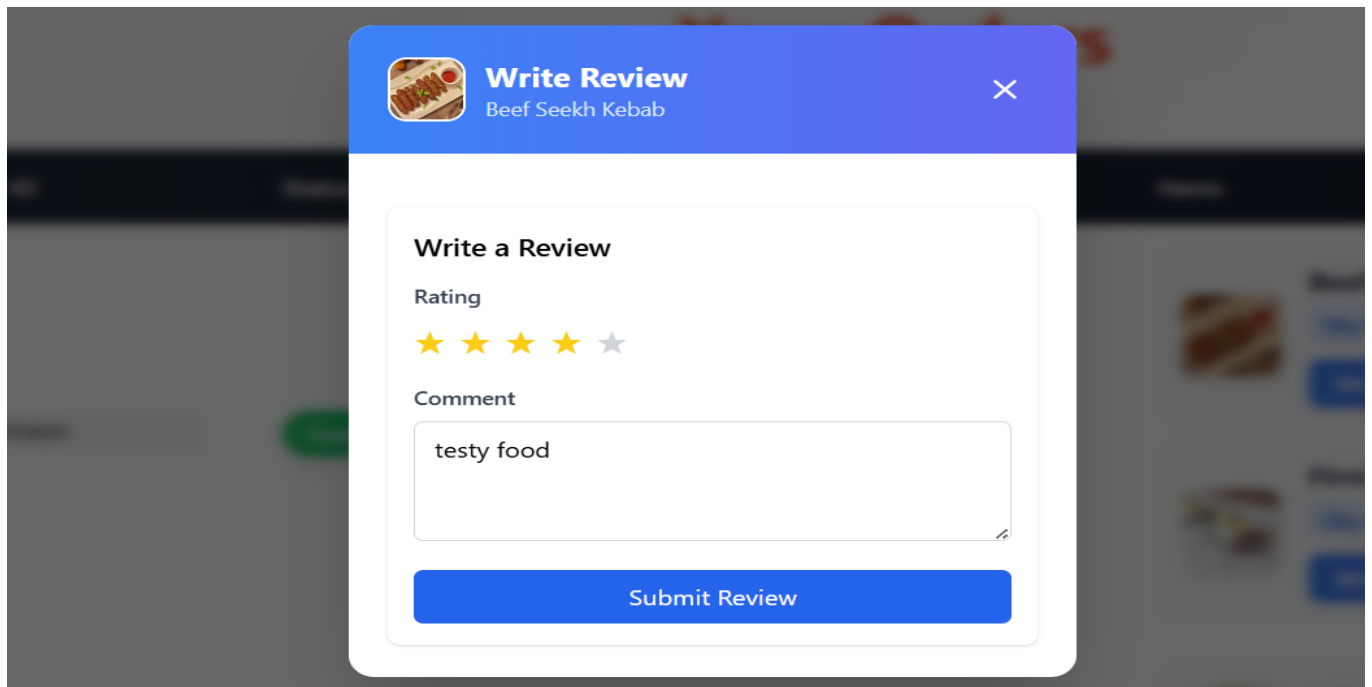


Figure 19: Review and Rating

Admin part of the system is designed to manage and monitor all activities within the food delivery application. It provides the administrator with full control over user accounts, restaurant information, orders, payments, and reviews. The admin dashboard ensures smooth operations and helps maintain data integrity across the platform.

4.6 Admin Part

The admin can log in and add restaurants to the system. On the Add Restaurant page, the admin fills in details such as the restaurant name, location. When the form is submitted, the backend checks the data and saves it in the MongoDB database. The admin can also add or delete restaurants from the database anytime as needed.

Made a form to add new food items. We designed a form with fields like Food Name, Price, Description, Category, Image URL, Stock, and Discount. This form helps restaurant owners add all details of a food item in one place. It also includes **Add** and **Cancel** buttons for user control.

4.6.1 Create Restaurant

Made a form to add new food items. We designed a form with fields like Food Name, Price, Description, Category, Image URL, Stock and Discount. This form helps restaurant owners add all details of a food item in one place. It also includes Add and Cancel buttons for user control.

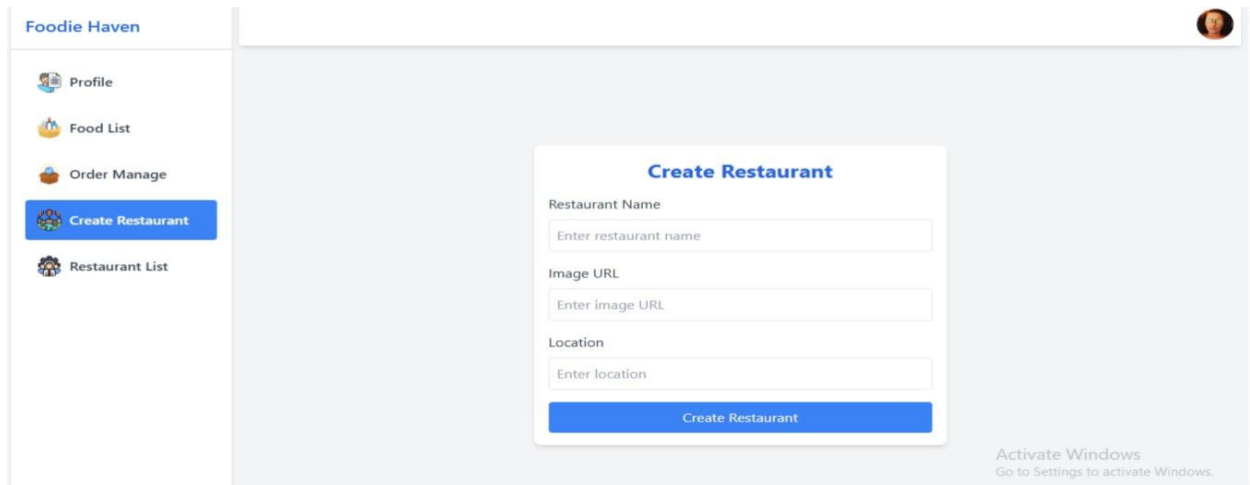
The screenshot shows a web application interface for 'Foodie Haven'. On the left is a sidebar with navigation links: Profile, Food List, Order Manage, Create Restaurant (highlighted in blue), and Restaurant List. The main content area displays a 'Create Restaurant' form. The form has three input fields: 'Restaurant Name' with placeholder text 'Enter restaurant name', 'Image URL' with placeholder text 'Enter image URL', and 'Location' with placeholder text 'Enter location'. Below these fields is a blue 'Create Restaurant' button. In the bottom right corner of the application window, there is a Windows watermark that says 'Activate Windows Go to Settings to activate Windows.'

Figure 20:Create Restaurant

4.6.2 Add Form Validation

Checked that all form fields are filled correctly. We added validation using JavaScript to make sure users enter valid data, like price must be a number and image URL must not be empty. This helped prevent errors before saving the food item.

4.6.3 Add Food

Let users add food item picture. We added a feature **to** upload food images, either by using an image link or uploading a file. This made the menu items more attractive and helped use see what the food looks like. Admin can add food in the different restaurant.

The screenshot shows a web application interface for 'Foodie Haven'. On the left is a sidebar menu with icons and labels for 'Profile', 'Food List', 'Order Manage', 'Create Restaurant', and 'Restaurant List' (which is highlighted in blue). The main area displays a modal form titled 'Add Food to Radhuni Restaurant'. The form contains several input fields: 'Food Name' (with a placeholder 'Food Name'), 'Price (BDT)' (with a placeholder 'Price'), 'Description' (with a placeholder 'Short description'), 'Category' (with a placeholder 'Category'), 'Image URL' (with a placeholder 'Image URL'), 'Stock' (with a placeholder 'Stock'), and 'Discount (%)' (with a placeholder 'Discount'). At the bottom right of the form are two buttons: a grey 'Cancel' button and a green 'Add' button.

Figure 21:Add food

4.6.4 Connect Form to Backend

Linked form to backend to save and update food items. We used APIs to connect the form with the backend. This allowed restaurant owners to add, edit, or delete menu items, and all changes were saved directly to the database.

4.6.5 Order management

Gave admin access to view and update all user orders. Gives admin access to view and update all user orders. I made a system where the admin can see all users' order history, including what food items they ordered, their quantity, and total price.

From this panel, the admin can also change the order status to:

1.Pending 2. Delivery 3. Completed 4. Rejected

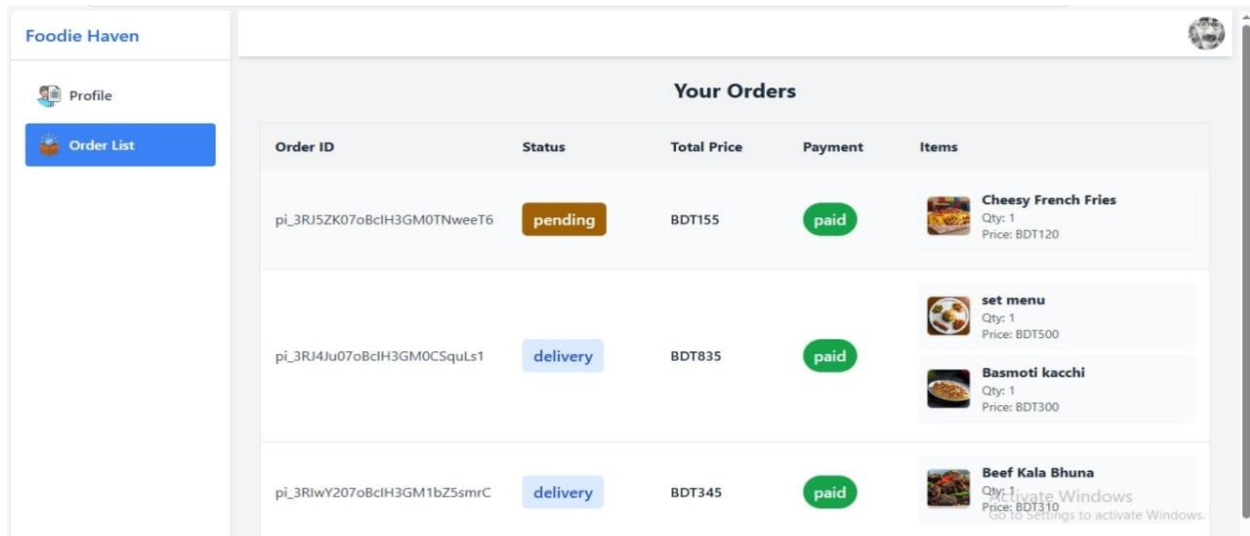


Figure 22:View Order

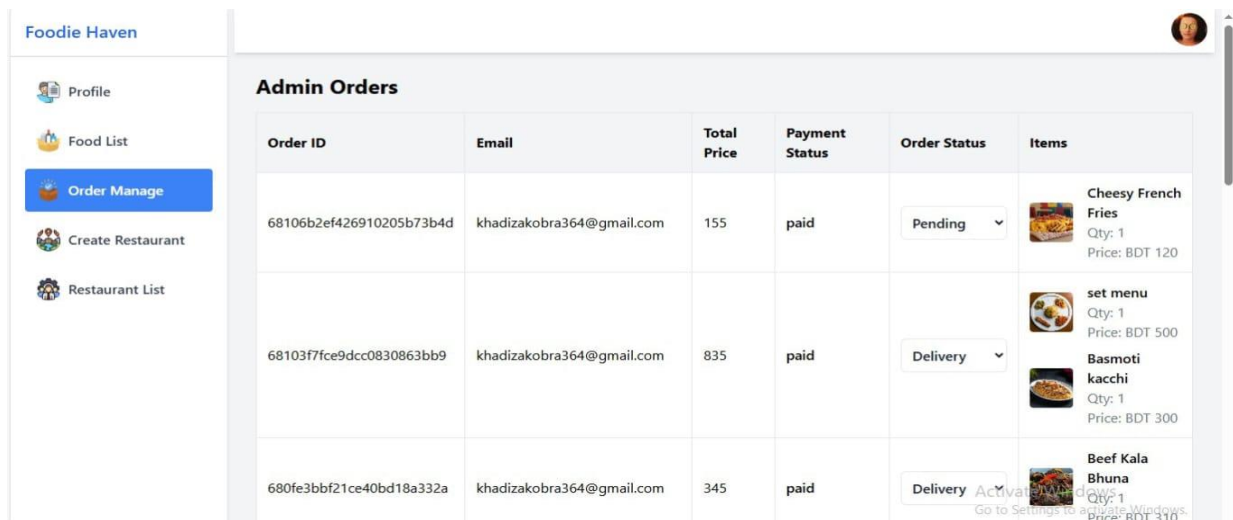


Figure 23: Manage order

4.7 GitLab

For the development of our food delivery application, we used **GitLab**, a web-based platform that provides powerful tools for version control, source code management,

and continuous integration. GitLab helped us maintain a collaborative environment where each team member could work on their own part of the project from their own computer, while still sharing progress seamlessly with the entire group.

Through GitLab, we managed our source code efficiently, tracked changes, and ensured that different modules of the food delivery system such as customer interface, restaurant management, and delivery partner module were properly integrated. The platform also supported continuous integration and deployment, which allowed us to test features and fix bugs faster. By using GitLab, we streamlined our development lifecycle, improved team communication, and ensured a more organized and efficient workflow for building the food delivery app.

Chapter 5 CHALLENGES

Throughout the development of the Foodie Haven project, several significant challenges were encountered:

- **Integrating Complex Features:** Implementing advanced functionalities such as real-time order tracking, secure payment processing, and an efficient admin panel proved to be more complicated than initially expected.
- **Limited Initial Knowledge:** At the beginning of the project, the team lacked a comprehensive understanding of how to design and develop a fully functional food delivery system.
- **Limited Programming Experience:** Since our team had relatively little prior experience with certain technologies like TypeScript, React Native, and backend frameworks, building a large-scale, integrated system posed difficulties.

Chapter 6 CONCLUSION

The Foodie Haven project effectively delivers a comprehensive meal delivery service with a focus on speed, simplicity, and customer happiness. We hope to improve the food ordering process for both customers with restaurant owners by combining a seamless user interface, a sophisticated admin the panel, and review and rating. Despite some problems owing to a lack of technical understanding, cooperation and devotion enabled us overcome them. In the future, we intend to develop recommendation system with new features to make it more smarter and more efficient.

References

1. Saad, A. T. (2020). An empirical investigation into the determinants influencing online food delivery services in Bangladesh. *The British Food Journal*.
<https://doi.org/10.1108/BFJ-05-2020-0449>
2. Halim, S. B. (2023). An analytical study on Bangladeshi consumers' willingness to use food delivery applications. *International Journal of Science and Business*. Retrieved from <https://www.ijstab.com/volume-21-issue-1/5412>

Appendices

Source code:

<https://github.com/khadizaqqqq/Foodie-Haven-main>