# INTRODUCTION

## Project Description

**Cricket teams often face challenges in deciding whether to bat or field after winning the toss, as the decision depends on multiple factors such as ground conditions, weather, and historical performances. Whereas Fans and teams lack accurate, real-time predictions of match outcomes based on evolving game scenarios and player/team performance metrics.**

**This project aims to take the analytical experience of ongoing cricket matches to the next level by providing real-time scores of all ongoing matches, dynamic statistics, and insights for enthusiasts, analysts, and fans. Through the display of live match updates together with changing metrics, the platform improves engagement of the game. Pre-match information provides strategic advice, e.g., whether to bat or bowl first, from historical venue statistics, weather conditions, and past performa nce records. The information seeks to provide teams and supporters with a better, data-driven understanding prior to the first ball being bowled.**

**Apart from pre-match analysis, the project proposes a machine learning model for predicting the chances of a successful chase in the second innings of a T20 match. The model updates continuously after each ball, using current data to modify win possibilities based on the match status. Through the use of Machine Learning, the platform not only enhances the viewing experience but also offers analysts more insightful, predictive understandings of matches, and revolutionizes the way T20 cricket is watched and comprehended.**

**While this project provides a robust solution for predicting and analyzing T20 cricket matches, it does have some limitations. Currently, predictions focus only on the second innings, assessing the chasing team's probability based on factors like required run rate, wickets in hand, and game dynamics. Additionally, toss recommendations rely on historical patterns, which might not always reflect sudden environmental changes. Expanding the model to include first-innings predictions and adaptive factors could further improve its accuracy and usefulness.**

**The purpose of this project is to build a real-time cricket scoring and prediction platform that keeps fans engaged with instant match updates and AI-powered win probability predictions. Fans can track live scores, follow dynamic match insights, and enjoy a more interactive viewing experience. Analysts benefit from data-driven performance metrics, while advertisers and industry stakeholders gain opportunities to reach a passionate audience.**

# LITERATURE SURVEY

## Related Work

* + 1. TITLE- *Winner Prediction in One-Day International Cricket Matches Using Machine Learning Framework: An Ensemble Approach* AUTHOR- Manoj Ishi, Dr. Jayantrao Patil et al.  
       PUBLISHED YEAR- 2022

SUMMARY- This research aims to develop machine learning models to predict the winner of one-day international (ODI) cricket matches before the game begins, providing insights for team management and sports analysts. Using historical data from 1693 ODI matches (2006–2019), the study evaluates batting and bowling strength, run-scoring patterns, and overall team metrics to build predictive models. Various machine learning algorithms, including Logistic Regression, SVM, and ensemble methods like voting and stacking classifiers, were employed, with feature selection techniques enhancing accuracy. The best models achieved up to 96.31% accuracy, demonstrating the potential of machine learning to handle cricket's unpredictability and offering practical value for strategy optimization. Future work includes incorporating additional features and extending the methodology to other cricket formats or sports analytics domains.

* + 1. TITLE- Prediction of IPL Match Outcome Using Machine Learning Techniques

AUTHOR- Srikantaiah K C, Aryan Khetan et al.

PUBLISHED YEAR- 2019

SUMMARY- This research focuses on predicting the outcomes of Indian Premier League (IPL) matches using machine learning models based on nine years of historical data. By analyzing datasets that include team performance (home/away), match details, player statistics, and ball-by-ball deliveries, features such as win percentages, toss impacts, and player metrics were extracted. Machine learning models like Random Forest, SVM, Logistic Regression, and K-Nearest Neighbor were implemented, with Random Forest achieving the highest accuracy of 88.10%. The models were trained on 70% of the data and tested on 30%, with cross-validation ensuring reliability. This study offers valuable insights for team managers, analysts, and fans, enabling informed decisions about match strategies and outcomes. Future work includes evaluating individual player performances and integrating additional features to improve prediction accuracy and versatility.

* + 1. TITLE*-* *Cricket Match Analytics and Prediction Using Machine Learning*

AUTHOR- Param Dalal, Hirak Shah, Tej Kanjariya, Dhananjay Joshi

PUBLISHED YEAR- 2024

SUMMARY- This research explores cricket match prediction using machine learning, focusing on outcomes during the second innings by analyzing factors like target, runs left, wickets fallen, and player-specific metrics. A key innovation is a custom "Player Consistency" formula, which combines traditional cricket statistics with dynamic ratings to enhance predictive accuracy. The study evaluates models such as Random Forest, SVM, Logistic Regression, and Naive Bayes, with Random Forest achieving the highest testing accuracy of 89.82%. By leveraging techniques like feature extraction and deep learning for weight optimization, the research demonstrates significant advancements in cricket analytics, providing actionable insights for players, teams, and fans while identifying gaps and opportunities for future enhancements in prediction models.

* + 1. TITLE*- Predicting IPL Victories: An Ensemble Modeling Approach Using Comprehensive Dataset Analysis.*

AUTHOR- Pritpal Singh, Dr. Jatinder Kaur, Lovedeep Singh

PUBLISHED YEAR- 2024

SUMMARY- This research applies machine learning techniques to predict IPL match outcomes using factors like player statistics, team performance, venue, and weather. Algorithms such as Decision Trees and Random Forest were tested, with the ensemble model achieving **99% accuracy**, outperforming individual models. The study demonstrates the potential of ensemble approaches in improving prediction accuracy and suggests integrating more contextual features for enhanced performance in future applications.

* + 1. TITLE*-**Outcome Prediction of ODI Cricket Matches Using Decision Trees and MLP Networks*

AUTHOR- Jalaz Kumar, Rajeev Kumar, Pushpender Kumar

PUBLISHED YEAR- 2018

SUMMARY- This research uses machine learning models like Decision Trees and Multilayer Perceptron (MLP) Networks to predict outcomes of ODI cricket matches based on factors such as past team performance, venue, innings order, and home advantage. Using data from 3933 ODI matches, the study trained models on pre-game features to develop a prediction tool named **CricAI**, which forecasts match results with reasonable accuracy. MLP Classifier achieved an accuracy of **57.4%**, slightly outperforming Decision Trees at **55.1%**. The findings highlight the effectiveness of these classifiers in capturing patterns from historical data and suggest potential improvements through team composition analysis and application to other sports.

## Existing Application

* + 1. TITLE- *CricViz*

RELEASE YEAR- 2015

SUMMARY- CricViz is a leading cricket analytics platform that provides in-depth insights, match predictions, and performance analysis using advanced statistical models and machine learning techniques. It offers real-time win probability, player performance metrics, and historical data analysis for various formats like Test, ODI, and T20. CricViz’s predictive models help broadcasters, analysts, and fans better understand match dynamics and player impact. Its features, like the Win Predictor and PitchViz, make it a popular tool in the cricket analytics domain.

* + 1. TITLE- *ESPNCricinfo StatsGuru*

RELEASE YEAR- 1993

SUMMARY- ESPNCricinfo StatsGuru is a comprehensive cricket statistics and analysis tool that provides detailed historical and live match data across all formats. It allows users to filter and analyze player performances, team statistics, and match outcomes with customizable queries. Widely used by analysts, broadcasters, and cricket enthusiasts, StatsGuru offers insights into trends, records, and performance patterns, making it a reliable resource for cricket-related research and prediction.

# HARDWARE AND SOFTWARE REQUIREMENTS

## Hardware requirements

To develop CricketVerse application processor i3 equivalent and above is required. RAM

should be 8GB. Hard disk should be 40GB.

## Software requirements

To develop CricketVerse application’s Front-end ReactJS 18.0.0)is required. To develop the Back-end of the application Node.js with Springboot(3.9) and Hybernate framework is required. To store information in database MySQL is required. Operating System windows 7 and above, macOS or Linux is required. It uses technologies like Java 1.8, Servlet, JSP and Tomcat Apache(8.5).

# SOFTWARE REQUIREMENTS SPECIFICATION

## Functional requirement

* + 1. **Users**

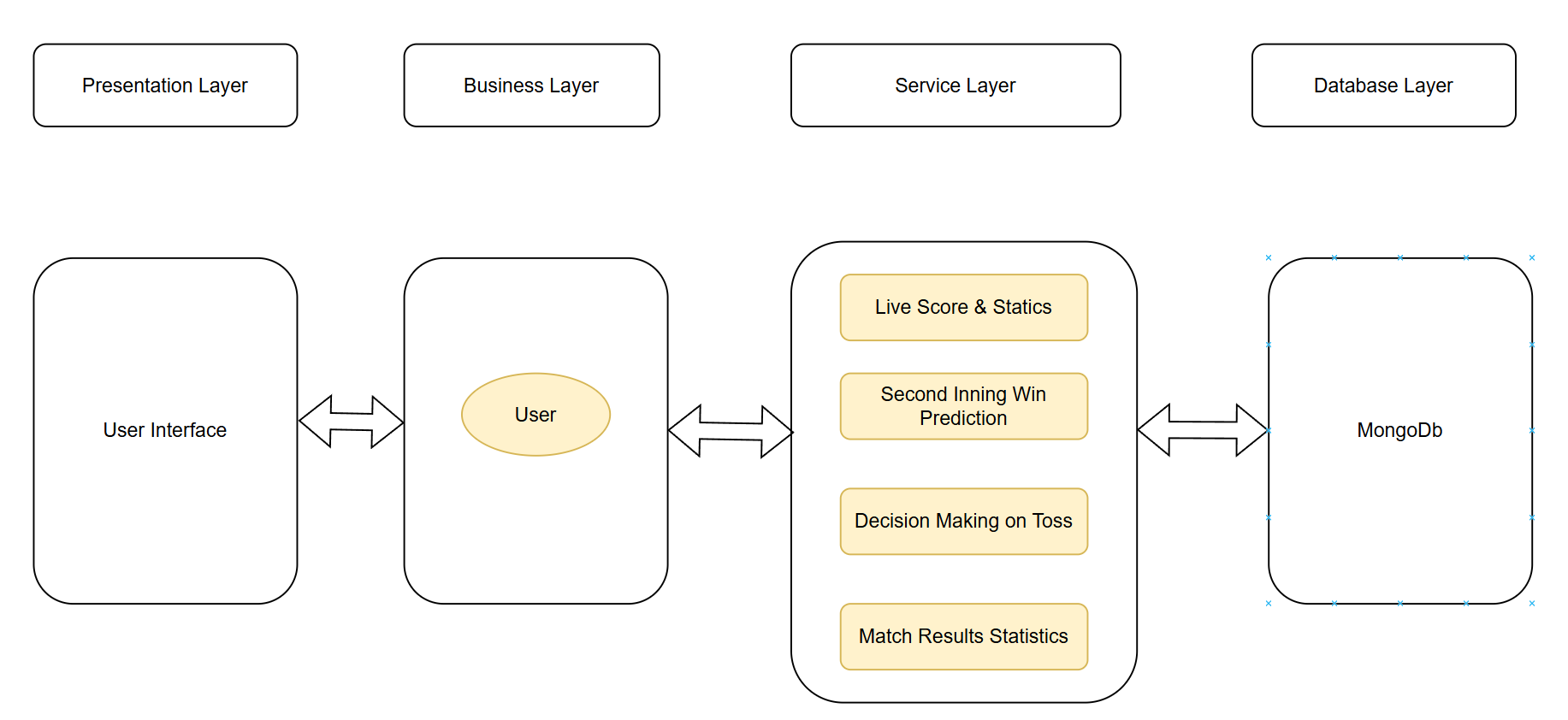
A platform exists that enables cricket fans to connect directly with their favorite games as they have never experienced before. CricketVerse brings this experience to life by supplying users with real-time updates together with insightful predictions and interactive match analytics. The platform delivers real-time Live Score and Statistics Display that shows ball-by-ball scores with essential match events along with metrics including run rates and required run rates and projected scores. CricketVerse gives users real-time delivery tracking for all games including team matches and high-profile final contests. The pleasure continues beyond this point. Our platform activates Second Inning Win Prediction because real-time data enables us to estimate whether the chasing team can achieve the target. The predictive model updates its assessment throughout the game by considering the number of wickets remaining along with current run rates and individual player performances for an informative outlook on the match progression. The Toss Decision Recommendation service from CricketVerse launches before the first ball is bowled. The platform examines multiple data points comprising historical performance data together with local match statistics along with environmental elements to determine which team should use bat or bowl first. CricketVerse generates Match Result Statistics which present finalized match summaries after determining every ball in the match has been delivered. Match enthusiasts along with scholars and sports analysts gain game reliving capabilities through detailed systems which provide crucial performance highlights that expose potential patterns for forthcoming matchups. The cricket domain provides enhanced cricket insights in addition to delivering both inquisitive analytics and up-to-date information suitable for enthusiasts.

## Non-functional requirement

When it comes to user experience, CRICKETVERSE prioritizes ease and accessibility. Its interface is designed to be intuitive and user-friendly, making it simple for anyone to navigate through menus, place orders, and track their food's journey. This eliminates frustration and ensures a smooth experience, regardless of technical expertise. But this application doesn't stop there. Understanding the crucial role of constant availability, the system leverages the power of the internet to be present 24/7. As long as you have a stable internet connection, you can access this application and satisfy your cravings anytime, anywhere. This eliminates the limitations of physical storefronts and caters to busy schedules or late-night hunger pangs. Furthermore, CRICKETVERSE is platform-agnostic, meaning it's not restricted to a specific device or operating system. You can access it seamlessly on any browser, be it on your laptop, phone, or tablet. This flexibility ensures wider reach and convenience, adapting to individual user preferences and technological comfort zones. However, even the most well-designed system requires continuous improvement. To keep users engaged and maintain its competitive edge, this application acknowledges the need for regular upgrades. By incorporating user feedback and implementing innovative features, the platform can stay ahead of the curve and attract a wider audience. In essence, this application emphasis on usability, availability, and platform independence creates a user-centric experience that is accessible, convenient, and adaptable, while its commitment to continuous improvement ensures its long-term relevance and success.

# SYSTEM DESIGN

## Architecture Diagram

****

*Figure 1 Architecture diagram of CricketVerse system*

The application's architecture is built on a layered model, ensuring clarity, efficiency, and scalability. At the *Presentation Layer* there is user interface. This is the face of the app, where Customers and Admin interact with the platform. We prioritize a simplistic and intuitive design with clear navigation, user-friendly interfaces for both Customers and Admin, and optimized performance for a seamless experience. Next layer in architecture diagram is *Business Layer* This layer acts as the brain of the application, handling core functionalities and managing user interactions.

User Interface:

This module consists of two sub-modules:

*Customer UI*: This sub-module handles all interactions between the customer and the application. It allows customers to browse menus, select items, place orders, track their orders, and manage their account information.

*Admin UI:* This sub-module handles all interactions between the restaurant administrator and the application. It allows administrators to manage their menus, orders, and delivery partners.

Business Layer: This module is responsible for handling all of the business logic of the application. It includes sub-modules for:

Order Management: This sub-module handles all aspects of order processing, from taking orders to sending them to the restaurant and delivery partner.

Inventory Management: This sub-module keeps track of the inventory of food items at each restaurant.

Payment Processing: This sub-module handles all payment processing, including online payments and cash on delivery.

Delivery Management: This sub-module assigns orders to delivery partners and tracks their progress.

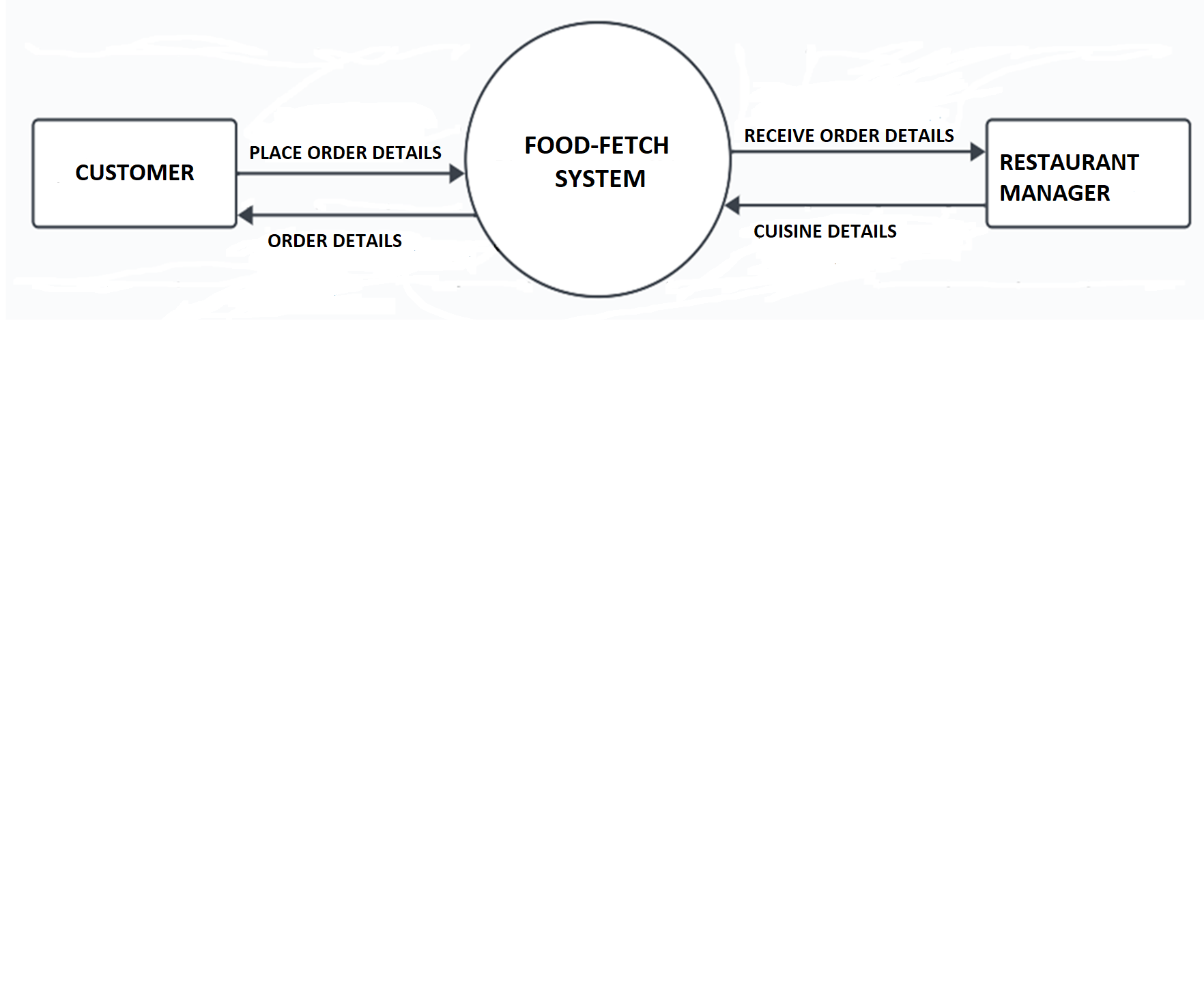
Notification: This sub-module sends notifications to customers and restaurants about the status of their orders.

Search: This module allows customers to search for restaurants and food items.

Data Service: This module is responsible for interacting with the database. It includes : Customer Data, Restaurant Data and Order Data

The clear separation of functionalities ensures ease of development, maintenance, and future enhancements. This architecture empowers the app to fulfill its mission of fostering impactful giving and creating a positive social impact.

## Context Flow Diagram

****

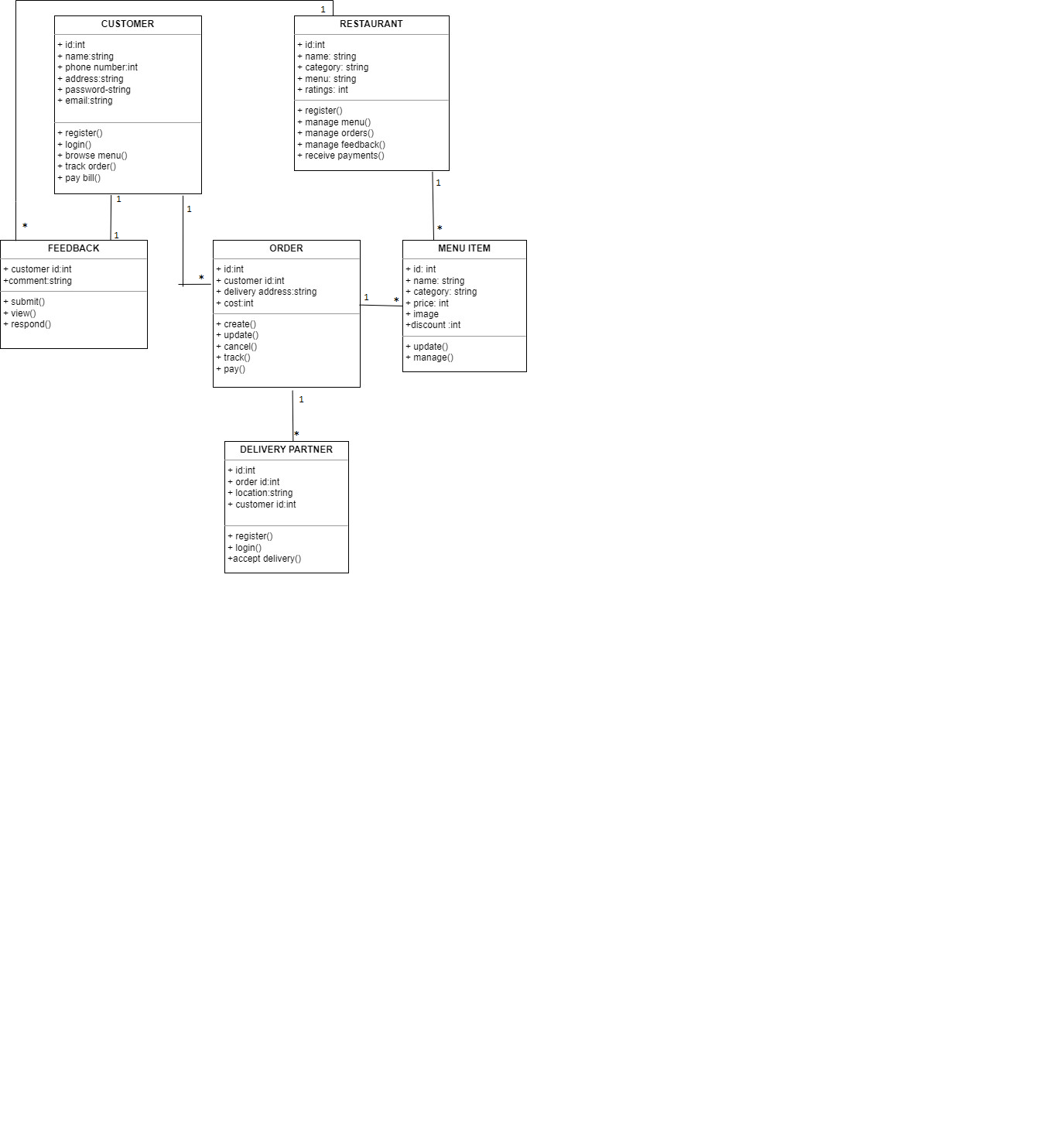
*Figure 2 Context Flow Diagram of Food Fetch System*

Customers are the initiators of the system, placing orders through the CricketVerse mobile app or website. This order information, including details like the items ordered, desired delivery location, and any special requests, is then sent to the CricketVerse system. The CricketVerse system plays a central role in facilitating communication and data exchange. It receives orders from customers and transmits them to the relevant restaurants. Additionally, it receives restaurant data, such as menu items, availability, and pricing, which it stores and makes accessible to customers through its user interface. This allows customers to browse menus, compare options, and make informed decisions before placing their orders.

Restaurants, upon receiving orders from the CricketVerse system, begin preparing the food items. Once the order is ready, the restaurant transmits its completion status back to the CricketVerse system. The system then updates the customer with the order's status and estimated delivery time. Finally, when the food is prepared, the restaurant hands it over to a delivery personnel, who is responsible for delivering it to the customer's specified location.

# DETAILED DESIGN

## Class Diagram

****

*Figure 3 Class Diagram of Food Fetch*

The class diagram of CricketVerse illustrates the core entities and their interactions within the application's architecture. It provides a structural foundation for understanding the application's functionality and data management. Key Classes and Their Functions:

Class Customer has Attributes like id, name ,phone number, address, password ,email and has Functions like register(): Creates a new customer account, login(): Allows a customer to access their account, browse menu(): Enables customers to view restaurant menus, select item(): Lets customers choose specific food items,

place order(): Initiates the order process, track order(): Provides updates on the order's status, pay

bill(): Facilitates payment for the order.

Class Restaurant has Attributes like id, name, category, menu and ratings and Functions like manage menu(): Allows restaurants to create, edit, and update their menus, manage orders(): Enables restaurants to view and process incoming orders, manage feedback(): Allows restaurants to address customer reviews, receive payments(): Processes payments from customers.

Class Feedback has Attributes like customer id, comment and Functions like submit(): Allows customers to submit their feedback, view(): Enables restaurants to view customer feedback ,respond(): Allows restaurants

to address feedback directly

Class Order has Attributes like id, customer id, delivery address and cost and Functions like create(): Initiates a new order, update(): Allows for modifications to the order, track(): Provides real-time status updates on the order, manage(): Enables restaurants to oversee order processing

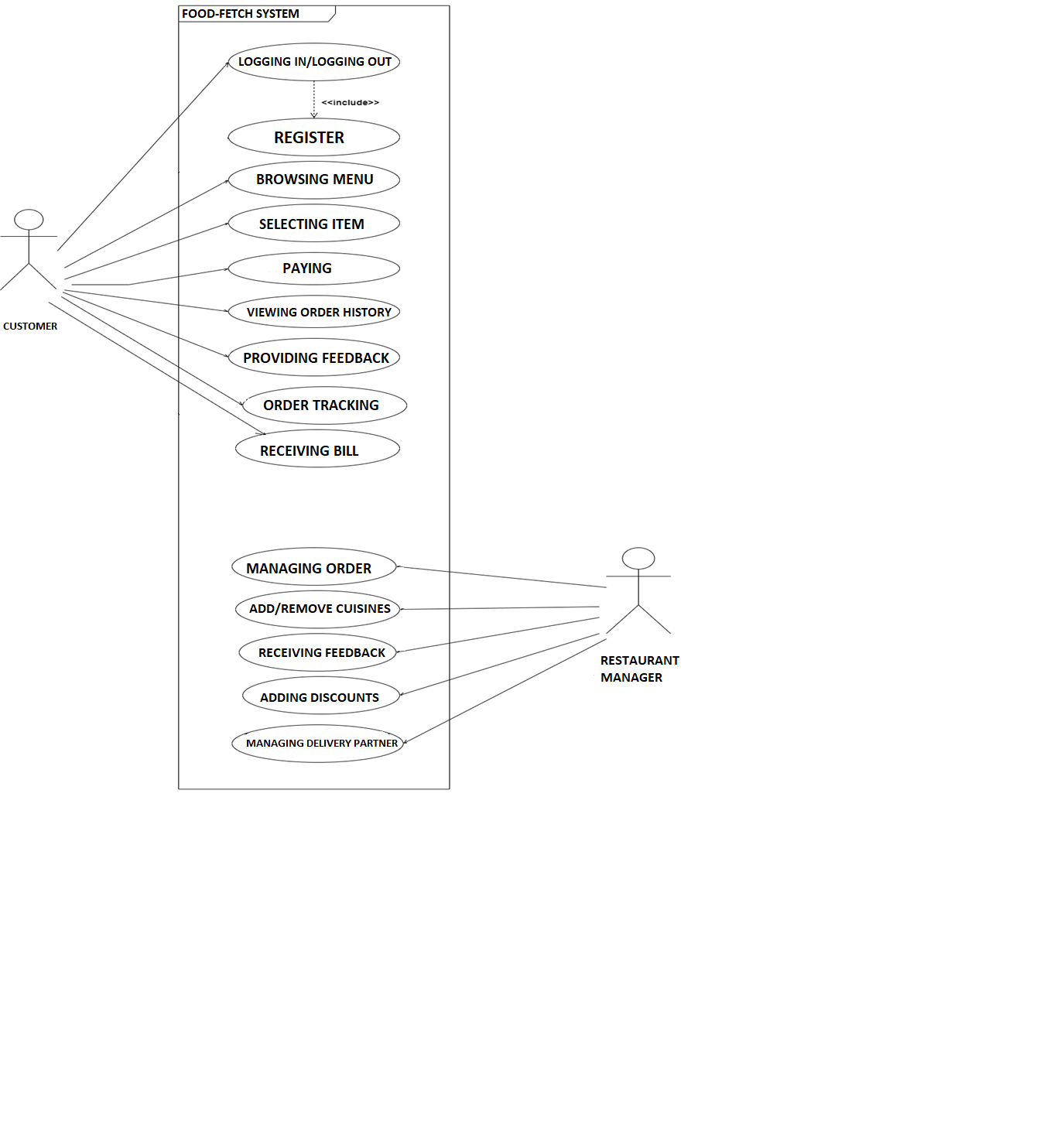
Class Delivery Partner has Attributes like id, order id and location and Functions like register(): Allows delivery partners to sign up with the system, login(): Enables delivery partners to access their account,

accept delivery(): Assigns a delivery partner to an order

Class Item has Attributes like id, price, name, category, discount, image and has functions like update() and manage().

Relationships Between Classes: Customers place Orders. Restaurants manage Menus and Orders. Feedback is associated with Customers and Restaurants. Delivery Partners deliver Orders. Orders include multiple Items.

## Use Case Diagram

****

*Figure 4 Use Case Diagram of CricketVerse System*

The use case diagram of Food Fetch provides a high-level overview of the functionalities available to “Customer” and “Restaurant Manager” actors within the Food Fetch Application. It highlights the primary interactions and benefits for each user type. The figure consists of 2 Actors: *Customer* and *Restaurant Manager.* Customers can initiate the following actions: Browse Menu and Select Items: Customers can browse through restaurant menus categorized by cuisine or other relevant criteria, and select the food items they want to order. Place Order: Once they've selected their items, customers can place their order, specifying their delivery address and any special requests. Track Order: Customers can track the progress of their order in real-time, seeing its estimated delivery time

and location. Provide Feedback: Customers can leave feedback about their experience with the restaurant and the Food Fetch system

Restaurants have their own set of use cases like Manage Menu: Restaurants can add, edit, and remove

items from their menus, update prices, and set availability. Manage Orders: Restaurants can receive and

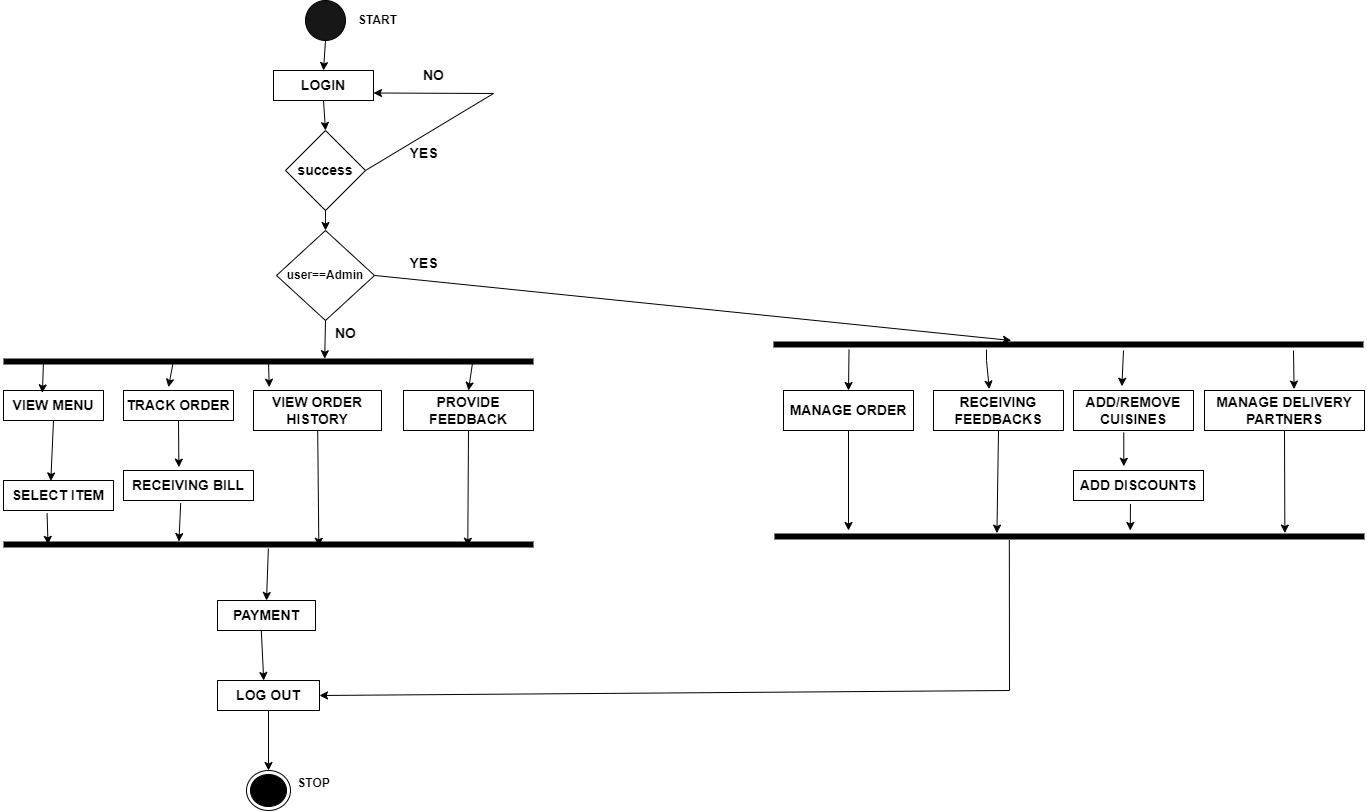
view incoming orders, confirm them, prepare the food, and mark them as complete for delivery. Manage

Delivery Partners: Restaurants can collaborate with delivery partners to manage delivery

schedules, track deliveries, and ensure timely delivery to customers. View Feedback: Restaurants can

view and respond to customer feedback to improve their service .

## Activity Diagram

****

*Figure 5 Activity Diagram of Food Fetch system*

It starts with the customer entering their **email address** and **phone number**. The application verifies the login

information and grants access to the **main menu**.

From the main menu, the customer can browse through **restaurants** or **search for specific items**. Once they

find a restaurant or item they like, they can view the **details**, including the menu, prices, and ratings. If the

customer decides to order, they add items to their **cart**. They can then proceed to **checkout**, where they specify

their **delivery address** and **payment method**. The application confirms the order details and sends it to the

**restaurant**. The restaurant receives the order and starts preparing the food. Once the food is ready, the

restaurant marks the order as **complete** and sends it to a **delivery partner**.

The delivery partner picks up the food from the restaurant and delivers it to the customer's address. The

customer receives a **notification** when their order is on the way and another notification when it arrives.

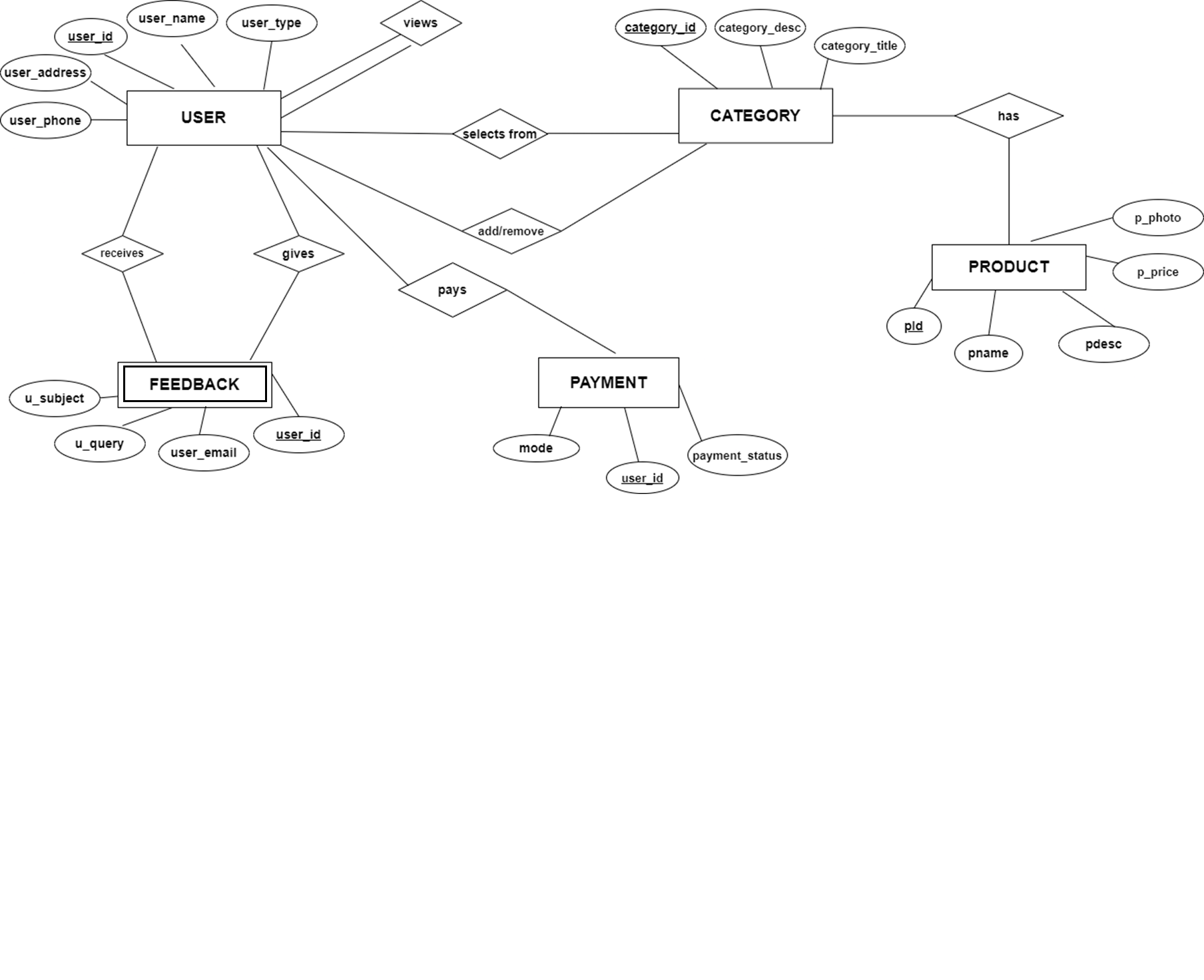
Throughout the process, the customer can **track their order** in real-time, see its estimated delivery time, and

contact the restaurant or delivery partner if needed.

Overall, the activity diagram provides a clear visual representation of the steps involved in placing an order on

the Food Fetch application, highlighting the interactions between the customer, restaurant, and delivery partner.

## Entity Relationship Diagram

****

*Figure 6 Entity Relationship diagram for Food Fetch*

The ER diagram represents the core entities and their relationships within the Food Fetch application.

These entities include:

User: This entity represents the users of the application, either customers or restaurant owners. It has

attributes like user\_id (primary key), user\_name, user\_type (customer or restaurant), phone\_number,

email, and address. The user\_type attribute determines the specific functionalities each user can access

within the application.

Category: This entity represents the different categories of food items available on the platform.

It has attributes like category\_id (primary key), category\_title, and category\_desc. Each category can

have many food items associated with it.

Product: This entity represents the individual food items offered by restaurants. It has attributes

like product\_id (primary key), product\_name, product\_description, product\_price, and category\_id

(foreign key referencing Category). A product belongs to one category, and many products can be

offered by a restaurant.

Restaurant: This entity represents the restaurants participating in the Food Fetch application. It has

attributes like restaurant\_id (primary key), restaurant\_name, restaurant\_address, and category\_id

(foreign key referencing Category). A restaurant can belong to one category and offer many products.

Order: This entity represents the orders placed by customers through the application. It has attributes

like order\_id (primary key), customer\_id (foreign key referencing User), restaurant\_id (foreign

key referencing Restaurant), order\_date, order\_status, and total\_price. An order is placed by one

customer, for one restaurant, and includes one or more products.

Feedback: This entity represents the feedback provided by customers about their experience with

restaurants. It has attributes like feedback\_id (primary key), customer\_id (foreign key referencing

User), restaurant\_id (foreign key referencing Restaurant), feedback\_text, and rating. A customer

can leave feedback for a specific restaurant, and a restaurant can receive feedback from multiple

customers.

The relationships between these entities are as follows: User can place many Orders. Order is placed

by one User. Order includes many Products. Product belongs to one Category. Product is offered

by one Restaurant. Restaurant belongs to one Category. Restaurant receives many Orders. Customer

(user type) can provide Feedback for Restaurant. Restaurant can receive Feedback from many Customers.

In essence, the ER diagram captures the essential data elements and their connections within the Food

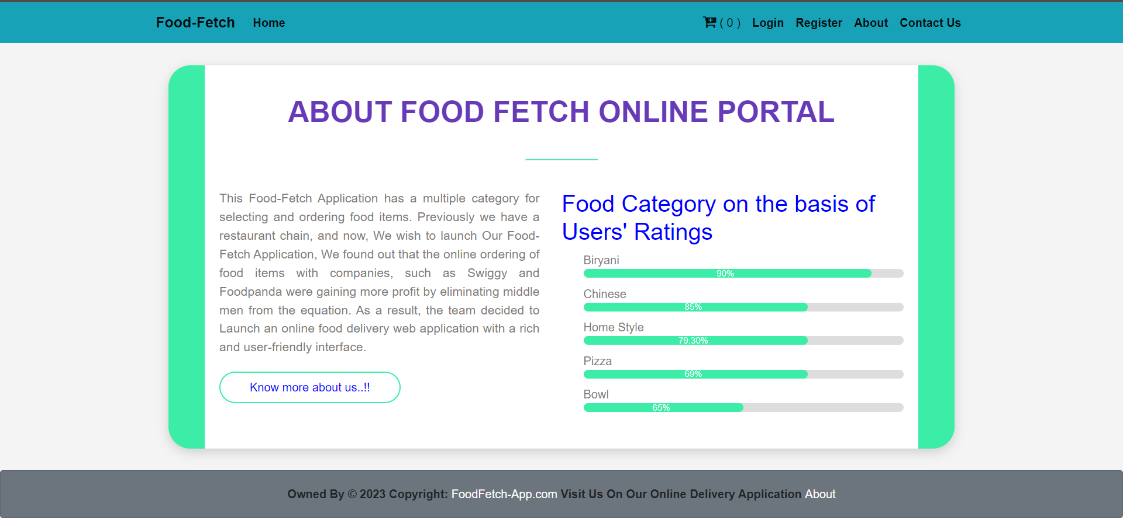
Fetch application, providing a foundational model for its database design and ensuring data integrity and

consistency.

**CHAPTER 7**

**7.1 SCREENSHOTS**

**7.1.1** **About us page**

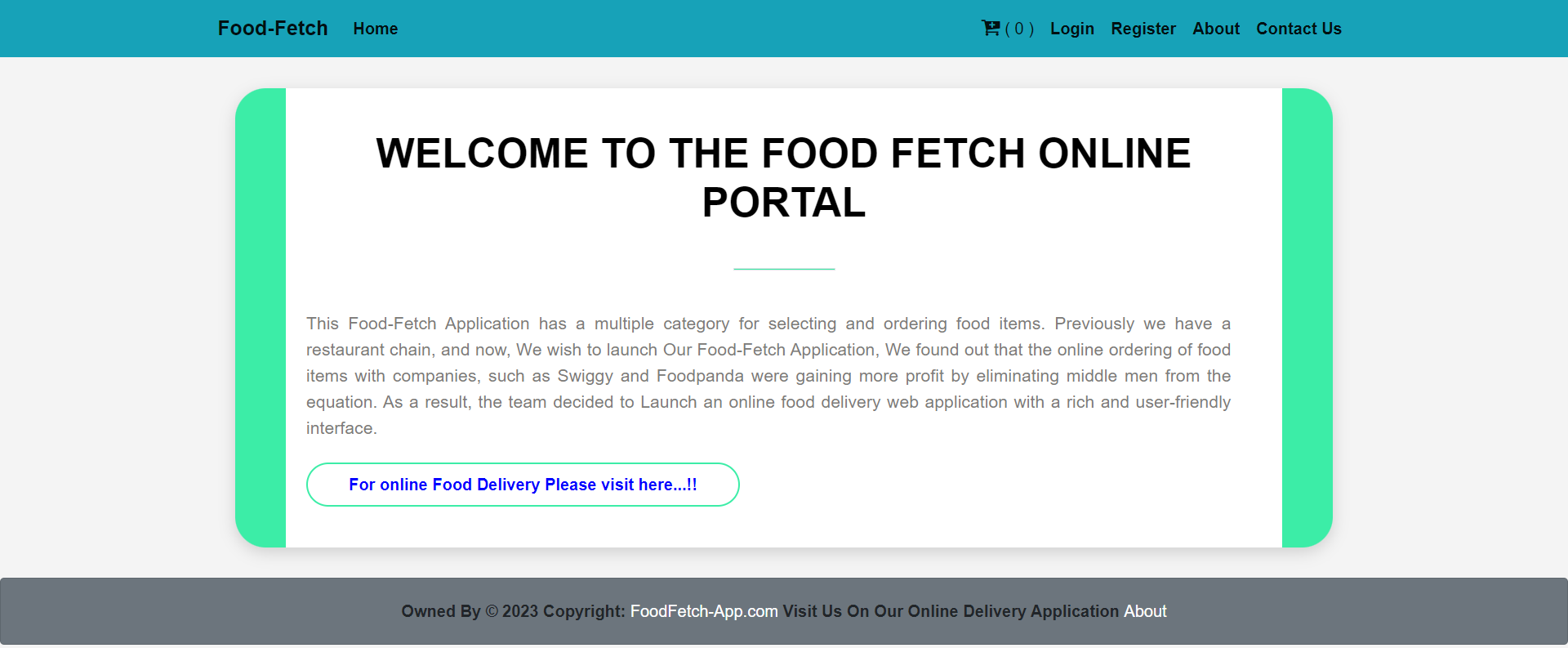


*Figure 7.1: About us*

This page explains how food fetch came into existence “This Application has a multiple category for selecting and ordering food items. We found out that the online ordering of food items with companies, such as Swiggy and Foodpanda were gaining more profit by eliminating middle men from the equation. As a result, the team decided to launch an online food delivery web application with a rich and user-friendly interface.”

**7.1.2 Welcome page**

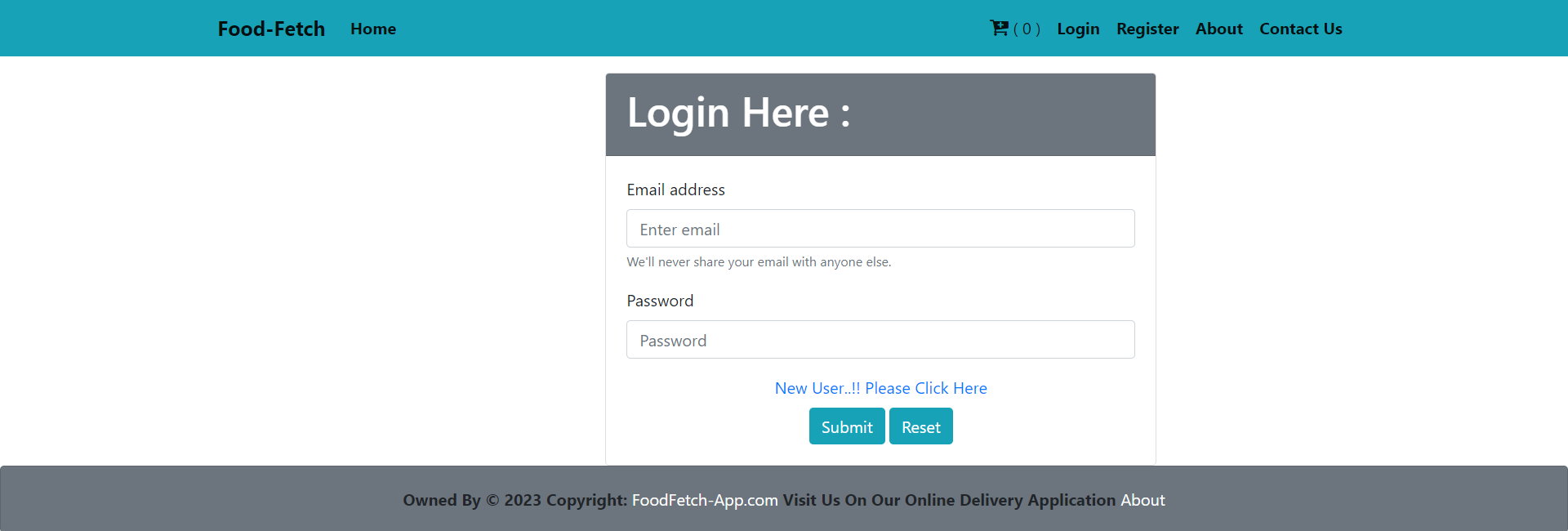
*Figure 7.2: Welcome page*



It features a banner with the company’s logo and slogan, “Welcome to the Food Fetch Online Portal,” followed by a brief welcome message that describes the company’s services and mission. It mentions they have expanded from a restaurant chain to an online food ordering platform, aiming to eliminate middlemen and deliver food directly from a variety of restaurants to the customer’s doorstep. It highlights that their online food delivery application has a rich and user-friendly interface, making the food ordering process convenient.

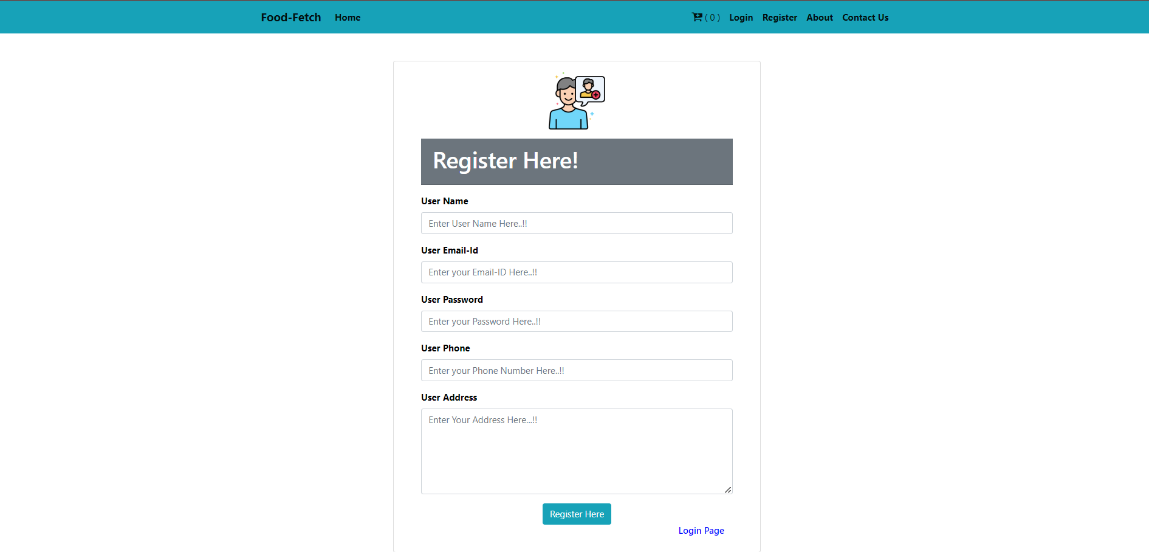
**7.1.3** **Login Page**

The login page is designed to be simple and easy to use. It has a white background with a green and blue logo at the top. There are two input fields on the page: one for the user's email address and one for their password. There is also a "Forgot Password?" link below the password field. Below the input fields, there is a green "Login" button. There is also a "New User? Please Click Here" link below the login button. Clicking on this link will take the user to the registration page, where they can create a new account.



*Figure 7.3: Login page*

**7.1.4 Registration page**

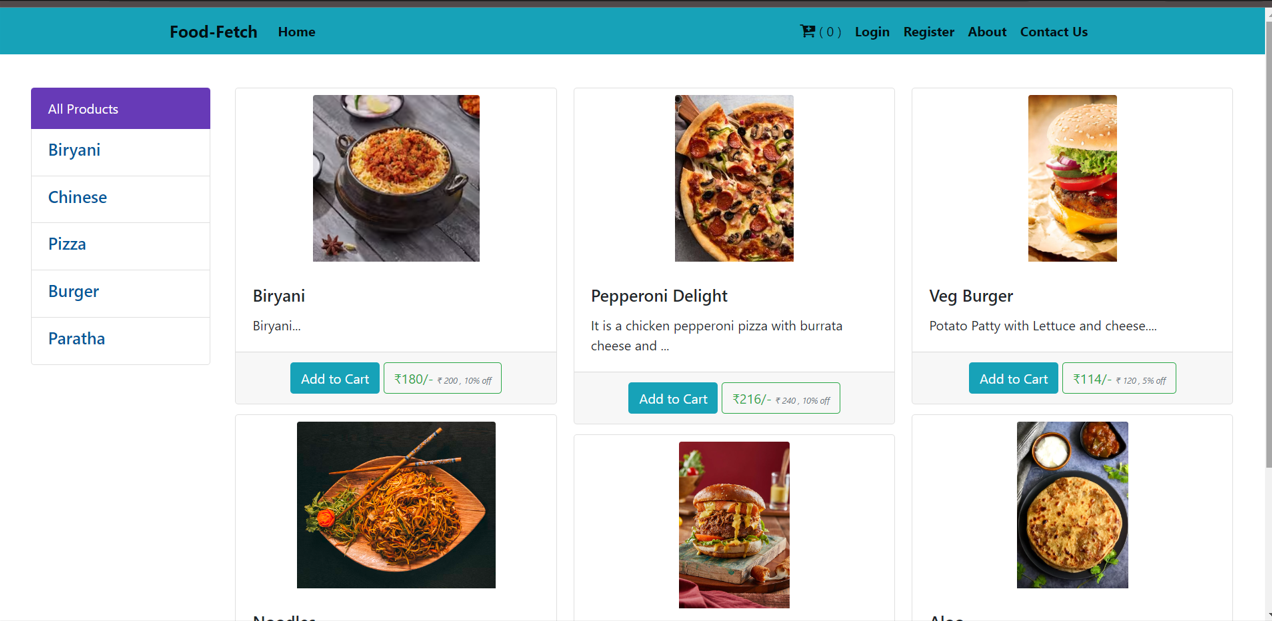


*Figure 7.4: Registration*

The registration page is designed to be simple and easy to use. It has a white background with a green and blue logo at the top. There are five input fields on the page:

Username, Email Address, Password, Phone Number, Address. There is also a green "Register Here" button below the address field. There is a "Login Page" link at the bottom of the page, which takes users to the login page.

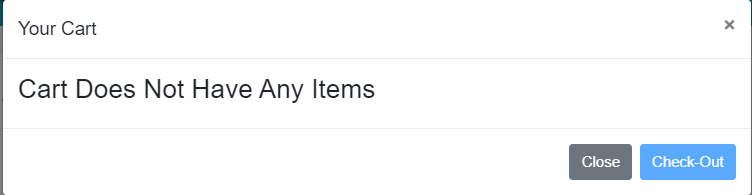
**7.1.5** **Home page**



*Figure 7.5: Home page*

It highlights different food items from various categories, including biryani, paratha, pizza, and burger. Each item has a description and price listed below it. For example, the description for the biryani says, "It is a chicken pepperoni pizza with burrata cheese and cheese." The price for the biryani is ₹180. There are also two buttons at the bottom of the image. The first button, "Add to Cart", allows users to add the selected food item to their shopping cart. The second button, "View All Products", likely takes users to a page where they can see all of the food items available on the Foodtech Home website.

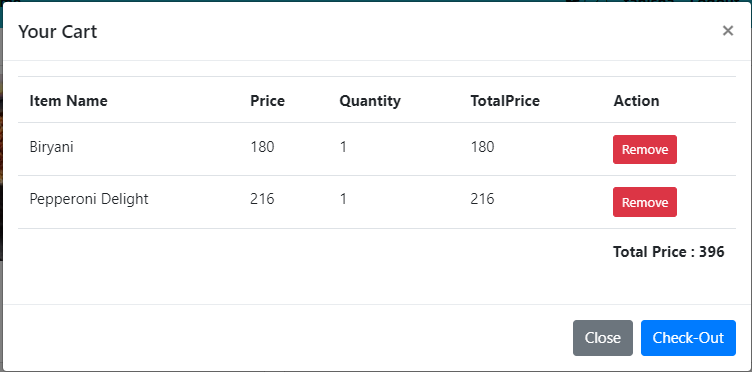
**7.1.6 Empty Cart page**



*Figure 7.6: Empty cart*

It is an empty cart image where user has not put anything in the cart.

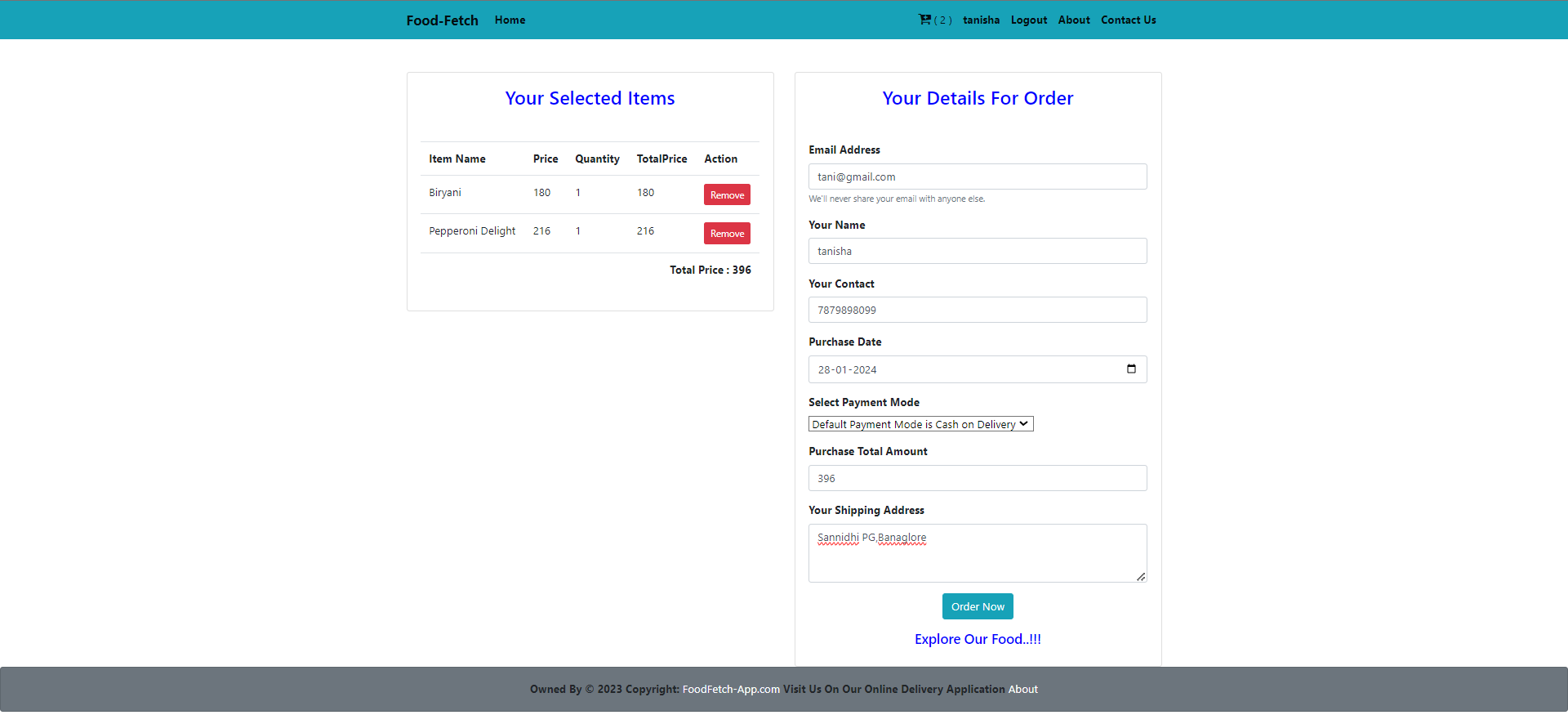
**7.1.7 Full cart page**



*Figure 7.7: Full cart*

It is a full cart image where user has put something in the cart.

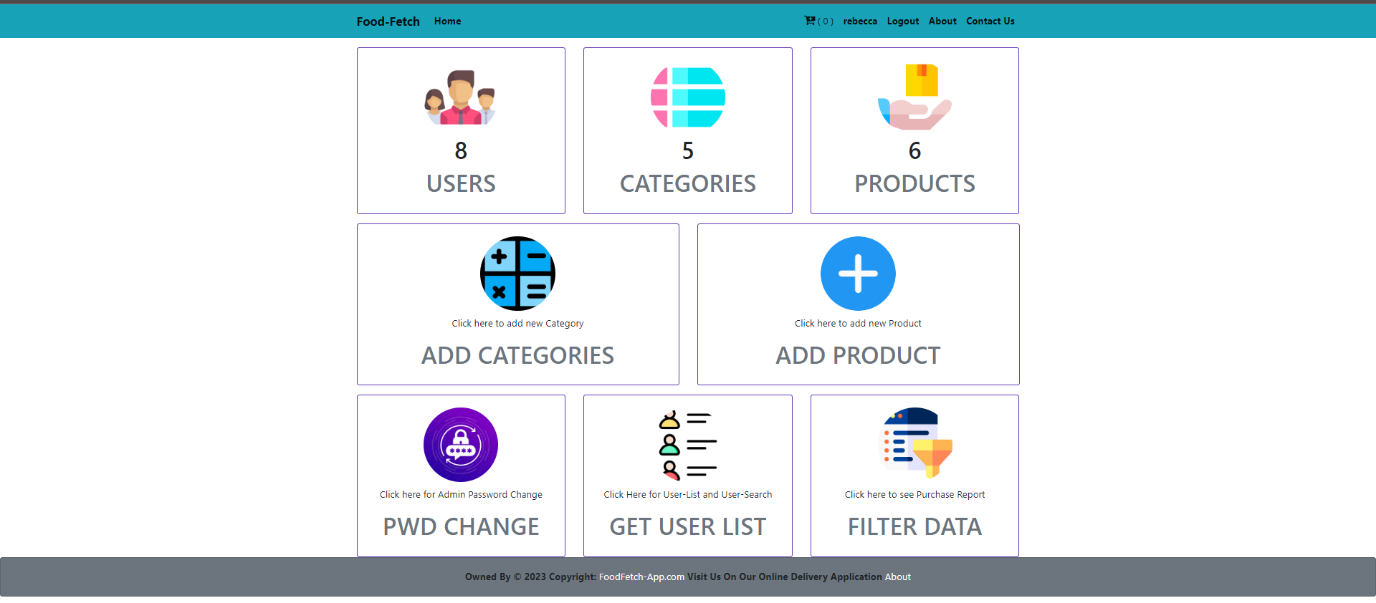
**7.1.8 Check out page**



*Figure 7.8: Check out*

The checkout page allows users to review their order details and enter their payment information before placing their order. The top section of the checkout page displays the user's selected items, including the item name, quantity, price, and total price. The middle section of the checkout page displays the user's contact information, including their name, email address, and phone number. The user can also enter a different shipping address if needed. The bottom section of the checkout page displays the user's payment information.

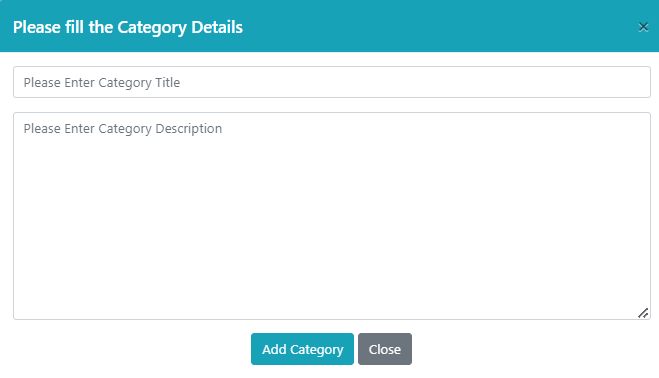
**7.1.9 Admin dashboard page**



*Figure 7.9: Admin dashboard*

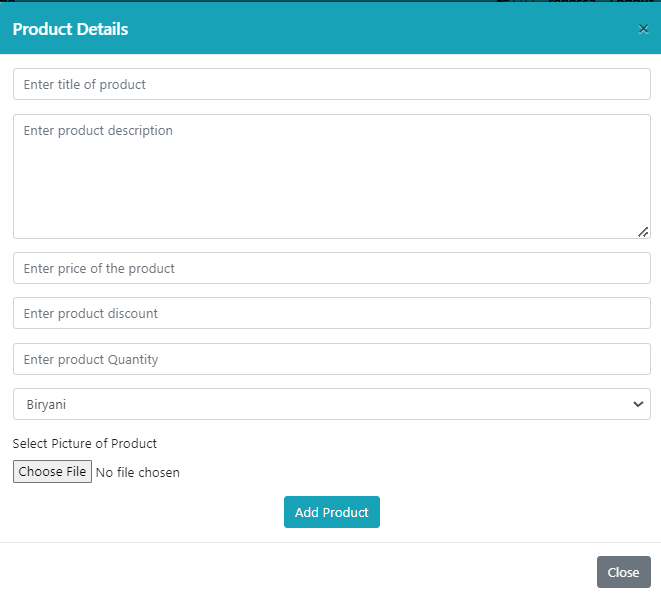
The dashboard is designed to give website administrators an overview of the website's activity and to allow them to manage the website's content and settings. The dashboard is divided into several sections. The top section contains the website's logo and the administrator's name. The left-hand side of the dashboard contains a navigation menu with links to different sections of the dashboard, such as "Users", "Categories", "Products" and "Orders."

**7.1.10 Category page**



*Figure 7.10: Category page*

**7.1.11 Product addition page**

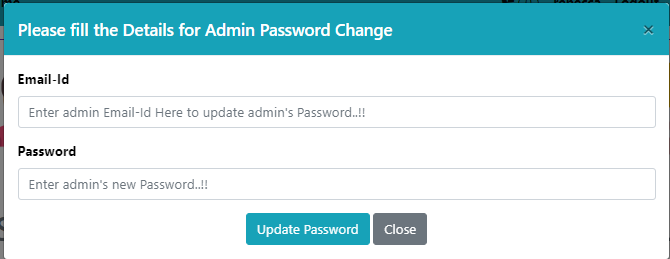


*Figure 7.11: Product addition*

This page adds different food item based on category and images for the specific item is also added and discount is added too by the admin.

**7.1.12** **Admin’s Password change** **page**

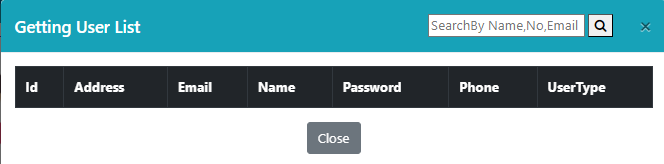
*Figure 7.12: Password change*



Admin can change their current password.

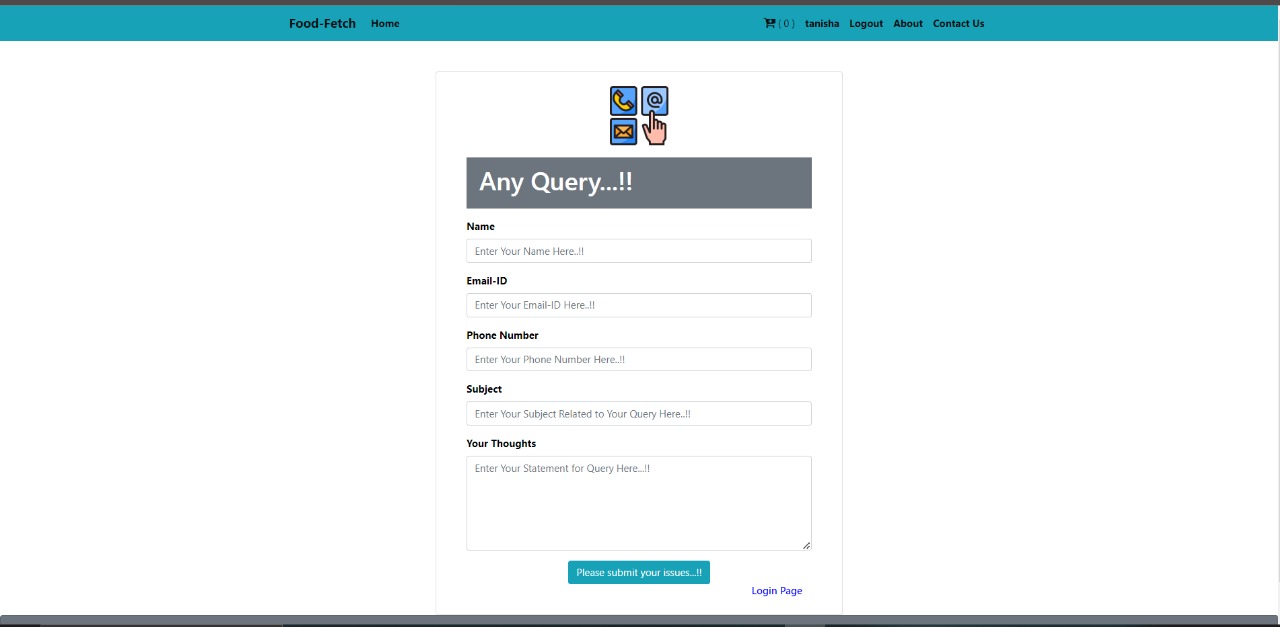
**7.1.13 Get user list page**

*Figure 7.13: Get user list*



It shows the users’ list that have registered themselves in the application

**7.1.14 Feedback page**



*Figure 7.1.4:feedback page*

Users can write their query and give any feedbacks to improve the quality of food.