System Requirements Statement (SRS) –

VynjanVibes- Food Ordering System

**Group No.: G88**

**Roll No. Name**

243178 Pradnya Tukaram Patil

243184 Pratibha Hanumant Tonpe

1. **Introduction**
2. **Functional Requirements**
   * 2.1 User Module
   * 2.2 Restaurant Module
   * 2.3 Admin Module
3. **NON-FUNCTIONAL REQIREMENTS**

# Introduction

# Document Purpose

The purpose of our project, "-Food Ordering System," is the process of ordering food from restaurants, enhancing both convenience and efficiency for users. This web application is designed to facilitate the seamless management of food orders, from browsing menus to placing orders and managing payments. The system aims to minimize the effort involved in ordering food, making it easier for users to find restaurants, view menu items, and complete transactions. By centralizing restaurant and item management, "FoodOrder" provides a user-friendly interface for customers and robust administrative tools for restaurant operators. The goal of this project is to offer a comprehensive solution that simplifies the food ordering process, reduces wait times, and improves overall customer satisfaction..

# 1.2Goals of the project

The main objective of this project is to develop a website that provides users with a convenient platform to order food from a variety of restaurants. Users will be able to browse through different restaurants, view detailed menu items, and select their preferred dishes and place orders directly through the platform. The aim of this project is to simplify the food ordering process, bridging the gap between customers and restaurants, and enhancing overall convenience and user satisfaction.

# Functional Requirements Overview

System consists of four modules described as below.

1. Customer/User Module
2. Restaurant Module
3. Admin Module

# 1. Patient/User Module

* Customer/User can register and update his own account.
* Customer/User Customers can search for restaurants and cuisine.
* Customer/User can view detailed information about menu items offered by restaurants.
* Customer/User can select menu items, customize their orders, and place food orders directly through the platform.
* Customer/User can check status of their current orders
* Customer/User can cancel order.
* Customer/User can view their past orders and order
* Customers can pay for their orders online through secure payment gateways.
* Customer/User can receive updates and notifications about their orders via email or mobile SMS.

# 2. Restaurant Module

* Restaurant can register and update their own account
* Restaurant owners can create, update, and manage their menu items and prices.
* Restaurant owners can view and manage incoming orders, including updating order statuses.
* Restaurant owners can view their past orders and order history.

# 3. Admin Module

* System should provide all function to admin how to handle the System.
* Admins can oversee and manage all aspects of the system,
* Could able to maintain/know all the record/history.
* Register/delete/block Restaurant

**Functional Description**

The Account part of – Food ordering System has three modules which are divided -- processes described as below.

|  |  |  |
| --- | --- | --- |
| **No** | **SRS Requirement ID** | **Description** |
| **2.1** | **User Module** | |
| **2.1.1** | F1 | Registration Process |
| **2.1.2** | F2 | Login Process |
| **2.1.3** | F3 | Forgot Password Process |
| **2.1.4** | F4 | Change Password Process |
| **2.1.5** | F5 | Update Account Process |
| **2.1.6** | F6 | Browse Restaurants |
| **2.1.7** | F7 | Place Order Process |
| **2.1.8** | F8 | Check Order Status |
| **2.1.9** | F9 | Cancel Order Process |
| **2.1.10** | F10 | View Order History |
| **2.1.11** | F11 | Online Payment Process |
| **2.1.12** | F12 | Receive Notifications via Email and SMS |
| **2.2** | **Restaurant Module** | |
| **2.2.1** | F13 | Login Process |
| **2.2.2** | F14 | Forgot Password Process |
| **2.2.3** | F15 | Reset Password Process |
| **2.2.4** | F16 | Manage Orders Process |
| **2.2.5** | F17 | Update Menu Process |
| **2.2.6** | F18 | Update Profile Process |
| **2.2.7** | F19 | View Order History Process |
| **2.2.8** | F20 | Accept Online Payments |
| **2.2.9** | F21 | Receive Notifications via SMS and Email |
| **2.3** | **Admin Module** |  |
| **2.3.1** | F22 | Login Process |
| **2.3.2** | F23 | New Restaurant Registration Process |
| **2.3.3** | F24 | Manage Customer List |
| **2.3.4** | F25 | Manage Restaurant List |
| **2.3.5** | F26 | Generate Reports Process |
| **2.3.6** | F27 | Manage Accounts (Restaurants, Customers) |

* **Customer/User Module**
* Customer is the user of System who can browse through available restaurants and food items.
* Users can place orders for food items from selected restaurants.
* **Registration Process**
* - Food Ordering System compels to create the account for order food. So, the System should provide the function which makes user create new account.
* When User creates new account, the function demands information described as below.
* Login information
* Contact Details
* Login information

The Login information consists of some items described as below

* UID Auto – Generated
* User Name
* Password
* E-mail address
* Phone number
* Address
* All items are compulsory demanded.
* Email
* The Email should be unique. If the email corresponds with not case-sensitive to other which is previously registered, the email should not be registered as an account.
* Password
* The Password has constraints which makes the Password consists of more than or equal 8 and less than or equal 16 characteristics including characters described as below.
* Numeric figure (at least one)
* Capital alphabet (A-Z) (at least one)
* Small alphabet (a-z) (at least one)
* Special character (#, $, %, &, etc.) (at least one)
* The password is masked by dummy characters. The re-entering Password is demanded.
* The password must be encrypted in Food ordering System
* Contact Details
* Contact Number
* All items are compulsory demanded.
* Address
* Address should be filled.
* **Login Process**
* - Food Ordering System always compels user authentication for using web application.
* The user authentication demands Email and Password. The Email and the Password should be checked in two ways.
* First, The Email and the Password should be existed and correct.
* If The Email and the Password are not equal to what the user has registered, the user authentication cannot be provided.
* Second, Email should be available.
* The Administrator can decide whether the Email is available or suspended – Refer to the SRS of the Admin part.
* If user is rejected, user authentication is not provided for System user.
* **Forgot Password Process**
* When System user lost their Password, the recovery method should be provided by Docpulse - Meet Your Doctor System.

The recovery method is described as below.

* First, System user enters their Email for - Food Ordering System.
* Next, - Food Ordering System will send OTP on the registered Email.
* Only when the Answer is correct, farmer get the new password by E-mail which also has been registered since when the Account was created.
* Only when OTP entered is match then user will able to change password.
* The new password is manually entered by user.
* Of course, the new password should consist of more than or equal 8 and less than or equal 16 characteristics including at least a numeric figure, a capital alphabet, a small alphabet, and a special character.
* **Change Password Process**
* When User wants to change their Password, the measure should be provided by - Food Ordering System.
* Therefore, System should provide the function which is available after getting the user authentication.
* The function demands the current password and the new password.
* Of course, the new password should consist of more than or equal 8 and less than or equal 16 characteristics including at least a numeric figure, a capital alphabet, a small alphabet, and a special character.
* The current password and the new password are masked by using dummy characters.
* The new password is demanded to enter twice to avoid a typing error.
* Only when the current password is correct, user could change their Password
* When the current password is changed into new password, System compels user authentication again.
* **Update Account Process**
* - Food Ordering System should provide the function which makes the account updated for user.
* The user can update the information as described below
* User information
* The User information

The updatable items as described below.

* Contact Number
* Password
* Address

All items are compulsory demanded, but updating is optional.

* **Search Restaurant**

After login user can search for a restaurant based on

* Location
* Specialization

If user select location as a criterion, then the list of all restaurant of respective location will get displayed.

If user will search for the restaurant without selecting the criteria (location or specialization) or user select specialization as a criterion without selecting the location, then System will give the message that user must select location to get list of restaurant

* **Place Order Process**

After getting the list of restaurant according to location and specialization user can select the particular restaurant.

After selecting the restaurant , all the slots will be shown to the User. User can see only the Restaurant which are available for ordering .

User can place order according to the available restaurants .

* **Check Order Status**

After successfully placing an order , the user can check ordering status.

* **Cancel Order**

After successfully placing an order if user wants to cancel the order, *Cancel Order* option is provided to User.

* **View Order History**

User could see the information of all the previous orders according to date select for the history.

* **Restaurant Module**

* **Login Process**
* -Food Ordering System always compels restaurant authentication before using itself except when a new account is successfully created.
* The user authentication demands Email and Password. The Email and the Password should be checked in two ways.
* First, The Email and the Password should be existed and correct.
* If The Email and the Password are not equal to what the user has registered, the user authentication cannot be provided.
* Second, Email should be available.
* The Administrator can decide whether the Email is available or suspended – Refer to the SRS of the Admin part.
* If user is rejected, user authentication is not provided for doctor.
* Only when the two checks are successfully completed, Doctor can be placed on respected page.
* The “Restaurant Home Page” provides some items described as below.
* A trigger to logout
* A trigger to Reset Password
* A trigger to Active Orders
* A trigger to Create slots
* A trigger to Update profile
* A trigger to Cancel Orders
* A trigger to View Orders History
* **Forgot Password Process**
* When Restaurant lost their Password, the recovery method should be provided by -Food Ordering System

The recovery method is described as below.

* First, Restaurant enters their Email for -Food Ordering System.
* Next, -Food Ordering System demands the Answer which has been registered since when the Account was created.
* Only when the Answer is correct ,Restaurant gets the new password by E-mail which also has been registered since when the Account was created.
* The new password is manually entered by doctor.
* Of course, the new password should consist of more than or equal 8 and less than or equal 16 characteristics including at least a numeric figure, a capital alphabet, a small alphabet, and a special character.
* If the Answer is not correct, otherwise, the correct Answer is demanded for doctor again.
* In that case, Of course, restaurant could not get the new password.
* **Reset Password Process**
* Restaurant can login using credentials given by the admin after successful registration.
* But for privacy Restaurant can reset password using Reset Password process.
* After reset password restaurant will have to login by using new password for every time.
* The function demands the current password and the new password.
* Of course, the new password should consist of more than or equal 8 and less than or equal 16 characteristics including at least a numeric figure, a capital alphabet, a small alphabet, and a special character.
* The current password and the new password are masked by using dummy characters.
* The new password is demanded to enter twice to avoid a typing error.
* Only when the current password is correct, Restaurant could change their Password.
* When the current password is changed into new password, Food ordering System compels Restaurant authentication again.
* **Update Profile Process**
* Food Ordering System should provide the function which makes the account updated for restaurant.
* The information restaurant could update is described below.
* Contact Number
* Address
* Category
* Food Items
* All items are compulsory demanded, but updating is optional.

.

* **View Order History**

• Restaurant can view all orders history.

* **Admin Module**

**•** Administratorshould be responsible for following activities**.**

* **Login Process**
* Food Ordering System always compels user authentication before using itself except when a new account is successfully created.
* The user authentication demands Email and Password. The Email and the Password should be checked as -
* The Email and the Password should be existed and correct.

If The Email and the Password are not equal to what the admin has registered, the admin authentication cannot be provided.

* **Add Restaurant (Registration)**

* Admin can create the new account for restaurant by verifying all the information given by the restaurant via email.
* Admin will verify all the details and create the account for restaurant .
* Admin will send first login credentials to restaurant (email and password).

Password sent by admin is in the form firstname@123.

* **View User List**

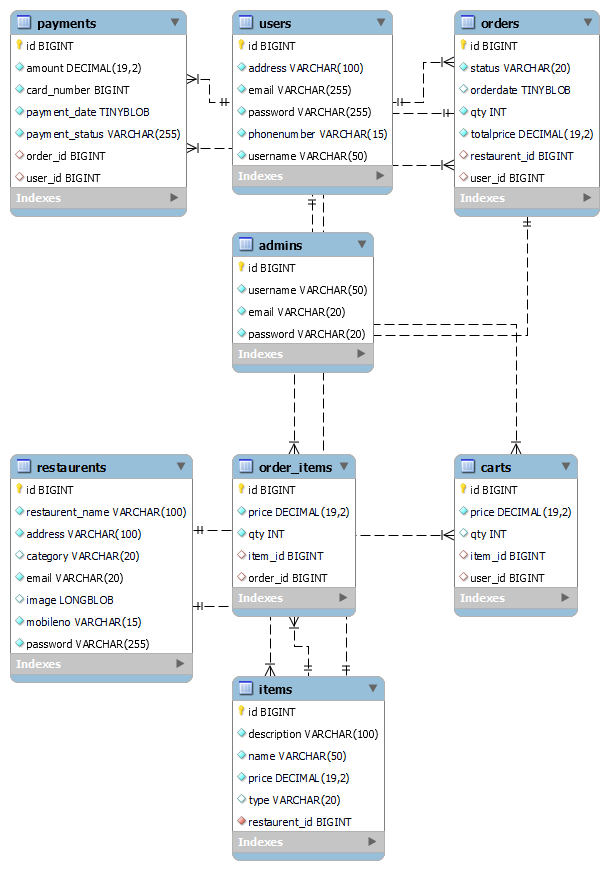
• Admin should be able to see all the user list according following criteria:

* All registered User
* **View Restaurant List**

• Admin should be able to see all the restaurant list according following criteria:

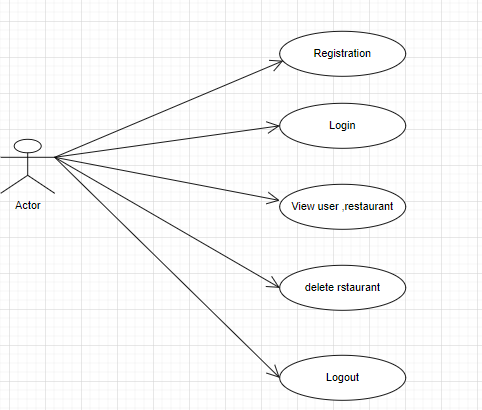
* All registered restaurant
* **Non-Functional Requirement**
  + Characteristic:
    - * UI should be localized with English-Marathi Language.
      * Platform should be secure, scalable, maintainable
  + Security:
    - * will implement Role based security to access content from Remote place.
      * System will provide secure communication between payment gateway applications (SSL)
      * System will expire session maintained for each user after 15 minutes.
      * System will not keep any sensitive data on user’s device.
      * Users’ information would protect.
      * All servers where System deployed would have been protected using firewall.
  + Safe:
    - * System will keep regular back up of data in incremental way.
      * System will be protected against malicious attack with proper cybersecurity rules.
      * System hosted servers would have been protected within restricted environment
  + Reliability:
    - * System will be always available 99.999%
      * System should have low downtime
      * System server would keep their availability using failover server
  + Scalability:
    - * System will provide consistent user experience irrespective of number users grow on particular events such as festival season,
      * Sale promotion. (Load balancing)
  + Modularity:
    - * System Web Application would be complex application due to its functionality expected.
      * System web Application would be composed using set of reusable modules.
  + Portability:
    - * System web application could be accessed from any personal device.
      * System will provide portable user interface using standard web technologies while building portal.
      * System web application could be deployed to any server Operating Environment.
      * System web application could be deployed to any Cloud such as Microsoft Azure or Amazon Web Service (AWS) or GCP
      * System web application could have been deployed to any other physical server or Virtualized server.
  + Compatibility:
    - * System could be installed on windows server, Linux server.
      * System will provide system browser compatible user interface

**ER-Diagram**



**Use Case Diagram**

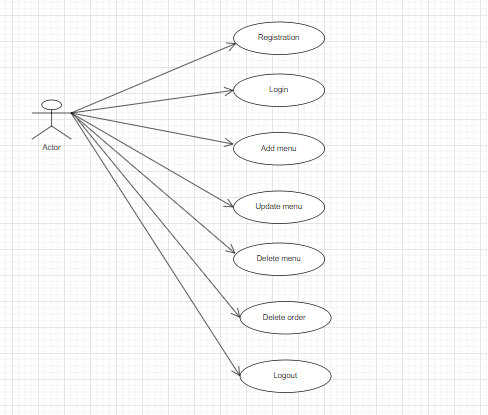
**Admin:**



*Fig. Use case diagram for admin*

* In Admin use case diagram Admin is the Actor.
* Admin can handle following use cases:
* Login
* Add Restaurant
* View User
* View Restaurant
* Delete Restaurant

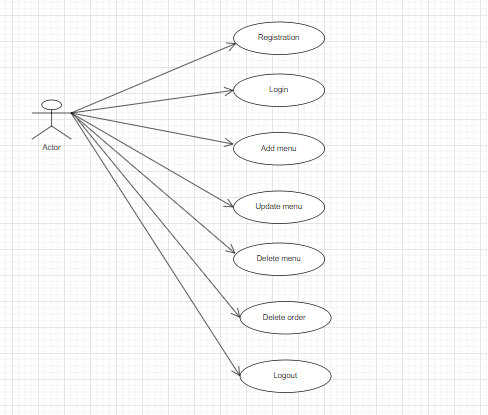
**Customer:**



*Fig. Use case diagram for Customer/User*

* In Customer use case diagram customer is the Actor.
* Customer can handle following use cases:
* Register
* Login
* Search Restaurant
* Order Food
* View Order Status
* Cancel Order
* View Order History

**Restaurant :**

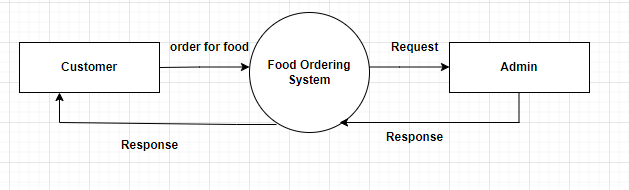


*Fig. Use case diagram for Restaurant*

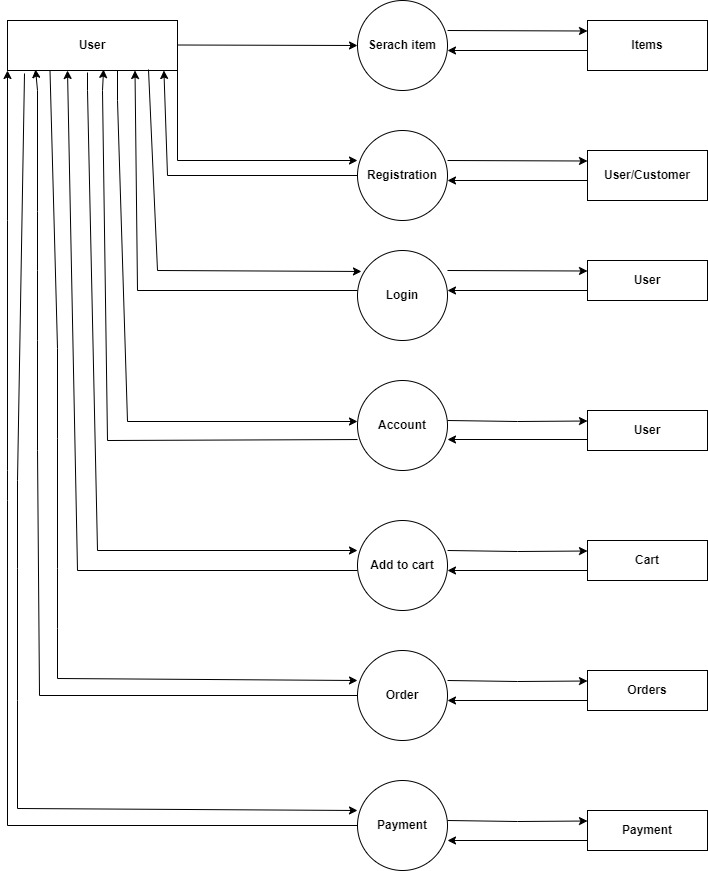
* In Restaurant use case diagram Restaurant is the Actor**.**
* Restaurant can handle following use cases:
* Login
* Add Menu
* Update Menu/Items
* Cancel Order
* View Order History
* Logout

**Data Flow Diagram:**

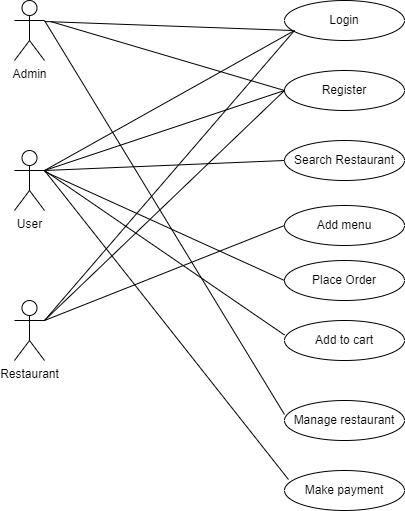
**Level 0:**

****

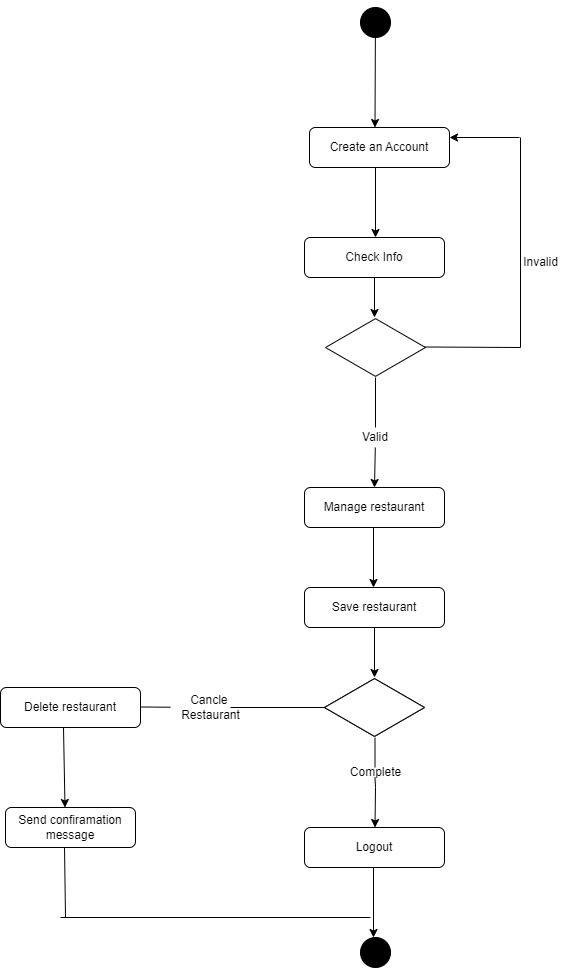
Level 1:



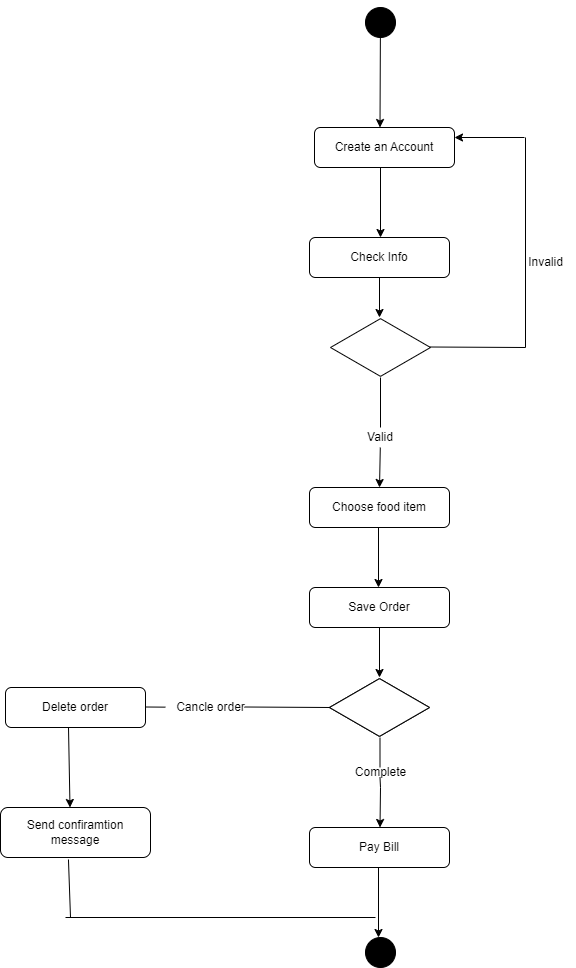
**Use Case Diagram:**



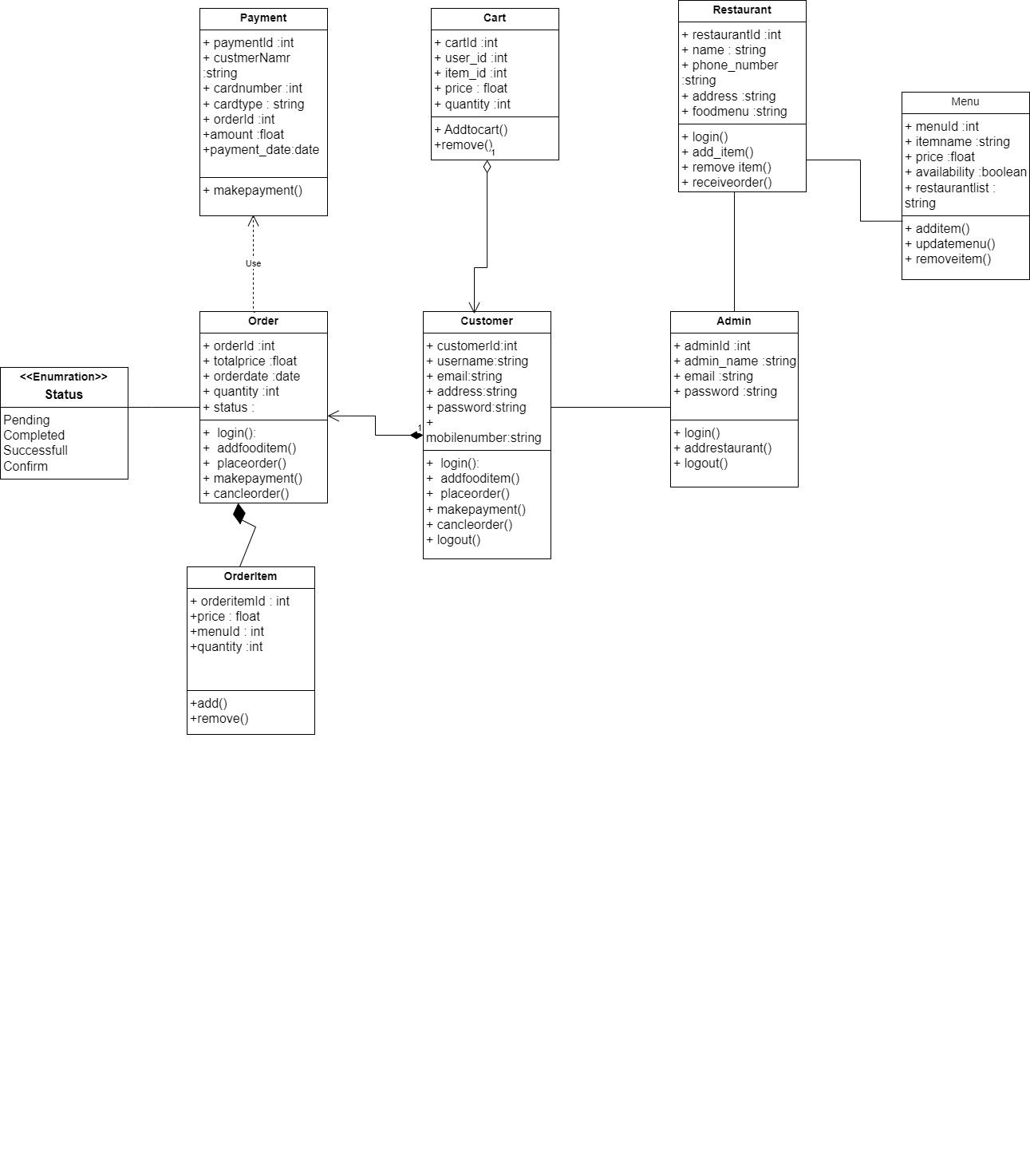
**Admin Activity Diagram :**



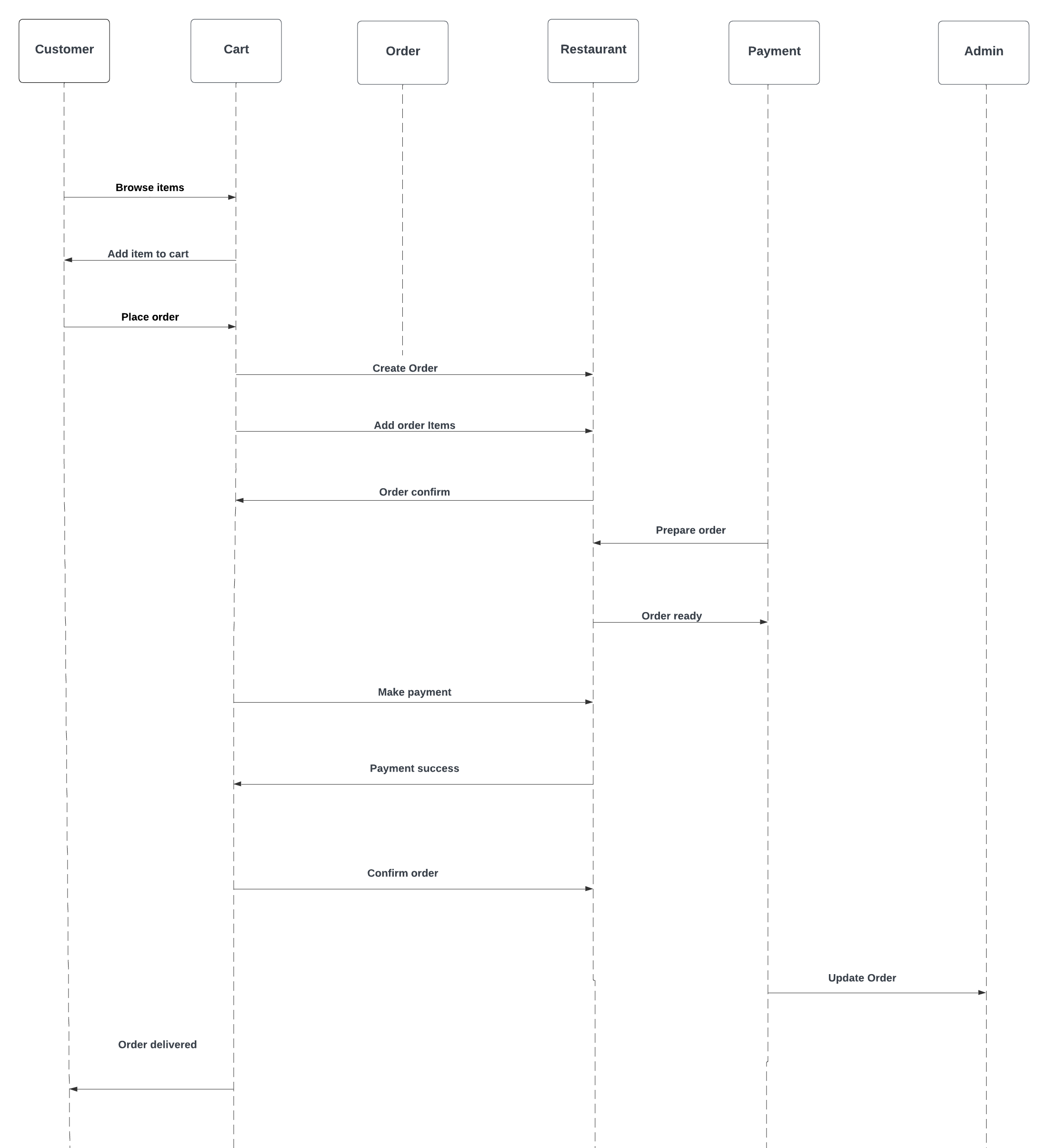
**User Activity Diagram :**



**Class diagram :**



**Sequence diagram :**



**Table Structure**

**Table :User/Customer Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key Type/ Constraint | Column Name | Data Type | Length | Allow Null (1=Yes;0=No) |
| PK | user-id | bigint |  | 0 |
|  | username | varchar | 50 | 0 |
|  | phonenumber | varchar | 15 | 0 |
|  | email | varchar | 30 | 0 |
|  | password | varchar | 14 | 0 |
|  | address | varchar | 100 | 0 |

**Table : Restaurant Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key Type/ Constraint | Column Name | Data Type | Length | Allow Null (1=Yes;0=No) |
| PK | id | bigint |  | 0 |
|  | restaurants\_name | varchar | 30 | 0 |
|  | address | varchar | 100 | 0 |
|  | category | varchar | 20 | 1 |
|  | email | varchar | 20 | 0 |
|  | image | longb lob |  | 1 |
|  | mobileno | varchar | 15 | 0 |
|  | password | varchar | 14 | 0 |

**Table : Admin Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key Type/ Constraint | Column Name | Data Type | Length | Allow Null (1=Yes;0=No) |
| PK | id | bigint |  | 0 |
|  | username | varchar | 50 | 0 |
|  | email | varchar | 20 | 0 |
|  | password | varchar | 20 | 0 |

**Table : Items Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key Type/ Constraint | Column Name | Data Type | Length | Allow Null (1=Yes;0=No) |
| PK | id | bigint |  | 0 |
|  | name | varchar | 50 | 0 |
|  | description | varchar | 100 | 0 |
|  | price | decimal |  | 0 |
|  | type | varchar | 20 | 1 |
| FK | restaurant\_id | bigint |  | 0 |

**Table : Orders Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key Type/ Constraint | Column Name | Data Type | Length | Allow Null (1=Yes;0=No) |
| PK | id | bigint |  | 0 |
|  | status | varchar | 20 | 0 |
|  | orderdate | tinyblob |  | 1 |
|  | qty | int |  | 0 |
|  | totalprice | decimal |  | 0 |
| FK | restaurant\_id | bigint |  | 0 |
| FK | user\_id | bigint |  | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key Type/ Constraint | Column Name | Data Type | Length | Allow Null (1=Yes;0=No) |
| PK | id | bigint |  | 0 |
|  | price | decimal |  | 0 |
|  | qty | int |  | 0 |
|  | item\_id | bigint |  | 0 |
|  | order\_id | bigint |  | 0 |

**Table : Order\_Items Table**

**Table : Carts Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key Type/ Constraint | Column Name | Data Type | Length | Allow Null (1=Yes;0=No) |
| PK | id | int |  | 0 |
|  | price | decimal |  | 0 |
|  | qty | int |  | 0 |
|  | item-id | bigint |  | 0 |
|  | user-id | bigint |  | 0 |

**Table : Payment Table**

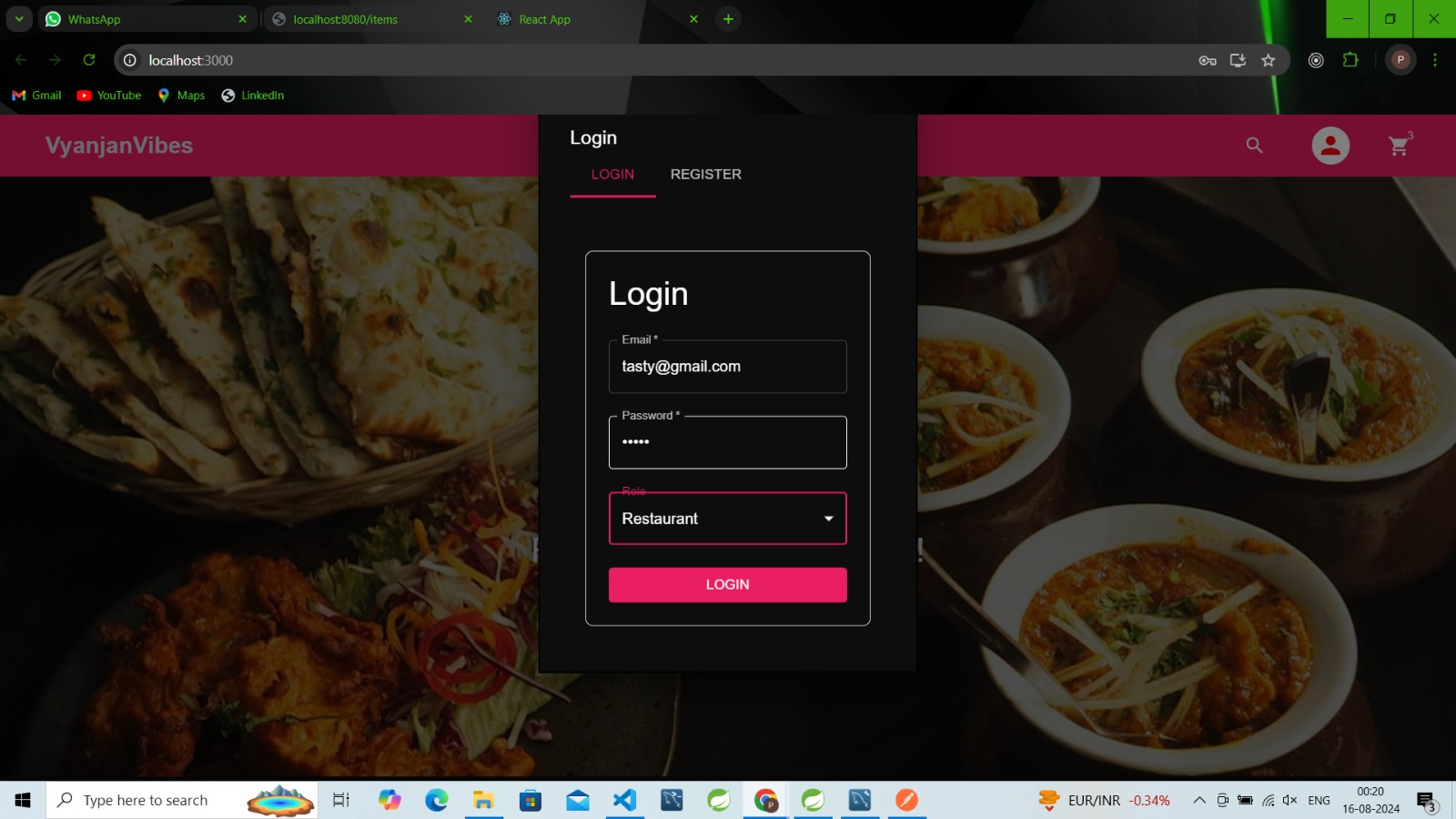
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key Type/ Constraint | Column Name | Data Type | Length | Allow Null (1=Yes;0=No) |
| PK | id | bigint |  | 0 |
|  | amount | decimal |  | 0 |
|  | card\_number | bigint |  | 0 |
|  | payment\_date | tinyblob |  | 0 |
|  | payment\_status | varchar | 30 | 0 |
| FK | order\_id | bigint |  | 0 |
| FK | user\_id | bigint |  | 0 |

**Screenshots**

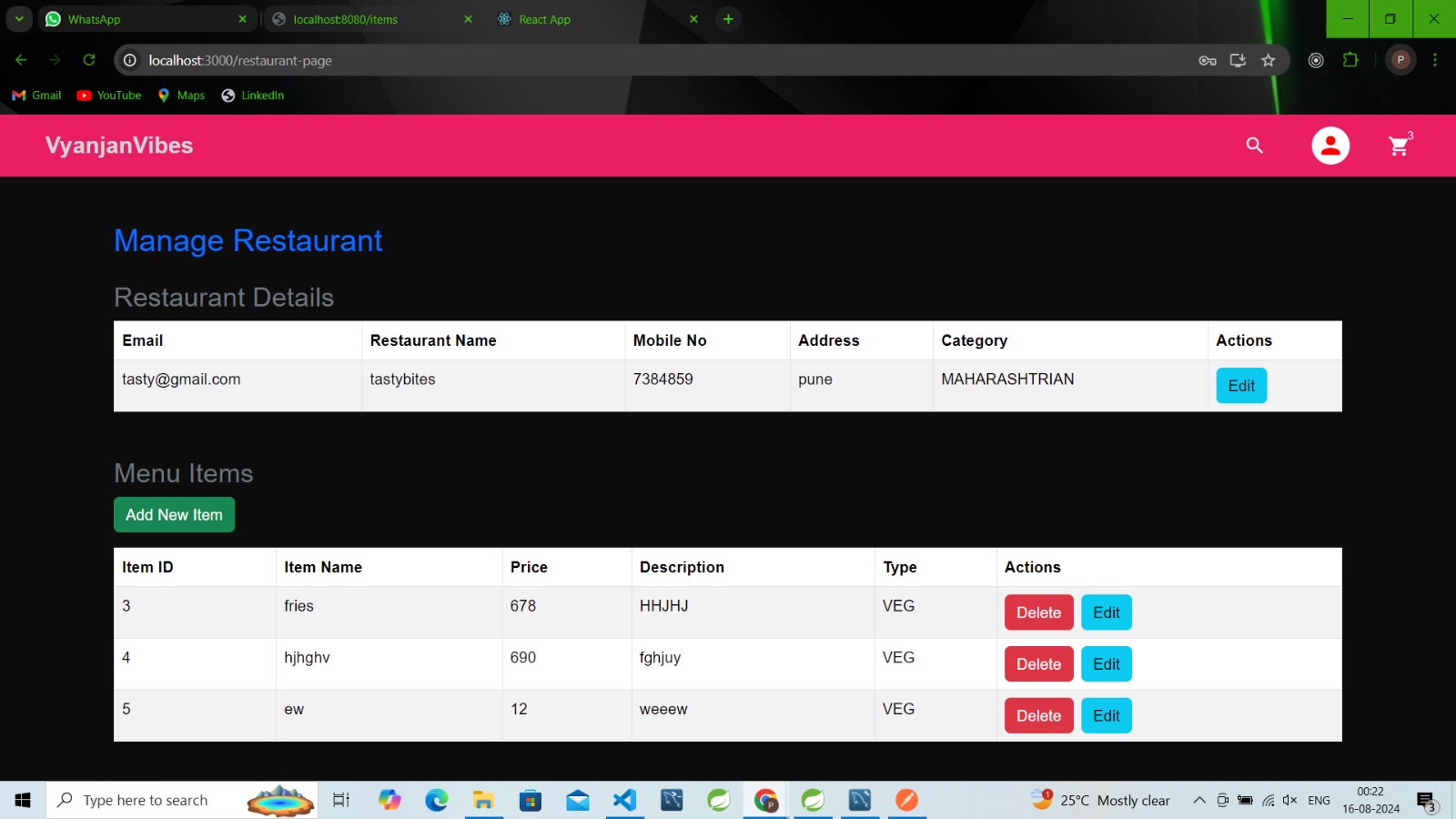
**Homepage:**

****

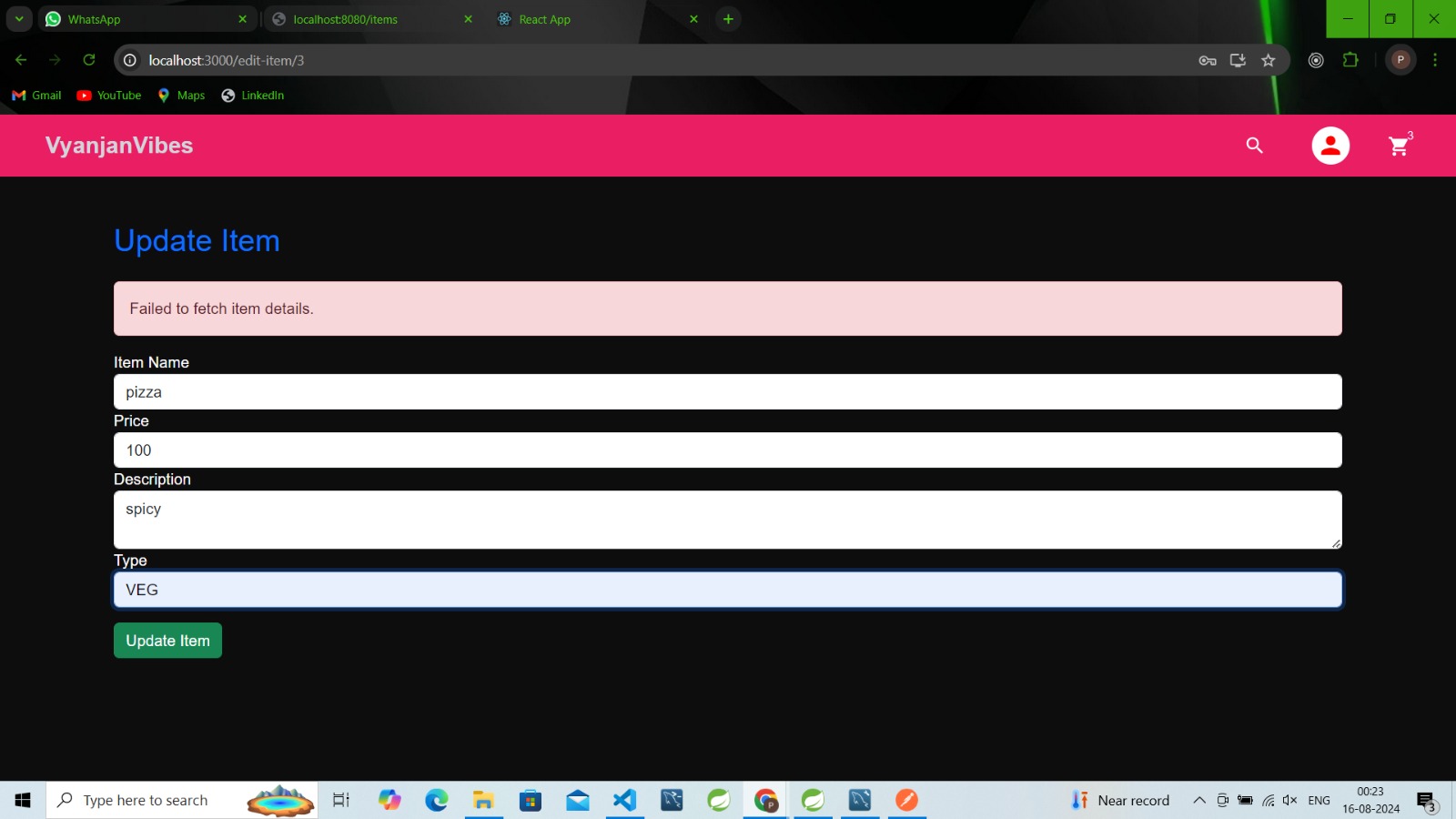
**Restaurant Registration:**

****

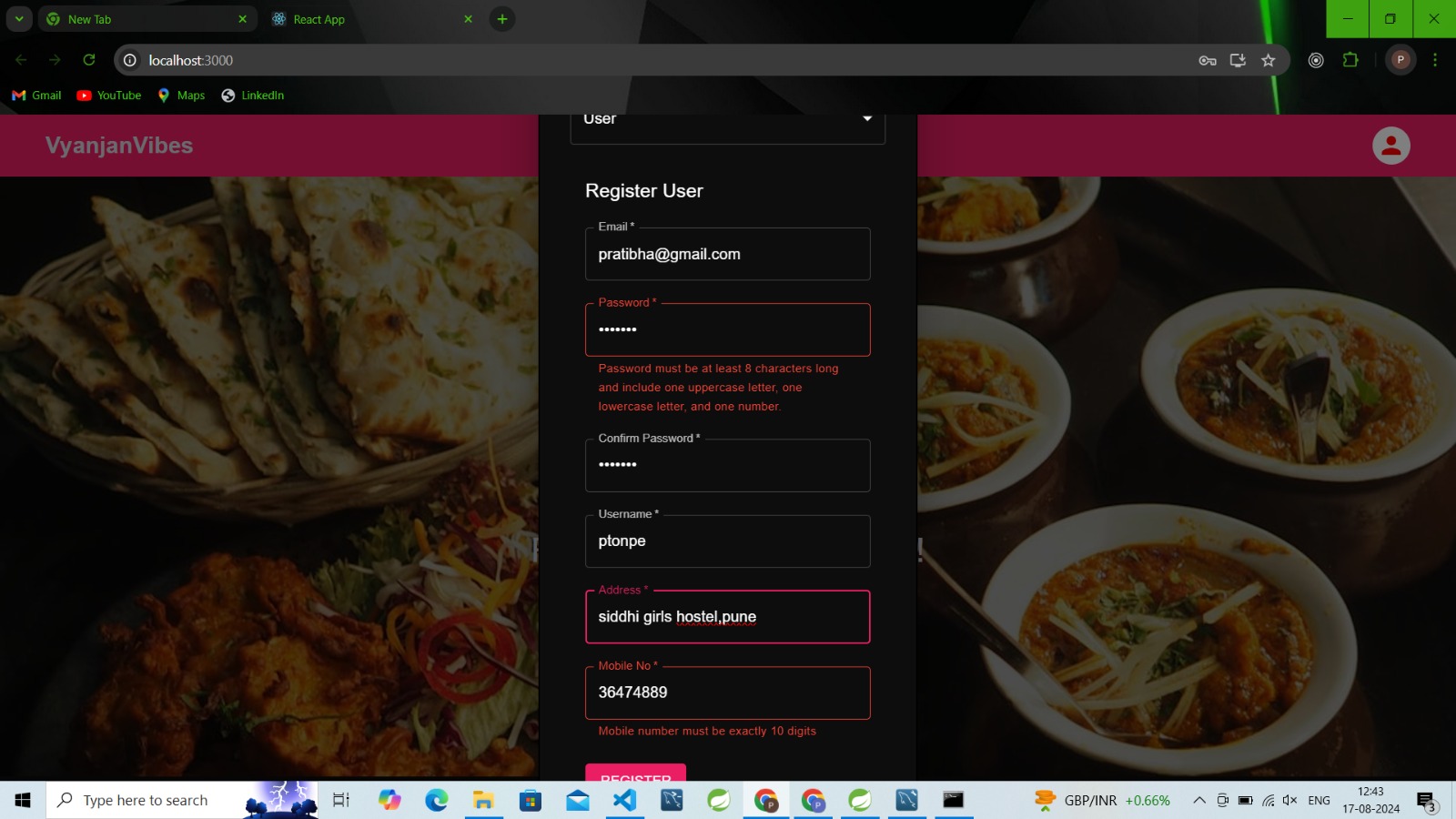
**Restaurant Dashboard:**

****

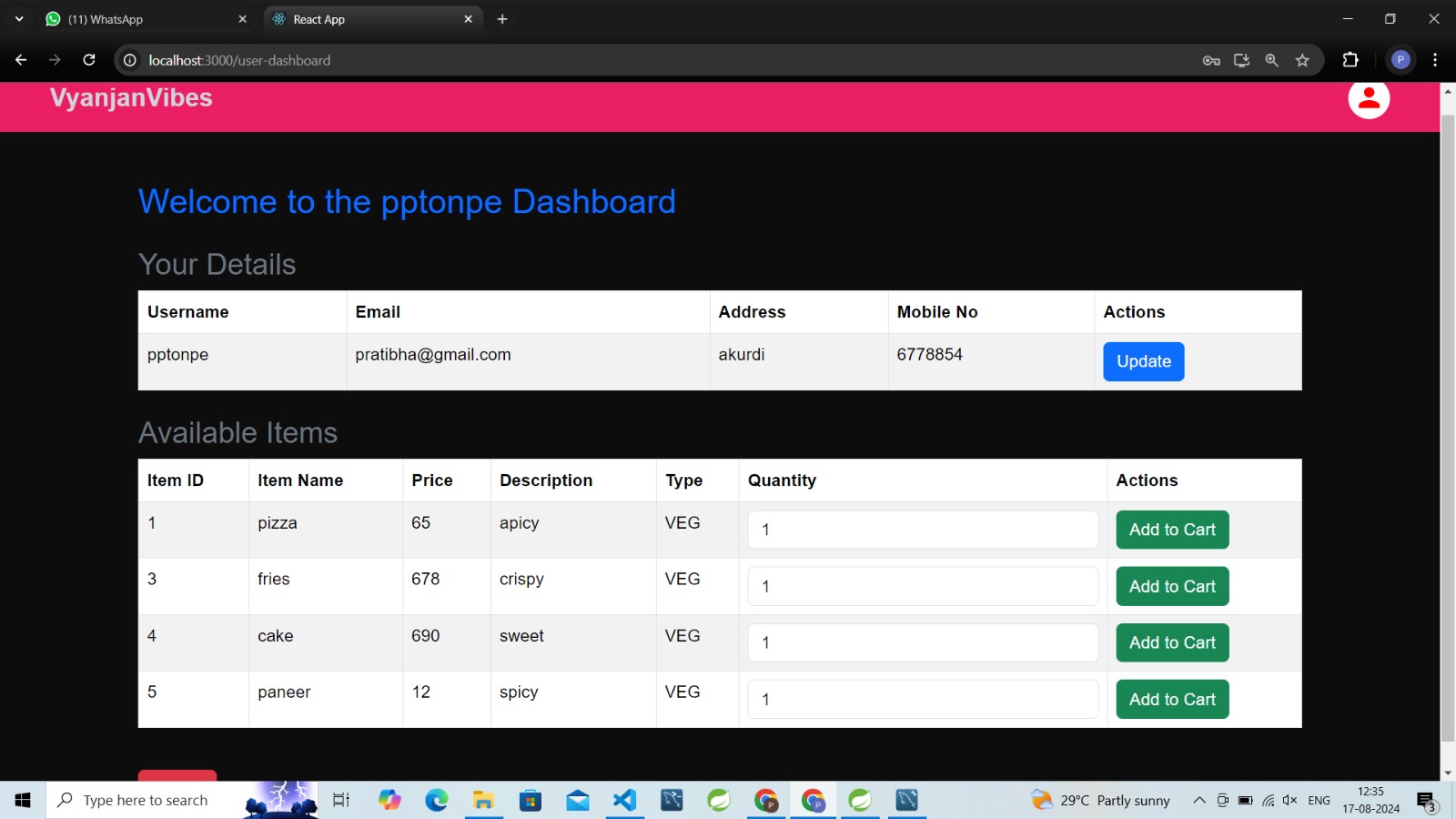
**Manage Restaurant:**

****

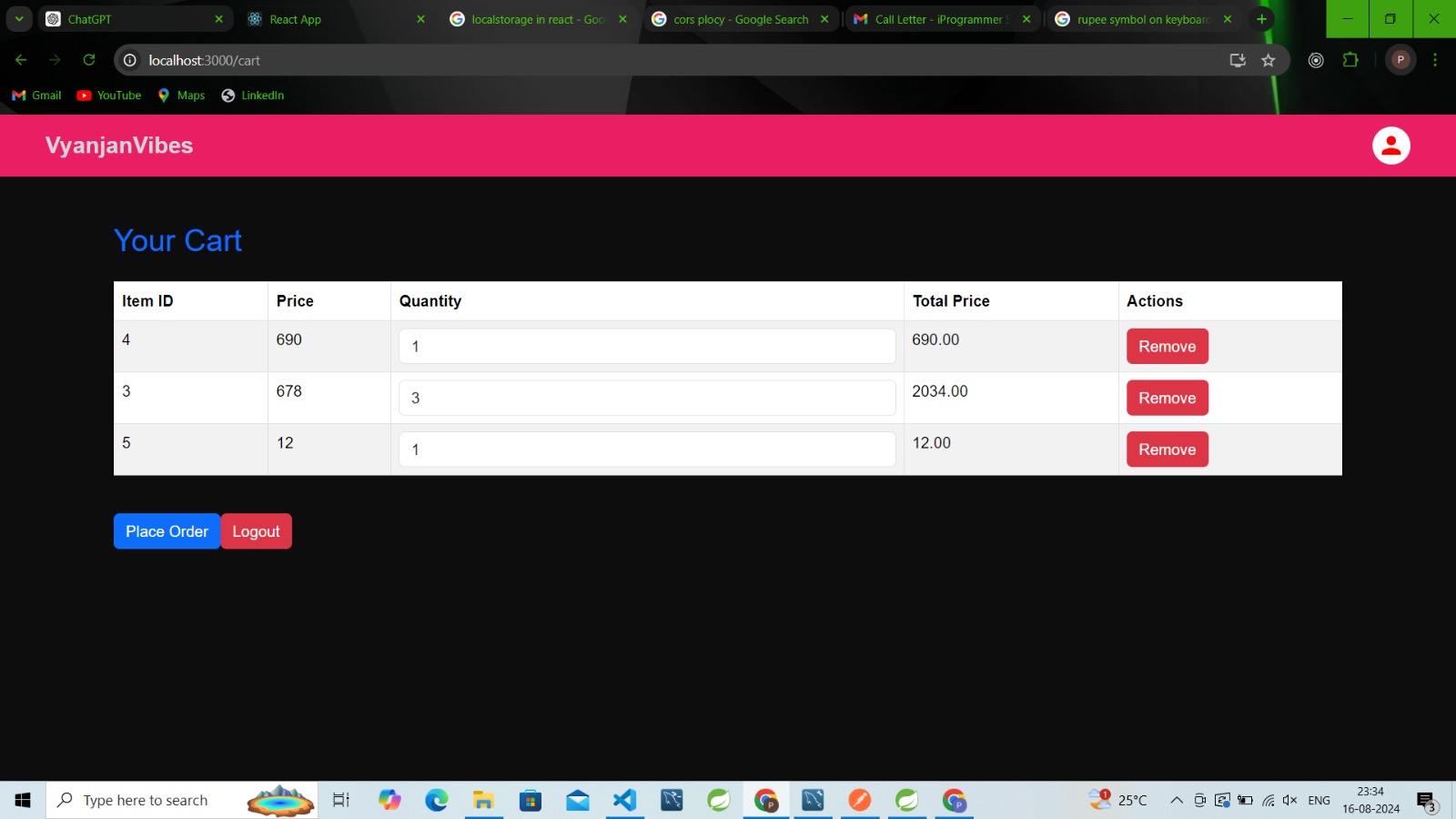
**User Registration:**

****

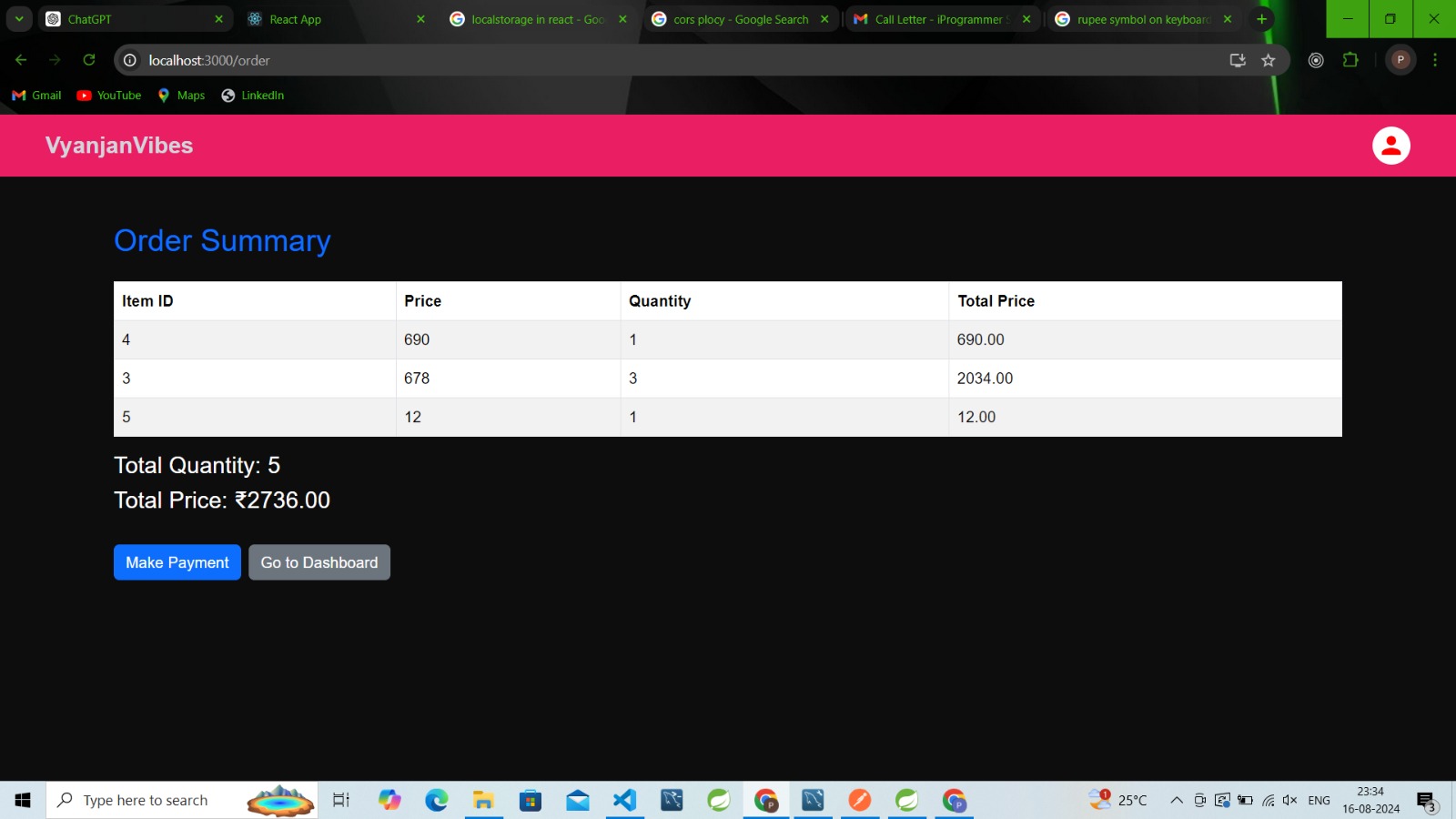
**User Dashboard:**

****

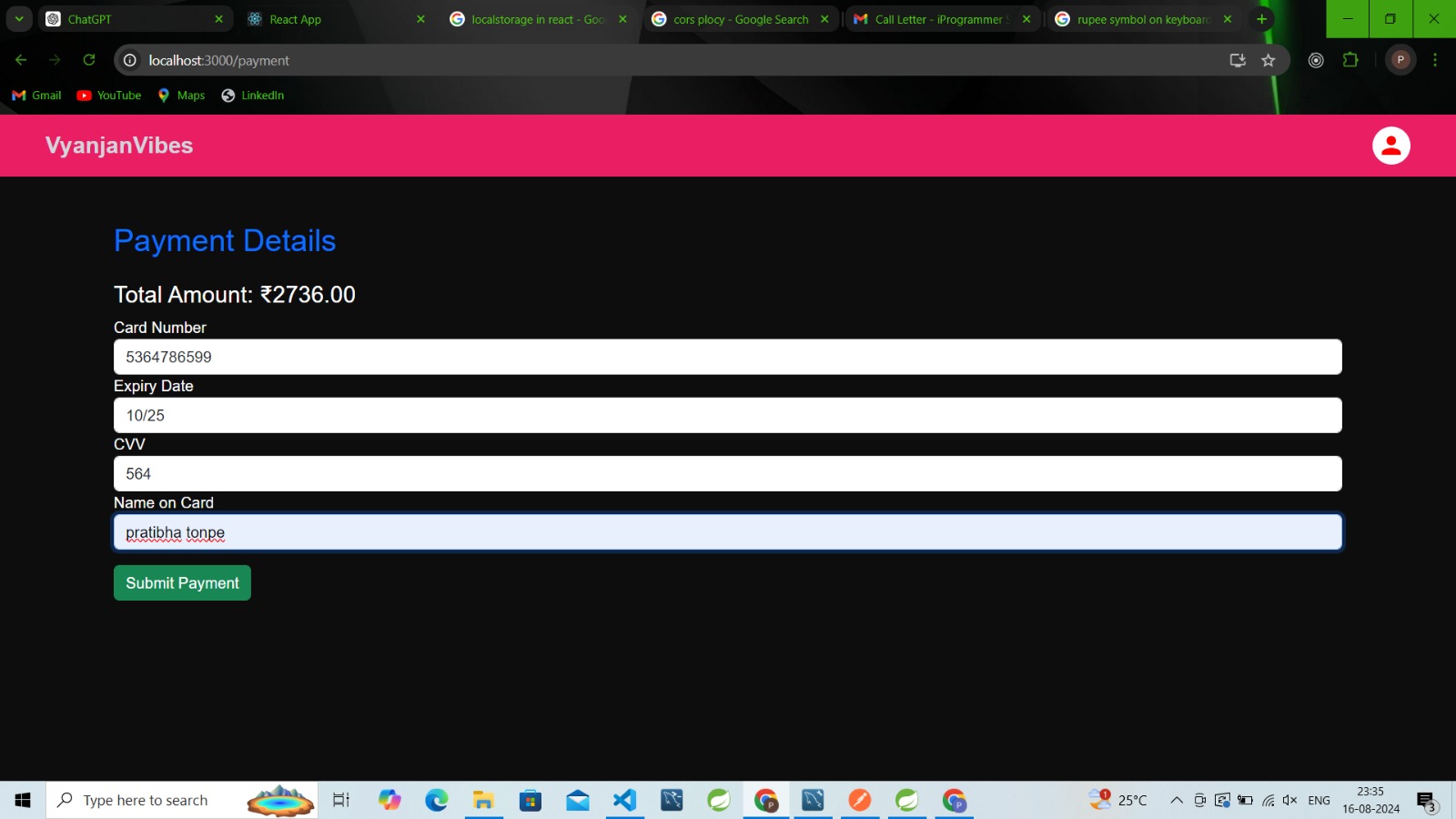
**Cart View;**

****

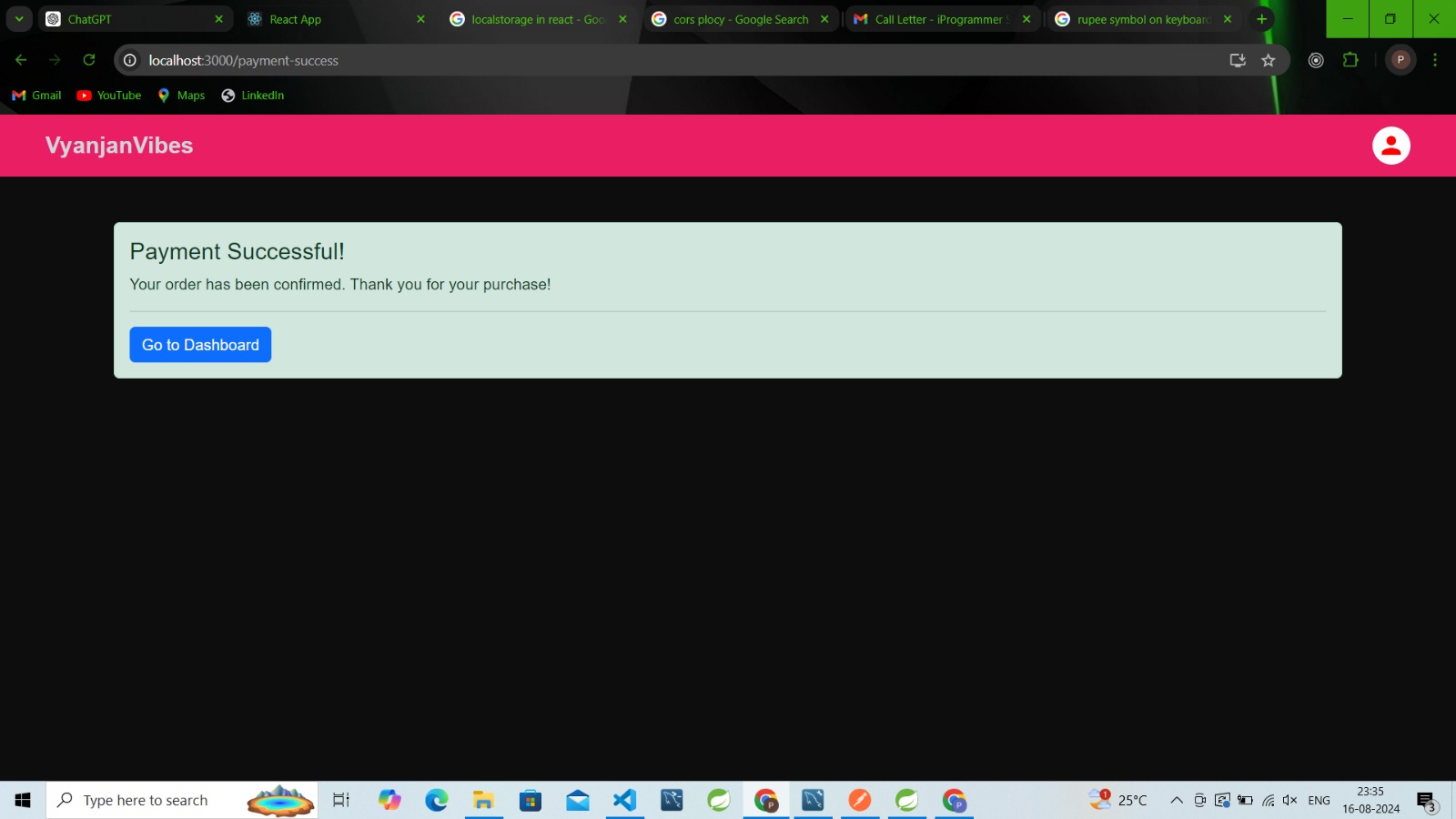
**Order View:**

****

**Payment View:**

****

**Confirmation Page:**

****

**Conclusion**

The food ordering system has revolutionized the way customers interact with restaurants, offering a seamless and efficient platform for both parties. By digitizing the ordering process, it eliminates the traditional barriers of time and location, allowing customers to access a wide range of dining options with just a few clicks. This system enhances customer satisfaction through convenience and reliability while providing restaurants with a scalable solution to manage orders and grow their business.

Looking forward, the system holds the potential for further innovations, such as incorporating personalized recommendations, real-time analytics for restaurants, and enhanced user experiences through AI and machine learning. The food ordering system is not just a tool for today's market but a foundation for future advancements in the food service industry.