UE22CS352B - Object Oriented Analysis & Design

Mini Project Report

Title

Submitted by:

Pragnan M U: *PES1UG22CS421*
Pavan Chandu C S: *PES1UG22CS410*
Pathipati Krishna Revanth: *PES1UG22CS409*
Payyavula Mahendra Chowdary: *PES1UG22CS413*

Semester: 6
Section: G

Facultly Name: Bhargavi Mokashi

January - May 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

Problem Statement:

Key Features:

1. User Authentication and Authorization
- Secure Login/Signup: Users can register and log in securely, ensuring that only authorized individuals access specific functionalities.
- Role-Based Access Control: Differentiates between various user roles such as customers, restaurant owners, and administrators, granting appropriate permissions to each.

2. Restaurant Management
- CRUD Operations: Administrators and restaurant owners can Create, Read, Update, and Delete restaurant profiles.
- Menu Management: Ability to add, modify, or remove food items, categorize them, and set prices.

3. Customer Interface
- Browse Restaurants and Menus: Customers can view a list of available restaurants and their respective menus.
- Search Functionality: Implement search features to find specific dishes or restaurants quickly.
- Order Placement: Users can select items, add them to a cart, and place orders with ease.

4. Order Management
- Real-Time Order Tracking: Customers can track the status of their orders from preparation to delivery.
- Order History: Maintain a record of past orders for reference and reordering.

5. Administrative Dashboard

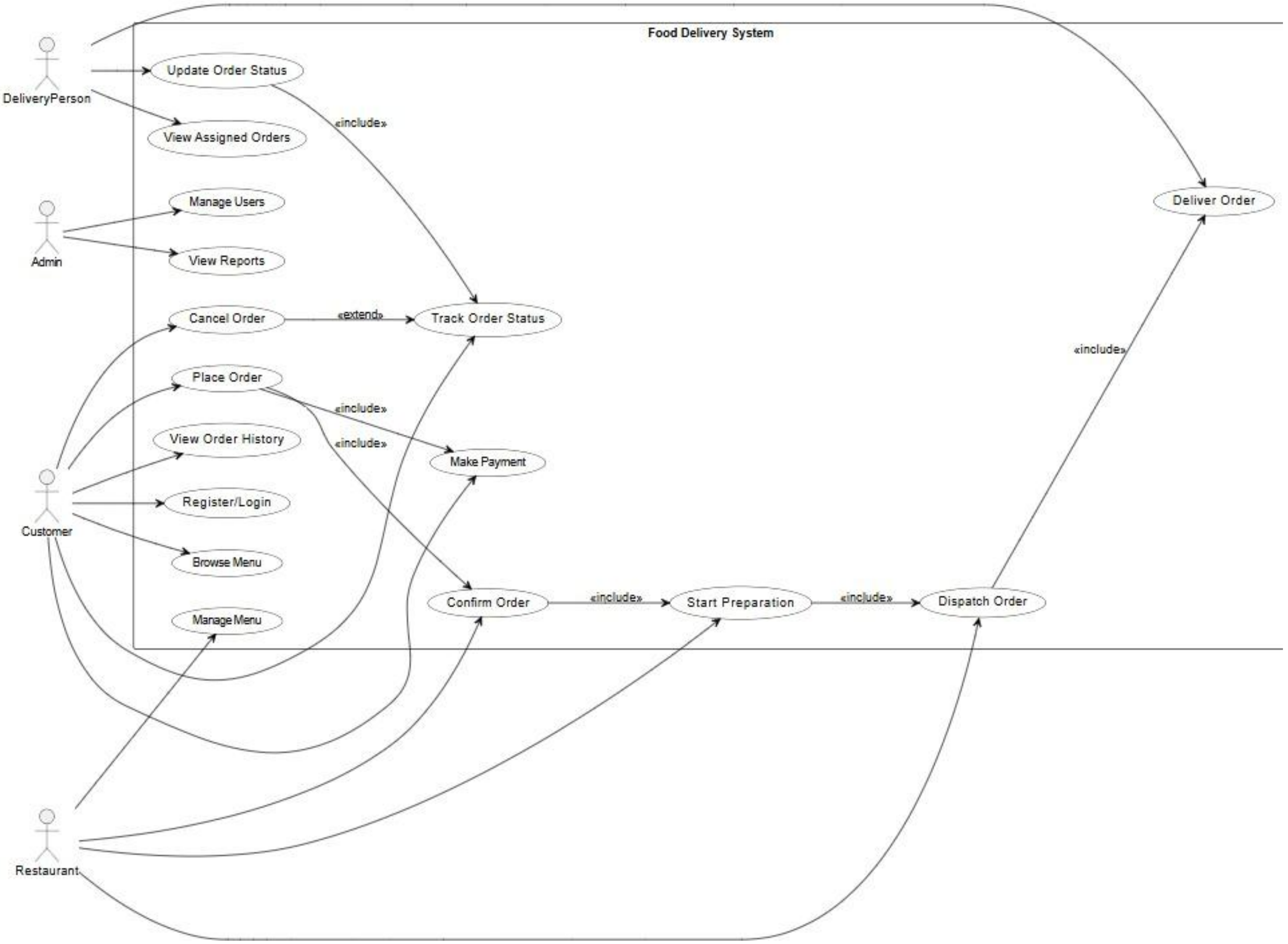
- User Management: Admins can manage user accounts, including activating or deactivating them.
- Analytics and Reporting: View reports on sales, popular dishes, and customer activity to make informed decisions.

6. Responsive Design

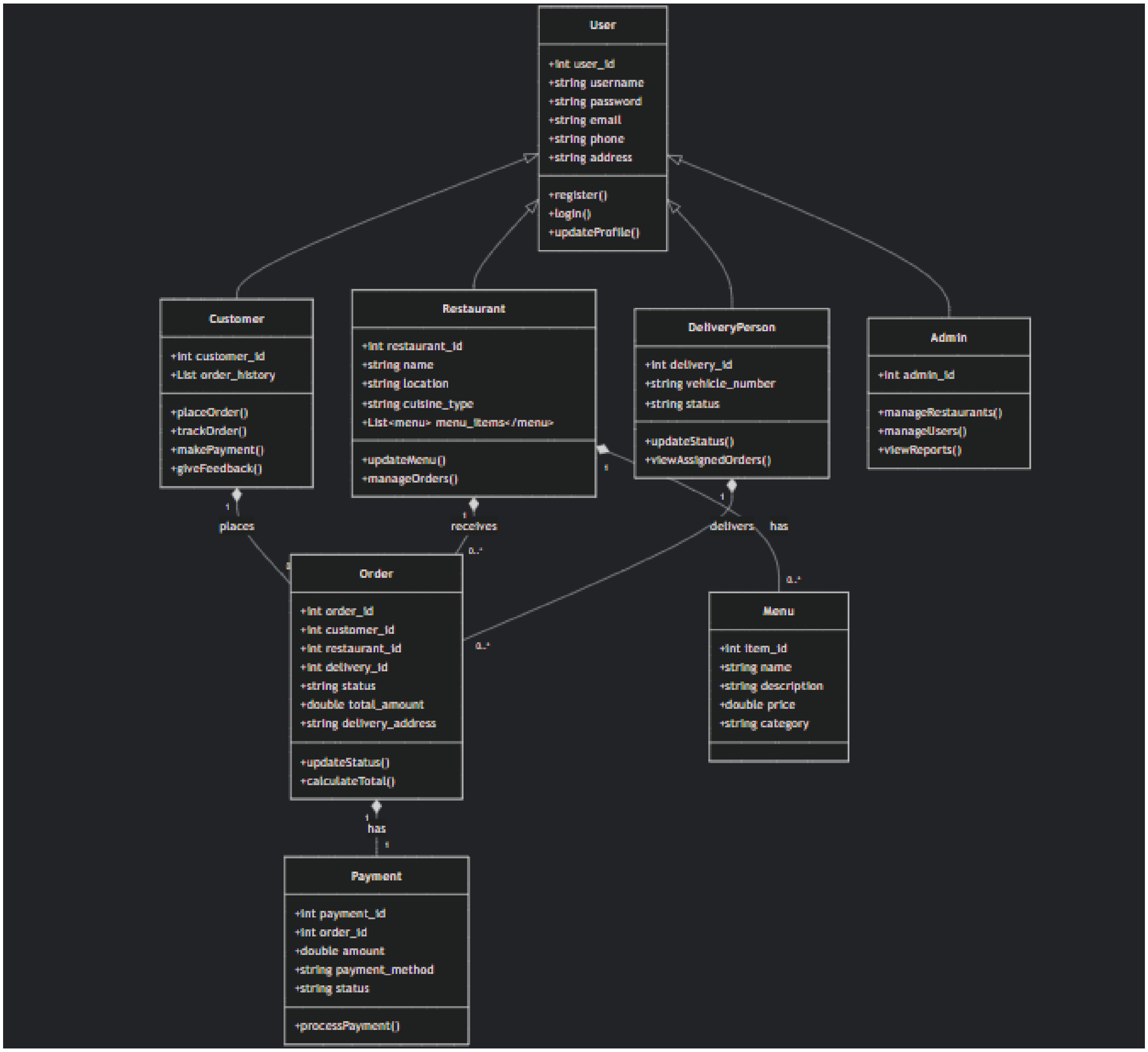
- Mobile-Friendly Interface: The frontend, built with React JS, ensures compatibility across various devices, providing a consistent user experience.

Models:

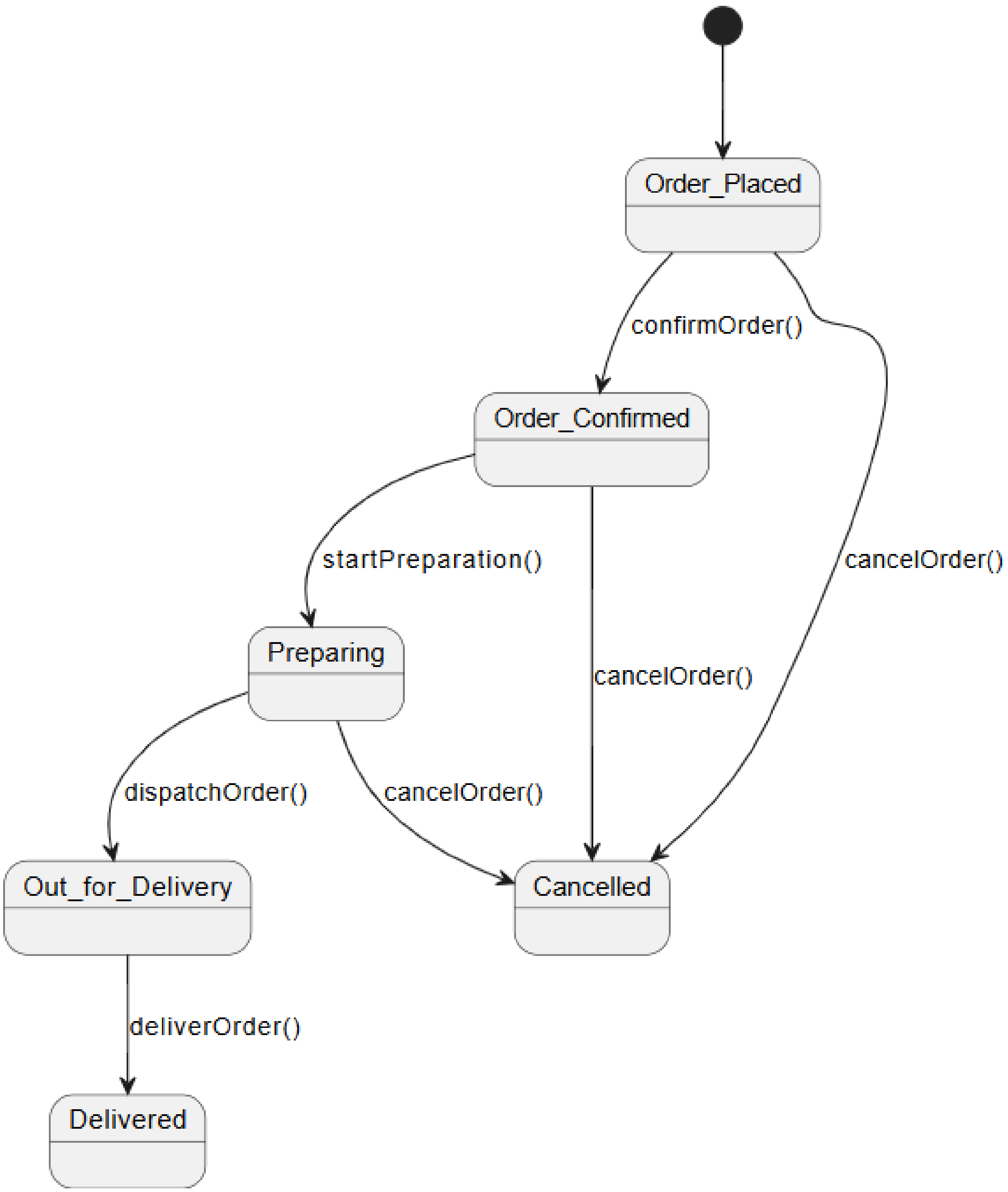
<u>Use Case Diagram:</u>



Class Diagram:

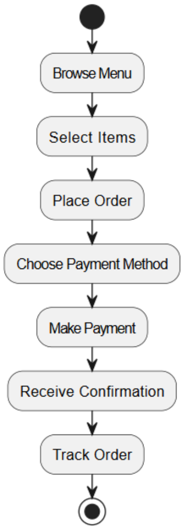


State Diagram:

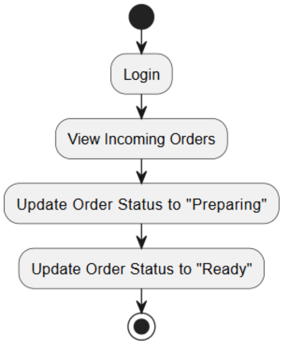


Activity Diagrams:

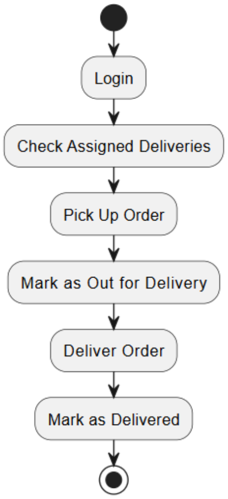
1. Major Usecase:
   Customer ordering flow:



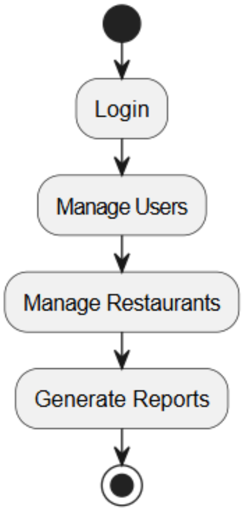
Restaurant Owner-Order Handling:


Login
View Incoming Orders
Update Order Status to "Preparing"
Update Order Status to "Ready"

Deliver person-Delivery flow:

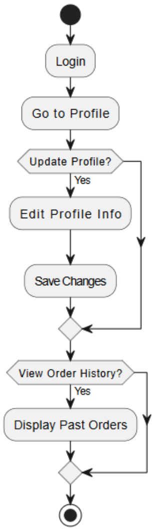


Admin-System maganement:

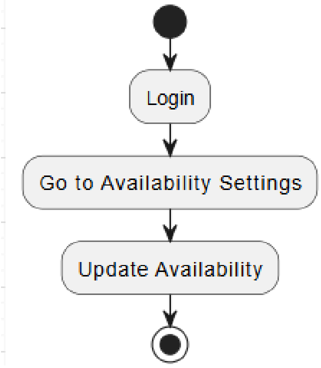


2. Minor Use case

View Order History & Update Profile:



Delivery Person - Update Availability:



Architecture Patterns, Design Principles, and Design Patterns:

Architecture Patterns

Model – View – Controller Pattern (MVC):

- Model: Represents the application's data and business logic. In this system, entities like User, Order, Menu Item, and Restaurant constitute the model layer.
- View: Handles the presentation layer. The React.js frontend serves as the view, rendering the user interface and interacting with users.
- Controller: Manages user input and interacts with the model to update the view accordingly. Spring Boot controllers process HTTP requests, invoke services, and return responses.

Design Principles:

- Separation of Concerns: By dividing the application into distinct layers (model, view, controller), each with specific responsibilities, the system promotes modularity and ease of maintenance.

- Single Responsibility Principle (SRP): Each class or module has a single, well-defined responsibility. For instance, service classes handle business logic, while controllers manage HTTP requests.

- Open/Closed Principle: The system is designed to be open for extension but closed for modification. This allows new features to be added with minimal changes to existing code.

- Dependency Injection: Utilizing Spring Boot's dependency injection facilitates loose coupling between components, enhancing testability and flexibility.

- Scalability and Performance: The architecture supports scalability to handle increasing user loads and ensures performance through efficient coding practices.

Design Patterns:

- Singleton Pattern: Ensures a class has only one instance and provides a global point of access to it. This is often used for configuration classes or logging mechanisms.

- Factory Pattern: Provides an interface for creating objects without specifying their concrete classes. This pattern is useful for creating instances of menu items or orders based on user input.

- Observer Pattern: Defines a one-to-many dependency between objects so that when one object changes state, all its dependents are notified. This is applicable in scenarios like order status updates, where customers and delivery personnel need real-time notifications.

- Strategy Pattern: Enables selecting an algorithm's behavior at runtime. For example, different payment strategies (credit card, PayPal, etc.) can be implemented using this pattern.

- Decorator Pattern: Allows behavior to be added to an individual object dynamically. This can be used to add features like promotional discounts to orders without modifying the original order class.

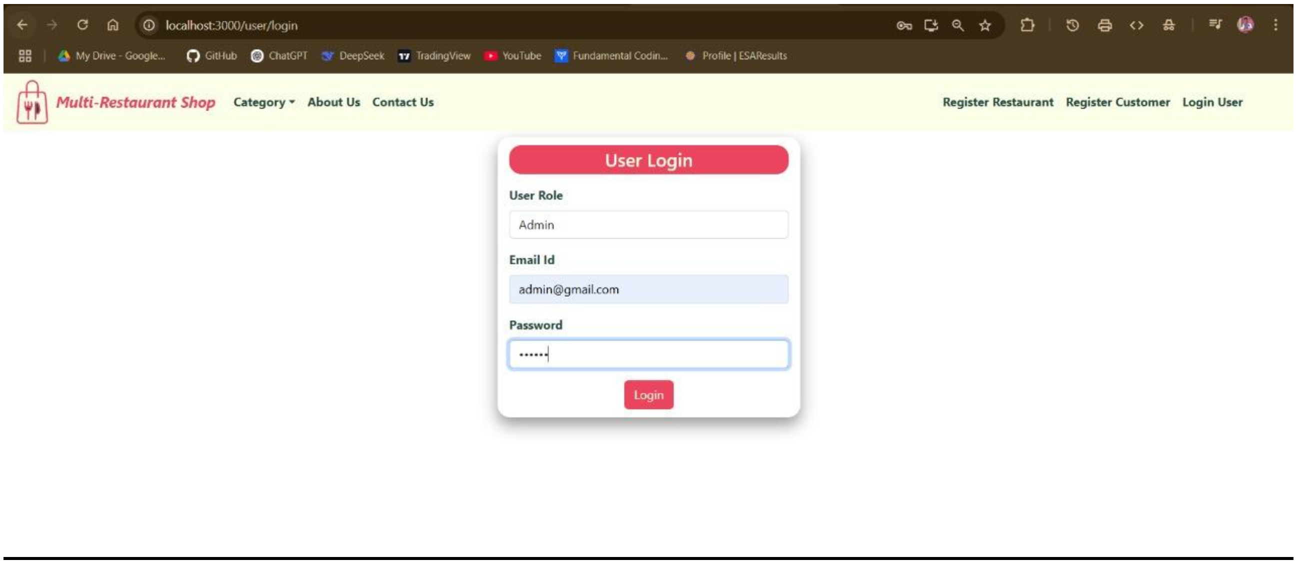
Github link to the Codebase:

https://github.com/PragnanMU/FoodDelivery

Screenshots

UI:

admin login:



User Login

User Role

Admin

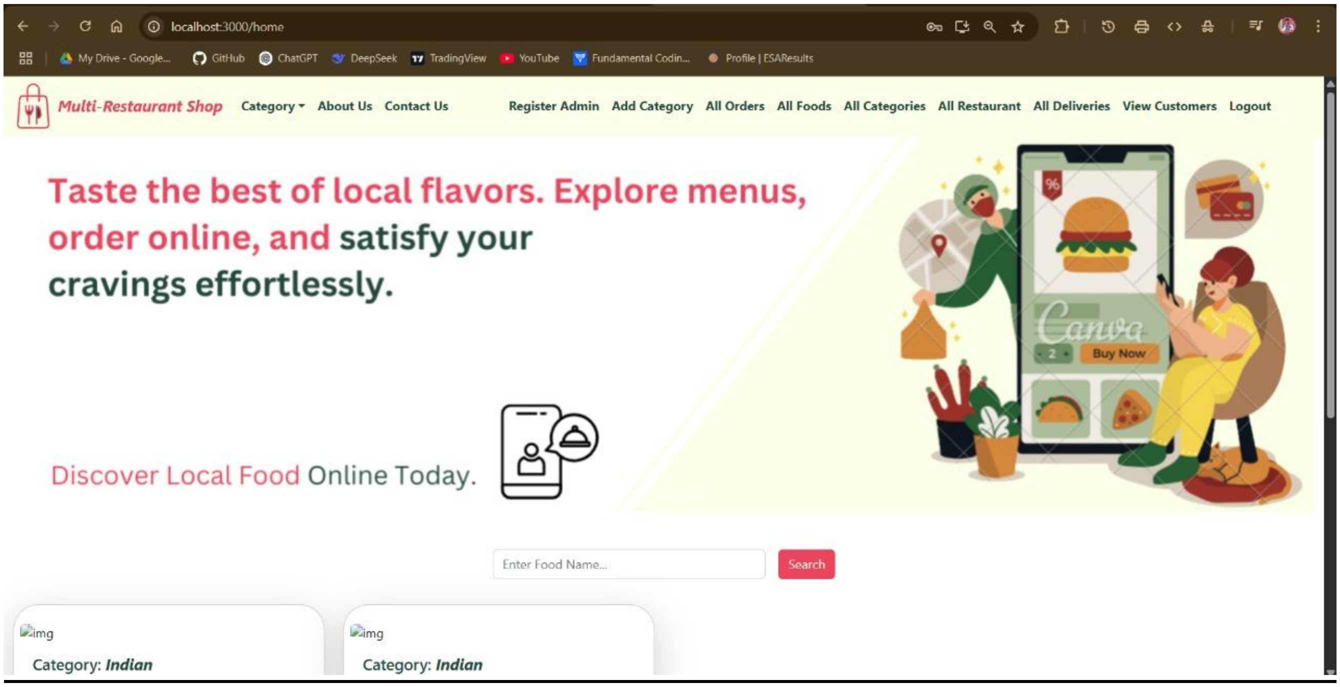
Email Id

admin@gmail.com

Password

••••••

Admin interface:



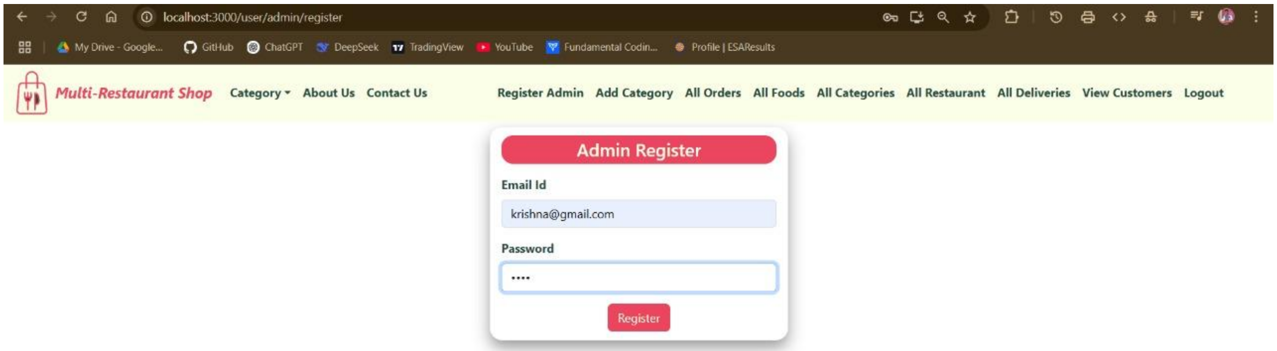
Taste the best of local flavors. Explore menus, order online, and satisfy your cravings effortlessly.

Discover Local Food Online Today.

Category: *Indian*

Category: *Indian*

Register admin:

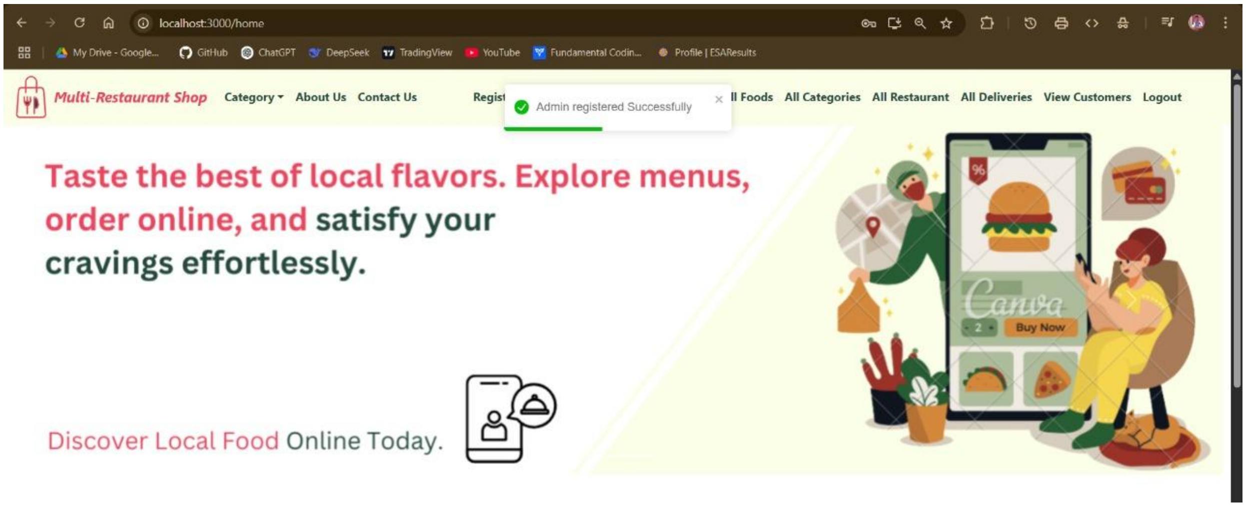

localhost:3000/user/admin/register
My Drive - Google... GitHub ChatGPT DeepSeek TradingView YouTube Fundamental Codin... Profile | ESAResults
Multi-Restaurant Shop
Category About Us Contact Us
Register Admin Add Category All Orders All Foods All Categories All Restaurant All Deliveries View Customers Logout
Register

Admin Register

Email Id: krishna@gmail.com

Password: ••••

Admin Registration successful:

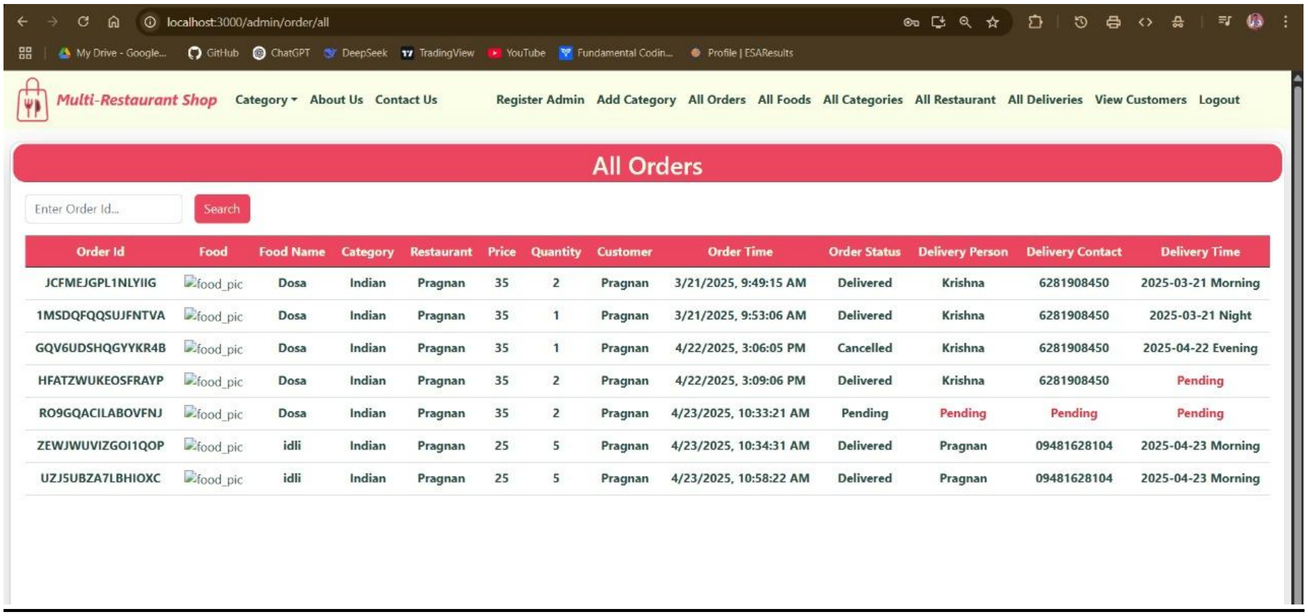

localhost:3000/home
My Drive - Google... GitHub ChatGPT DeepSeek TradingView YouTube Fundamental Codin... Profile | ESAResults
Multi-Restaurant Shop
Category About Us Contact Us
Regist ll Foods All Categories All Restaurant All Deliveries View Customers Logout
Canva
Buy Now

Admin registered Successfully

Taste the best of local flavors. Explore menus, order online, and satisfy your cravings effortlessly.

Discover Local Food Online Today.

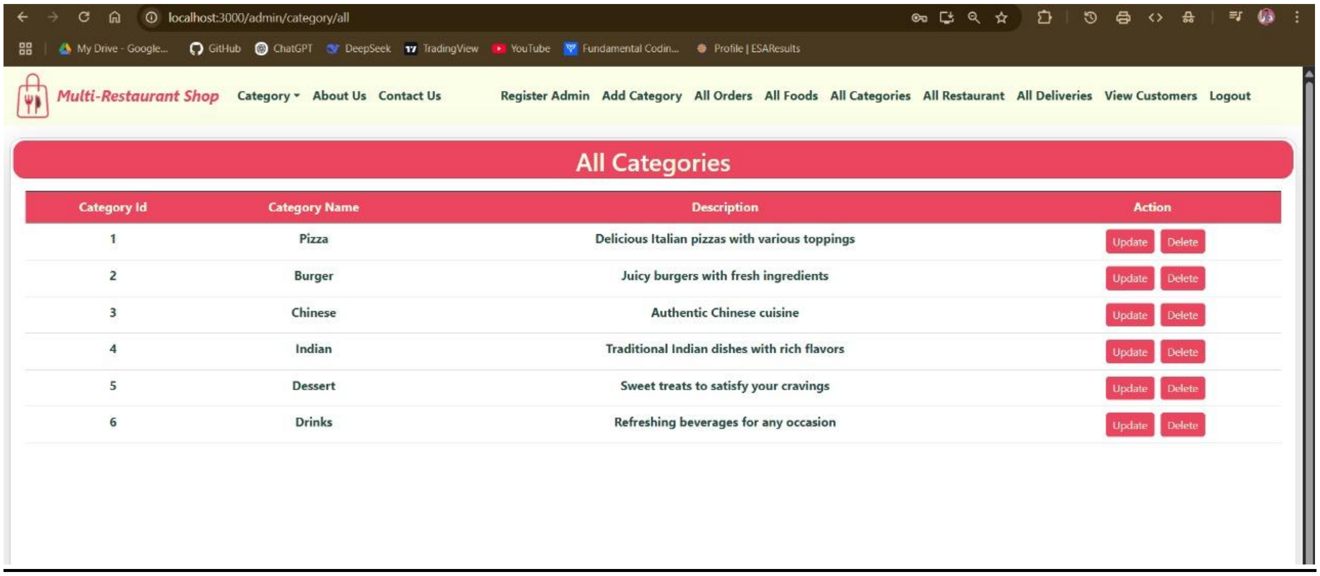
Admin viewing orders:



All Orders

Order Id	Food	Food Name	Category	Restaurant	Price	Quantity	Customer	Order Time	Order Status	Delivery Person	Delivery Contact	Delivery Time
JCFMEJGPL1NLYIIG	food_pic	Dosa	Indian	Pragnan	35	2	Pragnan	3/21/2025, 9:49:15 AM	Delivered	Krishna	6281908450	2025-03-21 Morning
1MSDQFQQSUJFNTVA	food_pic	Dosa	Indian	Pragnan	35	1	Pragnan	3/21/2025, 9:53:06 AM	Delivered	Krishna	6281908450	2025-03-21 Night
GQV6UDSHQGYYKR4B	food_pic	Dosa	Indian	Pragnan	35	1	Pragnan	4/22/2025, 3:06:05 PM	Cancelled	Krishna	6281908450	2025-04-22 Evening
HFATZWUKEOSFRAYP	food_pic	Dosa	Indian	Pragnan	35	2	Pragnan	4/22/2025, 3:09:06 PM	Delivered	Krishna	6281908450	Pending
RO9GQACILABOVFNJ	food_pic	Dosa	Indian	Pragnan	35	2	Pragnan	4/23/2025, 10:33:21 AM	Pending	Pending	Pending	Pending
ZEWJWUVIZGOI1QOP	food_pic	idli	Indian	Pragnan	25	5	Pragnan	4/23/2025, 10:34:31 AM	Delivered	Pragnan	09481628104	2025-04-23 Morning
UZJ5UBZA7LBHIOXC	food_pic	idli	Indian	Pragnan	25	5	Pragnan	4/23/2025, 10:58:22 AM	Delivered	Pragnan	09481628104	2025-04-23 Morning

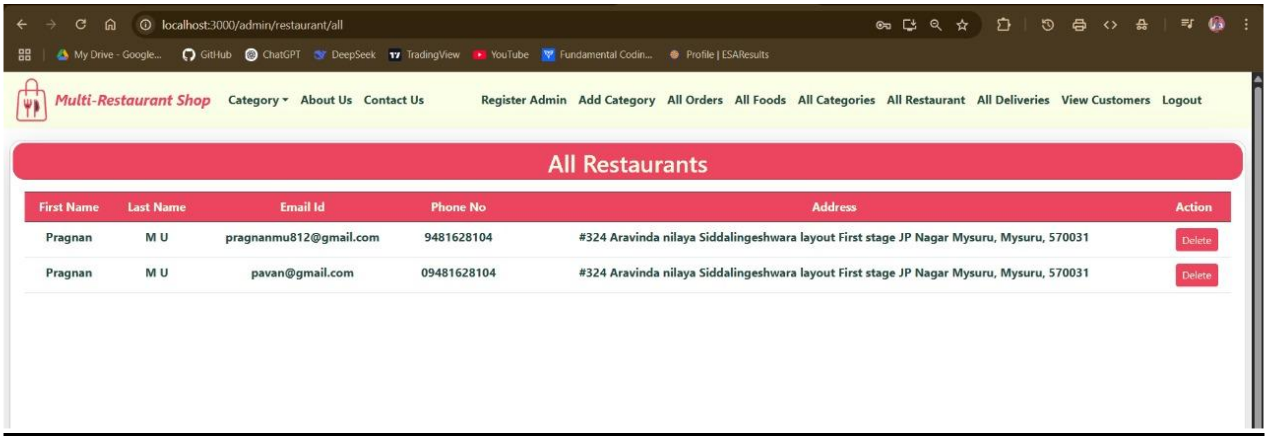
Admin Managing orders:



All Categories

Category Id	Category Name	Description	Action
1	Pizza	Delicious Italian pizzas with various toppings	Update Delete
2	Burger	Juicy burgers with fresh ingredients	Update Delete
3	Chinese	Authentic Chinese cuisine	Update Delete
4	Indian	Traditional Indian dishes with rich flavors	Update Delete
5	Dessert	Sweet treats to satisfy your cravings	Update Delete
6	Drinks	Refreshing beverages for any occasion	Update Delete

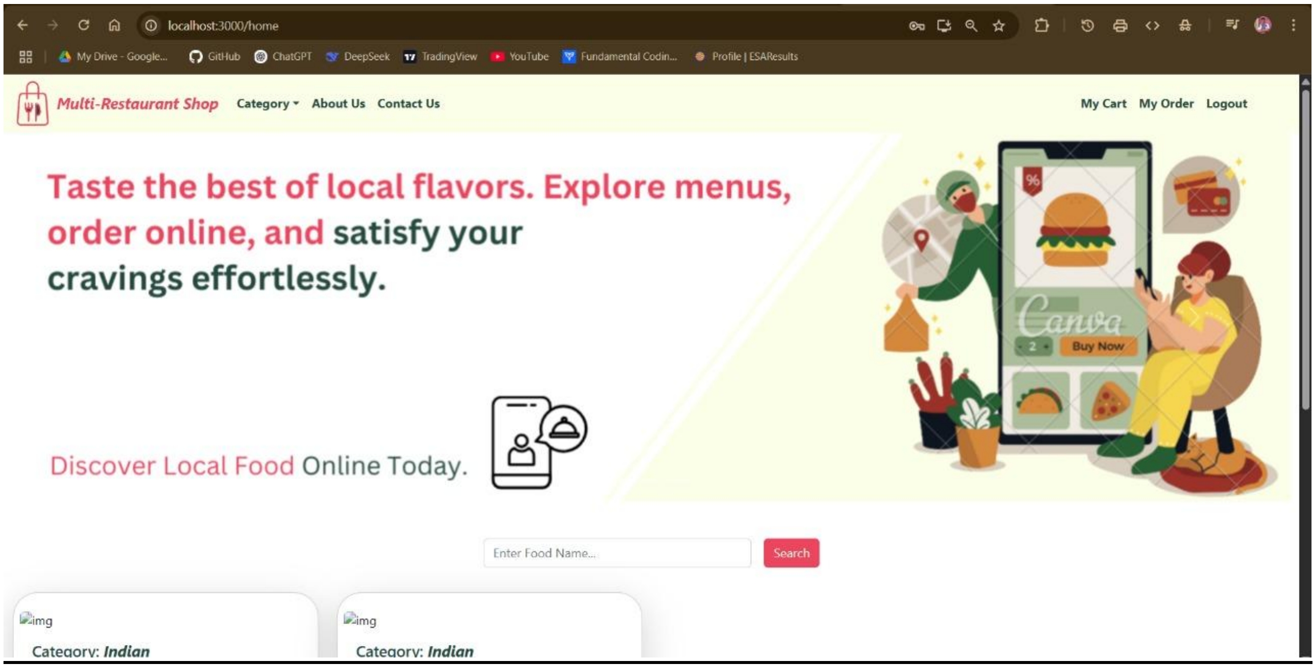
Number of Restaurants:



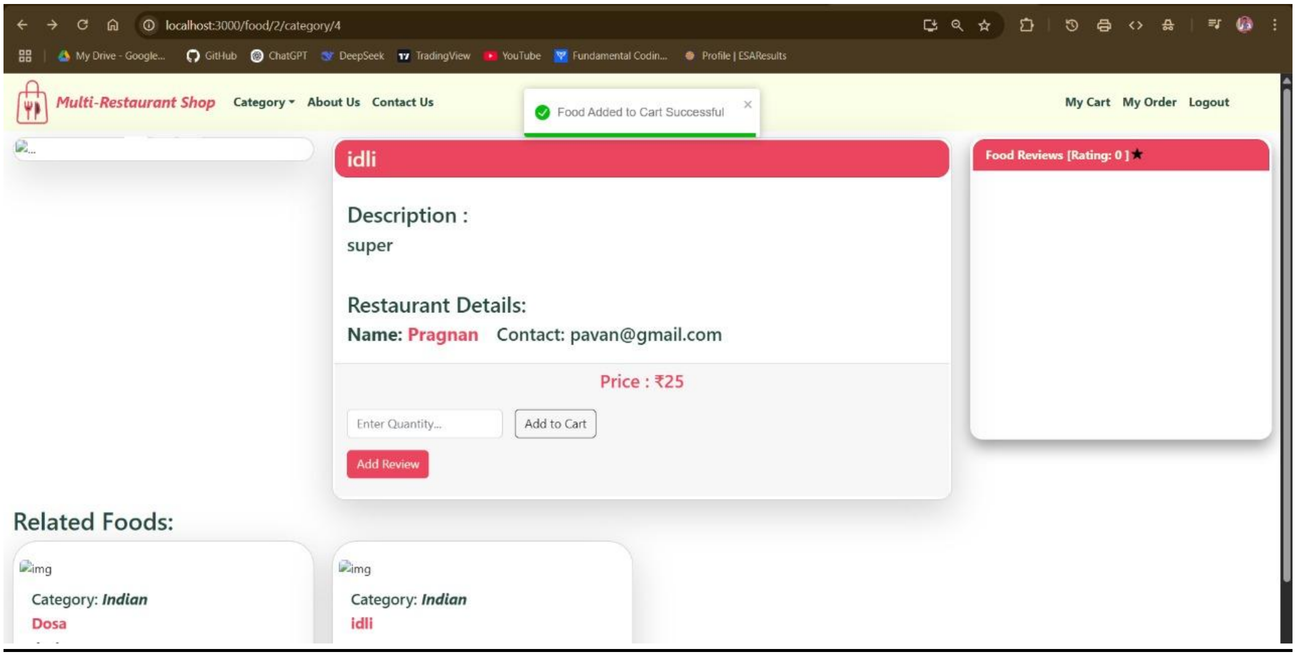
All Restaurants

First Name	Last Name	Email Id	Phone No	Address	Action
Pragnan	M U	pragnanmu812@gmail.com	9481628104	#324 Aravinda nilaya Siddalingeshwara layout First stage JP Nagar Mysuru, Mysuru, 570031	Delete
Pragnan	M U	pavan@gmail.com	09481628104	#324 Aravinda nilaya Siddalingeshwara layout First stage JP Nagar Mysuru, Mysuru, 570031	Delete

Customer login:



Customer Ordering food:



Food Added to Cart Successful

idli

Description :

super

Restaurant Details:

Name: Pragnan Contact: pavan@gmail.com

Price : ₹25

Enter Quantity... | Add to Cart

Add Review

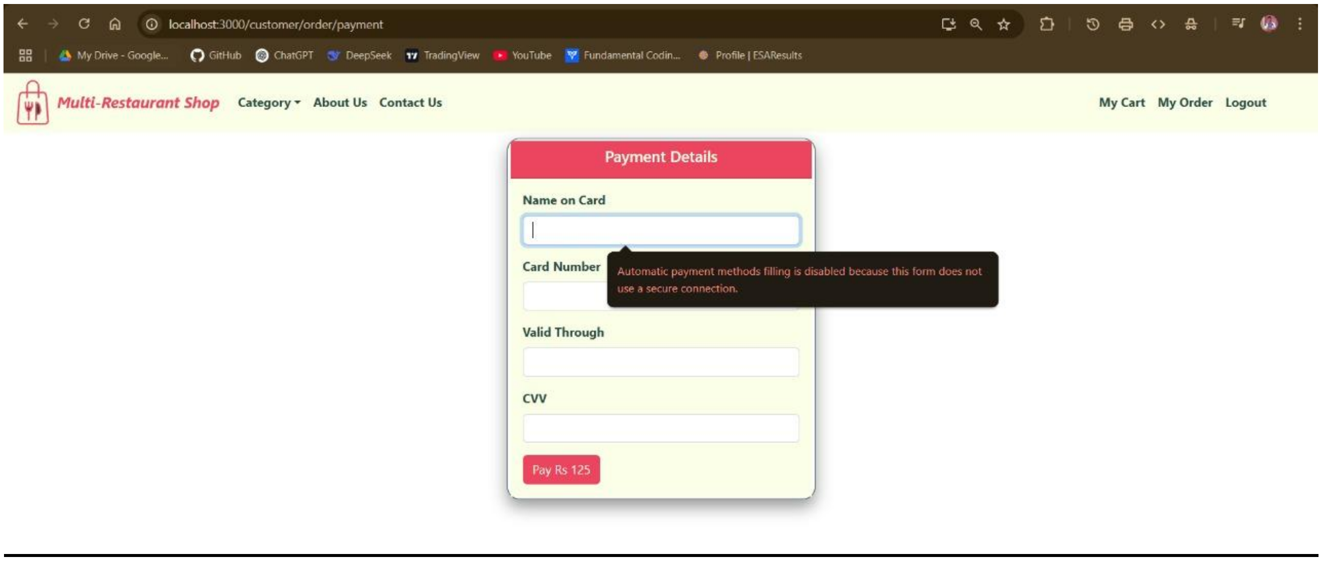
Food Reviews [Rating: 0]★

Related Foods:

Category: *Indian*
Dosa

Category: *Indian*
idli

Payment Interface:



Payment Details

Name on Card

Card Number

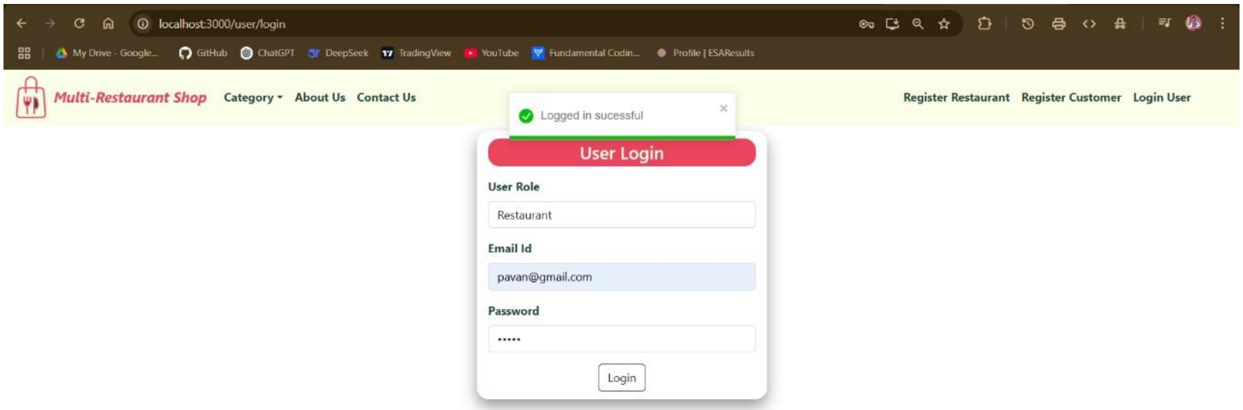
Automatic payment methods filling is disabled because this form does not use a secure connection.

Valid Through

CVV

Pay Rs 125

Restaurant Login:



User Login

User Role

Restaurant

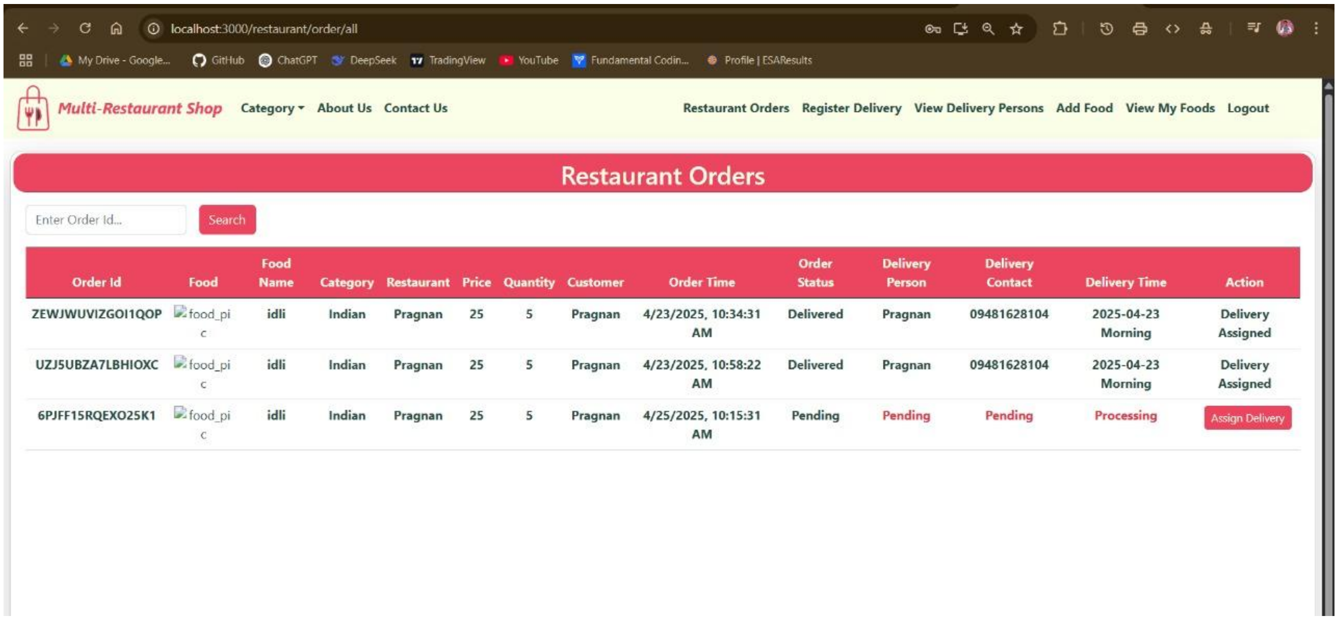
Email Id

pavan@gmail.com

Password

•••••

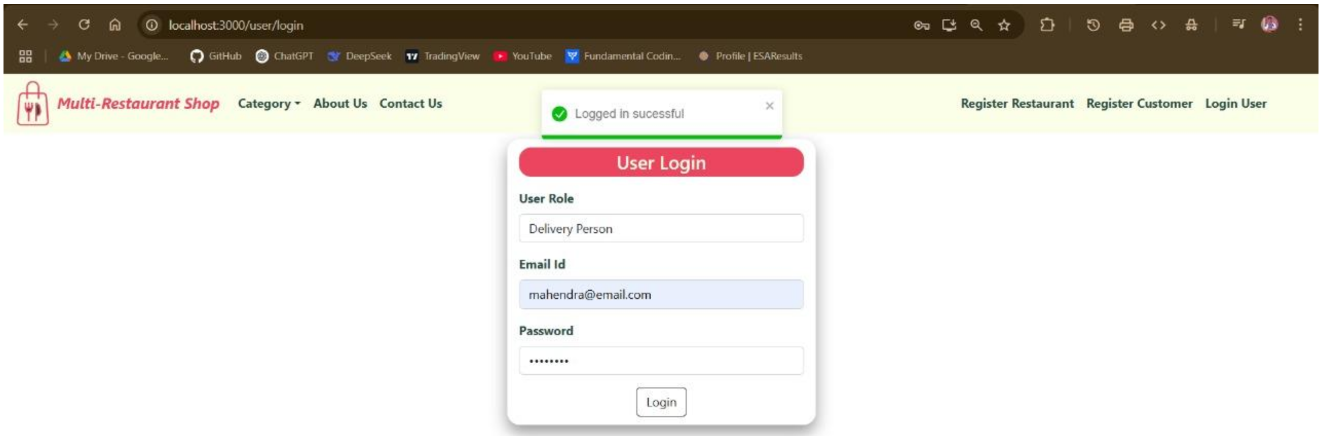
Restaurant Viewing Orders:



Restaurant Orders

Order Id	Food	Food Name	Category	Restaurant	Price	Quantity	Customer	Order Time	Order Status	Delivery Person	Delivery Contact	Delivery Time	Action
ZEWJWUVIZGOI1QOP	food_pic	idli	Indian	Pragnan	25	5	Pragnan	4/23/2025, 10:34:31 AM	Delivered	Pragnan	09481628104	2025-04-23 Morning	Delivery Assigned
UZJ5UBZA7LBHIOXC	food_pic	idli	Indian	Pragnan	25	5	Pragnan	4/23/2025, 10:58:22 AM	Delivered	Pragnan	09481628104	2025-04-23 Morning	Delivery Assigned
6PJFF15RQEXO25K1	food_pic	idli	Indian	Pragnan	25	5	Pragnan	4/25/2025, 10:15:31 AM	Pending	Pending	Pending	Processing	Assign Delivery

Delivery Partner login:


localhost:3000/user/login
My Drive - Google... GitHub ChatGPT DeepSeek TradingView YouTube Fundamental Codin... Profile | ESAResults
Multi-Restaurant Shop
Category
About Us
Contact Us
Register Restaurant
Register Customer
Login User

Logged in sucessful

User Login

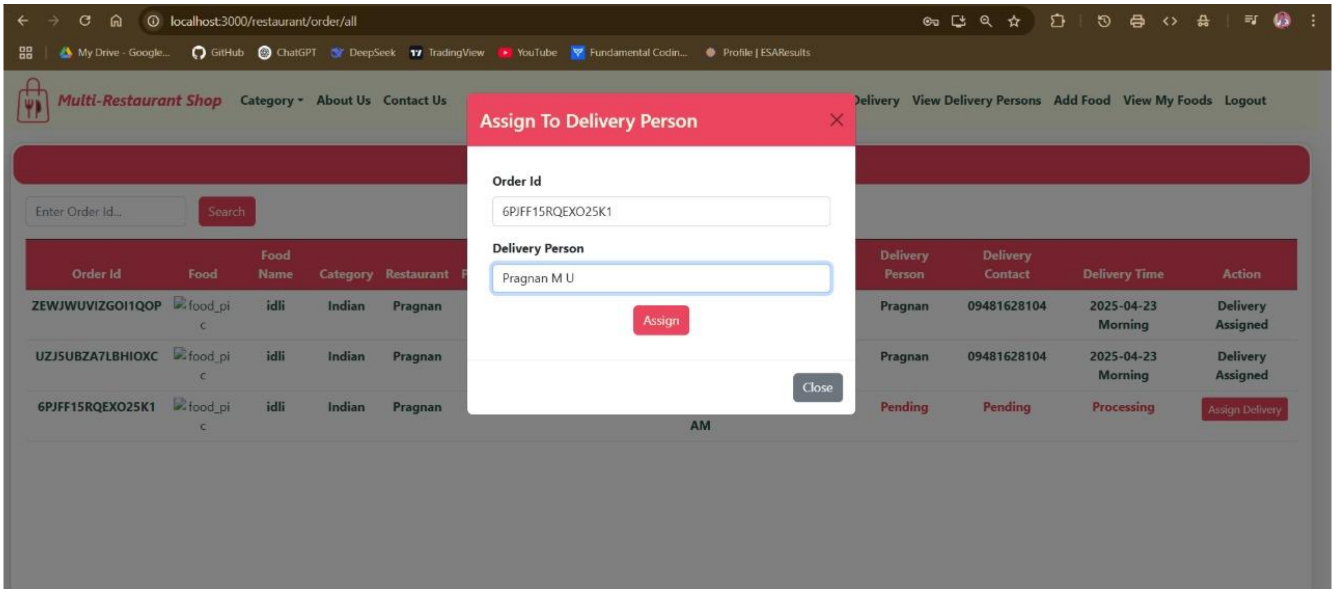
User Role: Delivery Person

Email Id: mahendra@email.com

Password: ••••••••

Login

Assigned for Delivery:


localhost:3000/restaurant/order/all
My Drive - Google... GitHub ChatGPT DeepSeek TradingView YouTube Fundamental Codin... Profile | ESAResults
Multi-Restaurant Shop
Category
About Us
Contact Us
elivery
View Delivery Persons
Add Food
View My Foods
Logout
Search

Assign To Delivery Person

Order Id: 6PJFF15RQEXO25K1

Delivery Person: Pragnan M U

Assign

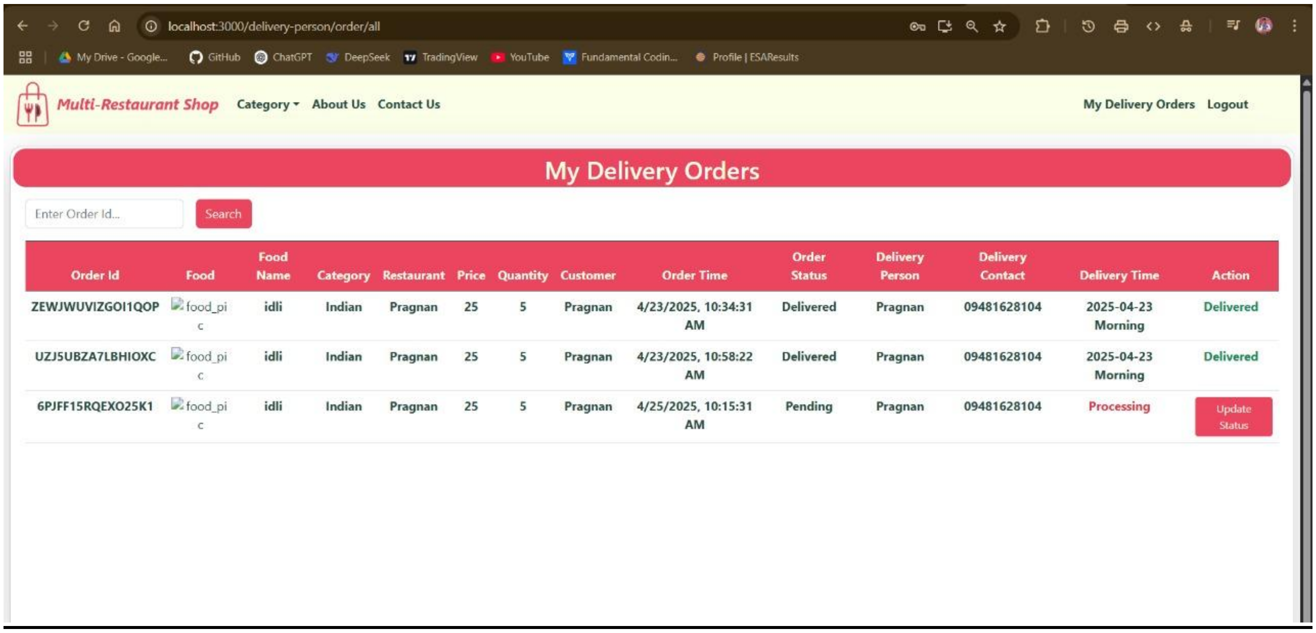
Close

Enter Order Id...

Order Id	Food	Food Name	Category	Restaurant	F		Delivery Person	Delivery Contact	Delivery Time	Action
ZEWJWUVIZGOI1QOP	food_pic	idli	Indian	Pragnan			Pragnan	09481628104	2025-04-23 Morning	Delivery Assigned
UZJ5UBZA7LBHIOXC	food_pic	idli	Indian	Pragnan			Pragnan	09481628104	2025-04-23 Morning	Delivery Assigned
6PJFF15RQEXO25K1	food_pic	idli	Indian	Pragnan		AM	Pending	Pending	Processing	Assign Delivery

---

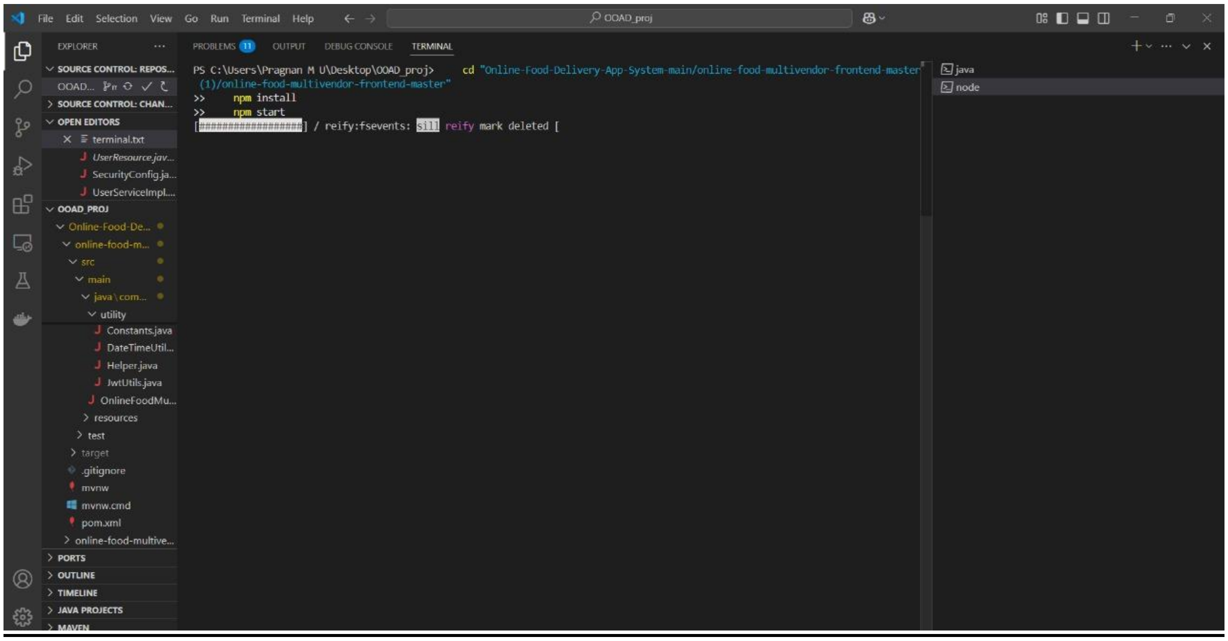
Delivery Partner Status update:



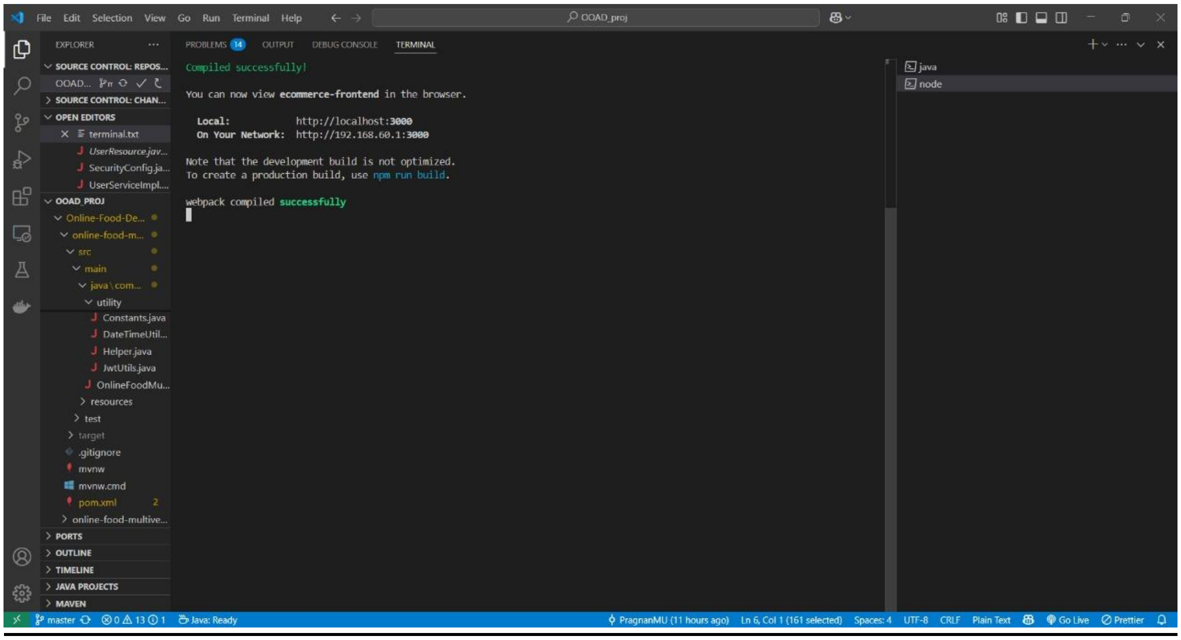
My Delivery Orders

Order Id	Food	Food Name	Category	Restaurant	Price	Quantity	Customer	Order Time	Order Status	Delivery Person	Delivery Contact	Delivery Time	Action
ZEWJWUVIZGOI1QOP	food_pic	idli	Indian	Pragnan	25	5	Pragnan	4/23/2025, 10:34:31 AM	Delivered	Pragnan	09481628104	2025-04-23 Morning	Delivered
UZJ5UBZA7LBHIOXC	food_pic	idli	Indian	Pragnan	25	5	Pragnan	4/23/2025, 10:58:22 AM	Delivered	Pragnan	09481628104	2025-04-23 Morning	Delivered
6PJFF15RQEXO25K1	food_pic	idli	Indian	Pragnan	25	5	Pragnan	4/25/2025, 10:15:31 AM	Pending	Pragnan	09481628104	Processing	Update Status

Frontend:

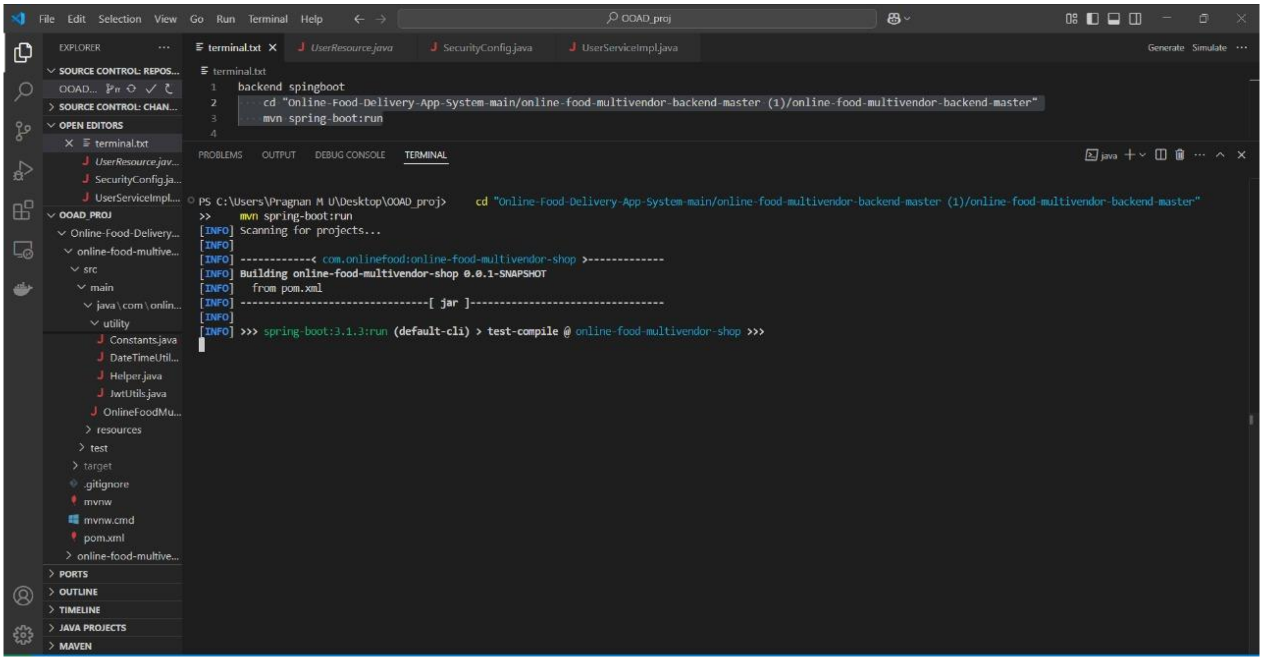


```
PS C:\Users\Pragnan M U\Desktop\OOAD_proj>     cd "Online-Food-Delivery-App-System-main/online-food-multivendor-frontend-master
 (1)/online-food-multivendor-frontend-master"
>>     npm install
>>     npm start
[##################] / reify:fsevents: sill reify mark deleted [
```



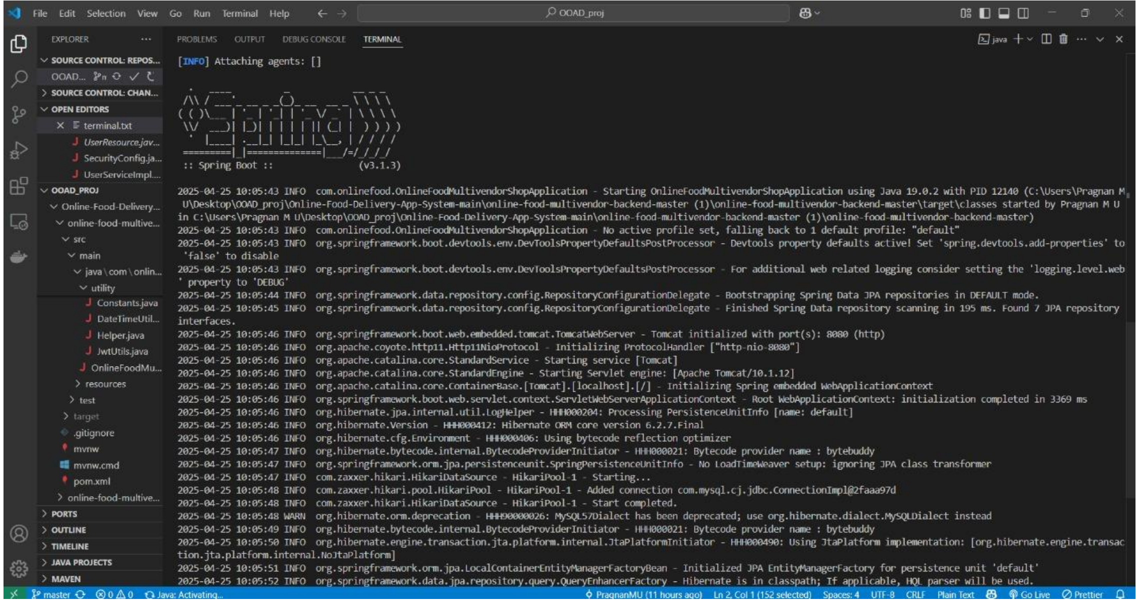
```
Compiled successfully!

You can now view ecommerce-frontend in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.60.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Backend Running:



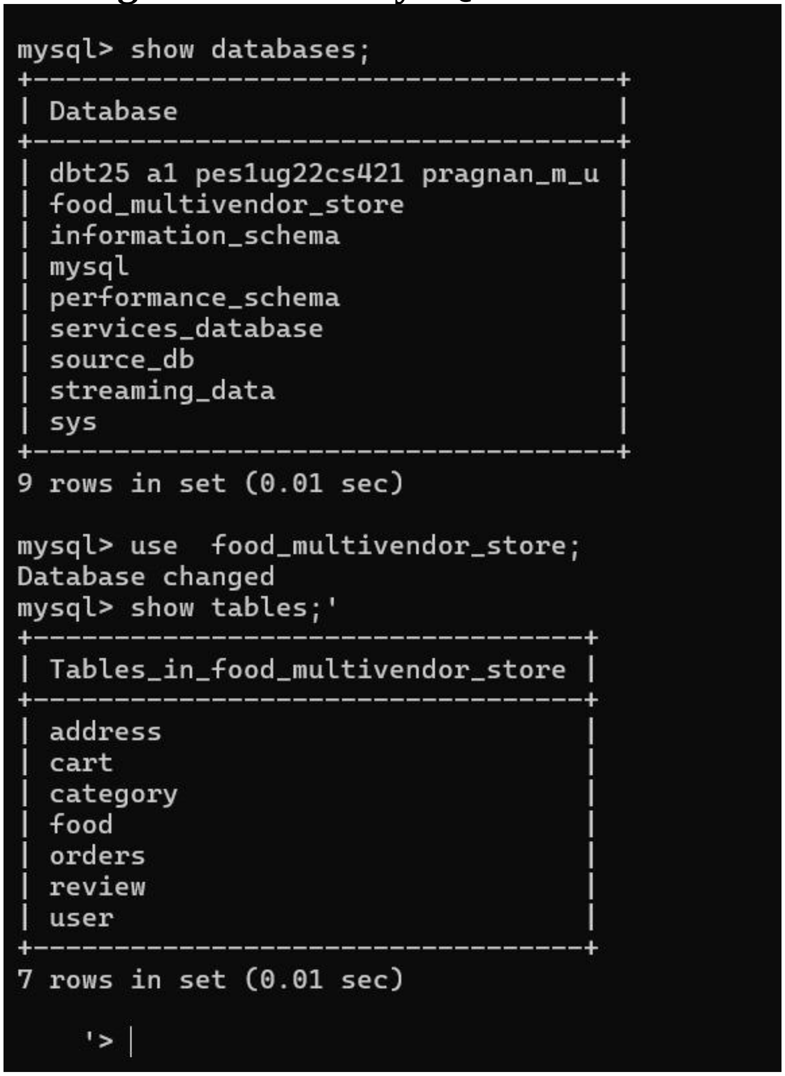
```
backend spingboot
    cd "Online-Food-Delivery-App-System-main/online-food-multivendor-backend-master (1)/online-food-multivendor-backend-master"
    mvn spring-boot:run
```

```
PS C:\Users\Pragnan M U\Desktop\OOAD_proj>     cd "Online-Food-Delivery-App-System-main/online-food-multivendor-backend-master (1)/online-food-multivendor-backend-master"
>>     mvn spring-boot:run
[INFO] Scanning for projects...
[INFO]
[INFO] ------------< com.onlinefood:online-food-multivendor-shop >-------------
[INFO] Building online-food-multivendor-shop 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] >>> spring-boot:3.1.3:run (default-cli) > test-compile @ online-food-multivendor-shop >>>
```

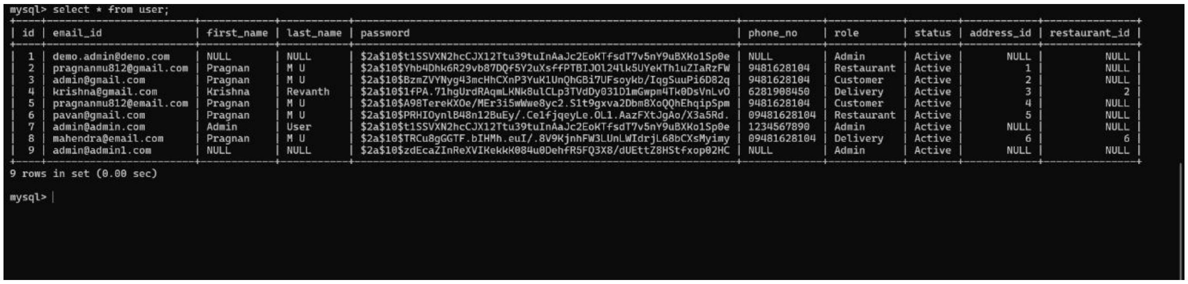


```
[INFO] Attaching agents: []

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v3.1.3)

2025-04-25 10:05:43 INFO  com.onlinefood.OnlineFoodMultivendorShopApplication - Starting OnlineFoodMultivendorShopApplication using Java 19.0.2 with PID 12140 (C:\Users\Pragnan M
 U\Desktop\OOAD_proj\Online-Food-Delivery-App-System-main\online-food-multivendor-backend-master (1)\online-food-multivendor-backend-master\target\classes started by Pragnan M U
in C:\Users\Pragnan M U\Desktop\OOAD_proj\Online-Food-Delivery-App-System-main\online-food-multivendor-backend-master (1)\online-food-multivendor-backend-master)
2025-04-25 10:05:43 INFO  com.onlinefood.OnlineFoodMultivendorShopApplication - No active profile set, falling back to 1 default profile: "default"
2025-04-25 10:05:43 INFO  org.springframework.boot.devtools.env.DevToolsPropertyDefaultsPostProcessor - Devtools property defaults active! Set 'spring.devtools.add-properties' to
 'false' to disable
2025-04-25 10:05:43 INFO  org.springframework.boot.devtools.env.DevToolsPropertyDefaultsPostProcessor - For additional web related logging consider setting the 'logging.level.web
' property to 'DEBUG'
2025-04-25 10:05:44 INFO  org.springframework.data.repository.config.RepositoryConfigurationDelegate - Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2025-04-25 10:05:45 INFO  org.springframework.data.repository.config.RepositoryConfigurationDelegate - Finished Spring Data repository scanning in 195 ms. Found 7 JPA repository
interfaces.
2025-04-25 10:05:46 INFO  org.springframework.boot.web.embedded.tomcat.TomcatWebServer - Tomcat initialized with port(s): 8080 (http)
2025-04-25 10:05:46 INFO  org.apache.coyote.http11.Http11NioProtocol - Initializing ProtocolHandler ["http-nio-8080"]
2025-04-25 10:05:46 INFO  org.apache.catalina.core.StandardService - Starting service [Tomcat]
2025-04-25 10:05:46 INFO  org.apache.catalina.core.StandardEngine - Starting Servlet engine: [Apache Tomcat/10.1.12]
2025-04-25 10:05:46 INFO  org.apache.catalina.core.ContainerBase.[Tomcat].[localhost].[/] - Initializing Spring embedded WebApplicationContext
2025-04-25 10:05:46 INFO  org.springframework.boot.web.servlet.context.ServletWebServerApplicationContext - Root WebApplicationContext: initialization completed in 3369 ms
2025-04-25 10:05:46 INFO  org.hibernate.jpa.internal.util.LogHelper - HHH000204: Processing PersistenceUnitInfo [name: default]
2025-04-25 10:05:46 INFO  org.hibernate.Version - HHH000412: Hibernate ORM core version 6.2.7.Final
2025-04-25 10:05:46 INFO  org.hibernate.cfg.Environment - HHH000406: Using bytecode reflection optimizer
2025-04-25 10:05:47 INFO  org.hibernate.bytecode.internal.BytecodeProviderInitiator - HHH000021: Bytecode provider name : bytebuddy
2025-04-25 10:05:47 INFO  org.springframework.orm.jpa.persistenceunit.SpringPersistenceUnitInfo - No LoadTimeWeaver setup: ignoring JPA class transformer
2025-04-25 10:05:47 INFO  com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Starting...
2025-04-25 10:05:48 INFO  com.zaxxer.hikari.pool.HikariPool - HikariPool-1 - Added connection com.mysql.cj.jdbc.ConnectionImpl@2faaa97d
2025-04-25 10:05:48 INFO  com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Start completed.
2025-04-25 10:05:48 WARN  org.hibernate.orm.deprecation - HHH90000026: MySQL57Dialect has been deprecated; use org.hibernate.dialect.MySQLDialect instead
2025-04-25 10:05:49 INFO  org.hibernate.bytecode.internal.BytecodeProviderInitiator - HHH000021: Bytecode provider name : bytebuddy
2025-04-25 10:05:50 INFO  org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator - HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transac
tion.jta.platform.internal.NoJtaPlatform]
2025-04-25 10:05:51 INFO  org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean - Initialized JPA EntityManagerFactory for persistence unit 'default'
2025-04-25 10:05:52 INFO  org.springframework.data.jpa.repository.query.QueryEnhancerFactory - Hibernate is in classpath; If applicable, HQL parser will be used.
```

Storing the data in MySQL:



```
mysql> show databases;
+-------------------------------------+
| Database                            |
+-------------------------------------+
| dbt25 a1 pes1ug22cs421 pragnan_m_u  |
| food_multivendor_store              |
| information_schema                  |
| mysql                               |
| performance_schema                  |
| services_database                   |
| source_db                           |
| streaming_data                      |
| sys                                 |
+-------------------------------------+
9 rows in set (0.01 sec)

mysql> use  food_multivendor_store;
Database changed
mysql> show tables;'
+-----------------------------------+
| Tables_in_food_multivendor_store |
+-----------------------------------+
| address                           |
| cart                              |
| category                          |
| food                              |
| orders                            |
| review                            |
| user                              |
+-----------------------------------+
7 rows in set (0.01 sec)

    '> |
```



```
mysql> select * from user;
+----+------------------------+------------+-----------+--------------------------------------------------------------+-------------+------------+--------+------------+---------------+
| id | email_id               | first_name | last_name | password                                                     | phone_no    | role       | status | address_id | restaurant_id |
+----+------------------------+------------+-----------+--------------------------------------------------------------+-------------+------------+--------+------------+---------------+
|  1 | demo.admin@demo.com    | NULL       | NULL      | $2a$10$t1SSVXN2hcCJX12Ttu39tuInAaJc2EoKTfsdT7v5nY9uBXKo1Sp0e | NULL        | Admin      | Active |       NULL |          NULL |
|  2 | pragnanmu812@gmail.com | Pragnan    | M U       | $2a$10$Yhb4Dhk6R29vb87DQf5Y2uXsffPTBIJOl24lk5UYeKTh1uZIaRzFW | 9481628104  | Restaurant | Active |          1 |          NULL |
|  3 | admin@gmail.com        | Pragnan    | M U       | $2a$10$BzmZVYNyg43mcHhCXnP3YuK1UnQhGBi7UFsoykb/IqgSuuPi6D82q | 9481628104  | Customer   | Active |          2 |          NULL |
|  4 | krishna@gmail.com      | Krishna    | Revanth   | $2a$10$1fPA.71hgUrdRAqmLKNk8ulCLp3TVdDy031D1mGwpm4Tk0DsVnLvO | 6281908450  | Delivery   | Active |          3 |             2 |
|  5 | pragnanmu812@email.com | Pragnan    | M U       | $2a$10$A98TereKXOe/MEr3i5wWwe8yc2.S1t9gxva2Dbm8XoQQhEhqipSpm | 9481628104  | Customer   | Active |          4 |          NULL |
|  6 | pavan@gmail.com        | Pragnan    | M U       | $2a$10$PRHIOynlB48n12BuEy/.Ce1fjqeyLe.OL1.AazFXtJgAo/X3a5Rd. | 09481628104 | Restaurant | Active |          5 |          NULL |
|  7 | admin@admin.com        | Admin      | User      | $2a$10$t1SSVXN2hcCJX12Ttu39tuInAaJc2EoKTfsdT7v5nY9uBXKo1Sp0e | 1234567890  | Admin      | Active |       NULL |          NULL |
|  8 | mahendra@email.com     | Pragnan    | M U       | $2a$10$TRCu8gGGTF.bIHMh.euI/.8V9KjnhFW3LUnLWIdrjL68bCXsMyimy | 09481628104 | Delivery   | Active |          6 |             6 |
|  9 | admin@admin1.com       | NULL       | NULL      | $2a$10$zdEcaZInReXVIKekkK084u0DehfR5FQ3X8/dUEttZ8HStfxop02HC | NULL        | Admin      | Active |       NULL |          NULL |
+----+------------------------+------------+-----------+--------------------------------------------------------------+-------------+------------+--------+------------+---------------+
9 rows in set (0.00 sec)

mysql> |
```

Individual contributions of the team members:

Name	**Module worked on**
Payyavula Mahendra Chowdary	Builder Pattern: Building response objects with method chaining in exception handling The Builder Design Pattern is another creational pattern designed to deal with the construction of comparatively complex objects.
Pragnan M U	Factory Pattern: The passwordEncoder() method in SecurityConfig.java creates and returns a BCryptPasswordEncoder PasswordEncoder interface is already defined in the Spring Security framework, and you just import and use it
Pavan Chandu C S	Singleton Pattern: Spring container ensures single instances of object (bean) like services and controllers
Pathipati Krishna Revanth	UserResource.java, FoodResource.java, OrderResource.java Simplifies interactions between controllers and multiple services