

CAPSTONE PROJECT

RESTAURANT MANAGEMENT SYSTEM



Project by,

S.GOPI KRISHNA

JAVA_FULLSTACK (FACEPREP)

43110270

B.E – CSE

1. INTRODUCTION

FoodFrenzy is a web-based food ordering and management system developed using Java, Spring Boot, Maven, Thymeleaf, HTML, CSS, and MySQL, designed to simplify and automate the process of food product management and ordering. The application allows users to register, log in, search for food products, place orders, and view their order history, while administrators can manage users, add, update, and delete products, control stock levels, and monitor all orders through an admin dashboard. By replacing manual and traditional ordering methods, FoodFrenzy reduces errors, improves efficiency, and enhances user experience through a structured and user-friendly interface. The system follows a layered architecture with controllers, services, repositories, and entities, ensuring better maintainability, scalability, and separation of concerns. Overall, FoodFrenzy demonstrates a practical implementation of full-stack web development concepts and provides an efficient solution for managing food ordering operations in a real-world scenario.

2. ABSTRACT

FoodFrenzy is a full-stack web application developed to provide an efficient and user-friendly platform for online food ordering and management. The system is built using Java, Spring Boot, Maven, Thymeleaf, HTML, CSS, and MySQL, following a layered architecture that ensures modularity, scalability, and ease of maintenance. The application supports two main roles: users and administrators. Users can search for available food products, place orders, and view their order history, while administrators can manage users, add and update products, maintain stock levels, and monitor all customer orders through a centralized admin panel. By automating product management and order processing, FoodFrenzy minimizes manual effort, reduces errors, and improves overall operational efficiency. This project demonstrates the practical application of modern web development technologies and showcases how a robust backend integrated with an interactive frontend can deliver a reliable food ordering solution.

3. OBJECTIVES OF THE PROJECT

- To develop an easy-to-use online food ordering system for users.
- To allow users to search products, place orders, and view order history.
- To provide an admin panel for managing users and products.

- To maintain and update product stock efficiently.
- To track and manage all customer orders in one place.
- To reduce manual work and improve accuracy using a Spring Boot-based system.

4. SCOPE

The scope of the FoodFrenzy project includes online product management, stock maintenance, and order processing through a web-based application. It supports user and admin roles to ensure secure access and efficient system control. The system can be further extended with online payments, delivery tracking, and mobile application support in the future.

5. TARGET USERS

Customers (Users)

Customers can register, log in, search for food products, place orders, view order history, and track their purchases easily through the application.

Administrators (Admins)

Administrators manage the entire system, including adding and updating products, maintaining stock levels, managing users, and monitoring all customer orders from the admin dashboard.

6.TECH STACK

Category	Technology Used	Purpose / Description
Programming Language	Java	Core backend logic and business processing
Backend Framework	Spring Boot	Application framework for building scalable Java applications
Web Framework	Spring MVC	Handles HTTP requests and responses using MVC pattern
ORM Framework	Hibernate / JPA	Manages database operations and object-relational mapping
Frontend Technologies	JSP, HTML, CSS	Used for designing and rendering user interface
Database	MySQL	Relational database for storing hospital data
Build Tool	Maven	Dependency management and project build automation
IDE	Spring Tool Suite (STS)	Development environment for Spring Boot applications
Server	Embedded Tomcat	Runs the Spring Boot web application

Table 1: Tech Stack

7. SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

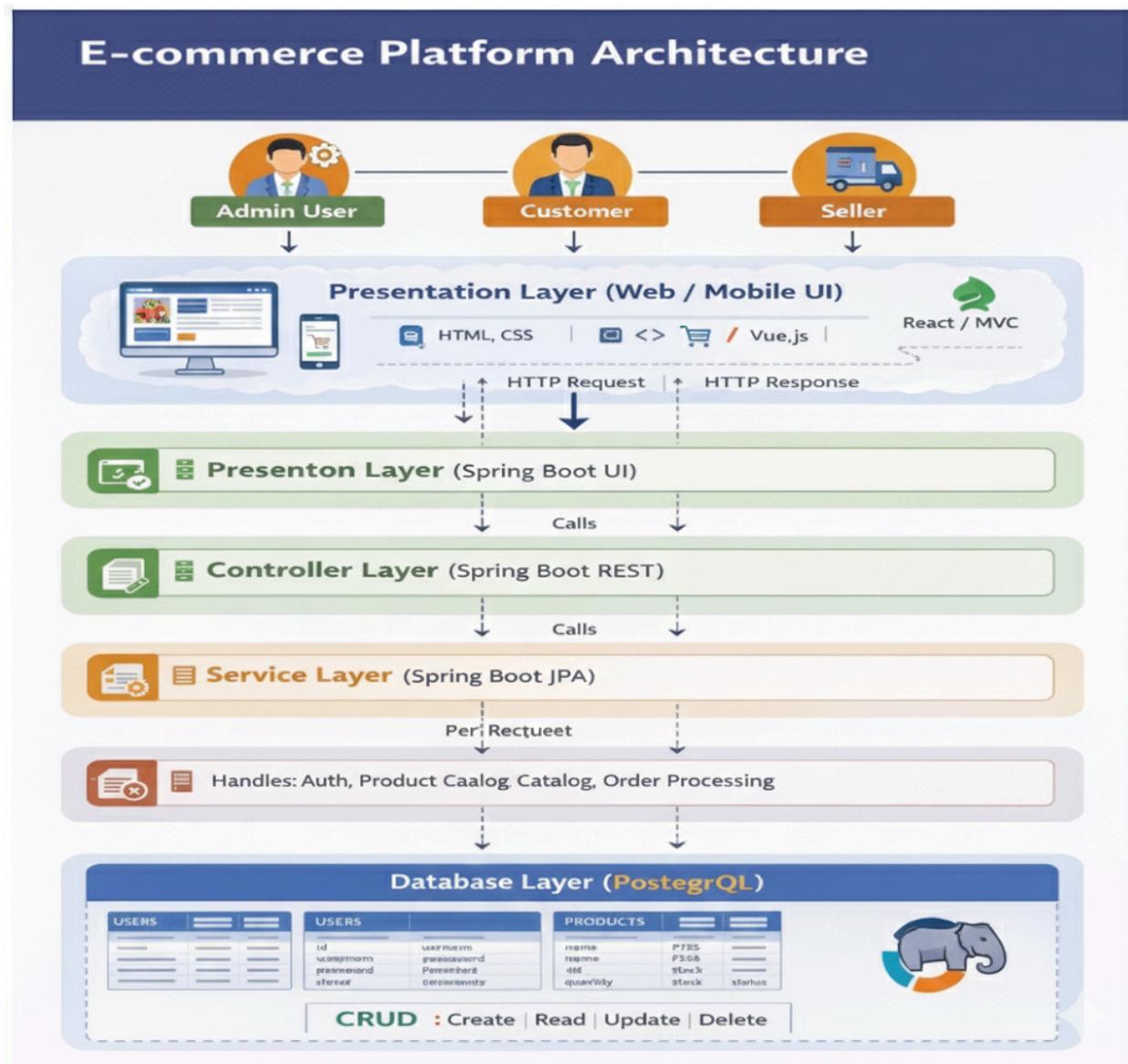
- **Processor:** Dual Core or higher
- **RAM:** Minimum 4 GB (Recommended 8 GB)
- **Hard Disk:** Minimum 5 GB free space
- **Display Resolution:** 1366 × 768 or higher
- **Internet Connection:** Required for development and deployment

SOFTWARE REQUIREMENTS

- **Operating System:** Windows 10 / Windows 11
- **Programming Language:** Java JDK 8 or above
- **Backend Framework:** Spring Boot 3.x
- **Web Framework:** Spring MVC (included with Spring Boot)
- **ORM Framework:** Hibernate / JPA (latest stable version)
- **Database:** MySQL Server 8.0 or above
- **IDE:** Spring Tool Suite (STS) 4.x
- **Build Tool:** Maven 3.6 or above
- **Application Server:** Embedded Apache Tomcat (comes with Spring Boot)
- **Frontend Technologies:** JSP, HTML5, CSS3
- **Web Browser:** Google Chrome (latest) / Microsoft Edge (latest)
- **API Testing Tool:** Postman (latest version)

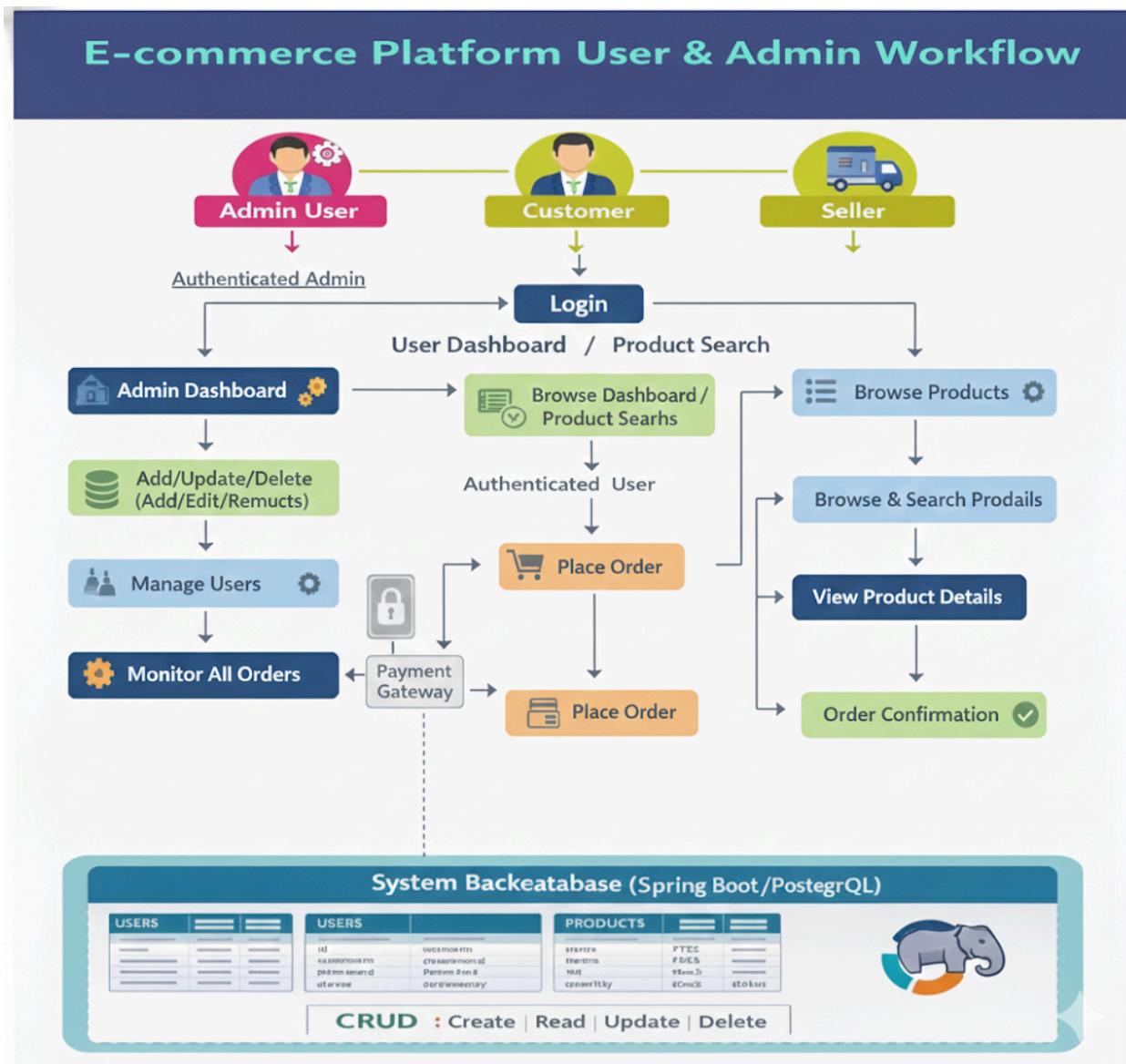
8.SYSTEM ARCHITECTURE

The FoodFrenzy system follows a layered architecture using Spring Boot. Users and administrators interact with the application through a web interface built with HTML, CSS, and Thymeleaf. Requests are processed by controllers, business logic is handled in the service layer, and data is managed using JPA repositories connected to a MySQL database. This architecture ensures better separation of concerns, scalability, and easy maintenance.



9.SYSTEM WORKFLOW

The user logs into the system and searches for available products. After selecting a product and quantity, the user places an order, and the system calculates and confirms the total amount. The order details are stored in the database and displayed in the user's order history. Administrators manage products, users, and monitor all orders through the admin dashboard.



10.DATABASE DESIGN

The database is designed using a relational model to efficiently store and manage application data. It consists of core tables such as **Admin**, **Members (Users)**, **Product**, and **Orders**, each with clearly defined primary keys to ensure data integrity. Relationships are established between users and orders, allowing each order to be associated with a specific user, while products are linked to orders to track purchases. This structured design supports fast data retrieval, secure storage, and smooth interaction between different modules of the system.

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar has sections for MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), and PERFORMANCE (Dashboard). The main area has tabs for Administration and Schemas, with Administration selected. A query editor window titled 'Query 1' contains the following SQL code:

```
1 • show databases;
2 • use foodfrenzy;
3 • show tables;
```

The results grid below shows the tables in the 'foodfrenzy' database:

Tables_in_foodfrenzy
admin
members
orders
orders_seq
product
...

The output pane shows the execution history:

Action Output	#	Time	Action	Message	Duration / Fetch
1	13:36:23	show tables	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...	0.000 sec	
2	13:36:27	show tables	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...	0.000 sec	
3	13:36:31	show tables	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...	0.000 sec	
4	13:36:51	show databases	26 row(s) returned	0.000 sec / 0.000 sec	
5	13:37:17	use foodfrenzy	0 row(s) affected	0.016 sec	
6	13:37:33	show tables	7 row(s) returned	0.016 sec / 0.000 sec	

11. HTTP REQUEST METHODS & API DOCUMENTATION

The Food Ordering Management System follows RESTful architecture principles and uses standard HTTP request methods to perform CRUD operations on system modules such as admins, users, products, and orders. Each endpoint is responsible for a specific operation and ensures smooth communication between the frontend and backend. Data is exchanged securely and efficiently, primarily using form data and server-rendered views.

11.1 API Mapping Table

Method	Endpoint	Description
POST	/admin-login	Authenticates admin and redirects to admin dashboard
POST	/user-login	Authenticates user and allows access to ordering features
POST	/user/register	Registers a new user into the system
GET	/admin/services	Displays admin dashboard with users, products, and orders
GET	/product/search	Loads product search page for users
POST	/product/search	Searches product by name
POST	/addingProduct	Adds a new product (Admin only)
GET	/editProduct/{id}	Opens edit page for selected product

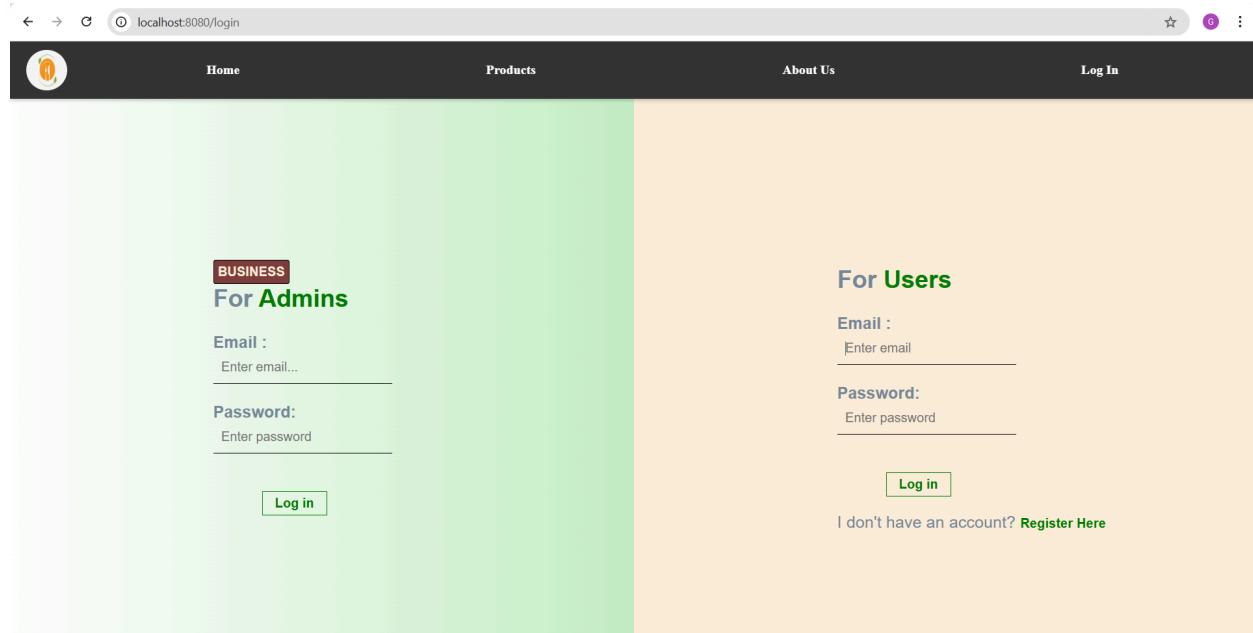
POST	/updatingProduct/{id}	Updates existing product details
GET	/deleteProduct/{id}	Deletes a product (Admin only)
POST	/user/product/order	Places an order for a selected product
GET	/order-success	Displays order success confirmation page
GET	/editUser/{id}	Opens edit page for a user (Admin only)
POST	/updatingUser/{id}	Updates user details
GET	/deleteUser/{id}	Deletes a user from the system

11.2 HTTP Methods Usage Summary

- **POST** – Used to create new records such as patients, doctors, and appointments
- **GET** – Used to retrieve data from the system
- **PUT** – Used to update appointment status and doctor availability
- **DELETE** – Used to remove records (optional / future scope)

12 PROJECT MODULES

12.1 Login Module

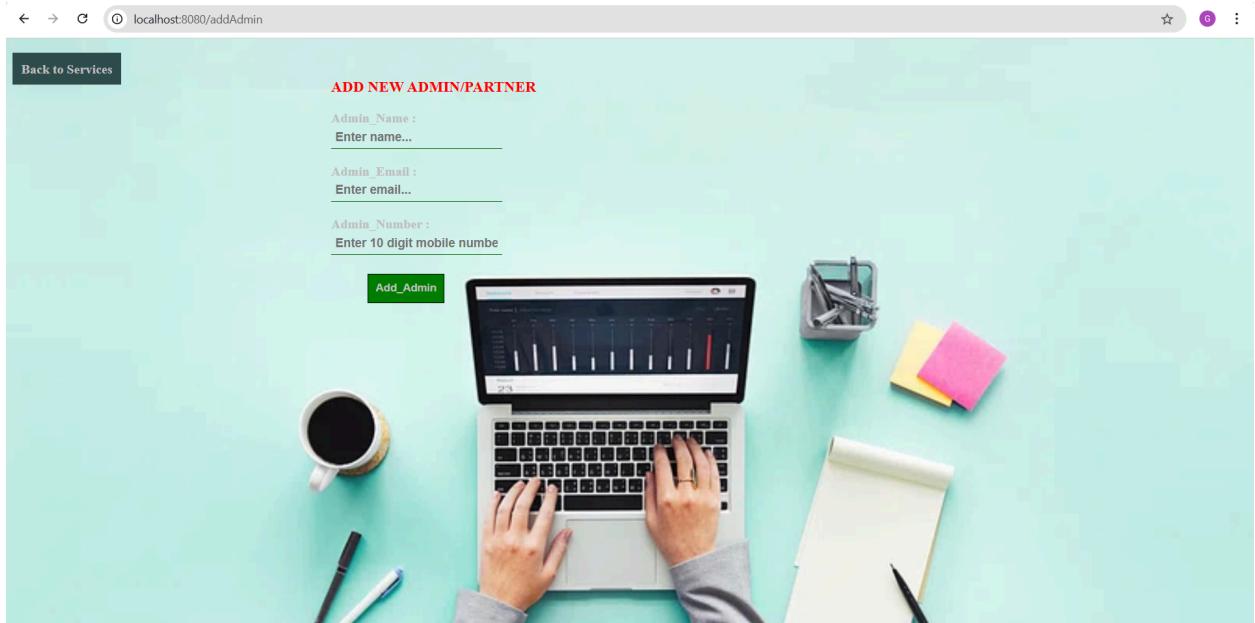


12.2 Admin Module:

The Admin Panel module acts as the central control unit of the FoodFrenzy system. It enables administrators to manage users, products, stock levels, and orders through a single dashboard. Admins can add, update, and delete product details, monitor inventory, view all customer orders, and manage user accounts. This module ensures smooth system administration, effective decision-making, and efficient overall operation of the application.

Services						
Back	Admin Id	Admin Name	Admin Email	Admin Number	Actions	
	1	Super Admin	admin@gmail.com	9778165514		
+ Add Admin						
Users:						
User Id	User Name	User Email	User Number	Actions		
1	gopi	gopi@gmail.com	9778165516			
2	krish	krish@gmail.com	9988776655			
5	user	user@gmail.com	9988776655			
+ Add User						
Products:						
Product Id	Product Name	Product Price	Stock	Actions		
1	Chicken Biryani	410.0	21			
2	Panner Butter	290.0	40			
3	Butter Chicken	380.0	35			
4	Chowmein	160.0	60			
5	Gulab Jamun	60.0	100			
6	Chicken Biryani	410.0	10			
7	butter	220.0	15			
+ Add Product						
Orders:						
User Name	User Email	Product Name	Price	Quantity	Total Amount	Date
gopi	gopi@gmail.com	Gulab Jamun	60.0	3	180.0	2025-12-18 15:11:32.198

ADD NEW ADMIN/PARTNER					
Back to Services					
Admin_Name :	<input type="text"/>	Enter name...			
Admin_Email :	<input type="text"/>	Enter email...			
Admin_Number :	<input type="text"/>	Enter 10 digit mobile number...			
Add_Admin					



localhost:8080/editAdmin/1

[Back to Services](#)

UPDATE EXISTING ADMIN / PARTNER DETAILS

Admin_Name :
Super Admin

Admin_Number :
8778165514

Admin_Email :
admin@gmail.com

Update_Admin

Pavan Tea And Beverage's
Taste Me Best....



localhost:8080/user/addUser

[Back to Services](#)

ADD NEW USER/CUSTOMER

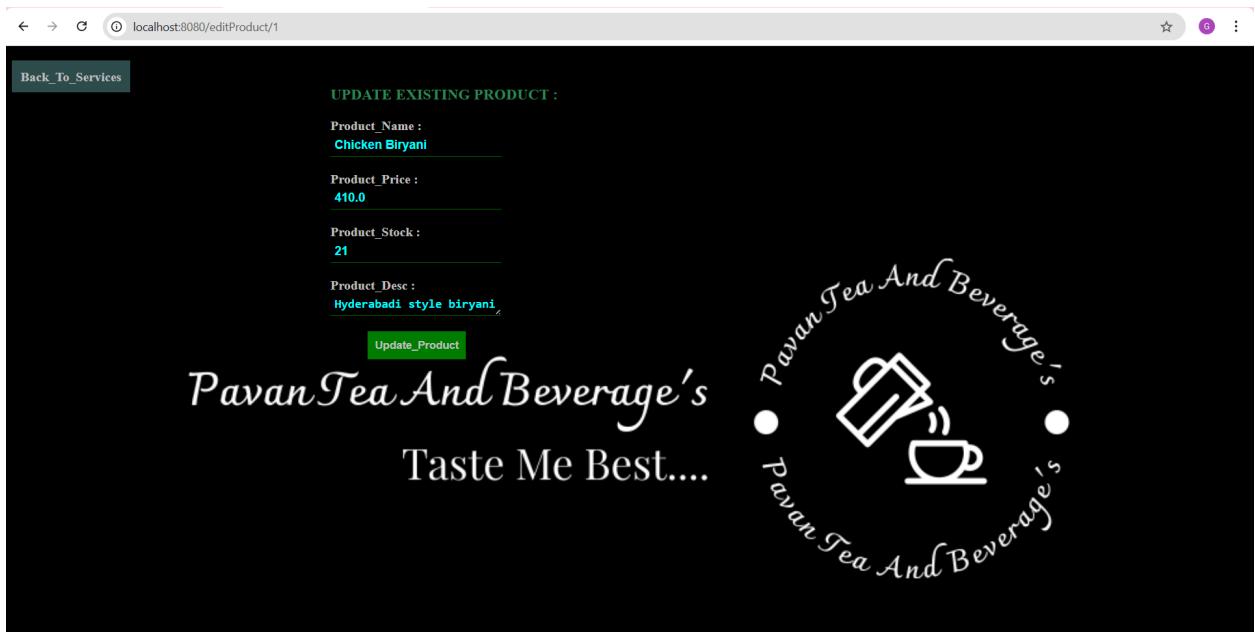
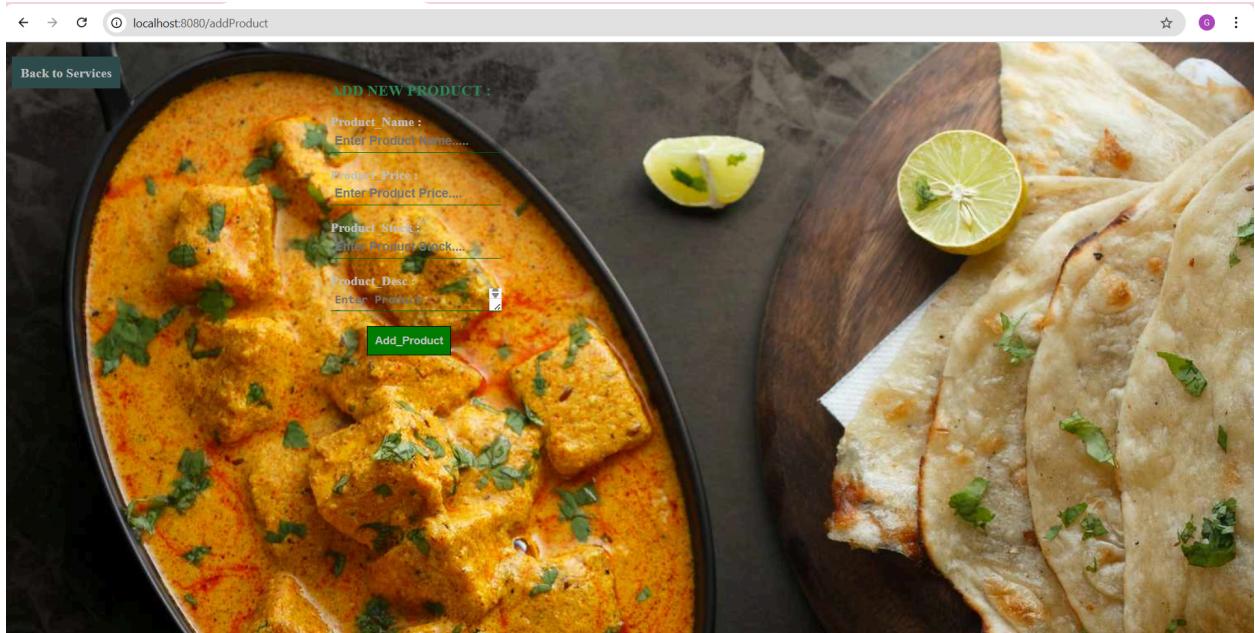
User_Name : _____

User_Email : _____

User_Number : _____

Add_User





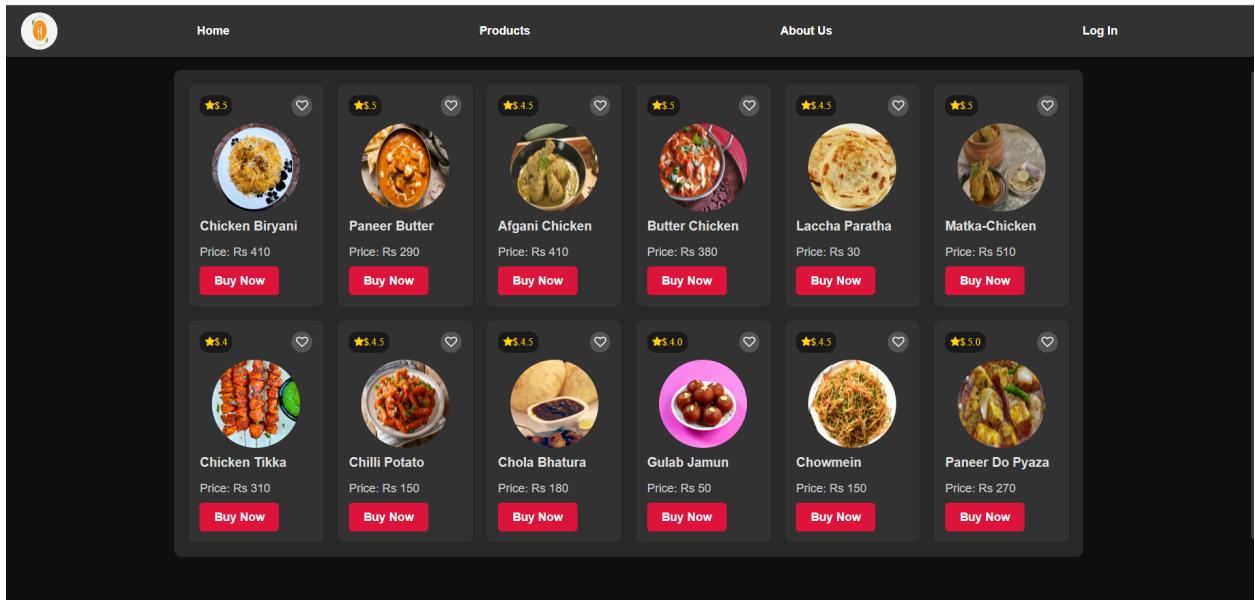
12.3 Register Module:

The User Registration module allows new users to create an account by providing essential details such as name, email, password, and contact number. The system validates user information to ensure secure and accurate registration. Once registered, users can log in to access features like product search, order placement, and order history, enabling a personalized and seamless experience within the FoodFrenzy platform.

The screenshot shows a web browser window with the URL `localhost:8080/register` in the address bar. The page has a dark header bar with a logo on the left, and navigation links for "Home", "Products", "About Us", and "Log In". On the right side of the header is a search icon and a notifications badge with the number "9". The main content area is a white box titled "User Registration". It contains four input fields: "Email", "Username", "Phone Number", and "Password", each with a corresponding placeholder text. Below these fields is a green "Register" button. The overall design is clean and modern.

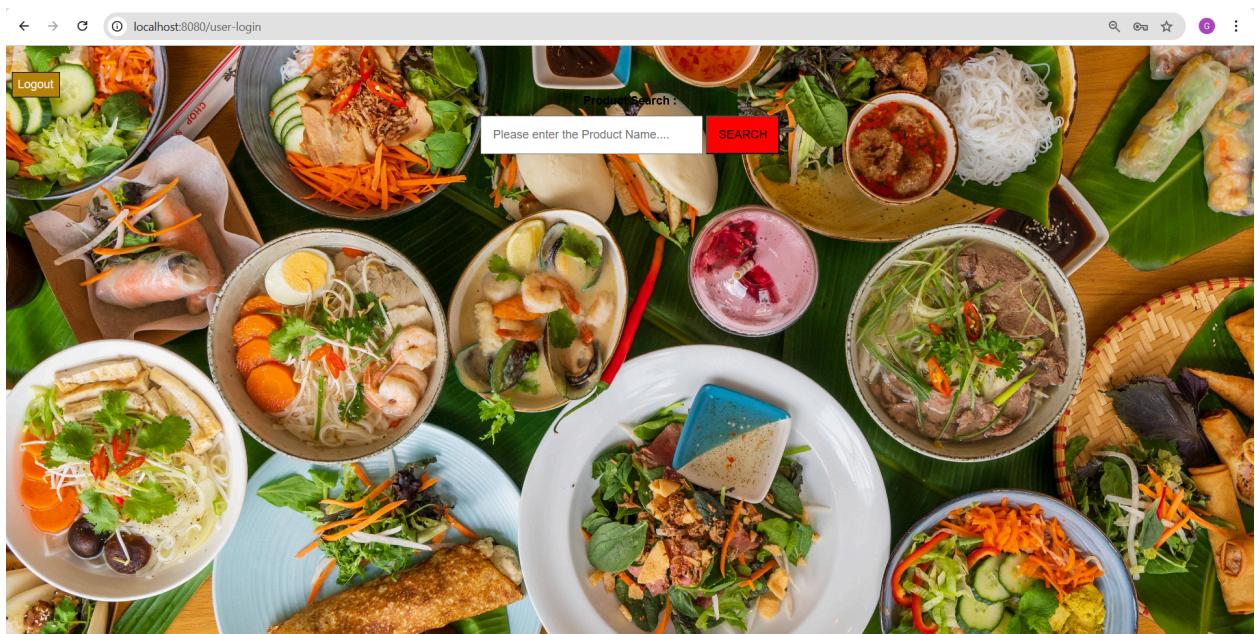
12.4 Product Module:

The Products module is responsible for managing all product-related information in the FoodFrenzy system. It allows administrators to add new products, update existing product details such as price, description, and stock, and remove products when required. This module ensures that accurate and up-to-date product information is available to users, supporting effective inventory management and a seamless ordering experience.



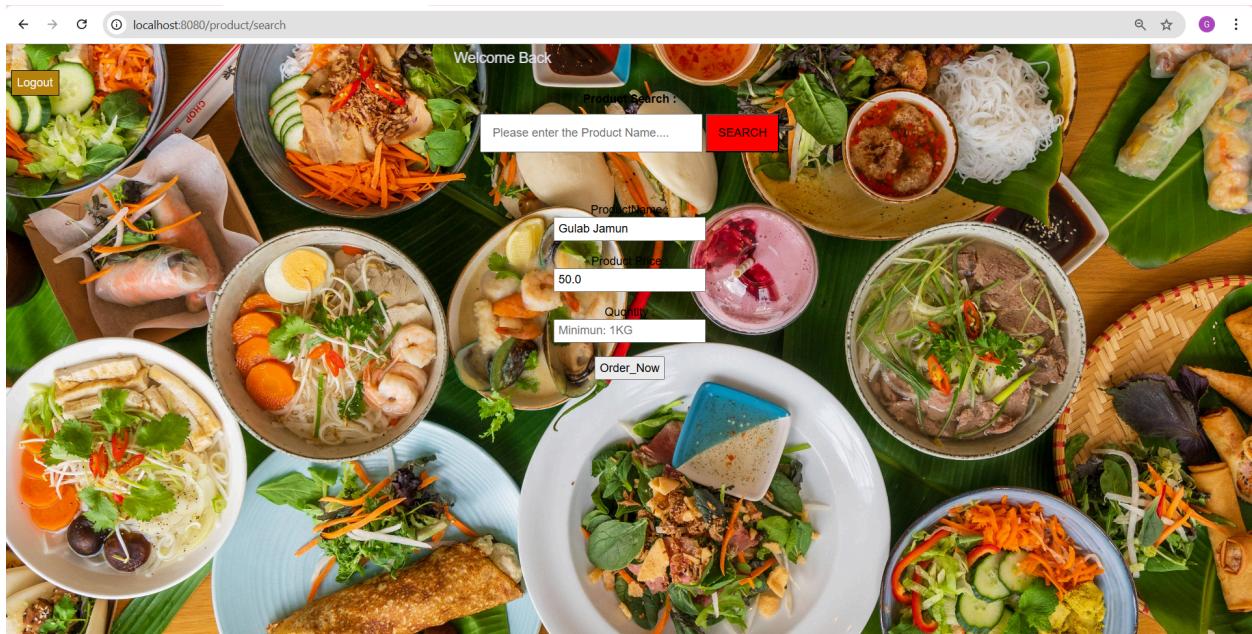
12.5 User Home & Buy Product

The User Home and Buy Product module provides a simple and interactive interface where users can search for available products, view product details, and place orders with ease. After logging in, users can browse products, select the desired quantity, and proceed to purchase. This module ensures a smooth shopping experience by displaying order confirmations and maintaining the user's order history for future reference.



12.6 Search And Order Product Module:

The Search and Order Product module allows users to easily find food products by entering the product name and viewing relevant details such as price and availability. Once a product is selected, users can choose the required quantity and place an order instantly. The system automatically calculates the total amount and confirms the order, providing a smooth and user-friendly purchasing experience.



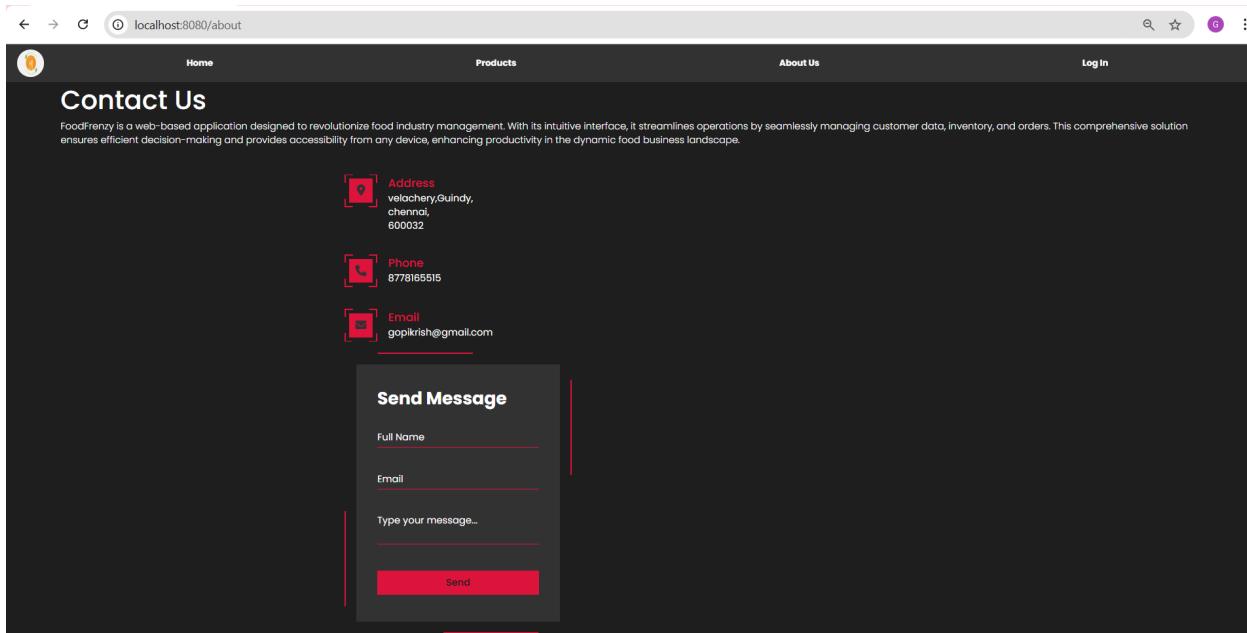
12.7 Order Success Module:

Your purchase has been successfully placed and is now being processed. We have received your order details, and our team will begin preparing it shortly. You can view your order information and total amount on this page, and we'll ensure a smooth and timely service. Thank you for choosing **FoodFrenzy**—we hope you enjoy your meal!



12.8 About Module:

FoodFrenzy is a user-friendly online food ordering and management platform designed to simplify the process of browsing products, managing orders, and handling inventory efficiently. Our goal is to bridge the gap between customers and food service providers by offering a reliable, secure, and easy-to-use system. With a focus on performance and scalability, FoodFrenzy empowers administrators to manage products, stock, and users effectively, while customers enjoy a smooth and convenient ordering experience.



13.DATABASE TABLES

13.1 Users Table:

The screenshot shows the MySQL Workbench interface with the 'user' table selected. The 'Query 1' tab contains the following SQL code:

```
1 • show databases;
2 • use foodfrenzy;
3 • show tables;
4 • select * from user;
```

The 'Result Grid' shows the following data:

u_id	unumber	uemail	uname	upassword
1	8778165516	gopi@gmail.com	gopi	-----
2	9968776655	krish@gmail.com	krish	\$2a\$10\$87edkHAn.X9U53mcloSUO400hsg9x...
5	9968776655	user@gmail.com	user	\$2a\$10\$8eeVKA3Xvhf6R.h.6T0vOkzsVU5p...

The 'Output' pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
2	13:36:27	show tables	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SC...	0.000 sec
3	13:36:31	show tables	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SC...	0.000 sec
4	13:36:51	show databases	26 row(s) returned	0.000 sec / 0.000 sec
5	13:37:17	use foodfrenzy	0 row(s) affected	0.016 sec
6	13:37:33	show tables	7 row(s) returned	0.016 sec / 0.000 sec
7	14:11:09	select * from user LIMIT 0, 1000	3 row(s) returned	0.015 sec / 0.000 sec

13.2 Orders Table:

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: MANAGEMENT INSTANCE PERFORMANCE Administration Schemas

Query 1:

```

1 • show databases;
2 • use Foodfrenzy;
3 • show tables;
4 • select * from user;
5 • select * from orders;
6
7
8
9
10

```

Result Grid:

o_id	o_price	o_quantity	total_amout	u_id	order_date	o_name	user_u_id
602	50	3	150	1	2025-12-18 13:11:32.198000	Gulab Jamun	NULL
652	410	4	1640	1	2025-12-18 14:16:40.711000	Chicken Biryani	NULL
653	410	1	410	1	2025-12-18 14:21:45.722000	Chicken Biryani	NULL
654	290	1	290	1	2025-12-18 14:26:19.086000	Paneer Butter	NULL
702	410	2	820	1	2025-12-18 14:48:17.047000	Chicken Biryani	NULL

orders 5 x

Action Output:

#	Time	Action	Message	Duration / Fetch
4	13:35:51	show databases	26 row(s) returned	0.000 sec / 0.000 sec
5	13:37:17	use foodfrenzy	0 row(s) affected	0.016 sec
6	13:37:33	show tables	7 row(s) returned	0.016 sec / 0.000 sec
7	14:11:05	select * from user LIMIT 0, 1000	3 row(s) returned	0.015 sec / 0.000 sec
8	14:11:57	show tables	7 row(s) returned	0.000 sec / 0.000 sec
9	14:12:04	select * from orders LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

13.3 Admin Table:

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: MANAGEMENT INSTANCE PERFORMANCE Administration Schemas

Query 1:

```

1 • show databases;
2 • use Foodfrenzy;
3 • show tables;
4 • select * from user;
5 • select * from orders;
6 • select * from admin;
7
8
9
10
11

```

Result Grid:

admin_id	admin_email	admin_name	admin_number	admin_password
1	admin@gmail.com	Super Admin	8778165514	admin123
NULL	NULL	NULL	NULL	NULL

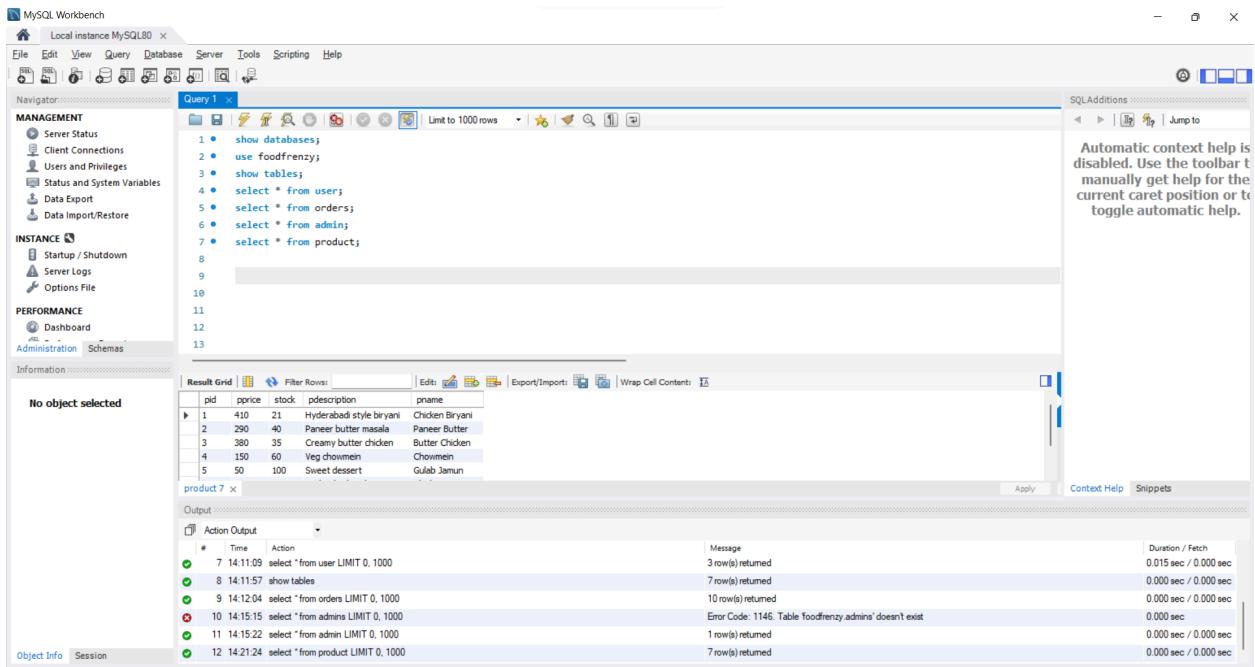
admin 6 x

Action Output:

#	Time	Action	Message	Duration / Fetch
6	13:37:33	show tables	7 row(s) returned	0.016 sec / 0.000 sec
7	14:11:05	select * from user LIMIT 0, 1000	3 row(s) returned	0.015 sec / 0.000 sec
8	14:11:57	show tables	7 row(s) returned	0.000 sec / 0.000 sec
9	14:12:04	select * from orders LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
10	14:15:15	select * from admin LIMIT 0, 1000	Error Code: 1146. Table 'Foodfrenzy.admins' doesn't exist	0.000 sec
11	14:15:22	select * from admin LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

13.3 Product Table:



14. TESTING

14.1 Spring Boot Testing:

Spring Boot testing plays a crucial role in ensuring the reliability and correctness of the FoodFrenzy application. The project uses Spring Boot's testing framework to validate different layers of the application, including controllers, services, and repositories. Unit tests are written to verify business logic such as product management, user authentication, stock handling, and order processing. Integration tests help ensure smooth interaction between the application and the database using an in-memory database where required. By using annotations like `@SpringBootTest`, `@WebMvcTest`, and `@DataJpaTest`, the application achieves better code quality, reduces runtime errors, and ensures stable performance before deployment.

15. CONCLUSION AND FUTURE ENHANCEMENTS

15.1 CONCLUSION:

The FoodFrenzy project successfully demonstrates the development of a full-stack web application using Java, Spring Boot, Thymeleaf, and MySQL to manage an end-to-end food ordering system. The application provides separate functionalities for administrators and users, enabling efficient product management, stock handling, user management, and order processing. By following a layered architecture and RESTful principles, the system ensures better scalability, maintainability, and security. Overall, this project enhances practical understanding of real-world web application development and serves as a strong foundation for future enhancements such as online payments, order tracking, and advanced analytics.

15.2 FUTURE ENHANCEMENTS:

Online Payment Integration – Integrate secure payment gateways like UPI, cards, or wallets to enable complete cashless transactions.

Order Tracking System – Allow users to track order status in real time (placed, preparing, dispatched, delivered).

Inventory Alerts – Automatically notify admins when product stock reaches a low level.

User Reviews & Ratings – Enable customers to rate and review products to improve service quality.

Role-Based Security – Enhance authentication and authorization using Spring Security and JWT.

Mobile Application Support – Develop an Android/iOS app to make the system more accessible and user-friendly.

Project Video Drive Link:

<https://drive.google.com/drive/folders/1eyMo3-ciOmDesIR957TGhx1srwQ2Rs-O>