

Online Food Ordering System

A MINI-PROJECT REPORT

Submitted by

ROHITH KUMAR S 240701436

ADITYA S 240701020

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING



RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project **“Online Food Ordering System”** is the Bonafide work of

“ROHITH KUMAR S, ADITYA S” who carried out the project work under my supervision.

SIGNATURE

Dr. S.SATHIYAVATHI

ASSISTANT PROFESSOR SG

Dept. of Computer Science and Eng,

Rajalakshmi Engineering College

Chennai

This mini project report is submitted for the viva voce examination to be held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

This project presents the design and development of a fully functional **Online Food Ordering and Delivery Management System** built using **Java Swing** for an interactive graphical user interface. The persistence layer leverages **Java Database Connectivity (JDBC)** to manage data within a **MySQL** database server ensuring seamless real-time synchronization between users, restaurants and delivery tracking simulation. A key architectural highlight is the implementation of **Data Access Objects(DAO)** such as User DAO and Order DAO which encapsulate all database interaction logic-including robust, transaction-safe handling of order , payments and delivery-status. This system effectively demonstrates essential food delivery functionalities, including **user authentication, restaurant, menu management, order placement, calorie checks, automated status updates from “Pending” to “Delivered”**. This structure provides both modularity and reliability aligning with real-world food delivery application standards.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M. THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head of The Department **Dr. E.M. MALATHY** and our Deputy Head of The Department **Dr. J. MANORANJINI** for being ever supporting force during our project work

We also extend our sincere and hearty thanks to our internal guide **Dr. S.SATHIYAVATHI** , for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

1. ROHITH KUMAR S

2. ADITYA S

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	3
1	INTRODUCTION	7
1.1	INTRODUCTION	
1.2	SCOPE OF THE WORK	
1.3	PROBLEM STATEMENT	
1.4	AIM AND OBJECTIVES OF THE PROJECT	
2	SYSTEM SPECIFICATIONS	9
2.1	HARDWARE SPECIFICATIONS	
2.2	SOFTWARE SPECIFICATIONS	
3	MODULE DESCRIPTION	10
4	CODING	12
5	OUTPUT	14
6	CONCLUSION AND FUTUR ENHANCEMENT	18
7	REFERENCES	19

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
5.1	WELCOME FRAME	14
5.2	CUSTOMER LOGIN	14
5.3	ADMIN LOGIN	15
5.4	PRODUCT FRAME	15
5.5	ADMIN FRAME	16
5.6	CART FRAME	16
5.7	CHECKOUT FRAME	17

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

With the increasing digital transformation of the food industry, managing online food orders efficiently has become essential. This project involves designing and implementing a **Food Ordering and Delivery Management System** that uses **MySQL** as the core **Database Management System**, integrated through JDBC. The system stores and manages all essential data, including customer details, restaurant menus, orders, and delivery status. Through a Java Swing-based front-end, customers can browse restaurants, select dishes, place orders, and track their delivery updates. On the administrative side, the system provides database-driven functionalities to handle incoming orders, update order status, and manage delivery operations. The objective is to develop a reliable and well-structured database application that ensures seamless data flow between customers, restaurants, and delivery services.

1.2 SCOPE OF THE WORK

The scope of this project includes the design and development of a fully functional **desktop-based food ordering and delivery application** using **Java and MySQL**. The primary deliverables include:

1. **Customer Module:**
Provides an intuitive interface for customers to register, log in, explore available restaurants, browse their fixed menus, place orders, and track order status until delivery.
2. **Administrator Module:**
Allows administrators to manage restaurant details, oversee incoming orders, update delivery status, and ensure smooth processing of all customer requests.
3. **Data Persistence:**
Utilizes **JDBC connectivity with MySQL** to maintain reliable storage and retrieval of user, restaurant, and order information across sessions.
4. **Order and Delivery Management:**
Automatically updates order status through defined stages — from *Placed* to *Accepted*

to *Delivered* — including a visual **delivery tracking feature** that simulates the movement of the delivery agent until the order reaches the customer.

1.3 PROBLEM STATEMENT

The main challenge is to design and develop a **desktop-based food ordering and delivery management system** that ensures **real-time order tracking**, **data consistency**, and **modular separation** between user functionalities. The system must:

- Manage customer, restaurant, and order details in a **persistent MySQL database**.
- Maintain **synchronized and accurate order status updates**, from placement to delivery.
- Allow customers to **browse restaurants and view available dishes** seamlessly.
- Provide **role-specific interfaces** — customers for ordering and tracking, and administrators/restaurants for managing orders and updating delivery status.
- Ensure that the delivery process automatically reflects changes in order status (e.g., from *Accepted* to *Delivered*) in real time.

1.4 AIM AND OBJECTIVES OF THE PROJECT

Aim:

To develop a reliable and user-friendly **Food Ordering and Delivery Management System** using **Java Swing** and **JDBC with MySQL** for seamless order handling and tracking.

Objectives:

1. To design a responsive interface for customers and restaurants to manage orders and deliveries efficiently.

2. To implement a robust database layer for storing and updating order, user, and delivery details securely.
3. To enable real-time order status updates and with calorie limit based within the application.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Component	Detailed specification
Processor	Intel Core I5 (2.0GHZ or Higher)
RAM	15.7 GB DDR4
Storage	Minimum 200MB free disk space
Network Connection	Not required (uses local MySQL Server)

2.2 SOFTWARE SPECIFICATIONS (Updated for MySQL)

Component	Detailed specification
Operating System	Windows 11
Programming language	JAVA
GUI library	JAVA Swing
Database management System	MYSQL Server
Database Connector driver	JDBC
Database utility Tool	MYSQL command-line Client

Integrated Development Environment (IDE)	Eclipse
---	---------

CHAPTER 3

MODULE DESCRIPTION

The application is structured into four primary modules: Data Model, Data Access Layer (DAO), Database Utility, and User Interfaces (UI).

3.1 DATA MODEL MODULE

The Data Model defines the main entities stored in the MySQL database and accessed through JDBC.

It includes four core tables:

- **Users:** Stores customer login and profile details.
- **Restaurants:** Contains restaurant information such as name and cuisine.
- **Orders:** Records customer orders with status (Pending, Accepted, Delivered) and total amount.
- **Order_Items:** Holds individual dish details linked to each order.

These tables are relationally linked using foreign keys to maintain data consistency across the application.

3.2 DATA ACCESS LAYER (DAO) MODULE

User.java module handles all database interactions using JDBC. It includes methods for inserting, updating, retrieving, and deleting records. Main operations:

- Manage user authentication and registration.
- Fetch restaurant details and menu items.
- Handle order placement, status updates, and order history retrieval.
- Maintain data integrity between orders and order_items tables.

3.3 DATABASE UTILITY MODULE

DBConnection.java module manages database connectivity using JDBC. It provides a single reusable connection to the MySQL database. Key functions:

- Establishes and maintains the connection using MySQL JDBC driver.
- Ensures secure and efficient query execution.
- Handles exceptions and closes resources properly to prevent leaks.

3.4 USER INTERFACE(UI) MODULE

The UI Module is built using **Java Swing** and comprises three main frames that manage the user experience and interaction.

1.mainpg.java: Acts as the entry point. Handles user login, displays available restaurants, and shows previous orders with their statuses.

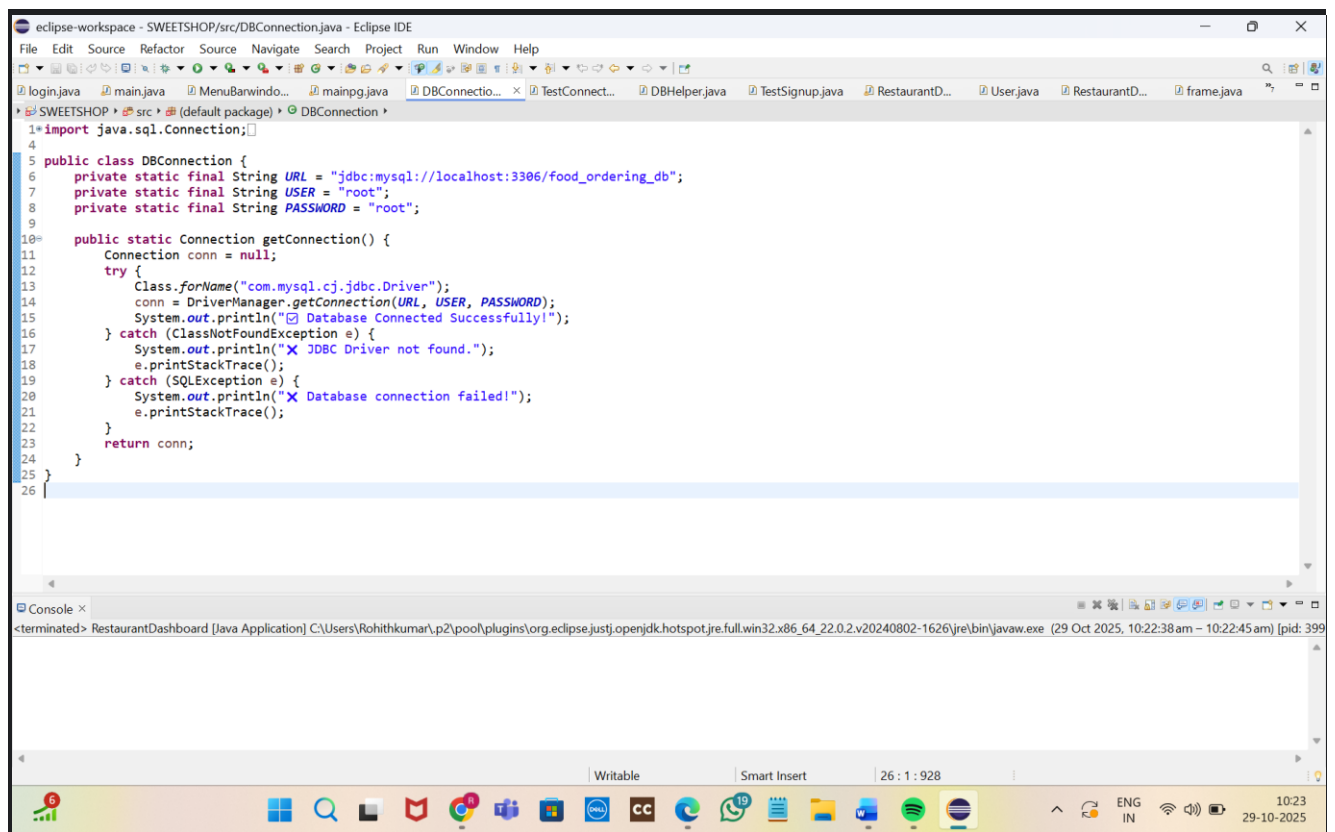
2.menubarwindow.java: Displays the restaurant's menu with categories and prices, allowing users to add items to the cart and proceed to checkout.

3.restaurantdashboard.java: Used by restaurants to view incoming orders, manage order status (e.g., accept, complete), and track active deliveries.

CHAPTER 4

SAMPLE CODING

DBCONNECTION CODE:



The screenshot shows the Eclipse IDE with the following components:

- Editor:** Displays the `DBConnection.java` file with the following code:


```

1 import java.sql.Connection;
2
3
4
5 public class DBConnection {
6     private static final String URL = "jdbc:mysql://localhost:3306/food_ordering_db";
7     private static final String USER = "root";
8     private static final String PASSWORD = "root";
9
10    public static Connection getConnection() {
11        Connection conn = null;
12        try {
13            Class.forName("com.mysql.cj.jdbc.Driver");
14            conn = DriverManager.getConnection(URL, USER, PASSWORD);
15            System.out.println("Database Connected Successfully!");
16        } catch (ClassNotFoundException e) {
17            System.out.println("X JDBC Driver not found.");
18            e.printStackTrace();
19        } catch (SQLException e) {
20            System.out.println("X Database connection failed!");
21            e.printStackTrace();
22        }
23        return conn;
24    }
25 }
26

```
- Console:** Shows a terminated message for the `RestaurantDashboard` application:


```

<terminated> RestaurantDashboard [Java Application] C:\Users\Rohithkumar\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1626\jre\bin\javaw.exe (29 Oct 2025, 10:22:38 am - 10:22:45 am) [pid: 399]

```
- Taskbar:** Displays the Windows taskbar with various application icons and the system clock showing 10:23 on 29-10-2025.

WELCOMEFRAME CODE:

```
eclipse-workspace - SWEETSHOP/src/mainpg.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
login.java main.java MenuBarWindow... mainpg.java DBConnectio... TestConnect... DBHelper.java TestSignup.java RestaurantD... User.java RestaurantD... frame.java
SWEETSHOP > src > (default package) > DeliveryTrackerWindow > startMovement() : void
import java.awt.*;

11
12 public class mainpg extends JFrame {
13     private JPanel mainPanel;
14     private JPanel ordersPanel;
15
16     private static final String DB_URL = "jdbc:mysql://localhost:3306/food_ordering_db";
17     private static final String DB_USER = "root";
18     private static final String DB_PASS = "root";
19     private CardLayout cardLayout;
20     private boolean isLoggedIn = false;
21     private String currentUser = "", currentPass = "";
22     private int currentCaloriesLeft = 0;
23     private String name;
24
25     public mainpg() {
26         setTitle("Ziggy Food Center - Online Food Booking");
27         this.setDefaultCloseOperation(JFrame.MAXIMIZED_BOTH);
28         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
29         setLayout(new BorderLayout());
30         cardLayout = new CardLayout();
31         mainPanel = new JPanel(cardLayout);
32
33         JPanel navbar = createNavbar();
34         mainPanel.add(createHomePage(), "Home");
35         mainPanel.add(createRestaurantsPage(), "Restaurants");
36         mainPanel.add(createAboutPage(), "About");
37         mainPanel.add(createContactPage(), "Contact");
38
39         ordersPanel = new JPanel();
40         ordersPanel.setBackground(new Color(255, 248, 248));
41         ordersPanel.setLayout(null);
42         JScrollPane scrollOrders = new JScrollPane(ordersPanel);
43         scrollOrders.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
44         scrollOrders.getVerticalScrollBar().setUnitIncrement(16);
45         mainPanel.add(scrollOrders, "Orders");
46
47         add(navbar, BorderLayout.NORTH);
48         add(mainPanel, BorderLayout.CENTER);
49
50         setVisible(true);
51     }
52
53     private JPanel createNavbar() {
54         JPanel navbar = new JPanel();
55         navbar.setBackground(new Color(228, 188, 58));
56         navbar.setPreferredSize(new Dimension(900, 60));
57         navbar.setLayout(new FlowLayout(FlowLayout.LEFT, 20, 15));
58
59         ImageIcon img = new ImageIcon("ziggy.png");
60         Image rrr = img.getScaledInstance(50, 30, Image.SCALE_SMOOTH);
61         ImageIcon scaledIcon = new ImageIcon(rrr);
62         JLabel logo = new JLabel("Ziggy Food Center", scaledIcon, JLabel.LEFT);
63         logo.setFont(new Font("Arial", Font.BOLD, 24));
64         logo.setBackground(new Color(40, 40, 40));
65         logo.setOpaque(true);
66         logo.setMargin(new Insets(5, 10, 5, 10));
67         navbar.add(logo);
68
69         navbar.add(Box.createHorizontalStrut(100));
70
71         String[] navItems = {"Home", "Restaurants", "About", "Contact"};
72         for (String item : navItems) {
73             JButton btn = new JButton(item);
74             btn.setFont(new Font("Arial", Font.PLAIN, 16));
75             btn.setBackground(new Color(40, 40, 40));
76             btn.setForeground(new Color(255, 255, 255));
77             btn.setBorderPainted(false);
78             btn.setFocusable(true);
79             btn.setCursor(new Cursor(Cursor.HAND_CURSOR));
80             btn.addActionListener(e -> cardLayout.show(mainPanel, item));
81             navbar.add(btn);
82         }
83
84         JButton loginBtn = new JButton(isLoggedIn ? "Hi, " + currentUser : "Login/Signup");
85         loginBtn.setFont(new Font("Arial", Font.PLAIN, 16));
86         loginBtn.setBackground(new Color(40, 40, 40));
87         loginBtn.setForeground(new Color(255, 255, 255));
88         loginBtn.setBorderPainted(false);
89         loginBtn.setCursor(new Cursor(Cursor.HAND_CURSOR));
90         loginBtn.addActionListener(e -> {
91             if (isLoggedIn) {
92                 showLoginPage();
93             } else {
94                 showLoginDialog();
95             }
96         });
97         navbar.add(loginBtn);
98
99         JButton ordersBtn = new JButton("My Orders");
100         ordersBtn.setFont(new Font("Arial", Font.PLAIN, 16));
101         ordersBtn.setBackground(new Color(40, 40, 40));
102         ordersBtn.setForeground(new Color(255, 255, 255));
103         ordersBtn.setBorderPainted(false);
104         ordersBtn.setCursor(new Cursor(Cursor.HAND_CURSOR));
105         ordersBtn.addActionListener(e -> {
106             populateUserOrders(currentUser);
107             cardLayout.show(mainPanel, "Orders");
108         });
109         navbar.add(ordersBtn);
110
111         return navbar;
112     }
113
114     private JPanel createHomePage() {
115         JPanel panel = new JPanel();
116         panel.setBackground(new Color(255, 248, 248));
117         panel.setLayout(new BorderLayout());
118
119         JPanel heroPanel = new JPanel();
120         heroPanel.setBackground(new Color(255, 235, 215));
121         heroPanel.setLayout(new BorderLayout(new BorderLayout(), BorderLayout.VERTICAL_1));
122         heroPanel.setBorder(BorderFactory.createEmptyBorder(40, 50, 50, 50));
123
124         JLabel title = new JLabel("Welcome to Ziggy Food Center");
125         title.setFont(new Font("Arial", Font.BOLD, 36));
126         title.setBackground(new Color(255, 255, 255));
127         title.setAlignmentX(Component.CENTER_ALIGNMENT);
128
129         JLabel subtitle = new JLabel("Your favorite meals, delivered to your doorstep");
130         subtitle.setFont(new Font("Arial", Font.PLAIN, 20));
131         subtitle.setBackground(new Color(255, 255, 255));
132         subtitle.setAlignmentX(Component.CENTER_ALIGNMENT);
133
134         heroPanel.add(title);
135         heroPanel.add(subtitle);
136     }
137
138     private void showLoginPage() {
139         // ...
140     }
141
142     private void showLoginDialog() {
143         // ...
144     }
145
146     private void populateUserOrders(String user) {
147         // ...
148     }
149 }
```

CHAPTER 5

OUTPUT SCREENSHOTS

Fig 5.1 LANDING PAGE

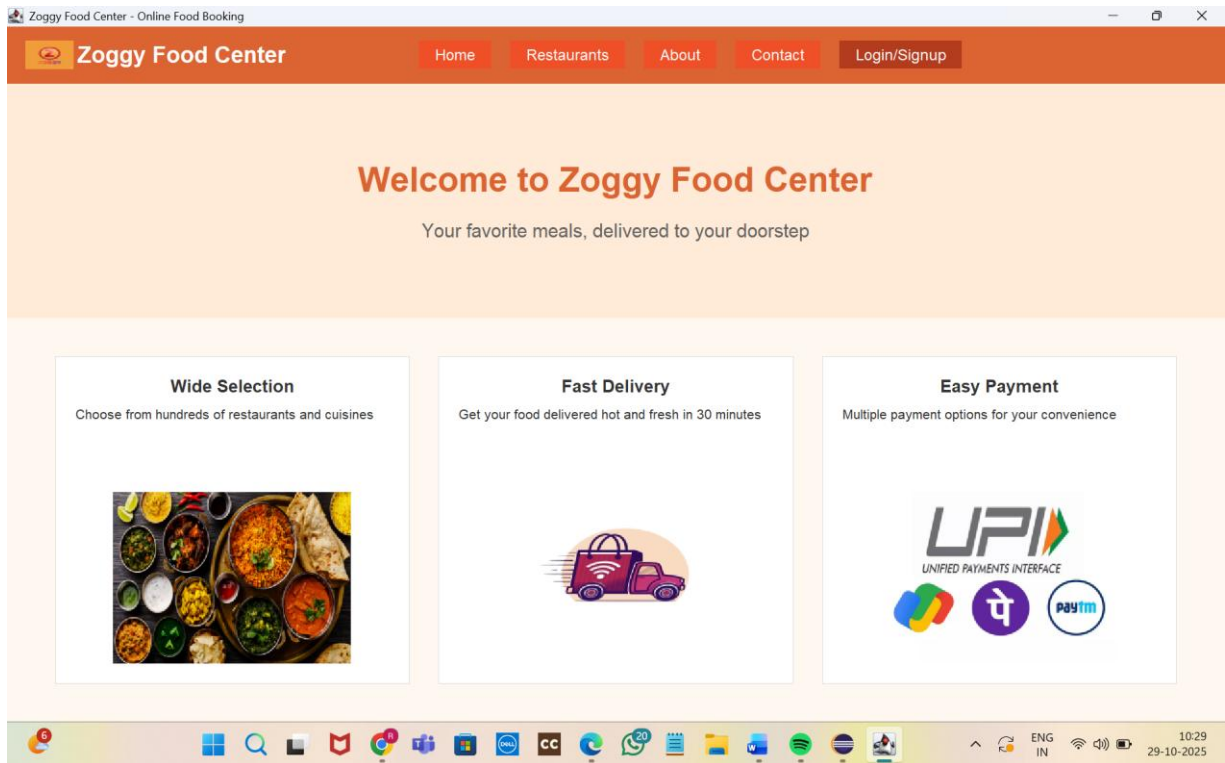


Fig 5.2 CUSTOMER LOGIN / SIGN UP

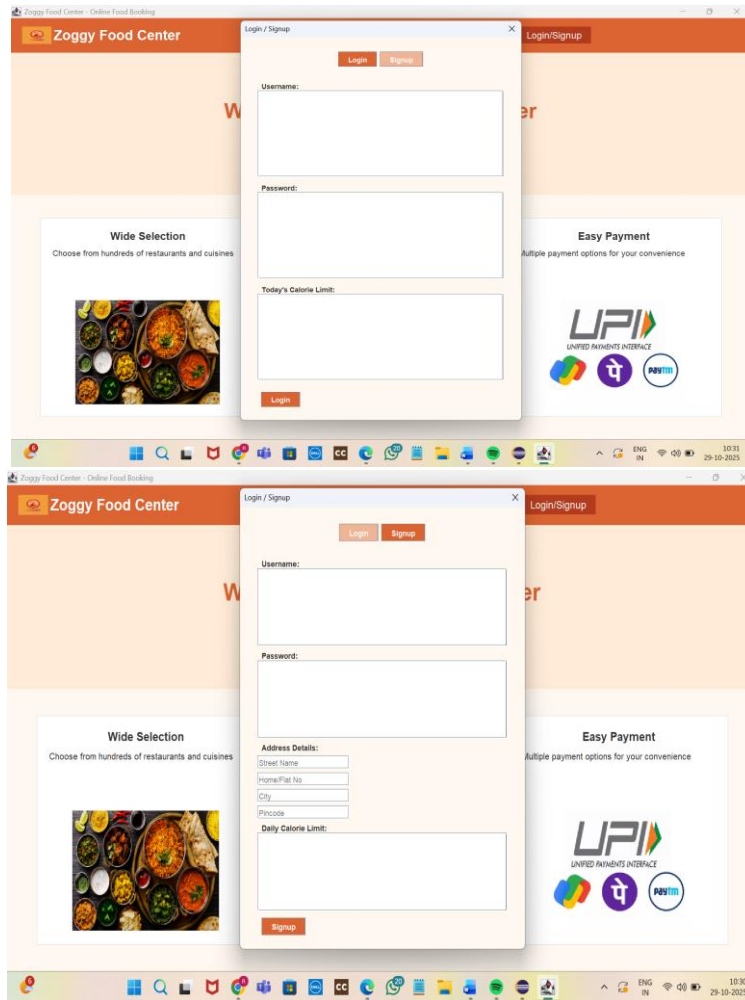


Fig 5.3 ADMIN LOGIN

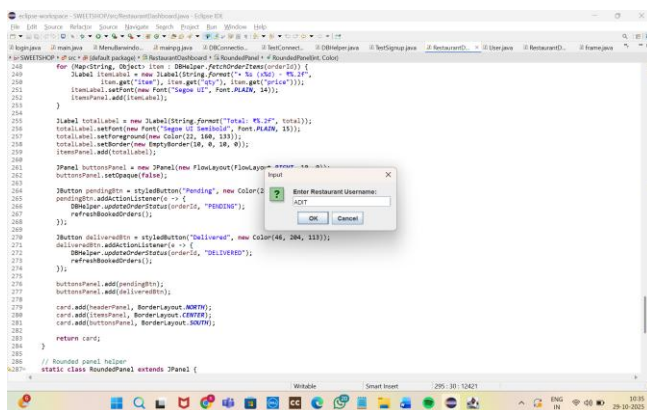


Fig 5.4 RESTAURANT FRAME

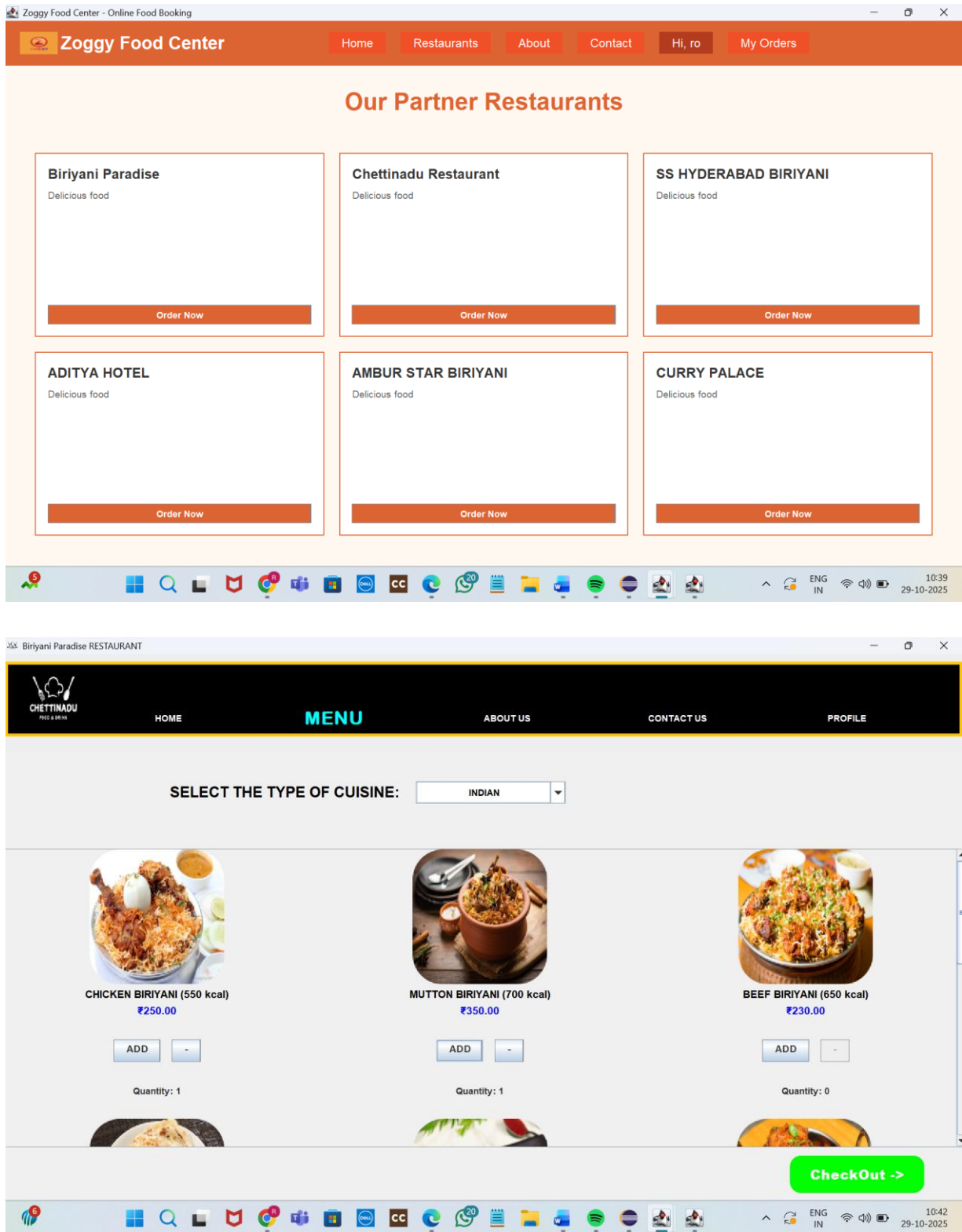


Fig 5.5 CHECKOUT FRAME

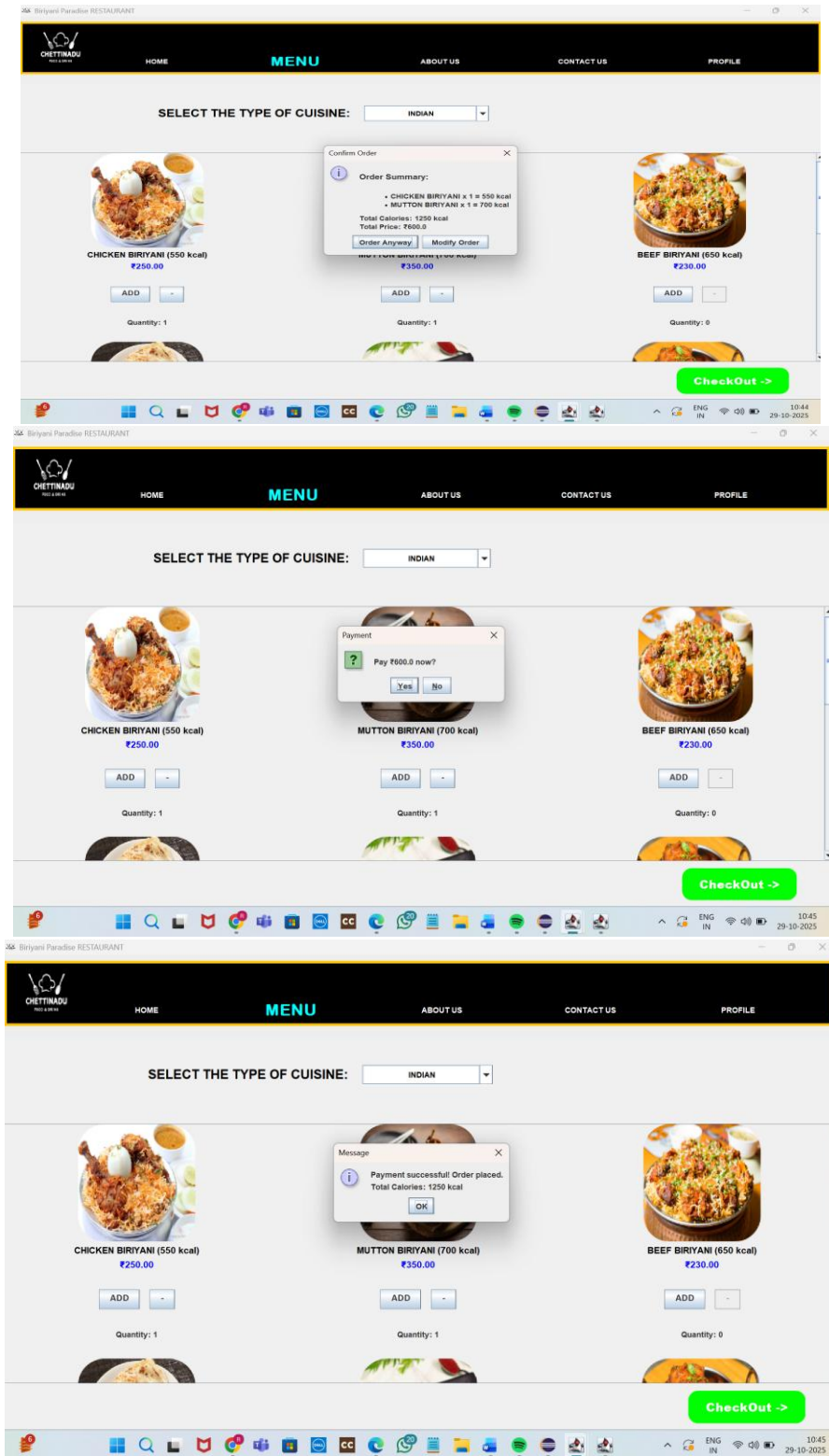


Fig 5.6 MY ORDERS FRAME

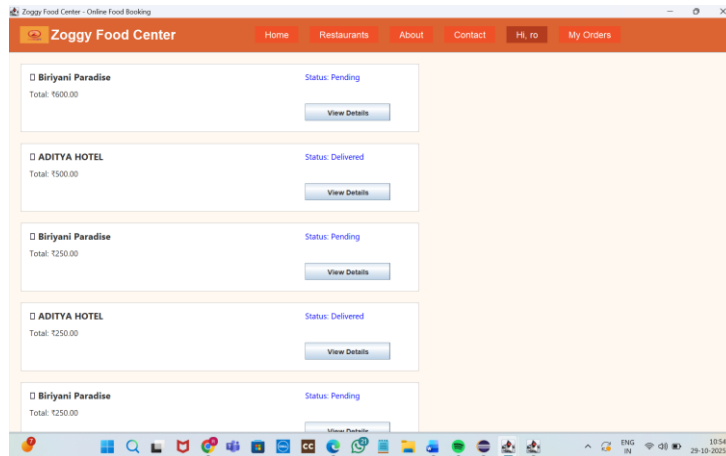


Fig 5.6 RESTAURANT FRAME

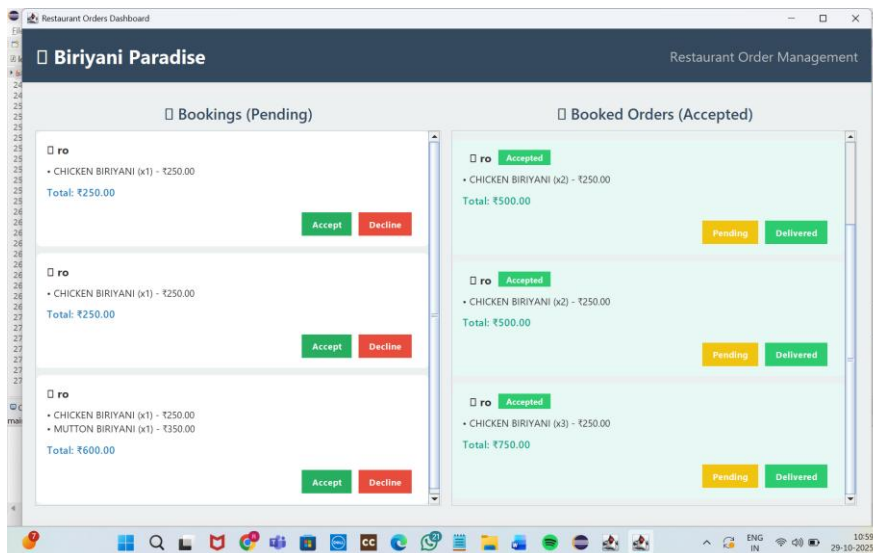


Fig 5.6 My Orders FRAME(After updated)

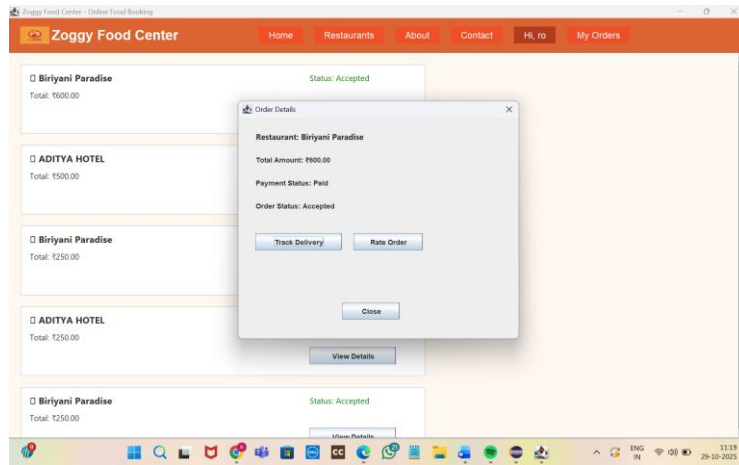
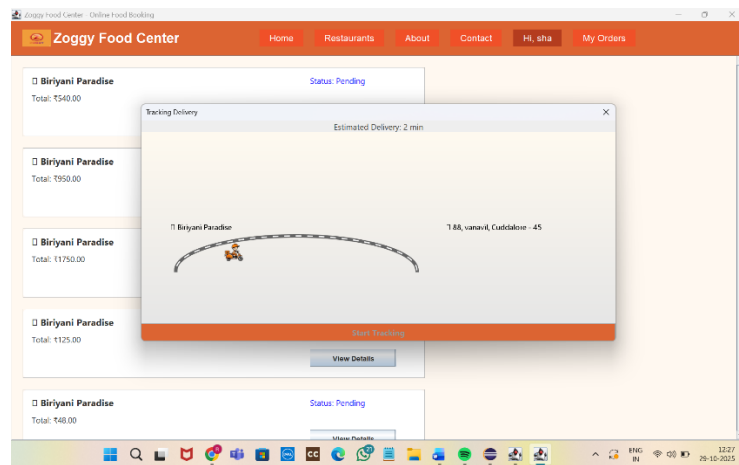


Fig 5.7 Food Tracking



CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

The *Online Food Ordering System* developed using **Java Swing** and **MySQL** effectively fulfills its objectives by providing a complete, database-driven desktop application for browsing restaurants, placing orders, and managing deliveries. The integration of **JDBC** ensures reliable data handling and smooth connectivity with the database. With separate interfaces for customers and restaurants (or administrators), the system offers clear functionality, user-friendly navigation, and maintainable architecture suitable for further enhancement.

FUTURE ENHANCEMENT

1. To make the system more feature-rich and scalable, the following improvements can be implemented in the future: Smart Cart and Payment System: Enhance the ordering process by adding a dynamic cart with quantity updates, item removal, and integration of secure online payment gateways.
2. Order Tracking and History: Introduce real-time order tracking and maintain complete order history for users and restaurants.
3. Enhanced User Accounts: Implement secure user authentication with password encryption and profile management for multiple customers and restaurants.
4. Data Analytics and Reports: Enable administrators to generate performance reports such as top-selling restaurants, customer statistics, and order trends.

REFERENCES

1. Oracle. *Java Platform, Standard Edition 8 API Specification — javax.swing Package Summary*.
<https://docs.oracle.com/javase/8/docs/api/javax/swing/packagesummary.html>
2. Oracle. *The Java Tutorials – JDBC Database Access*.
<https://docs.oracle.com/javase/tutorial/jdbc/index.html>
3. MySQL Documentation. *MySQL 8.0 Reference Manual*.
<https://dev.mysql.com/doc/refman/8.0/en/>
4. MySQL Documentation. *Connector/J 8.0 Developer Guide*.
<https://dev.mysql.com/doc/connector-j/8.0/en/>
5. Oracle. *Data Access Object (DAO) Design Pattern*.
<https://www.oracle.com/java/technologies/data-access-object.html>
6. GeeksforGeeks. *Java Swing Tutorial – Building GUI Applications*.
<https://www.geeksforgeeks.org/java-swing/>

