

## **Contents**

PROJECT SCOPE .....	2
GOALS AND OBJECTIVES .....	2
GUI FEATURES .....	4
Sign In Form: .....	4
Sign Up Form:.....	4
MENU ITEMS.....	5
DELIVERY.....	6
DATABASE CONNECTIVITY .....	8
CUSTOMER INFO .....	8
MENU_INFO .....	9
ORDER_INFO.....	10
DELIVERY_INFO .....	12
CONCLUSION.....	13

# ONLINE FOOD ORDERING SYSTEM

## PROJECT SCOPE

The convenience of ordering food from anywhere and having it delivered to your doorstep has made online food ordering systems highly desirable for modern consumers. These systems streamline the food ordering process, offering a vast range of restaurants and cuisines at users' fingertips. By lowering operational costs and leveraging technology, online food ordering platforms can provide more choices and competitive prices compared to traditional dining options.

Online food ordering data encompasses various information related to restaurant operations, customer preferences, sales performance, and delivery logistics. Companies with an online presence can use this data to gain insights into customer behavior and enhance their service offerings. This dataset, focusing on Pakistan's largest online food ordering platform, covers around six months of transactions and can be utilized to address multiple business and operational challenges:

- **Business performance:**  
Evaluate the performance of Pakistan's online food ordering sector.
- **Consumer behavior:**  
Understand customer preferences and purchasing patterns to optimize offerings.
- **Delivery and logistics:**  
Improve delivery efficiency and reduce operational costs.
- **Fraud and payments:**  
Monitor and prevent fraudulent activities.
- **Marketing campaign effectiveness:**  
Assess and enhance the success of marketing strategies.
- **Industry regulation compliance:**  
Ensure adherence to industry standards and consumer protection.
- **Product pricing:**  
Optimize pricing strategies to stay competitive.
- **Sales trend forecasting:**  
Identify and predict demand trends for better inventory management.
- **Inventory management:**  
Ensure adequate stock levels to meet customer demand.

## GOALS AND OBJECTIVES

These goals and objectives provide a comprehensive framework for analyzing Pakistan's Largest Online Food Ordering Dataset and addressing various aspects of the industry. By achieving these objectives, stakeholders can make informed decisions, enhance customer experiences, and contribute to the growth and efficiency of the sector.

### • **Business Performance:**

**Goal:** Evaluate the performance of Pakistan's online food ordering industry.

**Objectives:** Analyze sales and revenue data to identify operational strengths and weaknesses and improve profitability.

### • **Consumer Behavior:**

**Goal:** Understand customer behavior in Pakistan's online food ordering sector.

**Objectives:** Study customer preferences, purchasing habits, and histories to optimize inventory management, marketing strategies, and product development.

**• Delivery and Logistics:**

**Goal:** Enhance delivery and logistics services.

**Objectives:** Identify inefficiencies in shipping and delivery processes to reduce costs and improve delivery times.

**• Fraud and Payments:**

**Goal:** Monitor and prevent fraud.

**Objectives:** Analyze transaction data to identify potential fraud and take preventive actions.

**• Effectiveness of Marketing Campaigns:**

**Goal:** Assess the success of marketing initiatives.

**Objectives:** Evaluate marketing data to identify successful campaigns and improve ROI on future efforts.

**• Industry Regulation:**

**Goal:** Ensure compliance with industry regulations.

**Objectives:** Review transaction data and corporate practices to promote consumer protection and fair competition.

**• Product Pricing:**

**Goal:** Optimize product pricing strategies.

**Objectives:** Analyze price information and competition pricing methods to stay competitive and boost sales.

**• Forecasting Sales Trends:**

**Goal:** Predict future demand trends.

**Objectives:** Study sales data over time to identify patterns and make informed predictions.

**• Inventory Management:**

**Goal:** Improve inventory management practices.

**Objectives:** Anticipate product demand and ensure sufficient stock levels to fulfill customer orders efficiently.

## **BASIC FEATURES**

The basic features of an online food ordering system typically include:

**• Restaurant Management:**

Store and manage restaurant information, including name, location, menu, prices, availability, and ratings.

**• Order Management:**

Handle order details, including order ID, customer details, restaurant details, items ordered, quantity, total cost, delivery information, and order status.

**• Customer Management:**

Store and manage customer information, including customer ID, name, contact details, addresses, payment information, and order history.

**• Inventory Management:**

Track inventory levels, product stock, and availability, updating inventory based on orders and providing alerts for low stock items.

**• Shopping Cart:**

Allow users to add menu items to a virtual cart, review selections, and proceed to checkout.

**• Checkout and Payment Processing:**

Provide a secure checkout process with multiple payment options, ensuring secure payment validation and processing.

**• Search and Filtering:**

Enable users to search for restaurants or menu items based on keywords, categories, cuisine types, and other criteria, with filtering options for refined results.

**• Order History and Account Management:**

Allow users to view order history, track shipments, and manage account information

**• Analytics and Reporting:**

Collect and analyze data on customer behavior, sales performance, popular restaurants, and other metrics to generate reports for data-driven decisions.

## **GUI FEATURES**

The graphical user interface (GUI) of the Online Food Ordering System is designed to be intuitive and user-friendly.

### ***Sign In Form:***

#### **Fields:**

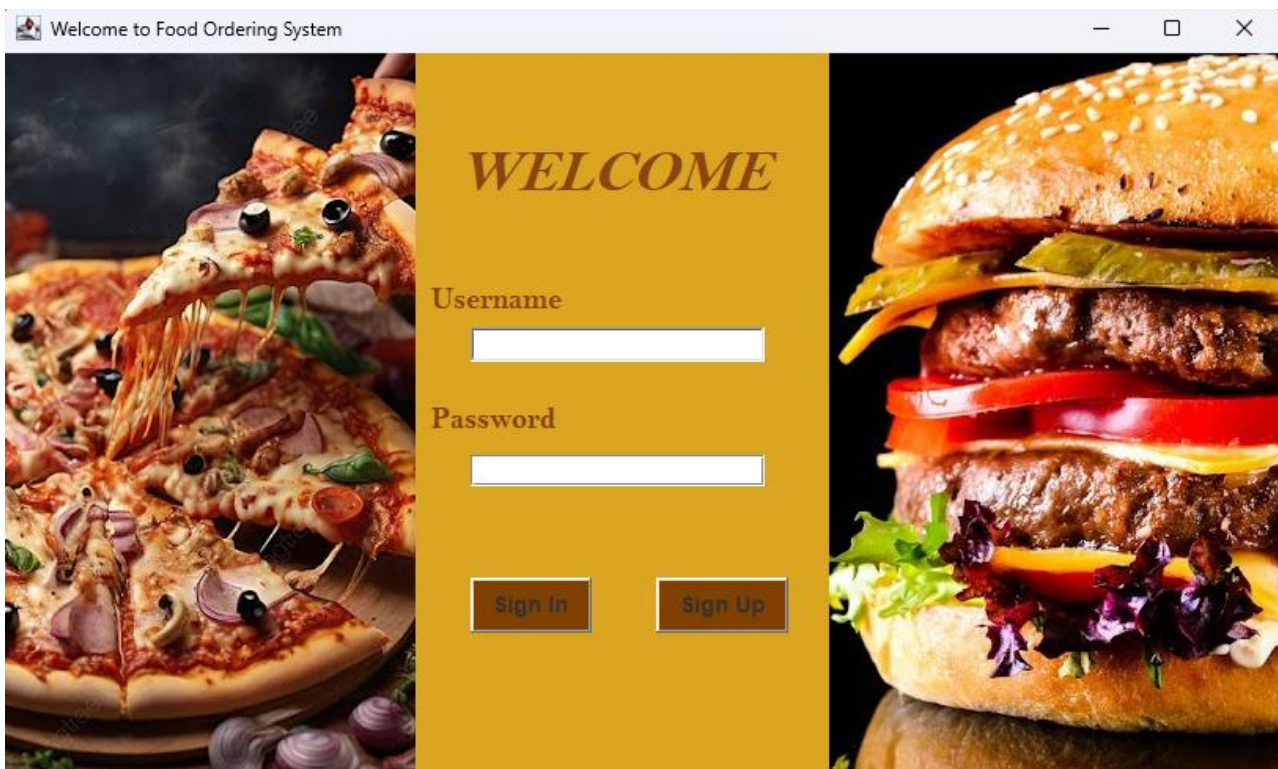
**Username:** Text field for entering the username.

**Password:** Password field for entering the password.

#### **Buttons:**

**Sign In:** Validates the credentials and grants access to the system.

**Sign Up:** Redirects to the Sign-Up form for new users.



### ***Sign Up Form:***

#### **Fields:**

**First Name:** Text field for entering the first name.

**Last Name:** Text field for entering the last name.

**Username:** Text field for entering the desired username.

**Password:** Password field for entering the desired password.

**Email:** Text field for entering the email address.

**Contact:** Text field for entering the phone number.

#### **Buttons:**

**OK:** Validates and submits the registration details.



Sign Up

## Create Your Account

First Name

Last Name

Username

Email

Password

Contact

OK

## MENU ITEMS

- **Menu Categories (Left Sidebar):**

- **Categories:**

- **Burger:** Button to display burger items.
    - **Pizza:** Button to display pizza items.
    - **Drinks:** Button to display drink items.

- **Menu Items Display (Center):**

- Each menu item is displayed with:
    - **Image:** Visual representation of the item.
    - **Name:** Text label displaying the name of the item.
    - **Price:** Text label displaying the price of the item.
    - **Add Button (+):** Button to add the item to the bill.

- **Bill Summary (Right Sidebar):**

- **Table:**

- **Columns:**

- **Item:** Name of the ordered item.
      - **Price:** Price of the ordered item.
      - **Qty:** Quantity of the ordered item.
      - **Total:** Total price for each ordered item (Price x Quantity).

- **Total Amount:**

- **Label (TOTAL):** Text label indicating the total amount.
    - **Value Field:** Field displaying the total calculated amount for the order.

- **Order Now Button:**


- **Order Now:** Button to finalize the order and proceed with the payment.

Menu Items

Burger

Pizza

Drinks



**Chicken Burger**  
Rs 500

+



**Zinger Burger**  
Rs 580

+



**Mighty Zinger**  
Rs 750

+



**Subway Burger**  
Rs 380

+



**Jalapeno Krunch**  
Rs 800

+



**Special Burger**  
Rs 800

+

BILL

Item	Price	Qty	Total
Zinger Burger	550	1	550

**TOTAL:** 550
 


ORDER NOW

Menu Items


Burger

Pizza


Drinks




**Vegetable Pizza**  
SELECT




**Chicken Fajita**  
Medium Rs...




**Pepporni Pizza**  
SELECT



**Bihari Kebab**  
SELECT



**Thick Crust Special**  
SELECT



**Crown Crust**  
SELECT

BILL

Item	Price	Qty	Total
Zinger Burger	550	1	550
Chicken Fajita(M)	1230	2	2460

**TOTAL:** 3010
 


ORDER NOW

Menu Items

Burger


Pizza

Drinks




**Juice**  
Rs 150

+



**Pepsi**  
Rs 100

+



**Water**  
Rs 70

+

BILL

Item	Price	Qty	Total
Zinger Burger	550	1	550
Chicken Fajita(M)	1230	2	2460
Pepsi	100	2	200

**TOTAL:** 3210
 

ORDER NOW

## DELIVERY

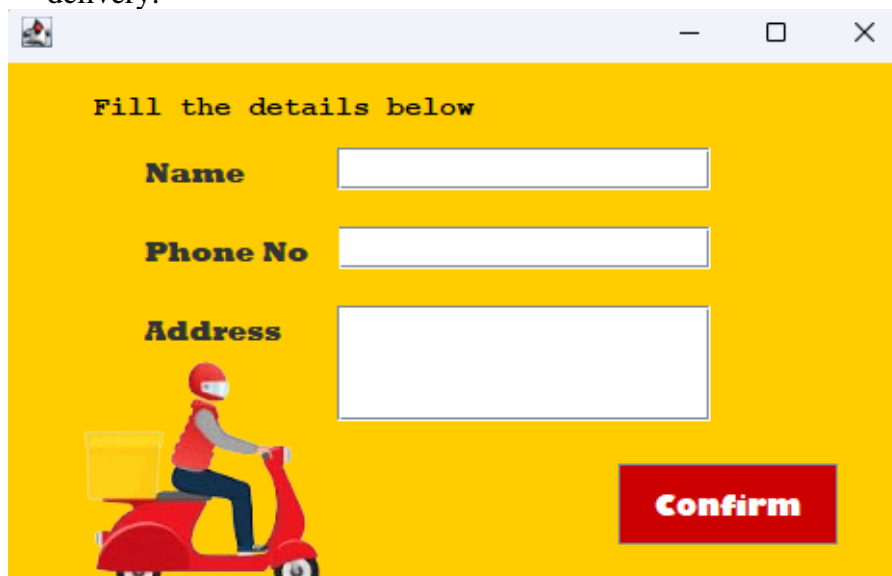
### 1. Form Fields:



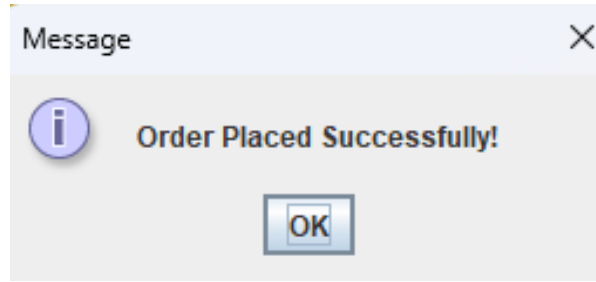
- **Name:**
    - Label: "Name"
    - Input: Text field for entering the customer's name.
  - **Phone No:**
    - Label: "Phone No"
    - Input: Text field for entering the customer's phone number.
  - **Address:**
    - Label: "Address"
    - Input: Multiline text field for entering the delivery address.
2. **Action Button:**
- **Confirm:**
    - Button labeled "Confirm" for submitting the delivery information.
3. **Visual Design:**
- **Background Color:**
    - Yellow background for a bright and inviting look.
  - **Text:**
    - Black bold text for labels to ensure readability.
  - **Image:**
    - Illustration of a delivery person on a scooter, enhancing the visual appeal and context of the form.

## Functional Features

1. **Input Fields:**
- **Name:**
    - User enters their name in this field.
  - **Phone No:**
    - User enters their phone number in this field.
  - **Address:**
    - User enters their delivery address in this field, with ample space for detailed information.
2. **Form Submission:**
- **Confirm Button:**
    - On clicking the "Confirm" button, the system will validate the input fields to ensure all necessary information is provided and correctly formatted.
    - Once validated, the information will be processed, and the order will be confirmed for delivery.



The screenshot shows a web form titled "Fill the details below" on a yellow background. It contains three input fields: "Name", "Phone No", and "Address". The "Address" field is a larger, multiline text area. At the bottom left, there is an illustration of a delivery person on a red scooter with a yellow box. At the bottom right, there is a red button labeled "Confirm". The form is displayed within a window with standard OS controls (minimize, maximize, close).



## **DATABASE CONNECTIVITY**

### ***CUSTOMER INFO***

```
private boolean authenticate(String username, String password) {
    boolean isAuthenticated = false;
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "sidra", "oracle");
        String query = "SELECT * FROM Customer_info WHERE username = ? AND password = ?";
        ps = conn.prepareStatement(query);
        ps.setString(1, username);
        ps.setString(2, password);
        System.out.println("Executing query: " + ps);
        rs = ps.executeQuery();
        if (rs.next()) {
            isAuthenticated = true;
            System.out.println("User found in the database.");
        } else {
            System.out.println("User not found in the database.");
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            if (rs != null) rs.close();
            if (ps != null) ps.close();
            if (conn != null) conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return isAuthenticated;
}
```



The screenshot shows a database management tool with two windows. The top window, titled 'CUSTOMER\_INFO', displays the table structure with columns: COLUMN\_ID, COLUMN\_NAME, DATA\_TYPE, NULLABLE, DATA\_DEFAULT, and COMMENTS. The bottom window shows the data of the table with columns: First Name, Last Name, Username, Email, and Contact.

COLUMN_ID	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COMMENTS
1	USERNAME	VARCHAR2 (20 BYTE)	No	(null)	1 (null)
2	PASSWORD	VARCHAR2 (20 BYTE)	No	(null)	2 (null)
3	LAST_NAME	VARCHAR2 (20 BYTE)	No	(null)	6 (null)
4	FIRST_NAME	VARCHAR2 (20 BYTE)	No	(null)	3 (null)
5	EMAIL	VARCHAR2 (50 BYTE)	No	(null)	4 (null)
6	CONTACT	NUMBER (38, 0)	No	(null)	5 (null)

First Name	Last Name	Username	Email	Contact
Sidra	Liaqat	sidra04	sidra04@gmail.com	12345
Ayesha	Tayyabah	aaeshhh	ayeshatayyabah@gmail.c...	3400500363
Tayyabah	Uzma	uzmaa	uzma97@gmail.com	3467812345
Amna	Khan	ak10	ak10@gmail.com	98765432

Buttons: Update, Delete, Add

## ***MENU\_INFO***

Connection conn = null;

PreparedStatement ps = null;

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "sidra", "oracle");

String query = "INSERT INTO Menu\_INFO (item, price, qty, total) VALUES (?, ?, ?, ?)";

ps = conn.prepareStatement(query);

DefaultTableModel model = (DefaultTableModel) Billl.getModel(); // Corrected variable name

for (int i = 0; i < model.getRowCount(); i++) {

String item = model.getValueAt(i, 0).toString();

int price = Integer.parseInt(model.getValueAt(i, 1).toString());

double qty = Double.parseDouble(model.getValueAt(i, 2).toString());

double total = Double.parseDouble(model.getValueAt(i, 3).toString());

ps.setString(1, item);

ps.setInt(2, price);

ps.setDouble(3, qty);

ps.setDouble(4, total);

```

int rowsInserted = ps.executeUpdate();
if (rowsInserted > 0) {
    System.out.println("Row " + (i+1) + " inserted successfully.");
} else {
    System.out.println("Failed to insert row " + (i+1) + ".");
}
}
} catch (Exception e1) {
    e1.printStackTrace();
} finally {
    try {
        if (ps != null) ps.close();
        if (conn != null) conn.close();
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}
}

```

The screenshot displays a database management interface. The top window shows the structure of the **MENU\_INFO** table with the following columns:

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 TOTAL	NUMBER (38, 0)	Yes	(null)	2	(null)
2 QTY	NUMBER	Yes	(null)	4	(null)
3 PRICE	NUMBER (38, 0)	Yes	(null)	3	(null)
4 ITEM	VARCHAR2 (20 BYTE)	Yes	(null)	1	(null)

The bottom window, titled "Menu Data Management", displays a table of menu items with their prices, quantities, and totals:

Item	Price	Quantity	Total
Pepsi	100.0	1	100.0
Special Burger	800.0	2	1600.0
Crown Crust(M)	1320.0	1	1320.0
Jalapeno Crunch	800.0	2	1600.0
Special Burger	800.0	3	2400.0
Chicken Burger	500.0	1	500.0
Chicken Burger	500.0	3	1500.0
Thick Crust(M)	1300.0	1	1300.0
Pepsi	100.0	1	100.0
Zinger Burger	550.0	1	550.0
Zinger Burger	550.0	1	550.0
Chicken Fajita(M)	1230.0	2	2460.0
Zinger Burger	550.0	1	550.0
Chicken Fajita(M)	1230.0	2	2460.0
Pepsi	100.0	2	200.0

At the bottom of the "Menu Data Management" window, there is a search bar and four buttons: Search, Update, Delete, and Add.

## ORDER\_INFO

```

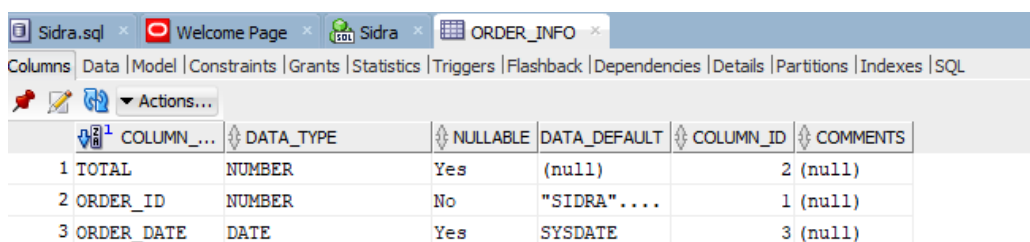
try {
    Class.forName("oracle.jdbc.driver.OracleDriver");

```

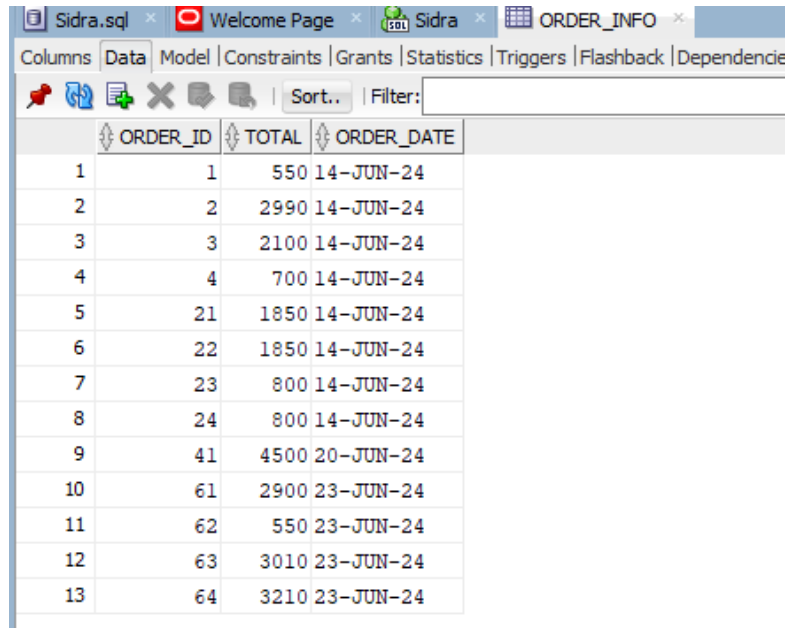
```

conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "sidra",
"oracle");
conn.setAutoCommit(false);
String query = "INSERT INTO ORDER_info (TOTAL) VALUES (?)";
ps = conn.prepareStatement(query);
ps.setString(1, gtotal.getText());
int rowsInserted = ps.executeUpdate();
if (rowsInserted > 0) {
    System.out.println("Order detail inserted successfully!");
    dispose();
}
conn.commit();
} catch (ClassNotFoundException ex) {
    ex.printStackTrace();
} catch (SQLException ex) {
    ex.printStackTrace();
    if (conn != null) {
        try {
            conn.rollback();
        } catch (SQLException se) {
            se.printStackTrace();
        }
    }
} finally {
    if (ps != null) {
        try {
            ps.close();
        } catch (SQLException se) {
            se.printStackTrace();
        }
    }
    if (conn != null) {
        try {
            conn.close();
        } catch (SQLException se) {
            se.printStackTrace();
        }
    }
}
}
});

```



	COLUMN_...	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	TOTAL	NUMBER	Yes	(null)	2 (null)	
2	ORDER_ID	NUMBER	No	"SIDRA"....	1 (null)	
3	ORDER_DATE	DATE	Yes	SYSDATE	3 (null)	



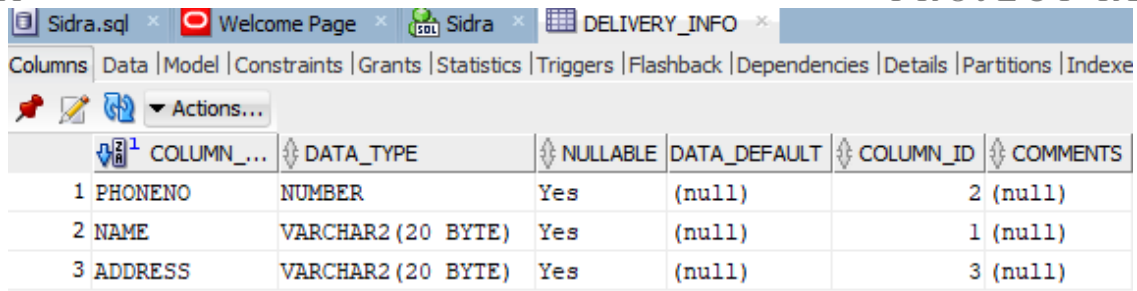
	ORDER_ID	TOTAL	ORDER_DATE
1	1	550	14-JUN-24
2	2	2990	14-JUN-24
3	3	2100	14-JUN-24
4	4	700	14-JUN-24
5	21	1850	14-JUN-24
6	22	1850	14-JUN-24
7	23	800	14-JUN-24
8	24	800	14-JUN-24
9	41	4500	20-JUN-24
10	61	2900	23-JUN-24
11	62	550	23-JUN-24
12	63	3010	23-JUN-24
13	64	3210	23-JUN-24

## ***DELIVERY\_INFO***

```

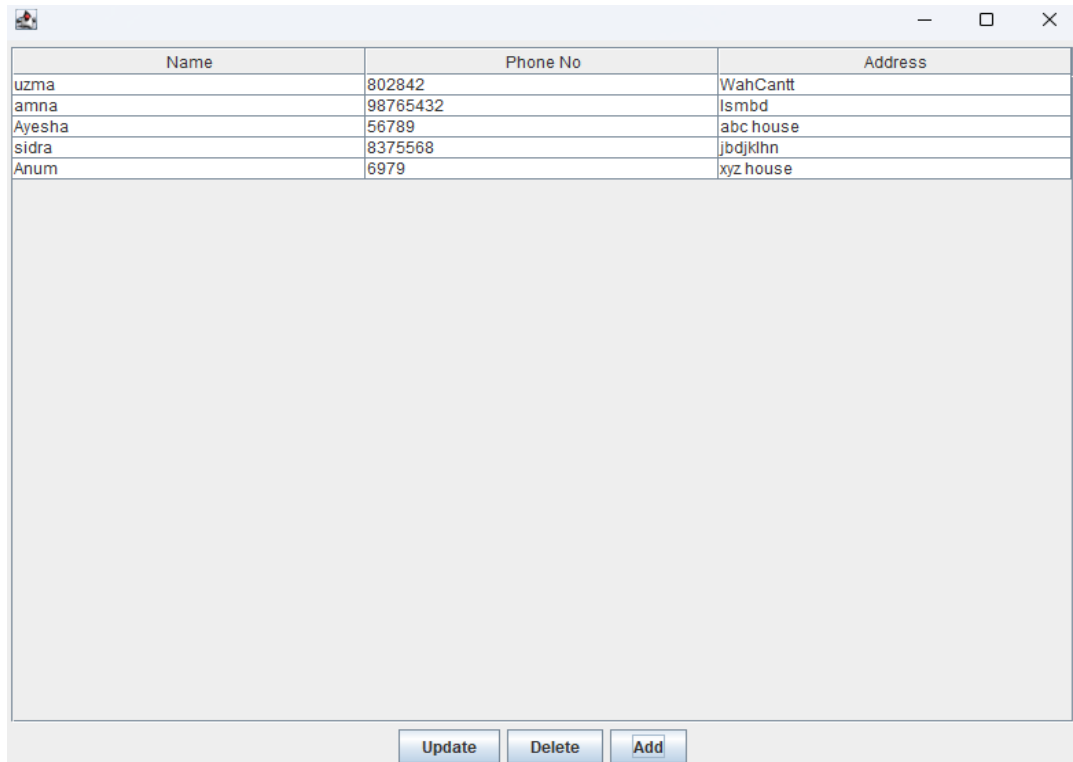
Connection conn = null;
PreparedStatement ps = null;
try {
    Class.forName("oracle.jdbc.driver.OracleDriver");
    conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "sidra", "oracle");
    String query = "INSERT INTO Delivery_INFO (name, Phoneno, Address) VALUES (?, ?, ?)";
    ps = conn.prepareStatement(query);
    ps.setString(1, name.getText());
    ps.setString(2, Phoneno.getText());
    ps.setString(3, Address.getText());
    int rowsInserted = ps.executeUpdate();
    if (rowsInserted > 0) {
        JOptionPane.showMessageDialog(OrderForm.this, "Order Placed Successfully!");
        System.out.println("Info added successfully.");
    } else {
        JOptionPane.showMessageDialog(OrderForm.this, "Order Placement Failed.");
    }
} catch (Exception e1) {
    e1.printStackTrace();
    JOptionPane.showMessageDialog(OrderForm.this, "Error: " + e1.getMessage());
} finally{
    try {
        if (ps != null) ps.close();
        if (conn != null) conn.close();
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}
dispose();
System.exit(0);
}

```



The screenshot shows the SQL Developer interface with the 'DELIVERY\_INFO' table selected. The 'Columns' tab is active, displaying the table's structure. The table has three columns: PHONENO (NUMBER), NAME (VARCHAR2(20 BYTE)), and ADDRESS (VARCHAR2(20 BYTE)). All columns are nullable and have a default value of (null). The column IDs are 2, 1, and 3 respectively.

	COLUMN_...	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PHONENO	NUMBER	Yes	(null)	2	(null)
2	NAME	VARCHAR2 (20 BYTE)	Yes	(null)	1	(null)
3	ADDRESS	VARCHAR2 (20 BYTE)	Yes	(null)	3	(null)



The screenshot shows a web application window with a table containing user information. The table has three columns: Name, Phone No, and Address. The data rows are as follows:

Name	Phone No	Address
uzma	802842	WahCantt
amna	98765432	Ismbd
Ayesha	56789	abc house
sidra	8375568	jbdjklhn
Anum	6979	xyz house

At the bottom of the window, there are three buttons: 'Update', 'Delete', and 'Add'.

## CONCLUSION

The food ordering system has significantly improved efficiency, customer satisfaction, and operational insights. It enhances user experience, reduces errors, and supports business growth through scalable and data-driven solutions. Adopting such technology is pivotal for staying competitive in the evolving food service industry.