

This Java program is an interactive command-line application for managing a food ordering system. It connects to a PostgreSQL database to perform user authentication, order processing, and administrative tasks. Users are classified into three roles: **Customer**, **Courier**, and **Admin**, each with specific functionalities.

## 1. Prerequisites

1. **Database:** A PostgreSQL database running on localhost with:
  - a. Database name: fodo
  - b. Port: 5433
  - c. Username: postgres
  - d. Password: 6789
2. **Tables in the Database:**
  - a. **users:** Contains user credentials and roles.
    - i. Columns: user\_id, username, password, role
  - b. **fooditems:** Contains food items available for ordering.
    - i. Columns: food\_id, food\_name, price, stock
  - c. **orders:** Stores customer orders.
    - i. Columns: order\_id, user\_id, total\_price, delivery\_address
3. **Java Setup:** Ensure you have:
  - a. Java installed.
  - b. PostgreSQL JDBC driver added to your classpath.

## 2. How to Use the Program

### *Program Flow*

1. **Login Page:**
  - a. Prompts for a username and password.
  - b. Validates credentials against the database.
  - c. If valid, directs the user to the appropriate menu based on their role:
    - i. **Customer**
    - ii. **Courier**
    - iii. **Admin**
2. **Roles and Their Functionalities**

## A. Customer

- **Objective:** Browse the menu, order food, and provide delivery details.
- **Steps:**
  - View available menu items along with prices and stock.
  - Enter the **food ID** and specify the **quantity**.
  - If the requested quantity exceeds stock, you'll be notified and prompted to re-enter.
  - Add more items or finalize your order.
  - Once finalized:
    - Enter your delivery address.
    - The order is stored in the database with the total price and address.
  - Confirmation of the order is displayed.

## B. Courier

- **Objective:** View orders assigned for delivery.
- **Steps:**
  - View all orders from the database.
  - Each order displays:
    - **Order ID**
    - **Delivery Address**
    - **Total Price**
  - No further action is required (this functionality is read-only).

## C. Admin

- **Objective:** Manage food stock and view orders.
- **Steps:**
  - Access the **Admin Menu** with three options:
    - **View Customer Orders:**
      - Displays a list of all orders placed, including:
        - Order ID, User ID, Total Price
    - **Manage Food Stock:**

- View all food items with their current stock levels.
  - Increase stock by entering the food ID and the amount to add.
  - Invalid food IDs result in an error message.
- **Exit:** Return to the login page.
- Update stock or view orders as needed.

### 3. Program Workflow

1. **Authentication:**
  - a. The authenticate method queries the users table to verify credentials.
  - b. If credentials match, the user's role determines the menu they access.
2. **Customer Workflow:**
  - a. Queries the fooditems table to display menu items.
  - b. Updates stock in the fooditems table after successful orders.
  - c. Inserts new records in the orders table to track purchases.
3. **Courier Workflow:**
  - a. Queries the orders table to retrieve order details.
4. **Admin Workflow:**
  - a. Queries the orders table to view all orders.
  - b. Queries and updates the fooditems table for stock management.

### 4. Key Features

- **Secure Authentication:** Prevents unauthorized access by validating credentials.
- **Dynamic Menu Display:** Displays only food items with available stock.
- **Order Management:**
  - Updates stock dynamically.
  - Calculates total order cost.
  - Records delivery addresses.
- **Role-Based Access:** Separates functionality for customers, couriers, and admins.
- **Database Interaction:** Uses SQL queries to fetch and modify data in real-time.

### 5. Potential Enhancements

1. **Password Security:** Use hashed passwords for storage and validation.

2. **Input Validation:** Add stricter checks for user input to prevent SQL injection and runtime errors.
3. **Order History:** Allow customers to view previous orders.
4. **Courier Actions:** Enable couriers to mark orders as delivered.