

JAVA CODE FILES:

Orderdao.java:

```
package com.restaurant.dao;

import java.util.List;
import com.restaurant.model.Order;

public interface Orderdao {
    void addOrder(Order o);
    List<Order> getAllOrders();
    Order getOrderById(int id);
    void updateOrder(Order o);
    void deleteOrder(int id);
}
```

OrderdaoImpl.java:

```
package com.restaurant.dao;

import java.sql.*;
import java.util.*;
import com.restaurant.model.Order;
import com.restaurant.util.DBConnection;

public class OrderdaoImpl implements Orderdao {

    public void addOrder(Order o) {
        try {
            Connection con = DBConnection.getConnection();
            PreparedStatement ps = con.prepareStatement(
                "INSERT INTO orders(customername, tablenumber, items) VALUES (?,?,?)"
            );
            ps.setString(1, o.getCustomerName());
            ps.setInt(2, o.getTableNumber());
            ps.setString(3, o.getItems());
            ps.executeUpdate();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public List<Order> getAllOrders() {
        List<Order> list = new ArrayList<>();
        try {
            Connection con = DBConnection.getConnection();
            ResultSet rs = con.createStatement()
                .executeQuery("SELECT * FROM orders");
            while (rs.next()) {
```

```

        Order o = new Order();
        o.setOrderId(rs.getInt("orderid"));
        o.setCustomerName(rs.getString("customername"));
        o.setTableNumber(rs.getInt("tablenumber"));
        o.setItems(rs.getString("items"));
        list.add(o);
    }
} catch (Exception e) {
    e.printStackTrace();
}
return list;
}

public Order getOrderById(int id) {
    Order o = null;
    try {
        Connection con = DBConnection.getConnection();
        PreparedStatement ps = con.prepareStatement(
            "SELECT * FROM orders WHERE orderid=?"
        );
        ps.setInt(1, id);
        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
            o = new Order();
            o.setOrderId(rs.getInt("orderid"));
            o.setCustomerName(rs.getString("customername"));
            o.setTableNumber(rs.getInt("tablenumber"));
            o.setItems(rs.getString("items"));
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return o;
}

public void updateOrder(Order o) {
    try {
        Connection con = DBConnection.getConnection();
        PreparedStatement ps = con.prepareStatement(
            "UPDATE orders SET customername=?, tablenumber=?, items=? WHERE orderid=?"
        );
        ps.setString(1, o.getCustomerName());
        ps.setInt(2, o.getTableNumber());
        ps.setString(3, o.getItems());
        ps.setInt(4, o.getOrderId());
        ps.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```
public void deleteOrder(int id) {  
    try {  
        Connection con = DBConnection.getConnection();  
        PreparedStatement ps = con.prepareStatement(  
            "DELETE FROM orders WHERE orderid=?"  
        );  
        ps.setInt(1, id);  
        ps.executeUpdate();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Order.java:

```
package com.restaurant.model;  
  
public class Order {  
  
    private int orderId;  
    private String customerName;  
    private int tableNumber;  
    private String items;  
  
    public Order() {}  
  
    public Order(String customerName, int tableNumber, String items) {  
        this.customerName = customerName;  
        this.tableNumber = tableNumber;  
        this.items = items;  
    }  
  
    public int getOrderId() {  
        return orderId;  
    }  
    public void setOrderId(int orderId) {  
        this.orderId = orderId;  
    }  
  
    public String getCustomerName() {  
        return customerName;  
    }  
    public void setCustomerName(String customerName) {  
        this.customerName = customerName;  
    }
```

```

public int getTableNumber() {
    return tableNumber;
}
public void setTableNumber(int tableNumber) {
    this.tableNumber = tableNumber;
}

public String getItems() {
    return items;
}
public void setItems(String items) {
    this.items = items;
}
}

```

SERVLETS:

CreateOrderServlet.java:

```

package com.restaurant.servlet;

import jakarta.servlet.*;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;
import java.io.IOException;
import com.restaurant.dao.*;
import com.restaurant.model.Order;

@WebServlet("/CreateOrderServlet")
public class CreateOrderServlet extends HttpServlet {

    protected void doPost(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {

        String name = req.getParameter("customerName");
        int table = Integer.parseInt(req.getParameter("tableNumber"));
        String items = req.getParameter("items");

        Order o = new Order(name, table, items);
        new OrderdaoImpl().addOrder(o);

        res.sendRedirect("ListOrderServlet");
    }
}

```

EditOrderServlet.java:

```
package com.restaurant.servlet;

import java.io.IOException;
import jakarta.servlet.*;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;
import com.restaurant.dao.*;
import com.restaurant.model.Order;

@WebServlet("/EditOrderServlet")
public class EditOrderServlet extends HttpServlet {

    protected void doPost(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {

        Order o = new Order();
        o.setOrderId(Integer.parseInt(req.getParameter("orderId")));
        o.setCustomerName(req.getParameter("customerName"));
        o.setTableNumber(Integer.parseInt(req.getParameter("tableNumber")));
        o.setItems(req.getParameter("items"));

        new OrderdaoImpl().updateOrder(o);
        res.sendRedirect("ListOrderServlet");
    }
}
```

ListOrderServlet.java:

```
package com.restaurant.servlet;

import java.io.IOException;
import java.util.List;
import jakarta.servlet.*;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;
import com.restaurant.dao.*;
import com.restaurant.model.Order;

@WebServlet("/ListOrderServlet")
public class ListOrderServlet extends HttpServlet {

    protected void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {

        req.setAttribute("orders", new OrderdaoImpl().getAllOrders());
        req.getRequestDispatcher("view_orders.jsp").forward(req, res);
    }
}
```

RemoveOrderServlet.java:

```
package com.restaurant.servlet;

import java.io.IOException;
import jakarta.servlet.*;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;
import com.restaurant.dao.*;

@WebServlet("/RemoveOrderServlet")
public class RemoveOrderServlet extends HttpServlet {

    protected void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {

        int id = Integer.parseInt(req.getParameter("id"));
        new OrderdaoImpl().deleteOrder(id);

        res.sendRedirect("ListOrderServlet");
    }
}
```

JSP FILES:

index.jsp:

```
<%@ page contentType="text/html; charset=UTF-8" %>
<!DOCTYPE html>
<html>
<head>
<title>Recruitment Management</title>
</head>
<body>

<h2>Restaurant Order System</h2>
<a href="add_order.jsp">Add Order</a> |
<a href="ListOrderServlet">View Orders</a>

</body>
</html>
```

add_order.jsp:

```
<%@ page contentType="text/html; charset=UTF-8" %>
<!DOCTYPE html>
<html>
<head>
<title>Add Order</title>
```

```

</head>
<body>

<h2>Add Order</h2>
<form action="CreateOrderServlet" method="post">
Customer Name: <input type="text" name="customerName"><br><br><br>
Table Number: <input type="number" name="tableNumber"><br><br><br>
Items: <textarea name="items" rows="4" cols="30"></textarea><br><br><br>
<input type="submit" value="Save">
</form>

</body>
</html>

```

view_order.jsp:

```

<%@ page contentType="text/html; charset=UTF-8" %>
<%@ page import="java.util.List" %>
<%@ page import="com.restaurant.model.Order" %>

<%
    List<Order> list = (List<Order>) request.getAttribute("orders");
%>

<!DOCTYPE html>
<html>
<head>
    <title>Order List</title>
</head>
<body>

<h2>Orders</h2>

<table border="1" cellpadding="8">
<tr>
    <th>ID</th>
    <th>Customer Name</th>
    <th>Table Number</th>
    <th>Items</th>
    <th>Action</th>
</tr>

<%
    if (list != null) {
        for (Order o : list) {
%>
<tr>
    <td><%= o.getId() %></td>
    <td><%= o.getCustomerName() %></td>
    <td><%= o.getTableNumber() %></td>

```

```

<td><%= o.getItems() %></td>
<td>
    <a href="update_order.jsp?id=<%= o.getOrderId() %>">Edit</a> |
    <a href="RemoveOrderServlet?id=<%= o.getOrderId() %>" 
        onclick="return confirm('Are you sure you want to delete?')">
        Delete
    </a>
</td>
</tr>
<%
    }
}
%>

</table>

<br>
<a href="index.jsp">Back to Home</a>

</body>
</html>

```

update_order.jsp:

```

<%@ page contentType="text/html; charset=UTF-8" %>
<%@ page import="com.restaurant.dao.Orderdao,com.restaurant.dao.OrderdaoImpl" %>
<%@ page import="com.restaurant.model.Order" %>

<%
    int id = Integer.parseInt(request.getParameter("id"));

    Orderdao dao = new OrderdaoImpl();
    Order o = dao.getOrderById(id);
%>

<!DOCTYPE html>
<html>
<head>
    <title>Edit Order</title>
</head>
<body>

<h2>Edit Order</h2>

<form action="EditOrderServlet" method="post">
    <input type="hidden" name="orderId" value="<%= o.getOrderId() %>">

    Customer Name:
    <input type="text" name="customerName"
        value="<%= o.getCustomerName() %>" required>

```

```
<br><br>
```

Table Number:

```
<input type="number" name="tableNumber"
      value="<% o.getTableNumber() %>" required>
<br><br>
```

Items:

```
<input type="text" name="items"
      value="<% o.getItems() %>" required>
<br><br>
```

```
<input type="submit" value="Update Order">
</form>
```

```
<br>
<a href="ListOrderServlet">Back to Orders</a>
```

```
</body>
</html>
```

DATABASE:

```
CREATE DATABASE IF NOT EXISTS restaurantdb;
USE restaurantdb;
```

```
CREATE TABLE IF NOT EXISTS orders (
    orderid INT PRIMARY KEY AUTO_INCREMENT,
    customername VARCHAR(100) NOT NULL,
    tablenumber INT NOT NULL,
    items TEXT NOT NULL
);
SELECT * FROM orders;
```

OUTPUT:

The screenshot shows a web browser window with the title "Recruitment Management". The address bar displays "localhost:8081/Restaurant/index.jsp". Below the title, the text "Restaurant Order System" is prominently displayed. At the bottom of the page, there are two links: "Add Order" and "View Orders".

The screenshot shows a web browser window with the title "Add Order". The address bar displays "localhost:8081/Restaurant/add_order.jsp". The page has a heading "Add Order". It contains fields for "Customer Name" (set to "Darshan") and "Table Number" (set to "225"). A text area labeled "Items:" contains the text "Parotta, Chilli Parotta, Chicken Rice, Ice cream". At the bottom is a "Save" button.

The screenshot shows a web browser window with the title "Order List". The address bar displays "localhost:8081/Restaurant/ListOrderServlet". The page has a heading "Orders". It displays a table with one row, showing ID 2, Customer Name Darshan, Table Number 225, Items Parotta, Chilli Parotta, Chicken Rice, and Action Edit | Delete. At the bottom is a "Back to Home" link.

Result Grid					Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
orderid	customername	tablenumber	items	totalbill				
2	Darshan	225	Parotta, Chilli Parotta, Chicken Rice, Ice cream	NULL				

Add Order

localhost:8081/Restaurant/add_order.jsp

Add Order

Customer Name: Hari

Table Number: 110

Chicken Rice, Parotta, Kulfi

Items:

Save

Order List

localhost:8081/Restaurant/ListOrderServlet

Orders

ID	Customer Name	Table Number	Items	Action
2	Darshan	225	Parotta, Chilli Parotta, Chicken Rice, Ice cream	Edit Delete
3	Hari	110	Chicken Rice, Parotta, Kulfi	Edit Delete

[Back to Home](#)

Result Grid					Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:	
	orderid	customername	tablenumber	items	totalbill				
▶	2	Darshan	225	Parotta, Chili Parotta, Chicken Rice, Ice cream	NULL				
▶	3	Hari	110	Chicken Rice, Parotta, Kulfi	NULL				
*	HULL	HULL	HULL	HULL	HULL				

Add Order

localhost:8081/Restaurant/add_order.jsp

Add Order

Customer Name: Kamesh

Table Number: 157

Chicken Rice, Chilli Parotta,
Desert

Items:

Save

Orders

ID	Customer Name	Table Number	Items	Action
2	Darshan	225	Parotta, Chilli Parotta, Chicken Rice, Ice cream	Edit Delete
3	Hari	110	Chicken Rice, Parotta, Kulfi	Edit Delete
4	Kamesh	157	Chicken Rice, Chilli Parotta, Desert	Edit Delete

[Back to Home](#)

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	orderid	customername	tablenumber	items	totalbill
▶	2	Darshan	225	Parotta, Chili Parotta, Chicken Rice, Ice cream	NULL
	3	Hari	110	Chicken Rice, Parotta, Kulfi	NULL
*	4	Kamesh	157	Chicken Rice, Chili Parotta, Desert	NULL
*	HULL	HULL	HULL	HULL	HULL