

SRK Food Delivery App (Project Guide)

"SRK Food Delivery App: User-Focused Solution with Servlets, JSP, JDBC, DAO, and MySQL Integration"

Project Overview:

The **SRK Food Delivery App** is a dynamic web-based application tailored to enhance the food ordering experience. Designed for users, it allows individuals to effortlessly browse the menu, add items to their cart, and place orders. Additionally, users can view their order history for easy reference. The application **utilizes Servlets, JSP, JDBC, and DAO** for efficient data management, with all user data securely stored in a **MySQL database**. This setup ensures a smooth and interactive experience for managing food orders and tracking order history.

Technologies Used:

- ✚ Java Servlets: For handling HTTP requests and responses.
- ✚ JSP (Java Server Pages): To generate dynamic web pages.
- ✚ DAO (Data Access Object): To abstract the database operations and interact with the MySQL database.
- ✚ MySQL: The relational database to store user, order and many other tables data.
- ✚ JDBC (Java Database Connectivity): For connecting and executing SQL queries on the MySQL database.

Project Structure:

Java Resources (src/main/java)-

- ✚ **com.srk.connector**: Handles database connectivity using JDBC.
- ✚ **com.srk.controller**: Contains servlet controllers that manage different user requests (e.g., Login, Menu, Orders).
- ✚ **com.srk.dao**: The Data Access Object layer that interacts with the database (e.g., UserDao, OrderDao, MenuDao).
- ✚ **com.srk.driver**: Manages the database driver (JDBC) setup and configuration.
- ✚ **com.srk.model**: Contains model classes such as User, Order, and MenuItem etc.. that represent the entities in the system.

Webapp Directory (src/main/webapp)-

- **JSP Files:**
 - home.jsp: The home page where users can view the different restaurants.
 - menu.jsp: Displays the list of food items.
 - cart.jsp: Manages the user's selected items.
 - checkout.jsp: The checkout page to confirm the order.

- login.jsp: User login page.
- signup.jsp: User registration page.
- orderhistory.jsp: Displays the order history for users.
- Other relevant JSPs like confirmorder.jsp, logout.jsp, and more.
- **CSS Files:**
 - style.css: Global styles for the application.
 - loginstyle.css and homestyle.css: Additional styles specific to login and home pages.
- **Images:** Contains the static images used in the application.

META-INF and WEB-INF:

- **META-INF:** Configuration metadata for the application.
- **WEB-INF:** Contains the web.xml configuration file, which defines the servlets and servlet-mappings.

Key Components:

1. DAO Layer

The DAO layer in com.srk.dao handles the database interactions. It includes:

- UserDAO: Handles user-related operations like login and registration.
- OrderDAO: Deals with creating and retrieving orders.
- MenuDAO: Manages the list of food items in the restaurant's menu.

2. Controller Layer

The controllers in com.srk.controller handle incoming HTTP requests and responses, manage the business logic, and decide which JSP page to forward the request to. For example:

- LoginController: Handles user authentication and redirects to the respective page.
- OrderController: Manages the creation of new orders and viewing order history.

3. Model Classes

Model classes in com.srk.model represent the core entities in the application, such as:

- User: Represents a user with fields like id, name, email, and password.
- Order: Represents an order, with details like orderID, userID, and orderTotal.
- MenuItem: Represents individual food items on the restaurant's menu.

Database Setup

You can use the following SQL code to set up the database tables for the application:

For user table:

```
CREATE TABLE `user` (  
  `user_id` int NOT NULL AUTO_INCREMENT,  
  `user_name` varchar(255) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `phone_no` varchar(15) DEFAULT NULL,  
  `password` varchar(255) NOT NULL,  
  `adress` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`user_id`),  
  UNIQUE KEY `email` (`email`)  
)
```

For restaurant table:

```
CREATE TABLE `restaurant` (  
  `restaurant_Id` int NOT NULL AUTO_INCREMENT,  
  `restaurant_Name` varchar(255) NOT NULL,  
  `cuisinType` varchar(100) DEFAULT NULL,  
  `deliveryTime` int DEFAULT NULL,  
  `adress` varchar(255) DEFAULT NULL,  
  `rating` double DEFAULT NULL,  
  `adminId` int DEFAULT NULL,  
  `isActive` varchar(10) DEFAULT NULL,  
  PRIMARY KEY (`restaurant_Id`)  
)
```

For menu table:

```
CREATE TABLE `menu` (
  `menuId` int NOT NULL AUTO_INCREMENT,
  `restaurantId` int DEFAULT NULL,
  `itemName` varchar(255) NOT NULL,
  `description` text,
  `price` double NOT NULL,
  `isAvailable` tinyint(1) NOT NULL,
  PRIMARY KEY (`menuId`),
  KEY `restaurantId` (`restaurantId`),
  CONSTRAINT `menu_ibfk_1` FOREIGN KEY (`restaurantId`) REFERENCES
`restaurant` (`restaurant_Id`)
)
```

For order history:

```
CREATE TABLE `orderhistory` (
  `orderHistoryId` int NOT NULL AUTO_INCREMENT,
  `orderId` int DEFAULT NULL,
  `userId` int DEFAULT NULL,
  `orderDate` date DEFAULT NULL,
  `totalAmount` double NOT NULL,
  `status` varchar(50) NOT NULL,
  PRIMARY KEY (`orderHistoryId`),
  KEY `orderId` (`orderId`),
  KEY `userId` (`userId`),
  CONSTRAINT `orderhistory_ibfk_1` FOREIGN KEY (`orderId`) REFERENCES
`ordertable` (`orderId`),
```

```
CONSTRAINT `orderhistory_ibfk_2` FOREIGN KEY (`userId`) REFERENCES `user`  
(`user_id`)  
)
```

For orderitem:

```
CREATE TABLE `orderitem` (  
  `orderId` int NOT NULL AUTO_INCREMENT,  
  `menuId` int DEFAULT NULL,  
  `quantity` int NOT NULL,  
  `subTotal` double NOT NULL,  
  PRIMARY KEY (`orderId`),  
  KEY `orderId` (`orderId`),  
  KEY `menuId` (`menuId`),  
  CONSTRAINT `orderitem_ibfk_1` FOREIGN KEY (`orderId`) REFERENCES  
  `ordertable` (`orderId`),  
  CONSTRAINT `orderitem_ibfk_2` FOREIGN KEY (`menuId`) REFERENCES `menu`  
  (`menuId`)  
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

For order table:

```
CREATE TABLE `ordertable` (  
  `orderId` int NOT NULL AUTO_INCREMENT,  
  `userId` int DEFAULT NULL,  
  `restaurantId` int DEFAULT NULL,  
  `orderDate` datetime DEFAULT NULL,  
  `totalAmount` double NOT NULL,  
  `status` varchar(50) NOT NULL,
```

```
`paymentMode` varchar(50) NOT NULL,  
  
PRIMARY KEY (`orderId`),  
  
KEY `userId` (`userId`),  
  
KEY `restaurantId` (`restaurantId`),  
  
CONSTRAINT `ordertable_ibfk_1` FOREIGN KEY (`userId`) REFERENCES `user`  
(`user_id`),  
  
CONSTRAINT `ordertable_ibfk_2` FOREIGN KEY (`restaurantId`) REFERENCES  
`restaurant` (`restaurant_id`)  
)
```

Take care of servlets and data base before running the application 😊