Online Food Ordering System – Project Documentation

---

1. Project Title

Online Food Ordering System (OFOS)

---

2. Project Overview

The Online Food Ordering System (OFOS) is a full-stack web application focused on **CRUD operations**. It allows the management of **users, restaurants, menus, and orders**. This version does not include authentication or security layers. The system provides a digital platform for customers to browse menus and place orders, while admins and restaurant staff can efficiently manage restaurant operations.

---

3. Technology Stack

Layer	Technology
Frontend	React.js
Backend	Spring Boot (Java)
Database	MySQL

---

4. User Roles (Logical)

Role	Responsibilities
Admin	Manage users and restaurants
Restaurant Staff	Manage menu items and orders
Customer	Browse menus and place orders

---

5. Core Functional Requirements

5.1 User Management

- Create users
- View all users or specific users
- Update user information
- Delete users

5.2 Restaurant Management

- Add new restaurants
- View all restaurants
- Update restaurant details
- Delete restaurants

5.3 Menu Management

- Add menu items for restaurants
- View menu items by restaurant
- Update menu item details
- Delete menu items

5.4 Order Management

- Create orders for customers
- View all orders or specific orders
- Update order status
- Delete orders

---

6. Database Schema

Users

Field	Description
id	Primary key
username	User login name
email	User email
role	ADMIN / RESTAURANT_STAFF / CUSTOMER

Restaurants

Field	Description
id	Primary key
name	Restaurant name
address	Restaurant address
contact_email	Contact email
contact_phone	Contact phone number

Menu_Items

Field	Description
id	Primary key
restaurant_id	Reference to restaurant
name	Menu item name
description	Menu item description
category	Food category (Starter, Main, Dessert)
price	Price of the item
availability_status	Available / Not Available

Orders

Field	Description
id	Primary key
customer_id	Reference to customer
restaurant_id	Reference to restaurant
order_date	Order date
status	Order status (Pending, Completed, Cancelled)
total_amount	Total order amount

Order_Items

Field	Description
id	Primary key
order_id	Reference to order
menu_item_id	Reference to menu item
quantity	Quantity ordered
price	Price of the item

---

7. User Stories *(Optional, if needed)*

- As an admin, I want to add, view, update, and delete users so that the system has correct user data.
- As a restaurant staff member, I want to manage menu items and orders to keep restaurant operations up-to-date.
- As a customer, I want to browse menus and place orders easily.

---

8. API Endpoints (CRUD)

Users

Method	Endpoint	Description
POST	/api/users	Create user
GET	/api/users	Get all users
PUT	/api/users/{id}	Update user
DELETE	/api/users/{id}	Delete user

Restaurants

Method	Endpoint	Description
POST	/api/restaurants	Add restaurant
GET	/api/restaurants	Get all restaurants
PUT	/api/restaurants/{id}	Update restaurant

Method	Endpoint	Description
DELETE	/api/restaurants/{id}	Delete restaurant

Menu Items

Method	Endpoint	Description
POST	/api/menu-items	Add menu item
GET	/api/menu-items	Get all menu items
PUT	/api/menu-items/{id}	Update menu item
DELETE	/api/menu-items/{id}	Delete menu item

Orders

Method	Endpoint	Description
POST	/api/orders	Create order
GET	/api/orders	Get all orders
PUT	/api/orders/{id}	Update order
DELETE	/api/orders/{id}	Delete order
