Restaurant Ordering System Assignment

Overview

In this assignment, you will develop a **QR code-based restaurant ordering system** that allows customers to place orders directly from their table using their mobile devices. The system will streamline the ordering process by sending orders directly to the kitchen staff through a **message queue system**, minimizing human interaction and reducing waiting times.

System Requirements

1. Core Functionality

1.1. Customer Interface

- **Table QR Code**
  - Assign a **unique identifier** to each table.
  - Generate a **QR code** that redirects to the ordering web application with the table ID.
- **Menu Display** [5 pts]
  - Display **restaurant menu items** with descriptions and prices.
  - Organize the menu into basic **categories** (e.g., appetizers, main courses, drinks).
- **Order Management** [20 pts]
  - Allow customers to **add/remove** items from the order.
  - Support **quantity selection** for each item.
  - Display the **running total cost** of the order.
  - Allow customers to **check the status** of their orders anytime they want.

1.2. Staff Interface

- **Authentication System** [20 pts]
  - Implement basic **login functionality** for kitchen staff, managers, waiters.

    - Establish **role differentiation** (e.g., kitchen staff vs. manager).

- **Kitchen Display System** [5 pts]
    - Show a **list of incoming orders** for kitchen staff.
    - Provide the ability to mark orders as **"In Preparation"** and **"Ready"**.
- **Waiter Notification** [5 pts]
    - Send a **notification** when orders are ready for delivery.
    - Allow waiters to mark orders as **“Delivered”**.

---

2. Technical Implementation

2.1. Backend Development

- **Spring Framework Application** [5 pts]
    - Develop **basic API endpoints** for the ordering system.
    - Implement **error handling** for robustness.
- **Database Design** [5 pts]
    - Create and implement a **simple database schema** (SQL or NoSQL). Note that you may use both types of databases for different purposes for this project.
    - Store all the necessary data in the databases, e.g., **menu items, orders, table information, etc**.
- **Message Queue Integration** [15 pts]
    - Implement some asynchronous messaging for order processing.
    - Set up a basic producer/consumer model. Use push model or pull model as you see fit for receiving orders in the kitchen.
    - Ensure **message reliability and delivery**.

2.2. Frontend Development

- **React Application** [5 pts]

    - Develop **basic components** for each view.
    - Implement **state management** for order processing.
    - Ensure **mobile responsiveness**.
- **API Integration** [10 pts]
    - Connect the **frontend app with backend endpoints**.
    - Handle **loading states and errors** effectively.

3. Additional Features

Testing

- Unit tests for backend services [5 pts]
- [**BONUS**] Integration tests for API endpoints [5 pts]
- [**BONUS**] Frontend component tests [5 pts]

Bonus

- Customizable menu items [5 pts]
- Implement Redis for improving
    - Check Status features and Order session management. [5 pts]
    - User session management for login/logout (JWT tokens can be whitelisted/backlisted) [5 pts]

Submission Guidelines

Submission details will be shared soon.