بسم الله الرحمن الرحيم

Restaurant Management System

Introduction :

Dear Developer, I hope you're doing well! Based on the following user story, you are tasked with building a database application for a **Restaurant Management System** using **Microsoft SQL Server Database**. This system will streamline restaurant operations by managing menus, orders, reservations, and staff efficiently.

Abstract :

Hello, my name is Ahmad. I own a restaurant and want to digitalize my operations to improve efficiency and customer satisfaction. The system should include:

- A comprehensive menu management feature.
- A streamlined order and billing process.
- A reservation system for customers.
- A staff management feature for roles, shifts, and attendance.

Additionally, I want to track restaurant performance by analyzing sales, order trends, and staff productivity.

User Story

1. ## Entities **and Features**:
    - **Menu**: Includes items with a name, category, price, availability, and optional discounts.
    - **Order**: Includes table number, customer name (optional), order date, status (e.g., pending, completed), and total amount.
    - **Reservation**: Includes customer name, contact details, number of guests, reservation date, and table number.
    - **Staff**: Includes employee name, role (e.g., chef, server), shift timings, attendance, and contact details.
    - **Admin Features**:
        - Manage menu items, categories, and discounts.
        - Oversee staff roles, shifts, and attendance.
        - Generate reports on sales, staff performance, and reservations.
    - **Customer Features**:
        - View the menu and place orders.
        - Make table reservations.
        - Provide feedback and rate their dining experience.
2. **Functionalities**:
    - Menu items can belong to multiple categories (e.g., "Vegan" and "Appetizers").
    - Orders may include multiple items with quantities.
    - Reservations should prevent table overbooking.

Database Queries

1. **Retrieve Menu by Category**:
    - Write a query to fetch all items under a specific category, including name, price, and availability.
2. **Daily Sales Report**:
    - Implement a query to calculate total sales, broken down by menu items and categories, for a specific date.
3. **Reservation Schedule**:
    - Create a query to list all reservations for a selected date, including customer details and table assignments.
4. **Staff Attendance**:
    - Write a query to fetch attendance records for all staff over a specific timeframe.

Database Views

1. **Available Menu Items View**:
   - Create a view to display all currently available menu items, including category and price.
2. **Daily Reservations View**:
   - Implement a view to show all reservations for the current day, including table numbers and customer contact details.
3. **Sales Summary View**:
   - Create a view summarizing sales by menu item and category, along with total revenue.

Stored Procedures

1. **Add New Order**:
   - Write a stored procedure to add a customer's order, including items, quantities, and calculating the total amount.
2. **Manage Reservations**:
   - Implement a stored procedure to add or update reservations, ensuring no table is double-booked.
3. **Generate Staff Report**:
   - Create a stored procedure to generate a report of staff attendance and performance metrics.
4. **Update Menu Availability**:
   - Write a stored procedure to update the availability status of menu items based on inventory or business decisions.

Implementation Requirements : Please Follow this instruction while developed this Project

1- Build ER – Diagram

2- Create Word Document to implement following

a- The scope of the project

b- The Mandatory object

c- The Software Type and the functionality follow in the system

d- Summarize the Objects

e- Summarize the Objects Relationships

3- Build Data base System using SQL Command and Follow the Normalization Pattern
4- Build the necessary DB configuration and constraints depends on the system questions
5- Fell Free To add any additional feature to the system
6- remember to add dummy data for testing purpose

Acceptance Criteria

1- Each Entity Must have a Primary Key and Make it's Identity

2- Ensure about applying check constraints and Default at least for 2 properties in each table
3- Configure the relation using foreign key constraints and use cascade

4- Submit SQL Scripts

5- Submit SQL Backup File (Data tier applications)

6- Submit Word Document

7- Submit ER Diagram

With The best Wishes From Your Trainer **Jasser Alshaer**

***fell free to ask questions about any thing**
*** remember you will be the best**