INSTITUTE OF SPACE TECHNOLOGY
KICSIT, Kahuta Campus

Project Proposal

Restaurant Management System

Group Members:

- Safa Waseem — 242201054
- Sania Shabbir — 242201055
- Misbah Saeed — 242201059

Submitted to: Sir Uzair

Department of Computer Science
KICSIT

June 3, 2025

1. Project Description

The **Restaurant Management System** is a console-based C++ application that simulates a restaurant's core operations. It is designed using object-oriented programming principles to model real-world restaurant activities such as taking orders, managing tables and employees, billing, and maintaining daily sales records.

This system allows a user (e.g., a waiter or manager) to handle customer interactions in an efficient and organized manner through a menu-driven interface. All operations are performed using in-memory data structures and some persistent storage via file handling.

2. Main Features

- **Customer and Employee Management:**
  - Input and store basic customer details (name, contact number)
  - Maintain employee records with unique ID, name, and role
- **Menu Management:**
  - Static menu consisting of food items (Burger, Pizza, Fries, Cola, etc.)
  - Each item has an ID, name, and price
- **Order Processing:**
  - Add multiple items to an order by selecting item IDs
  - Display a complete order summary
  - Store order records in a file named `orders.txt`
- **Table Management:**
  - Assign free tables to new customers automatically
  - Show current table occupancy status
  - Free tables manually when customers leave
- **Billing System:**
  - Calculate and display total amount due
  - Generate readable order summaries and bills
- **Sales Report:**
  - Accumulate daily sales from all orders
  - Display total sales for the day
  - Save total daily sales to `sales.txt`

3. Object-Oriented Design

Classes Used

- **Person (base class)**
  Derived: Customer, Employee
- **MenuItem, Menu**
- **Table**
- **Order**
- **Billing**
- **Report**
- **Restaurant (aggregates all functionalities)**

Key OOP Concepts

- **Inheritance:** For reusing the Person class in Customer and Employee
- **Polymorphism:** Virtual functions for dynamic binding
- **Encapsulation:** Data hiding through private attributes and public methods
- **Abstraction:** Providing a simplified interface while hiding complex implementation details

4. Tools and Libraries

- **Language:** C++
- **Compiler:** g++, Code::Blocks, or Dev C++
- **Libraries:**
  - `<iostream>` for input/output
  - `<fstream>` for file handling
  - `<string>` for string manipulation

5. Learning Objectives

- Practice advanced C++ features and object-oriented programming design
- Implement real-world logic into a manageable software project
- Understand and apply file handling for data persistence
- Strengthen skills in debugging, testing, and code documentation

6. Requirements

Functional Requirements

- The system should allow users to take and manage customer orders.
- It should manage employee data including roles and IDs.
- It must handle table allocation and deallocation.
- The system should generate bills and store order history in a file.

Non-Functional Requirements

- The application must be responsive and execute operations within a few seconds.
- It should be user-friendly and menu-driven for easy navigation.
- The data stored in files must be consistent and accurate after each operation.
- The system must be maintainable and modular for future enhancements.

8. Conclusion

The Restaurant Management System project serves as a practical application of object-oriented programming concepts in C++. It not only enhances the understanding of class design, inheritance, and file handling, but also simulates real-world restaurant operations effectively. Through this project, we aim to build a robust and user-friendly system that helps manage orders, staff, tables, and sales efficiently. The project reinforces good software engineering practices and lays the groundwork for future enhancements such as GUI integration or database connectivity.