**Project Title**

**Cafeteria Management System**

# **Group Members:**

1. **Ayesha Tahir**

ID: F24CSC021

1. **Manal Lodhi**

ID: F24BSE012

1. **Bakhtawar Khan**

ID: F24CSC033

**Semester: First Semester**

**Department: Computer Science**

**University: Salim Habib University**

**Submission date : 28-January-2025**

**Programming Fundamentals**

**Instructor: Sir Mansoor**

## **Overview**

The BAM Cafe Management System is a C++ application designed to streamline basic cafe operations, allowing users to:

1. View a predefined menu.
2. Place orders and calculate the total cost.
3. Save order details to a file.
4. Play interactive games while waiting for their order, such as solving riddles and guessing teachers based on hints.

This program provides a user-friendly interface for cafe management and enhances customer engagement through interactive activities.

**Purpose of choosing Cafeteria Management System as a project**

The primary purpose of this project is to create a user-friendly interface for placing food orders and engaging in fun activities. It emphasizes the relevance of programming fundamentals by applying various programming concepts, including:

**Structures:** Using structs to manage menu items and user information.

**File handling:** Saving orders and user data to files for persistence.

**Control flow**: Implementing loops and conditionals for flow control.

**Functions**: Organizing code into reusable functions for clarity and maintainability.

**Objectives of the BAM Cafe Ordering System**

The BAM Cafe Ordering System was developed with clear goals in mind, focusing on both user engagement and educational value. Here are the primary objectives of the project:

**Enhance User Experience:** The program aims to provide a seamless and enjoyable experience for users by allowing them to easily place food orders and engage in interactive games. This combination of functionality and entertainment is designed to keep users engaged while they interact with the cafe.

**Demonstrate Programming Fundamentals**: This project serves as a practical application of core programming concepts such as data structures, file handling, input validation, and control flow. It provides a hands-on opportunity for users to understand how these fundamentals come together to create a functional software application.

## **System Requirements**

To run this program, you'll need:

* A C++ IDE (Integrated Development Environment) such as: Code block, Visual Studio
* Required files:
  + riddles.txt (contains riddles and their answers, separated by '|')
  + teachers.txt (contains teacher hints and their answers)
  + orders.txt (optional: generated when placing orders)

### **Code Explanation:**

### **Key Components**

#### **Structures**

1. **Menu Item**:
   * Represents an item on the cafe menu with two attributes: name (string) and price (double).
2. **User:**
   * Stores customer details with attributes: name, email, phone, and address.

#### **Functions**

1. **displayMenu:**
   * Displays the menu stored in a MenuItem array.
   * Takes no parameters and returns void.
2. **placeOrder:**
   * Allows the user to select menu items by item number.
   * Calculates the total cost and saves the order in orders.txt.
3. **Validation Functions**:
   * isValidName: Ensures the name contains only alphabets and spaces.
   * isValidEmail: Ensures the email ends with @gmail.com.
   * isValidPhone: Validates the phone number contains only digits.

These functions return a bool value indicating whether the input is valid.

1. **collectUserInfo:**
   * Collects user details, validates them, and returns a User object.
2. **Riddles Game**:
   * **loadRiddles**:
     + Reads riddles and answers from riddles.txt.
     + Returns a vector of riddle-answer pairs.
   * **playRiddlesGame**:
     + Selects a random riddle and prompts the user for an answer.
     + Displays whether the user's answer is correct.
3. **Guess the Teacher Game**:
   * **loadTeachers**:
     + Reads hints and answers from teachers.txt.
     + Returns a vector of hint-answer pairs.
   * **playTeachersGame**:
     + Selects a random hint and prompts the user to guess the teacher.
     + Displays whether the user's guess is correct.
4. **main Function:**
   * Displays the menu.
   * Allows users to place an order, play games, or exit the program.
   * Handles user choices and provides a loop for continuous operation.

#### **How the Program Works**

### **1. Menu Display**

* At the start of the program, the menu is displayed.
* Users can view available items and their prices.

### **2. Placing Orders**

* Users enter the item numbers to place their order.
* The program validates input, calculates the total price, and prompts for customer details.
* Orders and customer information are saved to orders.txt.

### **3. Playing Games**

* Users can choose to play one of two games:
  + **Riddles Game**: Solve a randomly selected riddle.
  + **Guess the Teacher Game**: Guess the teacher based on a hint.

### **4. Exiting**

* The program exits when the user selects option 3(Exit).

# **Program Limitations**

* Input validation is specific to basic formats (e.g., only @gmail.com emails are allowed).
* Limited scalability for menu updates (requires manual code changes).
* Relies on correct formatting of riddles.txt and teachers.txt.

**Result**

The BAM Cafe Ordering System successfully achieves the following:

**User Engagement**: It allows users to easily browse a menu, place orders, and engage in interactive games, creating an enjoyable experience.

**Data Management**: The program effectively collects user information and saves orders to a file, ensuring that data is organized and retrievable for future reference.

**Educational Value:** By integrating programming fundamentals such as data structures, file handling, and input validation, the project serves as a practical learning tool for those new to programming.

**Conclusion**

Working on the BAM Cafe Ordering System provided valuable insights into software development. We learned how to effectively implement user input validation, manage data through file operations, and create a user-friendly interface. This project reinforced my understanding of core programming concepts while allowing us to apply them in a real-world scenario. Overall, it was a rewarding experience that enhanced both our technical skills and creativity in problem-solving.