**Project Title**

**Cafeteria Management System**

# **Group Members:**

* **Ayesha Tahir**

ID: F24CSC021

* **Manal Lodhi**

ID: F24BSE012

* **Bakhtawar Khan**

ID: F24CSC033

**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. You can view a predefined menu.
2. Place orders and calculate the total cost.
3. Save order details to a file.
4. Play interactive games 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.

**Features:**

1. **Order Management:**
   * View a list of delicious items available for order.
   * Select items from the menu and save your order to a file.
   * Collect and save customer details securely.
2. **Fun Games Option:**
   * **Riddles Game:** Test your wits with a collection of riddles.
   * **Guess the Teacher Game:** Guess the teacher based on hints.
3. **File-Based Storage:**
   * All orders and customer details are saved to a text file for record keeping.
4. **User-Friendly Input Validation:**
   * It ensures proper names, email addresses, and phone numbers are entered.
5. **Order Confirmation:**
   * Displays the ordered items, total price, and estimated delivery time.
6. **Exception Handling:**
   * Ensures the program runs smoothly by handling invalid inputs and file errors.

## **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.