Project title

Restaurant Management System

Submitted to:

Nusrat Tasnim

Designation: Lecturer

Submitted By:

1. **Tasve Al Samir (232-35-315)**
2. **Samia Malik (232-35-316)**
3. **Supti Saha (232-35-729)**

Semester: Fall 2024

Batch: 41

Section: A1

Course Code: SE133

Course Name: Software Development Capstone Project

Submission Date: 14/12/2024

Restaurant Management System

Table of Contents

Introduction

A restaurant management system (RMS) is a software system that helps restaurants run more efficiently by streamlining operations, tracking data, and simplifying decision-making.

Overview

This C program is designed for managing restaurant operations such as menus, orders, employee records, and customer purchases. It supports functionalities for administrators, employees, and customers with various services, including food ordering, employee attendance, and purchase history management.

The Header Files we have use to make this project:

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdbool.h>
#include <stdlib.h>
#include <windows.h>
#include <time.h>
#include <math.h>
```

The Functions we have use to make this project:

```
void adminlogin();
void admin();
void employee();
void customer();
void adminservices();
```

```
void employeeservices();
void phnnumber();
void searchbyid();
void searchbyrole();
void employee_details();
void entrytime();
void exittime();
void displaymenuadminlogin();
void displaymenucustomer();
void display_The_bengali_menu();
void display_The_american_menu();
void display_The_english_menu();
void display_The_indian_menu();
void display_The_middle_eastern_menu();

void insertMenu();
void insert_The_bengali_menu();
void insert_The_american_menu();
void insert_The_english_menu();
void insert_The_indian_menu();
void insert_The_middle_eastern_menu();
void deleteItem();
void delete_The_bengali_menu();
void delete_The_american_menu();
void delete_The_english_menu();
void delete_The_indian_menu();
void delete_The_middle_eastern_menu();
void updateprice();
void update_The_bengali_menu();
void update_The_american_menu();
void update_The_english_menu();
void update_The_indian_menu();
void update_The_middle_eastern_menu();
void orderfood();
void selectitem(char menu);
void viewcart();
void cash_or_card();
void process_cash();
void process_card();
void customerpurchasehistory();
void alltimehistory();
void specificdatehistory();
void searchbyword();
void moveCursorRelative(int dx, int dy);
void setColor(int color);
void reset();
```

Key Functionalities

1. Administrator Services

- **Login/Logout**
  - Secure login using ID and password.
- **Menu Management**
  - View, add, delete, and update menu items for various cuisines:
    - Bengali
    - Indian
    - Middle-Eastern
    - English
    - American
- **Employee Management**
  - Search by ID or position.
  - View attendance for specific dates.
- **Customer Purchase History**
  - View all-time or specific-date purchase records.
- **View customers purchase history.**

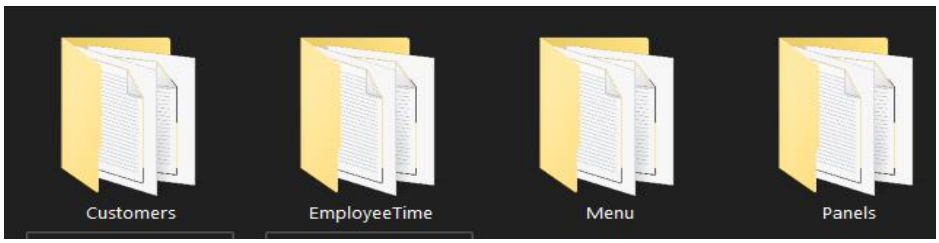
2. Employee Services

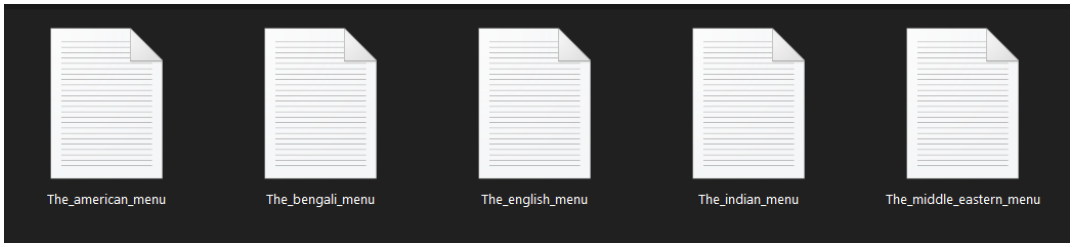
- Mark entry and exit times.
- Manage attendance records linked to specific dates.

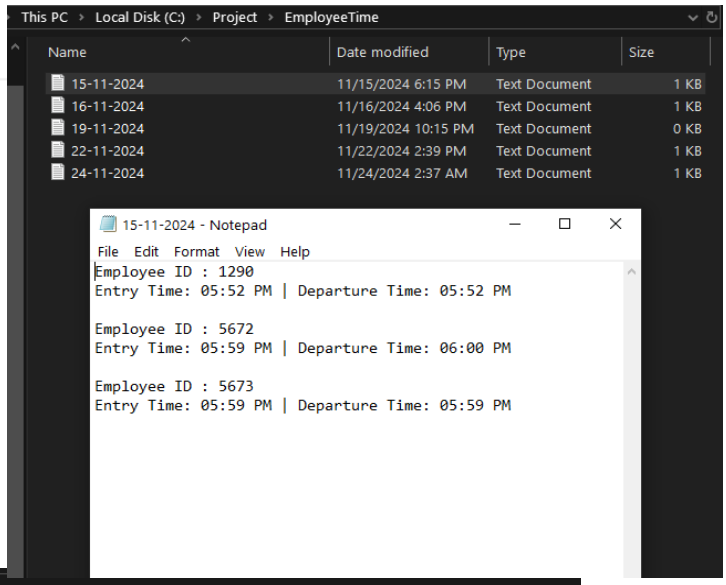
3. Customer Services

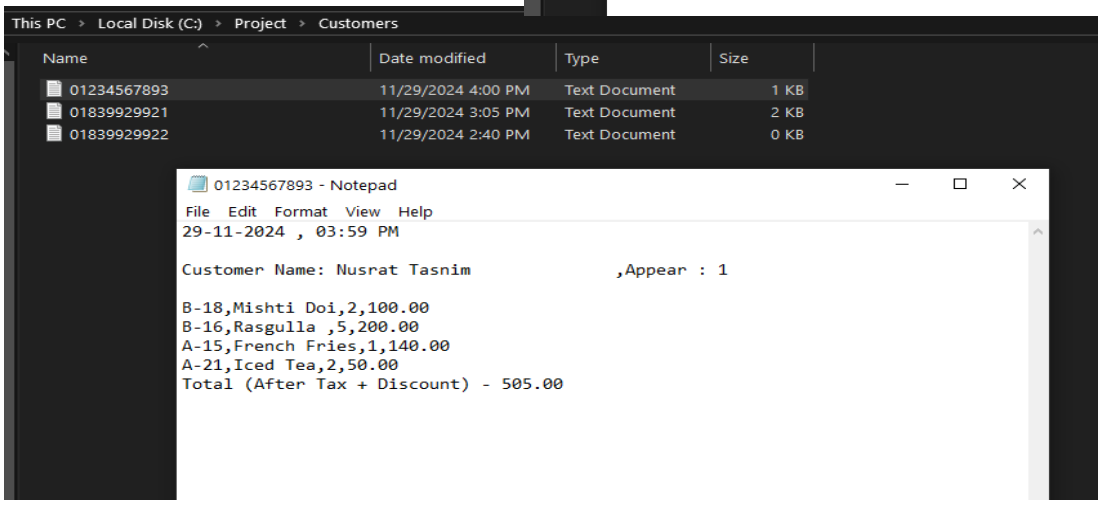
- Browse menus by cuisine.
- Order food and view the cart.
- Search for items by name.

The Files are:





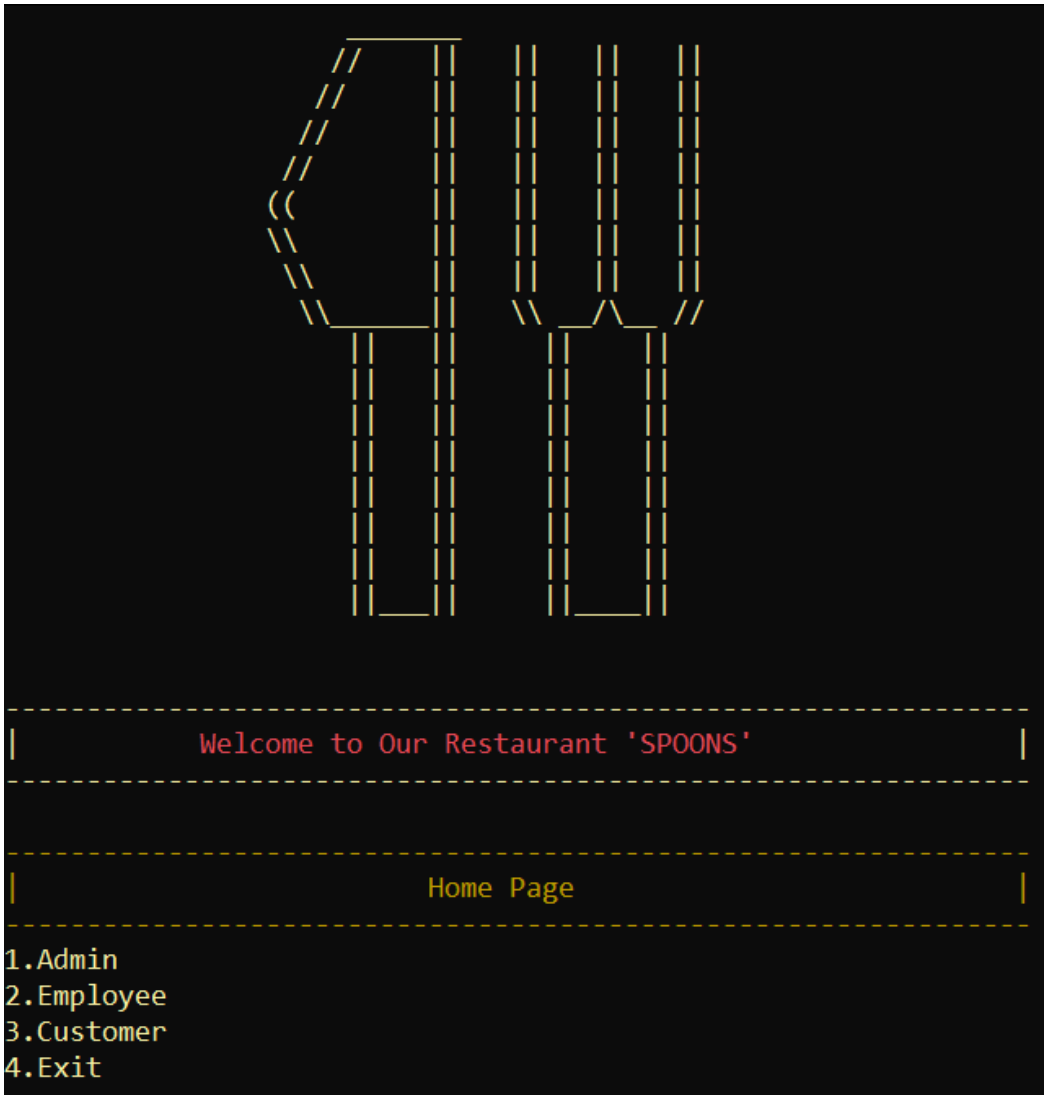




Services

1. The Homepage:

At first, we draw a knife and a fork as our logo. Because our restaurant name is “SPOONS”.
On the homepage, we include 4 options:
1.Admin
2.Employee
3.Customer
4.Exit



2. Admin Services:

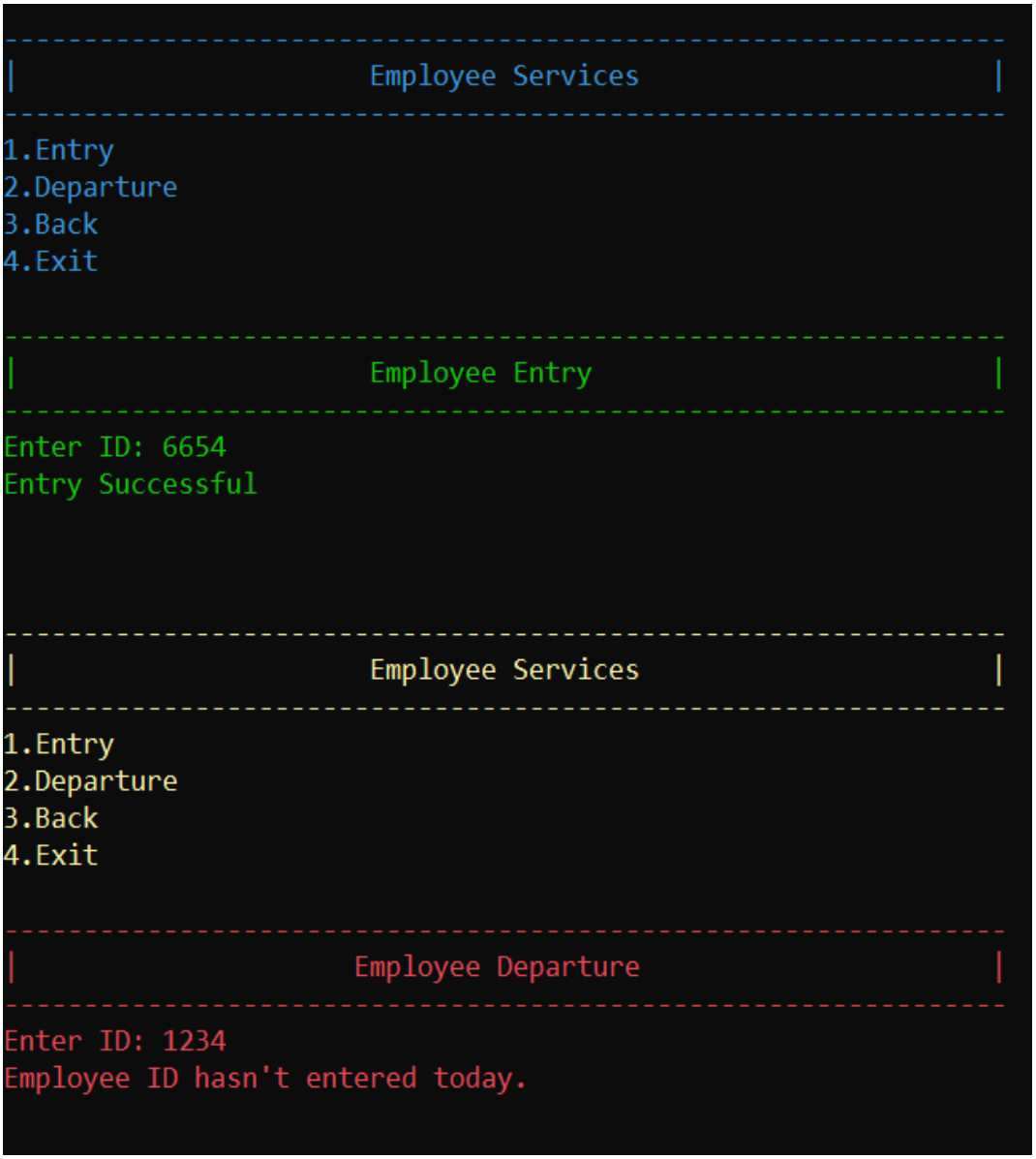
The admin services are allowed only for admins of this system.

It provides:

3. Employee Services:

This function tracks employee details. Each employee has a unique ID written in the file. Also, this information will be written on the file in real-time.

The Employee services provide:



File Structure

- **Header Files**: Standard libraries and additional headers (windows.h, time.h, math.h) for functionalities like file handling, text formatting, and system operations.
- **Files Used**:
  - Menu files (The_bengali_menu.txt, etc.)
  - Employee and customer files (e.g., employee.txt, Customers/<phone_number>.txt).

Core Data Structures

1. Product

- Represents a menu item.
- Fields: productId, productName, productQuantity, productPrice.

2. Admin

- Stores administrator credentials.
- Fields: id, password.

3. Cartitem

- Represents items in the customer's cart.
- Fields: num, productName, quantity, price.

4. Searchproduct

- Stores search results for products.
- Fields: p_id, p_name, p_price.

```
struct Product
{
    long productId;
    char productName[100];
    int productQuantity;
    float productPrice;
};
struct Product product[100];

struct Admin
{
    int id;
    char password[50];
};
```

```
struct cartitem
{
    int num;
    char productName[50];
    int quantity;
    float price;
};
struct cartitem cartproducts[100];

struct Searchproduct
{
    int p_id;
    char p_name[50];
    float p_price;
};
struct Searchproduct search[100];
```

Functions:

1. General

- setColor(int color) and reset(): Manage text color for output.
- moveCursorRelative(int dx, int dy): Adjust cursor position in the console.

2. Administrator Functions

- admin(), adminlogin(): Handle administrator actions.
- Menu management:
  - display_The_bengali_menu(), insert_The_bengali_menu(), delete_The_bengali_menu(), update_The_bengali_menu() etc.
- adminservices(): Navigate admin-specific operations.
- updatepass(): Change admin password.
- searchbyid(), searchbyrole(): Search employees by ID or role.
- customerpurchasehistory(), specificdatehistory(): Manage customer purchase records.

```
char choice;
choice = _getch();
switch (choice)
{
case '1':

    display_The_bengali_menu();
    break;
case '2':

    display_The_indian_menu();
    break;
case '3':

    display_The_middle_eastern_menu();
    break;
case '4':

    display_The_english_menu();
    break;
case '5':

    display_The_american_menu();
    break;
```

```
switch (choice)
{
case '1':

    insert_The_bengali_menu();
    break;
case '2':

    insert_The_indian_menu();
    break;
case '3':

    insert_The_middle_eastern_menu();
    break;
case '4':

    insert_The_english_menu();
    break;
case '5':

    insert_The_american_menu();
    break;
case '6':

    adminservices();
```

```
case '1':

    delete_The_bengali_menu();
    break;
case '2':

    delete_The_indian_menu();
    break;
case '3':

    delete_The_middle_eastern_menu();
    break;
case '4':

    delete_The_english_menu();
    break;
case '5':

    delete_The_american_menu();
    break;
```

```
case '1':

    update_The_bengali_menu();
    break;
case '2':

    update_The_indian_menu();
    break;
case '3':

    update_The_middle_eastern_menu();
    break;
case '4':

    update_The_english_menu();
    break;
case '5':

    update_The_american_menu();
    break;
```

3. Employee Functions:

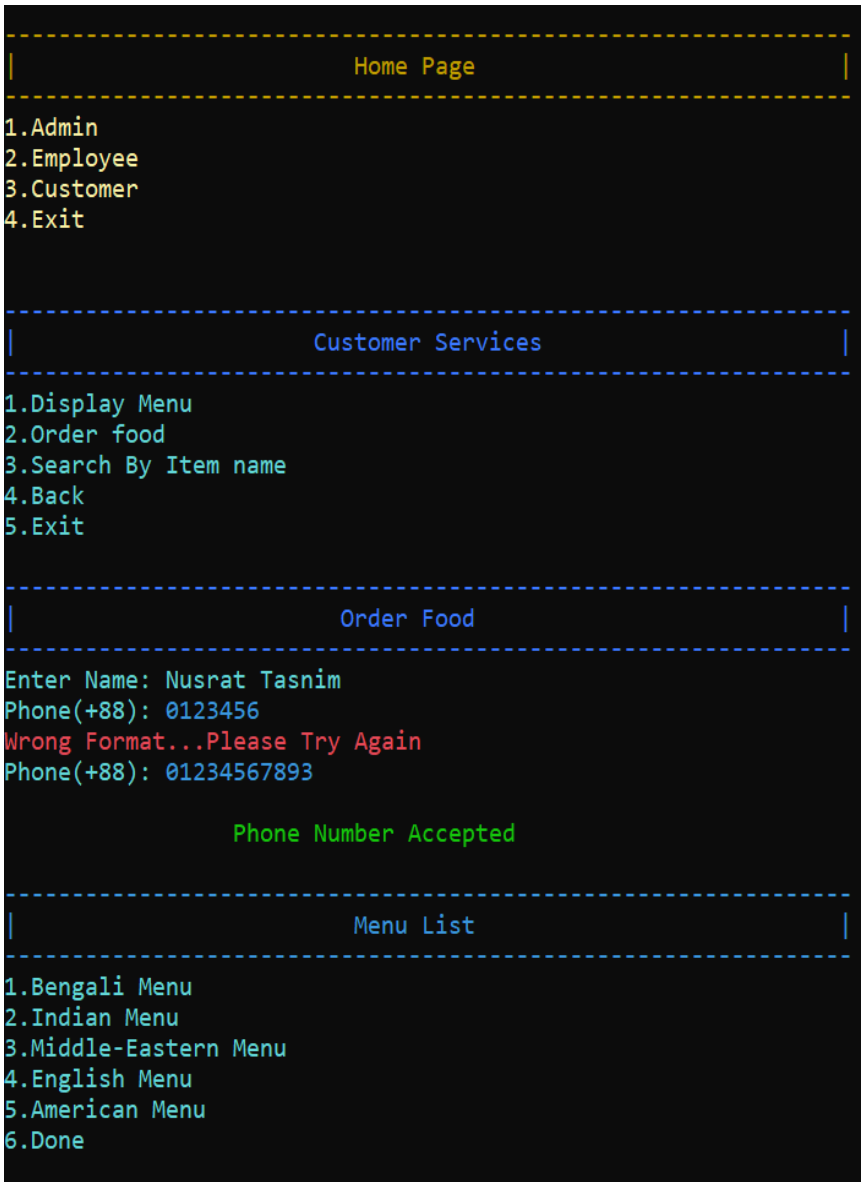
- employeeservices(): Manage employee actions (entry, exit).
- entrytime(), exittime(): Record attendance.

```
void employeeservices()
{
    int day, month, year, hour, min;
    char gmt[3];
    char buffer[200];
    time_t currentTime = time(NULL);
    struct tm *localTime = localtime(&currentTime);
    strftime(buffer, sizeof(buffer), "Entry Time: %d-%m-%Y , %I:%M %p\n", localTime);
    if (sscanf(buffer, "Entry Time: %d-%d-%d , %d:%d %s", &day, &month, &year, &hour, &min, gmt))
    {
        sprintf(store, "C:\\Project\\EmployeeTime\\%d-%d-%d.txt", day, month, year);
    }

    while (1)
    {
        setColor(9);
        printf("\n\n-------------------------------------------------------------------\n");
        printf("|                      Employee Services                          |\n");
        printf("-------------------------------------------------------------------\n");
        setColor(3);
        printf("1.Entry\n2.Departure\n3.Back\n4.Exit\n\n");
        char ch = _getch();
        switch (ch)
        {
        case '1':
            check = fopen(store, "a");
            fclose(check);
            entrytime();
            break;
        case '2':

            exittime();
            break;
```

4. Customer Functions:

- orderfood(): Allows customers to place orders.
- viewcart(): View items added to the cart.

How to order food from this system



```
-------------------------------------------------------------
|                      Bengali Food                         |
-------------------------------------------------------------
|  ID  |             Item Name             |       Price     |
=============================================================
|  20  |  Milk Cha(Tea)                    |      20.00      |
-------------------------------------------------------------

               1.Continue      |       0.Stop

-------------------------------------------------------------
|                      Add to List                          |
-------------------------------------------------------------
|  23  |  Fruit Juice                      |      60.00      |
-------------------------------------------------------------
|  24  |  Hot-Dog                          |      60.00      |
-------------------------------------------------------------

               1.Continue      |       0.Stop

-------------------------------------------------------------
|                      Add to List                          |
-------------------------------------------------------------
| ID |            Item Name          | Quantity |    Price   |
-------------------------------------------------------------
| 15 |  French Fries                 |     1    |  140.00    |
-------------------------------------------------------------
| 21 |  Iced Tea                     |     2    |  50.00     |
-------------------------------------------------------------
```

```
                      Customer Details

Customer Name: Nusrat Tasnim              Phone No: 01234567893

-------------------------------------------------------------
|                        View Cart                          |
-------------------------------------------------------------
| ID |            Item Name          | Quantity |    Price   |
-------------------------------------------------------------
| 01 | Mishti Doi                    |     2    |  100.00    |
-------------------------------------------------------------
| 02 | Rasgulla                      |     5    |  200.00    |
-------------------------------------------------------------
| 03 | French Fries                  |     1    |  140.00    |
-------------------------------------------------------------
| 04 | Iced Tea                      |     2    |  50.00     |
-------------------------------------------------------------
|                                  Base Total = 490.00      |
|                                      Tax(+) = 15.00       |
|                                       Total = 505.00      |
-------------------------------------------------------------


-------------------------------------------------------------
|                     Payment Options                       |
-------------------------------------------------------------
1.Cash
2.Card

-------------------------------------------------------------
|                          Cash                             |
-------------------------------------------------------------

Enter the amount of cash given: 1000
Change: 495.00

                  Transaction successful!

        Thanks for your purchase.....Come back soon.
```

How to Use:

Run the program: The main menu provides options for administrator, employee, and customer services.

Navigate using numeric inputs:

- **Admin**: Perform operations like menu management, employee, and customer record handling.
- **Employee**: Record entry and exit times.
- **Customer**: Browse menus, place orders, and check purchase history.

Summary of Functionality:

1. **User Roles:**
   - **Admin:** Can manage menus, update passwords, and view employee or customer details.
   - **Employee:** Can log entry and exit times.
   - **Customer:** Can browse menus, order food & search by item name.
2. **Menu Management:**
   - Separate menus for Bengali, Indian, Middle Eastern, English, and American cuisines.
   - Admin can view, add, delete, and update menu items for each cuisine.
3. **Order Management:**
   - Customers can place orders by selecting menu items and adding them to a cart.
   - Supports calculation of total, tax, and discounts based on customer loyalty.

4. **Employee Management:**
    - Logging entry and exit times for employees.
    - Searching employees by ID or role.
5. **Purchase History:**
    - Tracks purchase history for customers by phone number.
    - Allows searching for specific dates or viewing all-time history.
6. **File-Based Data Handling:**
    - Uses files to store data such as menu items, employee records, and purchase histories.
7. **Console Features:**
    - Use of color codes for better visual representation.
    - Simple animations and formatted displays.

Conclusions:

1. **Comprehensive System:**

   The program covers essential aspects of restaurant management, making it suitable for small or medium-sized businesses.

2. **File-Based Data Persistence:**

   The reliance on file-based storage may limit scalability and performance for larger operations.

3. **Security Concerns:**
    - Admin are selected and fixed for the code.
    - No user authentication beyond ID/password matching.

Project:

https://drive.google.com/drive/folders/1k6ENm3DoFf8zRGpetXQgTTA9dyzNMets?usp=drive_link