**Restaurant Management System**

**Table of Contents**

**1.**[**Introduction 2**](#_Toc183807738)

**2.**[**Overview 2**](#_Toc183807739)

**3.**[**The Header Files we have use to make this project: 2**](#_Toc183807740)

**4.**[**The Functions we have use to make this project: 2**](#_Toc183807741)**-3**

**5.**[**Key Functionalities 4**](#_Toc183807742)

[**a. Administrator Services 4**](#_Toc183807743)

[**b. Employee Services 4**](#_Toc183807744)

[**c. Customer Services 4**](#_Toc183807745)

**6.**[**The Files are: 5**](#_Toc183807746)

**7.**[**Services 6**](#_Toc183807747)

**a.** [**The Homepage: 6**](#_Toc183807748)

**b.** [**Admin Services: 7**](#_Toc183807749)

**c.** [**Employee Services: 8**](#_Toc183807750)

**8.**[**File Structure 9**](#_Toc183807751)

**9.**[**Core Data Structures 9**](#_Toc183807752)

[**a. Product 9**](#_Toc183807753)

[**b. Admin 9**](#_Toc183807754)

[**c. Cartitem 9**](#_Toc183807755)

[**d. Searchproduct 9**](#_Toc183807756)

**10.**[**Functions: 10**](#_Toc183807757)

[**a. General 10**](#_Toc183807758)

[**b. Administrator Functions 10**](#_Toc183807759)

[**c. Employee Functions: 11**](#_Toc183807760)

[**d. Customer Functions: 12**](#_Toc183807761)

**11.**[**How to order food from this system 13**](#_Toc183807762)

**12.**[**How to Use: 15**](#_Toc183807763)

**13.**[**Summary of Functionality: 15**](#_Toc183807764)

**14.**[**Conclusions: 16**](#_Toc183807765)

**15.Project link:…………...………………………………………………………………………………17**

# 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

## 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

## Admin Services:

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

It provides:

## 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.
* A screenshot of a computer program

  Description automatically generatedFields: p\_id, p\_name, p\_price.

# Functions:

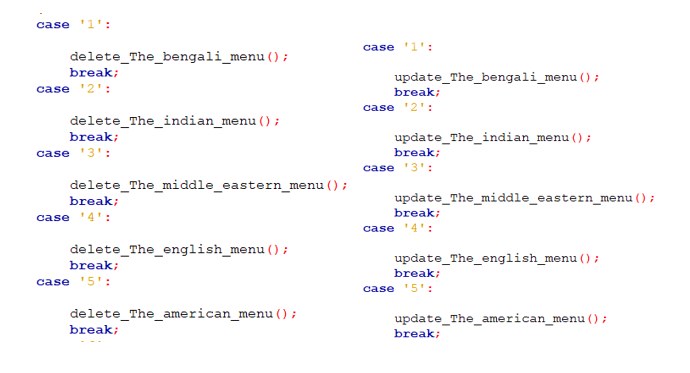
**1. General**

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

**2. Administrator Functions**

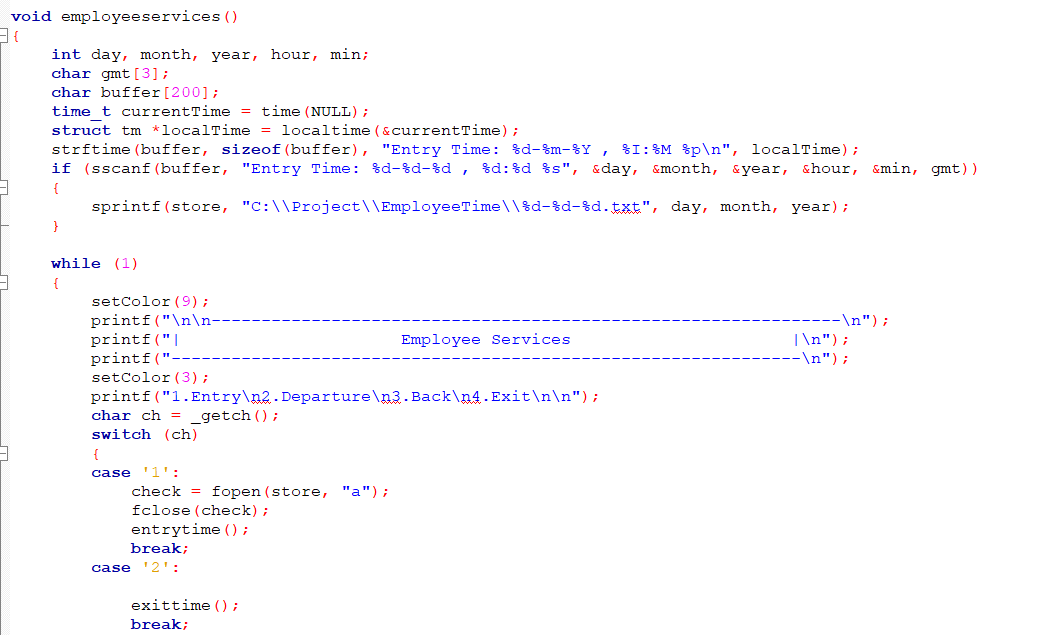
* admin(), adminlogin(): Handle administrator actions.
* Menu management:
  + display\_The\_bengali\_menu(), insert\_The\_bengali\_menu(), delete\_The\_bengali\_menu(), update\_The\_bengali\_menu() etc.
* adminservices(): Navigate admin-specific operations.
* updatepass(): Change admin password.
* searchbyid(), searchbyrole(): Search employees by ID or role.
* customerpurchasehistory(), specificdatehistory(): Manage customer purchase records.





**3. Employee Functions:**

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



**4. Customer Functions:**

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

# How to order food from this system

# 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.

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

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

1. **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**](https://drive.google.com/drive/folders/1k6ENm3DoFf8zRGpetXQgTTA9dyzNMets?usp=drive_link)