SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be **University** *u/s 3 of UGC Act, 1956)*

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

FACULTY OF ENGINEERING AND TECHNOLOGY

18CSS101J- PROGRAMMING FOR PROBLEM SOLVING

MINI PROJECT REPORT

PROJECT TITTLE -

EMPLOYEE RECORD MANAGEMENT SYSTEM

Submitted by:

Name & Reg. No - Anjali Jha & RA2111003011873

Name & Reg. No - Sanjana Kumari & RA2111003011864

Branch: CSE - CORE

Section: Z2

TABLE OF CONTENTS

Title **Page Number**

ABSTRACT

The project employee management system using C manages employee records using the file system. this help to register employee, display all list of an employee on the screen, search a particular employee record, modify and delete information of an employee. The best part of this project is it uses graphics to more look like real software designed in other higher-level programming languages.

PROBLEM STATEMENT

In this C Project we are going to cover create Employee Management System with C language. With this Employee Management System, we can manage the details of the employees working in a company or organization. Insert, Edit, and Delete file operations are used in this project, but the limitation of this project is that you can only list the data but cannot search for any data item in particular.

DESCRIPTION:

A Employee's Management System (EMS) is a software built to handle the primary housekeeping functions of a company. EMS help companies keep track of all the employees and their records. It is used to manage the company using computerized system.

Aim of Employee's Management System:

- ❖Built The Employee Table.
- ❖Insert New Entries.
- ❖Delete The Entries.
- ❖Search A Record.

OBJECTIVE:

The main objective of the system is to computerize the maintenance of the employee details and salary section in the company. Insert, Edit, and Delete file operations are used in this project, but the limitation of this project is that you can only list the data but cannot search for any data item in particular. You can include the searching techniques in c to modify this program if you have better knowledge.

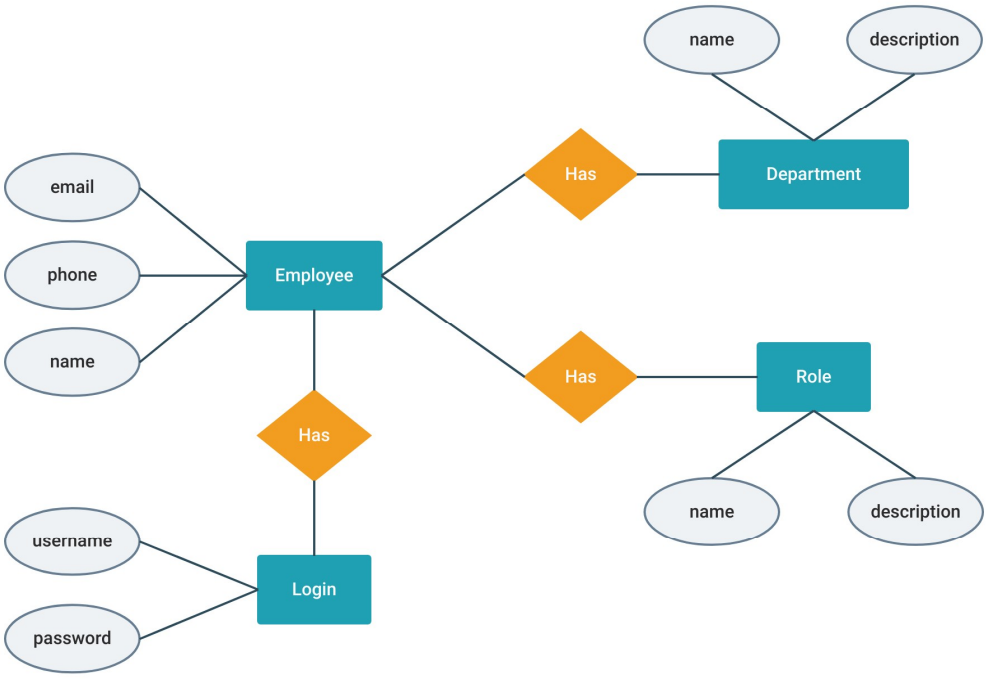
ALGORITHM:

For storing the data of the employee, create a user define datatype which will store the information regarding Employee.

For building the employee table the idea is to use the array of the above struct datatype which will use to store the information regarding employee.

Since we are using array to store the data, therefore to delete the data at any index shift all the data at that index by 1 and delete the last data of the array by decreasing the size of array by 1.

FLOWCHART:


name
description
Has
Department
email
phone
name
Employee
Has
Role
Has
name
description
username
Login
password

CODE:

```c
 #include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <windows.h>

// Structure of the employee
struct emp {
    char name[50];
    float salary;
    int age;
    int id;
};
struct emp e;

// size of the structure
long int size = sizeof(e);

// In the start coordinates
// will be 0, 0
COORD cord = { 0, 0 };

// function to set the
// coordinates
void gotoxy(int x, int y)
{
    cord.X = x;
    cord.Y = y;
    SetConsoleCursorPosition(
        GetStdHandle(STD_OUTPUT_HANDLE),
        cord);
}

FILE *fp, *ft;

// Function to add the records
void addrecord()
{
    system("cls");
    fseek(fp, 0, SEEK_END);
    char another = 'y';

    while (another == 'y') {
        printf("\nEnter Name : ");
        scanf("%s", e.name);

        printf("\nEnter Age : ");
```

```
        scanf("%d", &e.age);

        printf("\nEnter Salary : ");
        scanf("%f", &e.salary);

        printf("\nEnter EMP-ID : ");
        scanf("%d", &e.id);

        fwrite(&e, size, 1, fp);

        printf("\nWant to add another"
            " record (Y/N) : ");
        fflush(stdin);

        scanf("%c", &another);
    }
}

// Function to delete the records
void deleterecord()
{
    system("cls");
    char empname[50];
    char another = 'y';

    while (another == 'y') {
        printf("\nEnter employee "
            "name to delete : ");
        scanf("%s", empname);

        ft = fopen("temp.txt", "wb");
        rewind(fp);

        while (fread(&e, size,
                     1, fp)
            == 1) {
            if (strcmp(e.name,
                       empname)
                != 0)
                fwrite(&e, size, 1, ft);
        }

        fclose(fp);
        fclose(ft);
        remove("data.txt");
        rename("temp.txt", "data.txt");
        fp = fopen("data.txt", "rb+");
```

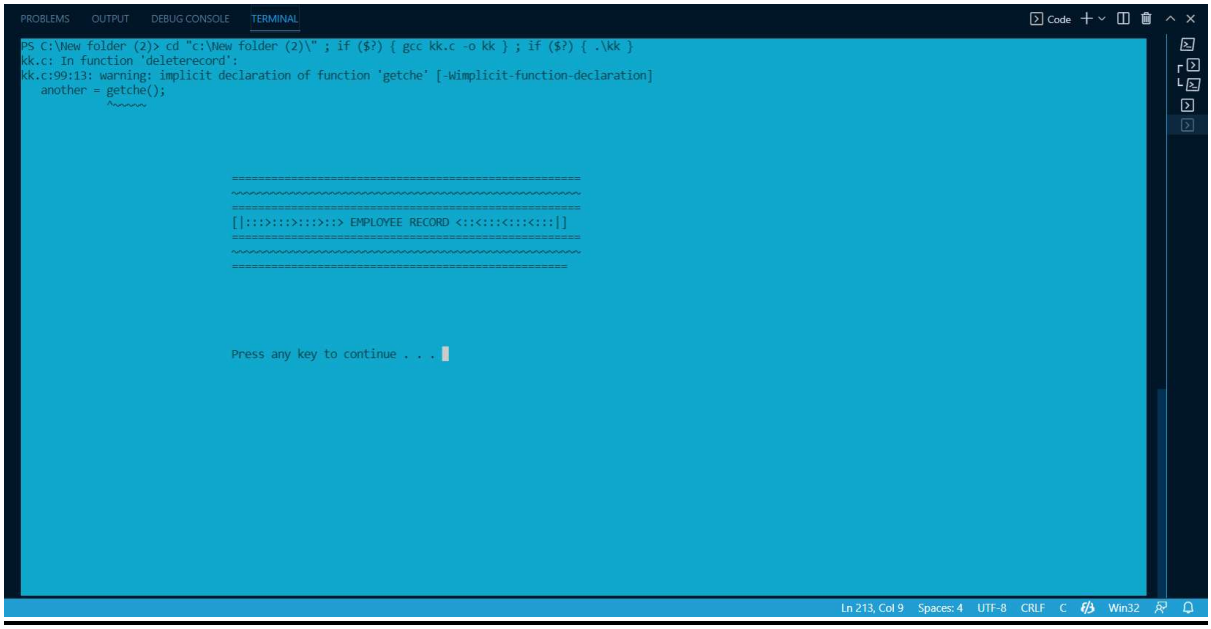
```
        printf("\nWant to delete another"
            " record (Y/N) :");
        fflush(stdin);
        another = getche();
    }
}

// Function to display the record
void displayrecord()
{
    system("cls");

    // sets pointer to start
    // of the file
    rewind(fp);

    printf("\n========================="
        "============================"
        "======");
    printf("\nNAME\t\tAGE\t\tSALARY\t\t"
        "\tID\n",
        e.name, e.age,
        e.salary, e.id);
    printf("============================"
        "============================"
        "====\n");

    while (fread(&e, size, 1, fp) == 1)
        printf("\n%s\t\t%d\t\t%.2f\t%10d",
            e.name, e.age, e.salary, e.id);

    printf("\n\n\n\t");
    system("pause");
}

// Function to modify the record
void modifyrecord()
{
    system("cls");
    char empname[50];
    char another = 'y';

    while (another == 'y') {
        printf("\nEnter employee name"
            " to modify : ");
        scanf("%s", empname);

        rewind(fp);
```

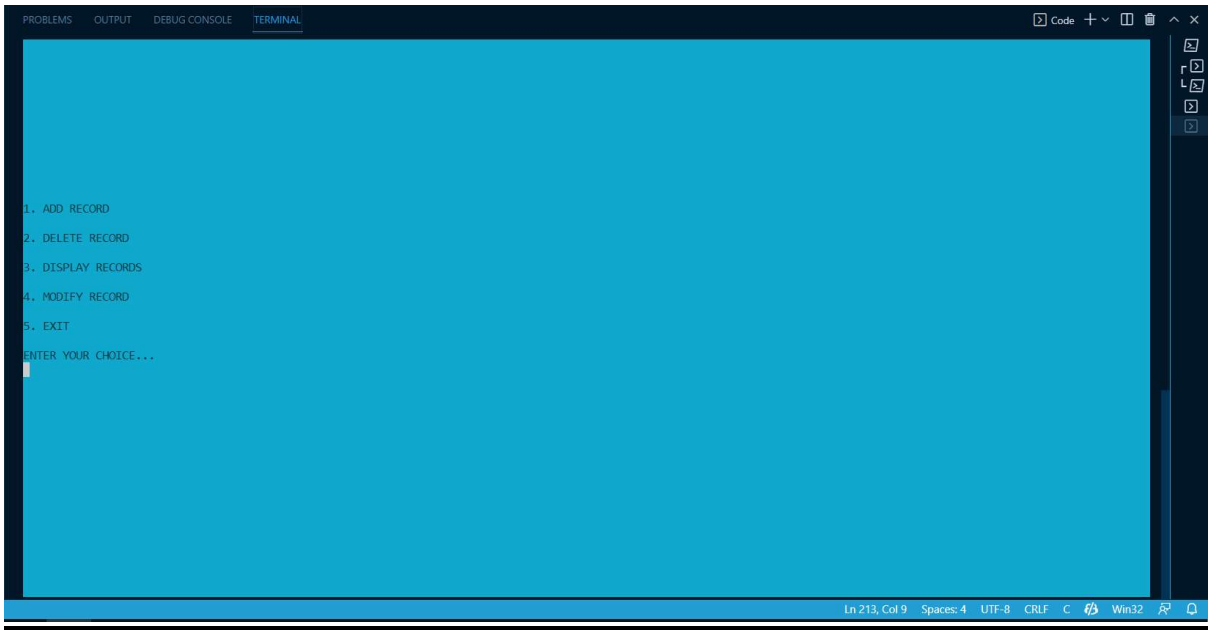
```
        // While File is open
        while (fread(&e, size, 1, fp) == 1) {
            // Compare the employee name
            // with ename
            if (strcmp(e.name, empname) == 0) {
                printf("\nEnter new name:");
                scanf("%s", e.name);
                printf("\nEnter new age :");
                scanf("%d", &e.age);
                printf("\nEnter new salary :");
                scanf("%f", &e.salary);
                printf("\nEnter new EMP-ID :");
                scanf("%d", &e.id);

                fseek(fp, -size, SEEK_CUR);
                fwrite(&e, size, 1, fp);
                break;
            }
        }

        // Ask for modifying another record
        printf("\nWant to modify another"
            " record (Y/N) :");
        fflush(stdin);
        scanf("%c", &another);
    }
}

// Driver code
int main()
{
    int choice;

    // opening the file
    fp = fopen("data.txt", "rb+");

    // showing error if file is
    // unable to open.
    if (fp == NULL) {
        fp = fopen("data.txt", "wb+");
        if (fp == NULL) {
            printf("\nCannot open file...");
            exit(1);
        }
    }

    system("Color 3F");
```

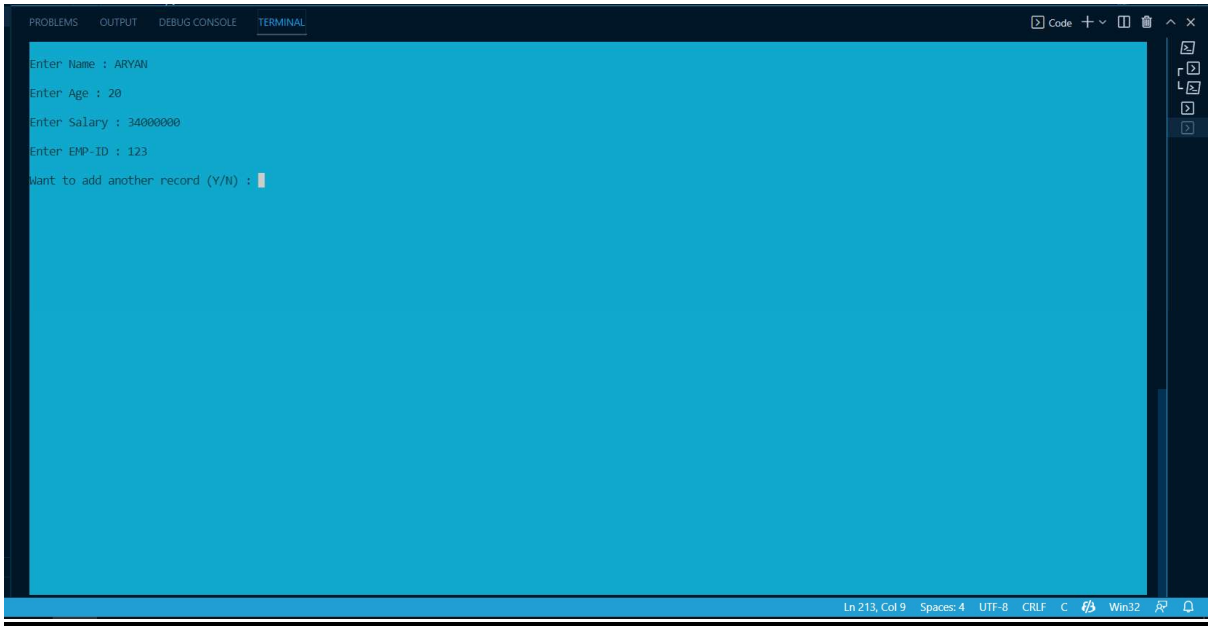
```
    printf("\n\n\n\n\t\t\t\t============"
        "============================"
        "===========");
    printf("\n\t\t\t\t~~~~~~~"
        "~~~~~~~~~~~"
        "~~~");
    printf("\n\t\t\t\t==================="
        "============================"
        "=====");
    printf("\n\t\t\t\t[|:::>:::>:::>::> "
        "EMPLOYEE RECORD <::<:::<:::"
        "<:::|]\t");
    printf("\n\t\t\t\t==================="
        "============================"
        "=====");
    printf("\n\t\t\t\t~~~~~~~~"
        "~~~~~~~~~~"
        "~");
    printf("\n\t\t\t\t====================="
        "============================\n");
    printf("\n\n\n\t\t\t\t\t\t\t\t\t\t\t\t\t"

        "\n\n\t\t\t\t");

    system("pause");

    while (1) {
        // Clearing console and asking the
        // user for input
        system("cls");
        gotoxy(30, 10);
        printf("\n1. ADD RECORD\n");
        gotoxy(30, 12);
        printf("\n2. DELETE RECORD\n");
        gotoxy(30, 14);
        printf("\n3. DISPLAY RECORDS\n");
        gotoxy(30, 16);
        printf("\n4. MODIFY RECORD\n");
        gotoxy(30, 18);
        printf("\n5. EXIT\n");
        gotoxy(30, 20);
        printf("\nENTER YOUR CHOICE...\n");
        fflush(stdin);
        scanf("%d", &choice);

        // Switch Case
        switch (choice) {
        case 1:
```

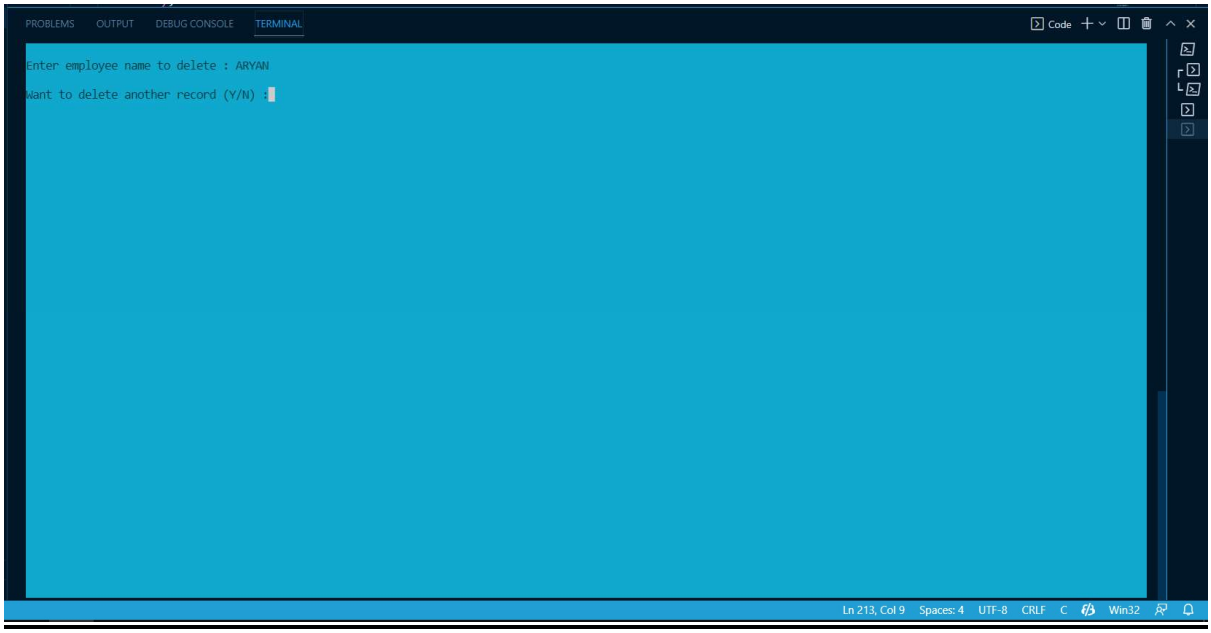
```
            // Add the records
            addrecord();
            break;

        case 2:

            // Delete the records
            deleterecord();
            break;

        case 3:

            // Display the records
            displayrecord();
            break;

        case 4:

            // Modify the records
            modifyrecord();
            break;

        case 5:
            fclose(fp);
            exit(0);
            break;

        default:
            printf("\nINVALID CHOICE...\n");
        }
    }

    return 0;
}
```

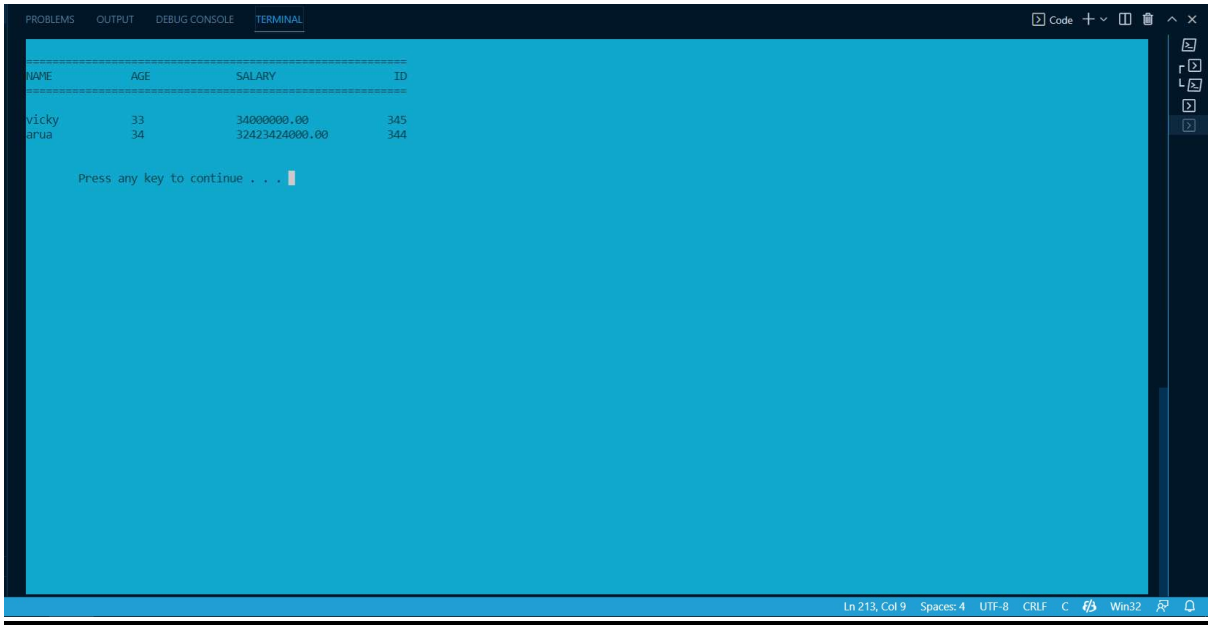
OUTPUT:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Code
PS C:\New folder (2)> cd "c:\New folder (2)\" ; if ($?) { gcc kk.c -o kk } ; if ($?) { .\kk }
kk.c: In function 'deleterecord':
kk.c:99:13: warning: implicit declaration of function 'getche' [-Wimplicit-function-declaration]
another = getche();
[|:::>:::>:::>::> EMPLOYEE RECORD <::<:::<:::<:::|]
Press any key to continue . . .
Ln 213, Col 9
Spaces: 4
UTF-8
CRLF
C
Win32


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Code
1. ADD RECORD
2. DELETE RECORD
3. DISPLAY RECORDS
4. MODIFY RECORD
5. EXIT
ENTER YOUR CHOICE...
Ln 213, Col 9
Spaces: 4
UTF-8
CRLF
C
Win32


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Code
Enter Name : ARYAN
Enter Age : 20
Enter Salary : 34000000
Enter EMP-ID : 123
Want to add another record (Y/N) :
Ln 213, Col 9
Spaces: 4
UTF-8
CRLF
C
Win32


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Code
Enter employee name to delete : ARYAN
Want to delete another record (Y/N) :
Ln 213, Col 9
Spaces: 4
UTF-8
CRLF
C
Win32



RESULT:

Employee Management system (EMS) is successfully evaluated.

CONCLUSION:

In this C Projects source code series, we create employee Management system (EMS). We can manage the details of the employees working in a company or organization.