A Micro Project Report On

EMPLOYEE MANAGEMENT SYSTEM

Submitted to the CMR Institute of Technology in partial fulfilment of the requirement of the award of the laboratory of

PROBLEM SOLVING WITH C PROGRAM

I-B.Tech. I Semester

DEPARTMENT OF FRESHMAN ENGINEERING

Submitted by

D.Sahith	24R01A6779
D.Thripura	24R01A6780
E.Sandeep	24R01A6781
G.Navya Sri	24R01A6782
G.Vyshnavi	24R01A6783
G.Satya Gandeep Varma	24R01A6784
G.Goutham	24R01A6785

Under The Guidance of

Mrs.K.Swathi Reddy

(Asst.Prof,H&S Department)

CMR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMOUS)

(Approved by AICTE, Affiliated to JNTU, Kukatpally, Hyderabad)

Kandlakoya, Medchal Road, Hyderabad.

2024-2025

CMR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMOUS)

(Approved by AICTE, Affiliated to JNTU, Kukatpally, Hyderabad)

Kandlakoya ,Medchal Road , Hyderabad.

DEPARTMENT OF FRESHMAN ENGINEERING

CERTIFICATE

This is to certify that a Micro Project entitled with

“EMPLOYEE MANAGEMENT SYSTEM”

submitted by

D.Sahith	24R01A6779
D.Thripura	24R01A6780
E.Sandeep	24R01A6781
G.Navya Sri	24R01A6782
G.Vyshnavi	24R01A6783
G.Satya Gandeep Varma	24R01A6784
G.Goutham	24R01A6785

In partial fulfillment of the requirement for the award of the laboratory of PROBLEM SOLVING WITH C PROGRAMMING of I-B.Tech. I Semester in CSE(DATA SCIENCE) towards a record of Bonafide work carried out under guidance and supervision.

Signature of Faculty
Mrs.K.Swathi Reddy
(Assistant Professor, H&S Department)

Signature of Head of the Department
Dr.M.Radha Krishna Reddy
(H&S Department)

ACKNOWLEDGEMENT

We are extremely grateful to **Dr. M. Janga Reddy** ,**Director**, **Dr. G.Madhusudhana Rao, Principal** and **Dr. M. Radha Krishna Reddy**, Head of Department, Department of FRESHMAN ENGINEERING, CMR Institute of Technology for their inspiration and valuable guidance during entire duration.
We are extremely thankful to our PSWC faculty **Mrs.K.Swathi Reddy (Assistant Professor, H&S Department),** CMR Institute of Technology for their constant guidance encouragement, and moral support throughout the project.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this Project successfully in time.

Finally, we are very much thankful to our parents and relatives who guided directly or indirectly for successful completion of project.

D.Sahith	24R01A6779
D.Thripura	24R01A6780
E.Sandeep	24R01A6781
G.Navya Sri	24R01A6782
G.Vyshnavi	24R01A6783
G.Satya Gandeep Varma	24R01A6784
G.Goutham	24R01A6785

TABLE OF CONTENTS

INTRODUCTION

Project Description:
Employee management system is a program to automate or computerize all employee management operations
Generally every company has different department systems (for example, accountant/admin/human resource/technical/vendors etc). for our project , consider the following departments. Due to the limited time we will not be implementing the feature of vendor department.
Employee management system is open to admins, managers and regular employees. Among all the user only admins have all privileges to access all information of EMS. So they can add or remove the emoployees , manager generates reports where as other user have limited access.once the users login they need to perform few tasks specific to their role.
Reports in emolyeement management system are categorized into different types of based on their roles.

NOTE: To make concept simple **,** assume the way we are storing all the information in binary files .

DESCRIPTION

1) Objective/ Vision:

This project is aimed at developing Employee Management System that allows to automate or computerize all employee management operations

2) Users of the System:

- Admins
- Employees
- Managers

3) Functional requirements

- Create initial setup which includes:
  Generating employee information (adding/deleting/view employee information)
  Generating unique employee ID for each employee
- User management
- Role-based user menus

4) Non-functional requirements

- Simple UI
- Generic Coding

5) User interface priorities:

console

6) Reports to be generated:

- Reports for Admins
- Reports for Employees
- Reports for Managers

7) Technologies to be used:

C, and binary Files for storing data

8) Tools to be Used:

Dev C++

9) Final Deliverable must include:

- Create initial setup mentioned as above
- User management
- Role-based user menus

RELATIONSHIP DIAGRAM

ADMIN	MANAGER	EMPLOYEE
1.Add or delete admin 1.Add admin 2.Delete admin	**1.**Add or delete employee 1.Add employee 2.Delete employee	**1.**Employee list
2. Add or delete manager 1.Add manager 2.Delete manager	**2.** Manager list	**2.**log out
3.Add or delete employee 1.Add employee 2.Delete employee	**3.** Employee list	
4. Admin list	**4.**log out	
5. Manager list		
6. Employee list		
7. log out		

SOURCE CODE

```
 #include <stdio.h >
#include < string.h>
#include < stdbool.h>
#include <stdlib.h>
#include <conio.h>
int p = 0, z = 0;
void admin();
void manager();
void employee();
void ainterface();
void minterface();
void einterface();
void A_D_admin();
void A_D_manager();
void A_D_employee();
void admin_list();
void manager_list();
void employee_list();
void main()
{
    int c;
    char w;
menu:
    clrscr();
    printf("\t\t********MENU********\n");
    printf("\t\t1.ADMIN\n\t\t2.MANAGER\n\t\t3.EMPLOYEE\n");
    printf("\t\tEnter your choice: ");
    scanf("%d", &c);
    switch (c)
    {
    case 1:
        admin();
        break;
    case 2:
        manager();
        break;
    case 3:
        employee();
        break;
    default:
        scanf("%c", &w);
        printf("\t\tinvalid character\n\n ");
        printf("\t\tpress enter for 'try again' ");
        scanf("%c", &w);
        goto menu;
    }
}
void admin()
{
    FILE *admin_p;
    int i, c = 0, k = 1, n;
    char pw[6], pwi[6], id[6], ch[6], w;
    admin_p = fopen("/storage/emulated/0/EMS/admin.txt", "r");
    clrscr();
    if (admin_p == NULL)
    {
        scanf("%c", &w);
        printf("\t\tAdmin list not found");
        scanf("%c", &w);
        main();
    }
    else
    {
    admin:
        clrscr();
        printf("\t\t   ADMIN LOGIN\n\n");
        printf("\t\tEnter  admin id :  ");
        scanf("%s", id);
        printf("\n");
        bool read_file = true;
        int c, line = 1;
        do
        {
            fgets(ch, 6, admin_p);
            if (feof(admin_p))
            {
                read_file = false;
 }
            else
            {
                c = 0;
                for (i = 0; i < 5; i++)
                {
                    if (ch[i] == id[i])
                    {
                        c++;
                    }
                }
            }
            if (line % 3 == 0)
            {
                k++;
            }
            if (c == 5)
            {
                break;
            }
            line++;
        } while (read_file);
        rewind(admin_p);
        if (c == 5)
        {
        pws:
            printf("\t\tEnter password [5 char] :  ");
            {
                scanf("%s", pw);
            }
            bool read_file = true;
            line = 1;
            do
            {
                fgets(ch, 6, admin_p);
                if (feof(admin_p))
                {
                    read_file = false;
                }
                else
                {
                    c = 0;
                    for (i = 0; i < 5; i++)
                    {
                        if (ch[i] == id[i])
                        {
                            c++;
                        }
                    }
                }
                if (c == 5)
                {
                    fgets(pwi, 6, admin_p);
                }
            } while (read_file);
            if (strcmp(pw, pwi) == 0)
```

```
            {
                scanf("%c", &w);
                printf("\n\t\tLogin successfull\n\n");
                printf("\t\tpress enter for admin interface");
                scanf("%c", &w);
                ainterface();
            }
            else
            {
                printf("\t\tIncorrect Password\n\n");
                goto pws;
            }
        }
        else
        {
            scanf("%c", &w);
            printf("\t\tadmin not found\n\n");
            printf("\t\tpress enter for 'try again' ");
            scanf("%c", &w);
            goto admin;
        }
    }
  fclose(admin_p);
}

void manager()
{
    FILE *manager_p;
    int i, c = 0;
    char pw[6], pwi[6], id[6], ch[6], w;
    manager_p = fopen("/storage/emulated/0/EMS/manager.txt", "r");
    system("cls");
    if (manager_p == NULL)
    {
        printf("\t\tmanager list not found");
    }
    else
    {
    manager:
        system("cls");
        printf("\t\t  MANAGER  LOGIN\n\n");
        printf("\t\tEnter  manager id :  ");
        scanf("%s", id);
        printf("\n");
        bool read_file = true;
        int c, line = 1;
```

```
do
{
    fgets(ch, 6, manager_p);
    if (feof(manager_p))
    {
        read_file = false;
    }
    else
    {
        c = 0;
        for (i = 0; i < 5; i++)
        {
            if (ch[i] == id[i])
            {
                c++;
            }
        }
    }
    if (line % 3 == 0)
    {
        int k;
        k++;
    }
    if (c == 5)
    {
        break;
    }
    line++;
} while (read_file);
rewind(manager_p);
if (c == 5)
{
pws:
    printf("\t\tEnter password [5 char] :  ");
    {
        scanf("%s", pw);
    }
    bool read_file = true;
    line = 1;
    do
    {
        fgets(ch, 6, manager_p);
        if (feof(manager_p))
        {
            read_file = false;
        }
```

```
                else
                {
                    c = 0;
                    for (i = 0; i < 5; i++)
                    {
                        if (ch[i] == id[i])
                        {
                            c++;
                        }
                    }
                }
                if (c == 5)
                {
                    fgets(pwi, 6, manager_p);
                }
            } while (read_file);
            if (strcmp(pw, pwi) == 0)
            {
                scanf("%c", &w);
                printf("\n\t\tLogin successfull\n\n");
                printf("\t\tpress enter for manager interfacr");
                scanf("%c", &w);
                minterface();
            }
            else
            {
                printf("\t\tIncorrect Password\n\n");
                goto pws;
            }
        }
        else
        {
            scanf("%c", &w);
            printf("\t\tmanager not found\n\n");
            printf("\t\tpress enter for 'try again' ");
            scanf("%c", &w);
            goto manager;
        }
    }
    fclose(manager_p);
}

void employee()
{
    FILE *employee_p;
    int i, c = 0;
```

```
char pw[6], pwi[6], id[6], ch[6], w;
employee_p = fopen("/storage/emulated/0/EMS/employee.txt", "r");
system("cls");
if (employee_p == NULL)
{
   printf("\t\temployee list not found");
}
else
{
employee:
   system("cls");
   printf("\t\t  EMPLOYEE  LOGIN\n\n");
   printf("\t\tEnter  employer id :  ");
 scanf("%s", id);
   printf("\n");
   bool read_file = true;
   int c, line = 1;
   do
   {
      fgets(ch, 6, employee_p);
      if (feof(employee_p))
      {
         read_file = false;
      }
      else
      {
         c = 0;
         for (i = 0; i < 5; i++)
         {
            if (ch[i] == id[i])
            {
               c++;
            }
         }
      }
      if (line % 3 == 0)
      {
         int k;
         k++;
      }
      if (c == 5)
      {
         break;
      }
      line++;
   } while (read_file);
```

```
rewind(employee_p);
if (c == 5)
{
pws:
    printf("\t\tEnter password [5 char] :  ");
    {
        scanf("%s", pw);
    }
    bool read_file = true;
    line = 1;
    do
    {
        fgets(ch, 6, employee_p);
        if (feof(employee_p))
        {
            read_file = false;
        }
        else
        {
            c = 0;
            for (i = 0; i < 5; i++)
            {
                if (ch[i] == id[i])
                {
                    c++;
                }
            }
        }
        if (c == 5)
        {
            fgets(pwi, 6, employee_p);
        }
    } while (read_file);
    if (strcmp(pw, pwi) == 0)
    {
        scanf("%c", &w);
        printf("\n\t\tLogin successfull\n\n");
        printf("\t\tpress enter for  employee menu");
        scanf("%c", &w);
        einterface();
    }
    else
    {
        printf("\t\tIncorrect Password\n\n");
        goto pws;
    }
```

```
        }
        else
        {
            scanf("%c", &w);
            printf("\t\temployee not found\n\n");
            printf("\t\tpress enter for 'try again' ");
            scanf("%c", &w);
            goto employee;
        }
    }
    fclose(employee_p);
}
void ainterface()
{
    int c, w;
ad_admin:
    clrscr();
    printf("\t\t***  ADMIN  MENU  ***\n\n");
    printf("\t\t1.Add or Delete Admin\n");
    printf("\t\t2.Add or Delete Manager\n");
    printf("\t\t3.Add or Delete Employee\n");
    printf("\t\t4.Admin list\n");
    printf("\t\t5.Manager list\n");
    printf("\t\t6.Employee list\n");
    printf("\t\t7.Log out\n\n");
    printf("\t\tEnter your choice :  ");
    scanf("%d", &c);
    switch (c)
    {
    case 1:
        A_D_admin();
        break;
    case 2:
        A_D_manager();
        break;
    case 3:
        p = 1;
        A_D_employee();
        break;
    case 4:
        admin_list();
        break;
    case 5:
        z = 1;
        manager_list();
        break;
    case 6:
        z = 1;
        employee_list();
        break;
    case 7:
        main();
        break;
    default:
        printf("\t\tinvalid character\n\n");
        printf("\t\tpress 1 for try again");
        scanf("%d", &w);
        goto ad_admin;
    }
}
void minterface()
{
    int c, w;
ad_manager:
    clrscr();
    printf("\t\t***  MANAGER  MENU  ***\n\n");
    printf("\t\t1.Add or Delete Employee\n");
    printf("\t\t2.Manager List\n");
    printf("\t\t3.Employee List\n");
    printf("\t\t4.Log out\n\n");
    printf("\t\tenter your choice :  ");
    scanf("%d", &c);
    switch (c)
    {
    case 1:
        p = 2;
        A_D_employee();
        break;
    case 2:
        z = 2;
        manager_list();
        break;
    case 3:
        z = 2;
        employee_list();
        break;
    case 4:
        main();
        break;
    default:
        printf("\t\tinvalid character\n");
        printf("\t\tpress 1 for manager menu");
        scanf("%d", &w);
        goto ad_manager;
    }
}
void einterface()
{
    int c;
    char w;
ad_employee:
    system("cls");
    printf("\t\t*** EMPLOYEE  MENU ***\n\n");
    printf("\t\t1.Employee List\n");
    printf("\t\t2.Log out\n\n");
    printf("\t\tenter choice :  ");
    scanf("%d", &c);
    switch (c)
    {
    case 1:
        z = 3;
        employee_list();
        break;
    case 2:
        main();
        break;
    default:
        scanf("%c", &w);
        printf("\t\tinvalid character\n");
        scanf("%c", &w);
        goto ad_employee;
    }
}
void A_D_admin()
{
    FILE *admin_p, *copy;
    char id[6], pw[6], add[6];
    int ch, w, i;
    admin_p = fopen("/storage/emulated/0/EMS/admin.txt", "r+");
   if (admin_p == NULL)
    {
        printf("Error opening file.\n");
        return;
    }
a1:
    clrscr();
    printf("\t\t--------------------------------------\n");
    printf("\t\t1.Add Admin\n\t\t2.Delete Admin\n\n");
    printf("\t\tenter your choice: ");
    scanf("%d", &ch);
    printf("\n");
    if (ch == 1 || ch == 2)
    {
        if (ch == 1)
        {
            clrscr();
            printf("\t\t1. Add Admin\n\n");
            printf("\t\tCreate admin ID: ");
            scanf("%s", id);
            fseek(admin_p, 0, 2);
            int i = 0;
            while (fgets(add, 6, admin_p))
            {
                add[strcspn(add, "\n")] = 0;
                if (strcmp(add, id) == 0)
                {
                    i = 1;
                    break;
                }
            }
            if (i == 1)
            {
                printf("\t\tAdmin ID already exists\n");
                printf("\t\tPress any key to try again...\n");
                scanf("%d", &w);
                goto a1;
            }
            printf("\t\tCreate password: ");
            scanf("%s", pw);
            fseek(admin_p, 0, 2);
            fflush(stdin);
            fprintf(admin_p, "\n%s", id);
            fprintf(admin_p, "%s", pw);
            printf("\n\t\tAdmin added\n\n");
            printf("\t\tPress 1 for admin menu\n");
            scanf("%d", &w);
            ainterface();
        }
        else if (ch == 2)
        {
            printf("\t\t2.Delete admin\n\n");
            char temp[] = "/storage/emulated/0/EMS/modified.txt", c;
            copy = fopen(temp, "w");
            printf("\t\tenter admin id :  ");
```

```
            scanf("%s", id);
            do
            {
                fgets(add, 6, admin_p);
                if (strcmp(add, id) == 0)
                {
                    continue;
                }
                else
                {
                    fputs(add, copy);
                }
                if (feof(admin_p))
                {
                    break;
                }
            } while (i == 1);
            fclose(admin_p);
            remove("/storage/emulated/0/EMS/admin.txt");
              rename("/storage/emulated/0/EMS/modified.txt","/storage/emulated/0/EMS/admin.txt");

            printf("\n\t\tdeleted successfully\n\n");
            printf("\t\tpress 1 admin menu");
            scanf("%d", &w);
            ainterface();
        }
    }
    else
    {
        printf("\t\tinvalid character");
        scanf("%d", &w);
        goto a1;
    }
}
void A_D_manager()
{
    FILE *manager_p, *copy;
    char id[6], pw[6], add[6], pwi[6];
    int i = 1, ch, w;
    manager_p = fopen("/storage/emulated/0/EMS/manager.txt", "a+");
a1:
clrscr();
    printf("\t\t-------------------------------------\n");
    printf("\t\t1.Add Admin\n\t\t2.Delete Admin\n\n");
    printf("\t\tenter your choice: ");
    scanf("%d", &ch);
```

```
printf("\n");
if (ch == 1 || ch == 2)
{
    if (ch == 1)
    {
        clrscr();
        printf("\t\t1. Add Msnager\n\n");
        printf("\t\tCreate manager ID: ");
        scanf("%s", id);
        fseek(manager_p, 0, 2);
        int i = 0;
        while (fgets(add, 6, manager_p))
        {
            add[strcspn(add, "\n")] = 0;
            if (strcmp(add, id) == 0)
            {
                i = 1;
                break;
            }
        }
        if (i == 1)
        {
            printf("\t\tManager ID already exists\n");
            printf("\t\tPress any key to try again...\n");
            scanf("%d", &w);
            goto a1;
        }
        printf("\t\tCreate password: ");
        scanf("%s", pw);
        fseek(manager_p, 0, 2);
        fprintf(manager_p, "\n%s", id);
        fprintf(manager_p, "%s", pw);
        printf("\n\t\tManager added\n\n");
        printf("\t\tPress 1 for manager menu\n");
        scanf("%d", &w);
        ainterface();
    }
    else if (ch == 2)
    {
        printf("\t\t2.Delete Manager\n\n");
        char temp[] = "/storage/emulated/0/EMS/modified.txt", c;
        copy = fopen(temp, "w");
        printf("\t\tenter manager id :  ");
        scanf("%s", id);
        do
        {
                fgets(add, 6, manager_p);
                if (strcmp(add, id) == 0)
                {
                    continue;
                }
                else
                {
                    fputs(add, copy);
                }
                if (feof(manager_p))
                {
                    break;
                }
            } while (i == 1);
            fclose(manager_p);
            remove("/storage/emulated/0/EMS/manager.txt");
          rename("/storage/emulated/0/EMS/modified.txt","/storage/emulated/0/EMS/manager.txt");

            printf("\n\t\tdeleted successfully\n\n");
            printf("\t\tpress 1 admin menu");
            scanf("%d", &w);
            ainterface();
        }
    }
    else
    {
        printf("\t\tinvalid character");
        scanf("%d", &w);
        goto a1;
    }
}
void A_D_employee()
{
    FILE *employee_p, *copy;
    char id[6], pw[6], add[6], pwi[6];
    int i = 1, ch, w;
    employee_p = fopen("/storage/emulated/0/EMS/employee.txt", "a+");
a1:
clrscr();
    printf("\t\t-------------------------------------\n");
    printf("\t\t1.Add Employer\n\t\t2.Delete Employee\n\n");
    printf("\t\tenter your choice: ");
    scanf("%d", &ch);
    printf("\n");
    if (ch == 1 || ch == 2)
    {
```

```
if (ch == 1)
{
    clrscr();
    printf("\t\t1. Add Employer\n\n");
    printf("\t\tCreate Employee ID: ");
    scanf("%s", id);
    fseek(employee_p, 0, 2);
    int i = 0;
    while (fgets(add, 6, employee_p))
    {
        add[strcspn(add, "\n")] = 0;
        if (strcmp(add, id) == 0)
        {
            i = 1;
            break;
        }
    }
    if (i == 1)
    {
        printf("\t\tEmployee ID already exists\n");
        printf("\t\tPress any key to try again...\n");
        scanf("%d", &w);
        goto a1;
    }
    printf("\t\tCreate password: ");
    scanf("%s", pw);
    fseek(employee_p, 0, 2);
    fprintf(employee_p, "\n%s", id);
    fprintf(employee_p, "%s", pw);
   printf("\n\t\tEmployer added\n\n");
    if (p == 1)
    {
        printf("\t\tpress 1 for admin menu");
        scanf("%d", &w);
        ainterface();
    }
    if (p == 2)
    {
        printf("\t\tpress 1 for manager menu");
        scanf("%d", &w);
        minterface();
    }
}
else if (ch == 2)
{
    clrscr();
```

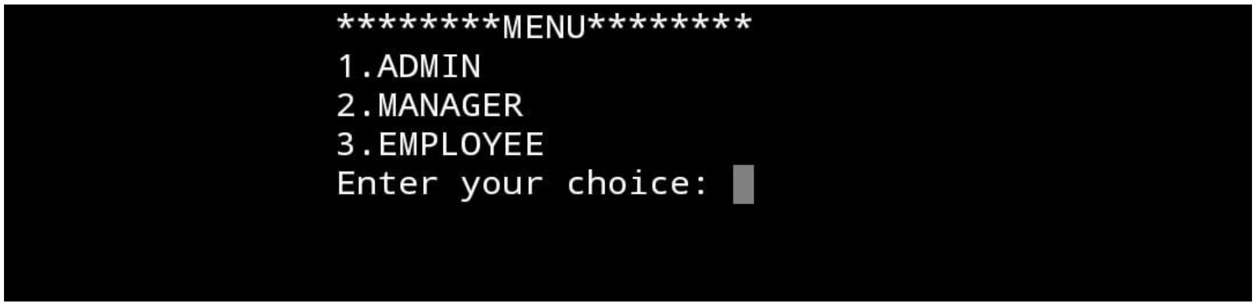
```
        printf("\t\t2.Delete Employee\n\n");
        char temp[] = "/storage/emulated/0/EMS/modified.txt", c;
        copy = fopen(temp, "w");
        printf("\t\tenter employer id :  ");
        scanf("%s", id);
        do
        {
            fgets(add, 6, employee_p);
            if (strcmp(add, id) == 0)
            {
                continue;
            }
            else
            {
                fputs(add, copy);
            }
            if (feof(employee_p))
            {
                break;
            }

        } while (i == 1);
        fclose(employee_p);
        remove("/storage/emulated/0/EMS/admin.txt");
         rename("/storage/emulated/0/EMS/modified.txt", "/storage/emulated/0/EMS/admin.txt");
        printf("\n\t\tdeleted successfully\n\n");
        if (p == 1)
        {
            printf("\t\tpress 1 for admin menu");
            scanf("%d", &w);
            ainterface();
        }
        if (p == 2)
        {
            printf("\t\tpress 1 for manager menu");
            scanf("%d", &w);
            minterface();
        }
    }
}
else
{
    printf("\t\tinvalid character");
    scanf("%d", &w);
    goto a1;
}
```

```
}

void admin_list()
{
    FILE *admin_p;
    char id[6], pw[6], spa[6];
    int x = 1, i = 1, m = 0, w;
    admin_p = fopen("/storage/emulated/0/EMS/admin.txt", "r");
    system("cls");
    printf("\t----------------------------------------\n\n");
    printf("\t\tADMIN    ID's   LIST\n\n");
    printf("\t-----------------------------------------\n\n");
    printf("\t\tID's\t\tpassword\n");
    printf("\t-----------------------------------------\n");
   while (x == 1)
    {
        fgets(id, 6, admin_p);
        if (i % 3 != 0)
        {
            printf("\t\t%s", id);
            m++;
            if (m % 2 == 0)
            {
                printf("\n");
                m = 0;
            }
        }
        if (feof(admin_p))
            break;
        i++;
    }
    fclose(admin_p);
    printf("\n\n\n\tpress enter to go to admin menu");
    scanf("%d", &w);
    ainterface();
}

void manager_list()
{
    FILE *manager_p;
    char id[6];
    int x = 1, i = 1, m = 0, w;
    manager_p = fopen("/storage/emulated/0/EMS/manager.txt", "r");
    system("cls");
    printf("\t----------------------------------------\n\n");
    printf("\t\tMANAGER    ID's   LIST\n\n");
    printf("\t-----------------------------------------\n\n");
    printf("\t\tID's\t\tpassword\n");
    printf("\t-----------------------------------------\n");
    while (x == 1)
    {
        fgets(id, 6, manager_p);
        if (i % 3 != 0)
        {
            printf("\t\t%s", id);
            m++;
            if (m % 2 == 0)
            {
                printf("\n");
                m = 0;
            }
        }
        if (feof(manager_p))
            break;
        i++;
    }
    fclose(manager_p);
    if (z == 1)
    {
        printf("\n\n\n\tpress 1 for admin menu");
        scanf("%d", &w);
        ainterface();
    }
    if (z == 2)
    {
        printf("\n\n\n\tpress 1 for  manager menu");
        scanf("%d", &w);
        minterface();
    }
}
void employee_list()
{
    FILE *employee_p;
    char id[6];
    int x = 1, i = 1, m = 0, w;
    employee_p = fopen("/storage/emulated/0/EMS/employee.txt", "r");
    system("cls");
    printf("\t----------------------------------------\n\n");
    printf("\t\tEMPLOYEE    ID's   LIST\n\n");
    printf("\t-----------------------------------------\n");
    printf("\t\tID's\t\tpassword\n");
    printf("\t-----------------------------------------\n");
    while (x == 1)
    {
        fgets(id, 6, employee_p);
        if (i % 3 != 0)
        {
            printf("\t\t%s", id);
            m++;
            if (m % 2 == 0)
            {
                printf("\n");
                m = 0;
            }
        }
        if (feof(employee_p))
            break;
        i++;
    }
    fclose(employee_p);
    if (z == 1)
    {
        printf("\n\n\n\tpress 1 for admin menu");
        scanf("%d", &w);
        ainterface();
    }
    if (z == 2)
    {
        printf("\n\n\n\tpress 1 for manager menu");
        scanf("%d", &w);
        minterface();
    }
    if (z == 3)
    {
        printf("\n\n\n\tpress 1 for employee menu");
        scanf("%d", &w);
        einterface();
    }
}
```

TOTAL LINES = 937

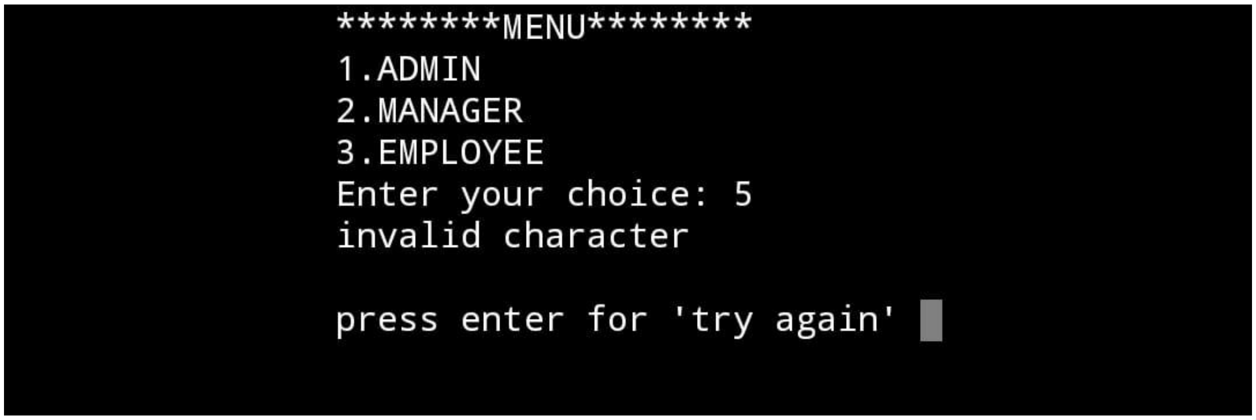
OUTPUT

1. MENU



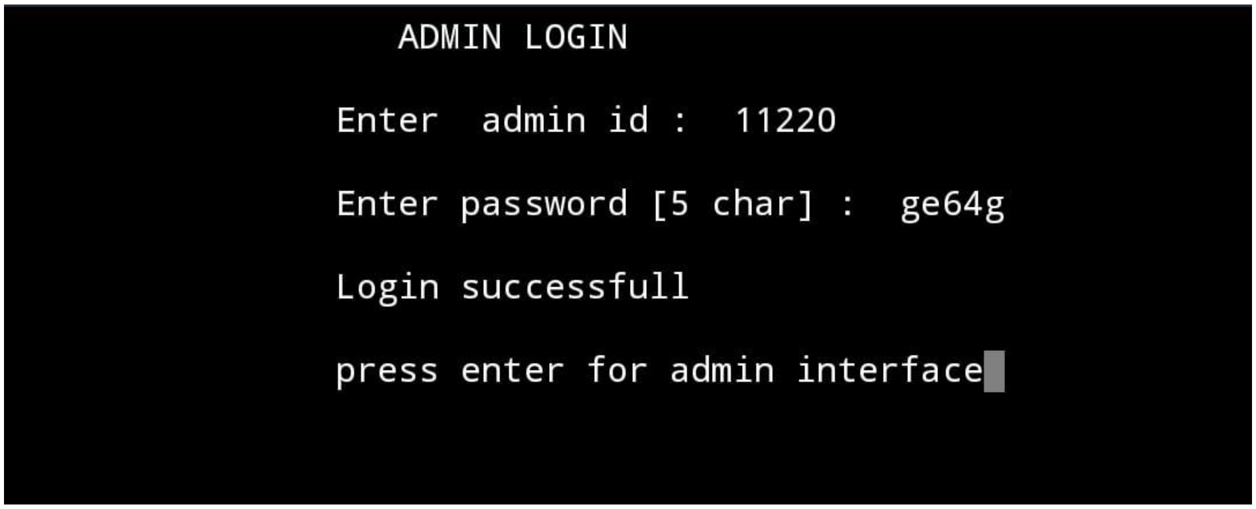
```
********MENU********
1.ADMIN
2.MANAGER
3.EMPLOYEE
Enter your choice: 
```

2. MENU FOR INCORRECT CHOICE



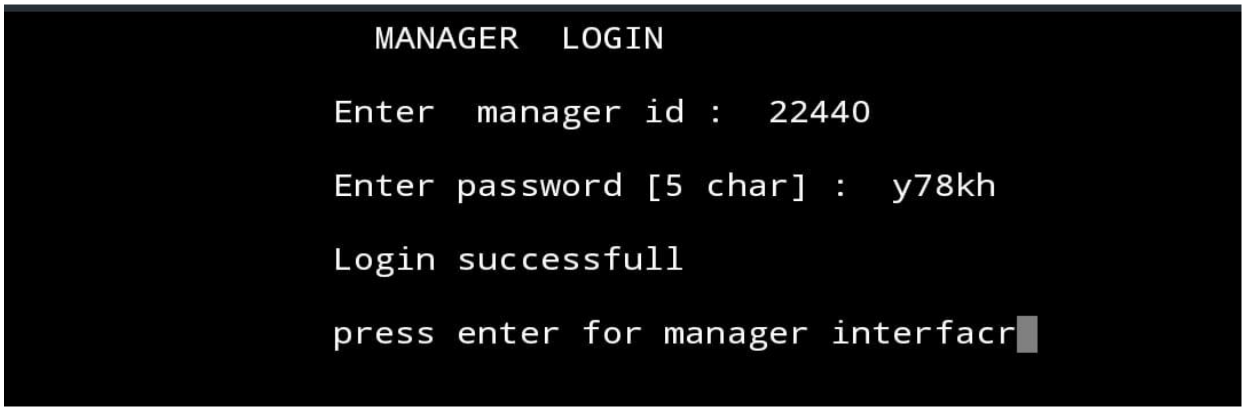
```
********MENU********
1.ADMIN
2.MANAGER
3.EMPLOYEE
Enter your choice: 5
invalid character

press enter for 'try again' 
```

3. ADMIN LOGIN



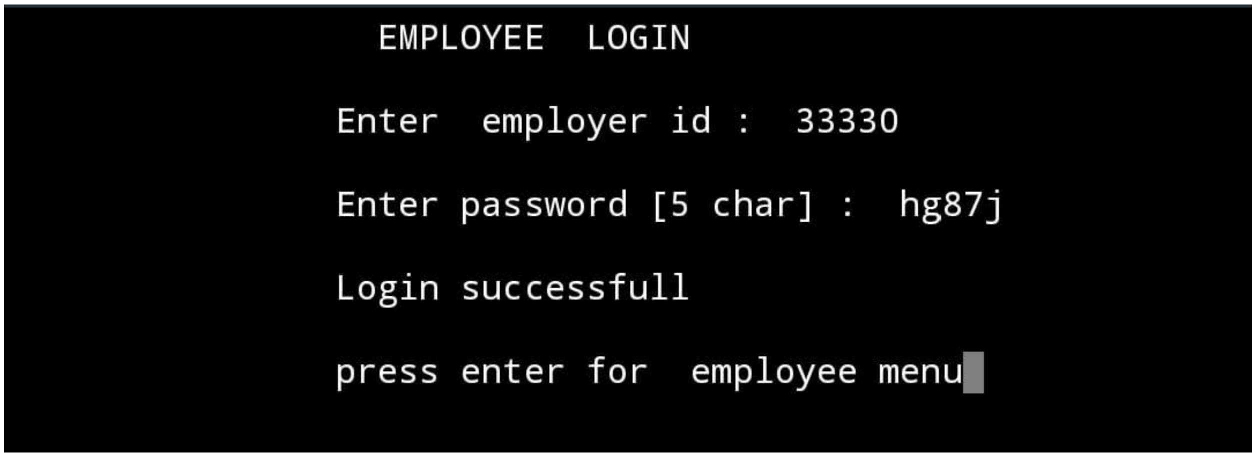
```
   ADMIN LOGIN

Enter  admin id :  11220

Enter password [5 char] :  ge64g

Login successfull

press enter for admin interface
```

4. MANAGER LOGIN



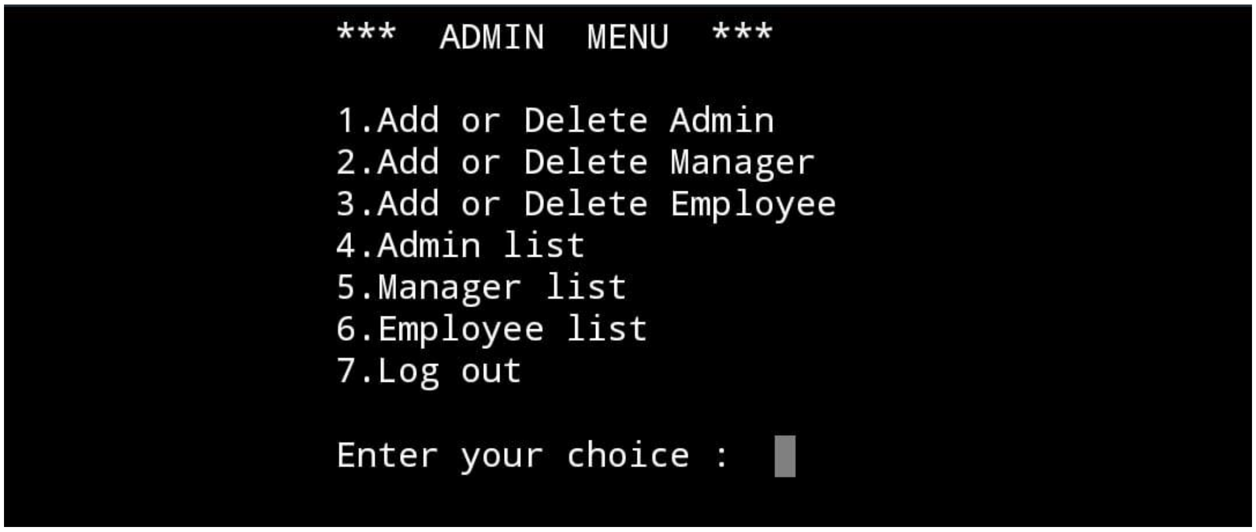
```
  MANAGER  LOGIN

Enter  manager id :  22440

Enter password [5 char] :  y78kh

Login successfull

press enter for manager interfacr
```

5. EMPLOYEE LOGIN



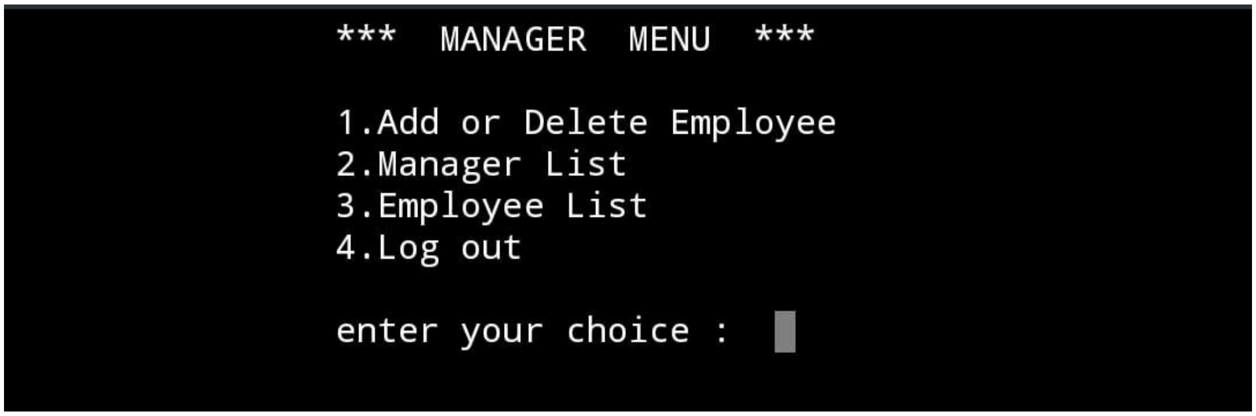
```
  EMPLOYEE  LOGIN

Enter  employer id :  33330

Enter password [5 char] :  hg87j

Login successfull

press enter for  employee menu
```

6. ADMIN MENU



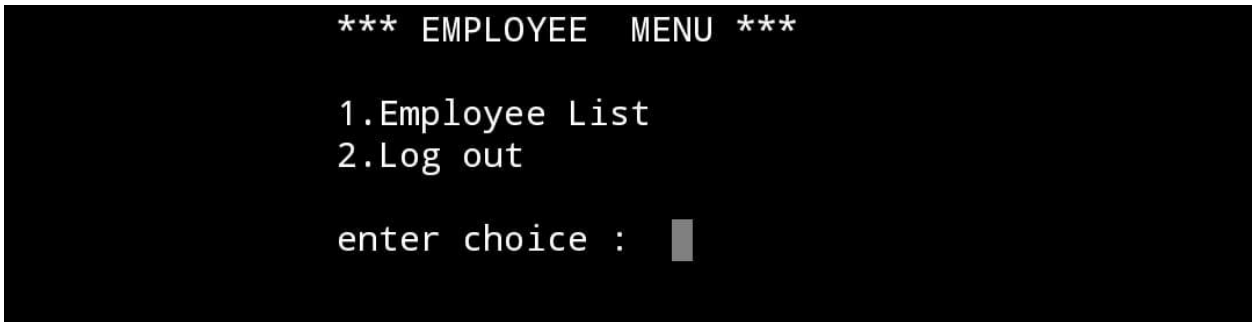
```
***  ADMIN  MENU  ***

1.Add or Delete Admin
2.Add or Delete Manager
3.Add or Delete Employee
4.Admin list
5.Manager list
6.Employee list
7.Log out

Enter your choice :
```

7. MANAGER MENU



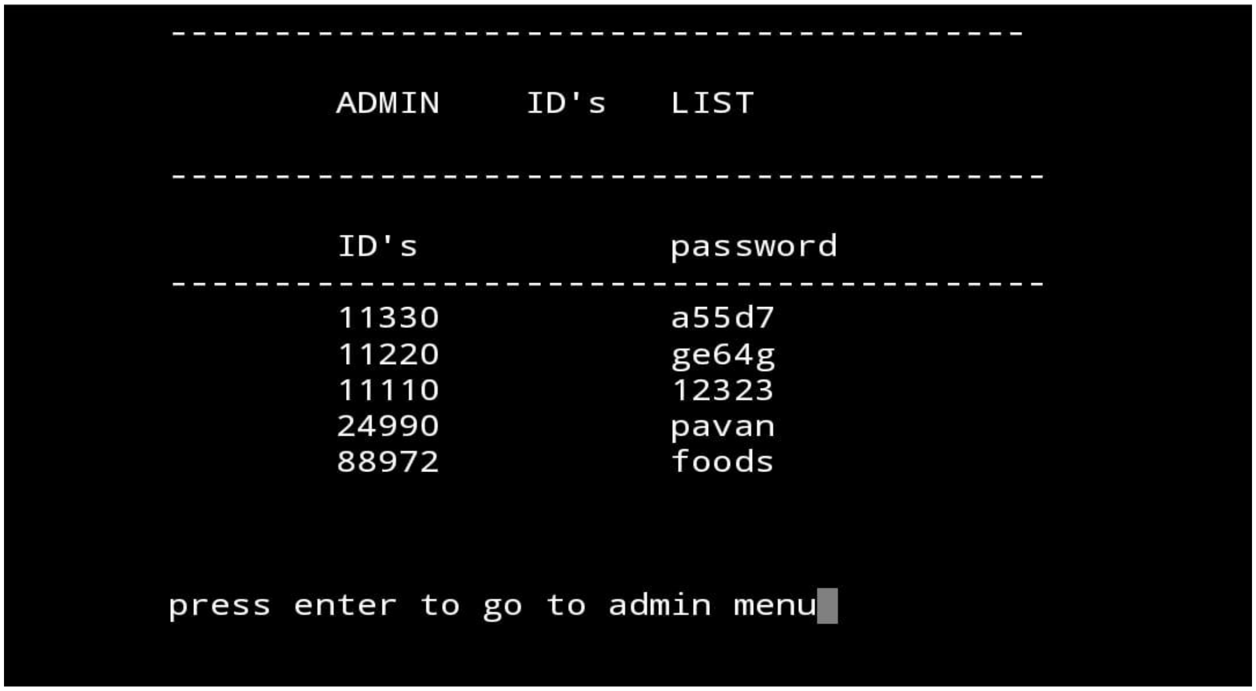
```
        ***  MANAGER  MENU  ***

        1.Add or Delete Employee
        2.Manager List
        3.Employee List
        4.Log out

        enter your choice :
```

8. EMPLOYEE MENU



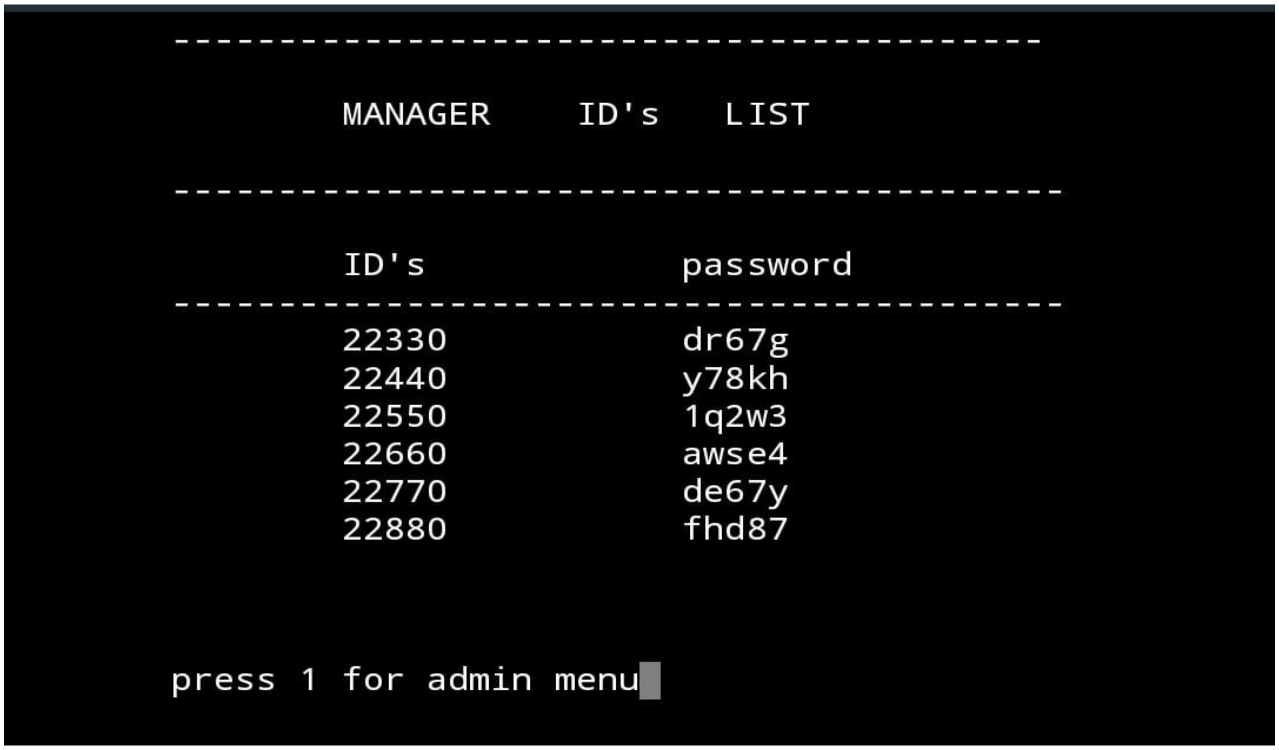
```
        *** EMPLOYEE  MENU ***

        1.Employee List
        2.Log out

        enter choice :
```

9. ADMIN LIST



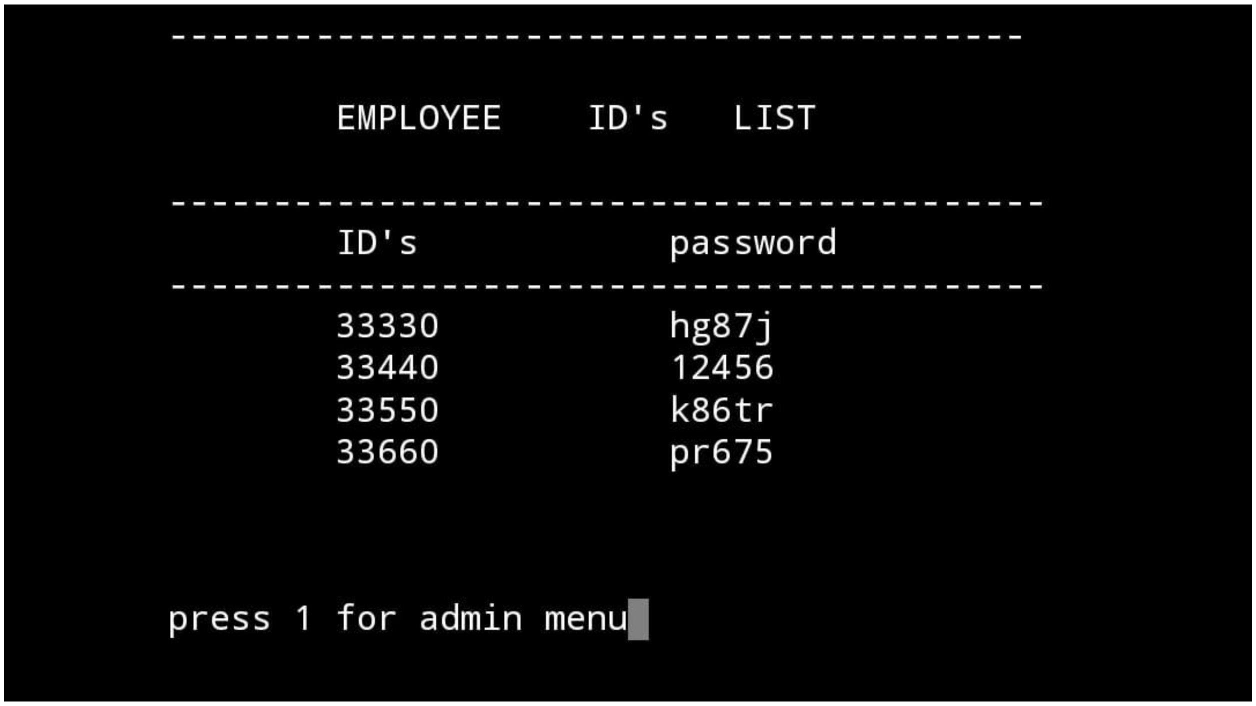
```
    ------------------------------------------

        ADMIN    ID's   LIST

    -------------------------------------------

        ID's            password
    -------------------------------------------
        11330           a55d7
        11220           ge64g
        11110           12323
        24990           pavan
        88972           foods



    press enter to go to admin menu
```

10. MANAGER LIST



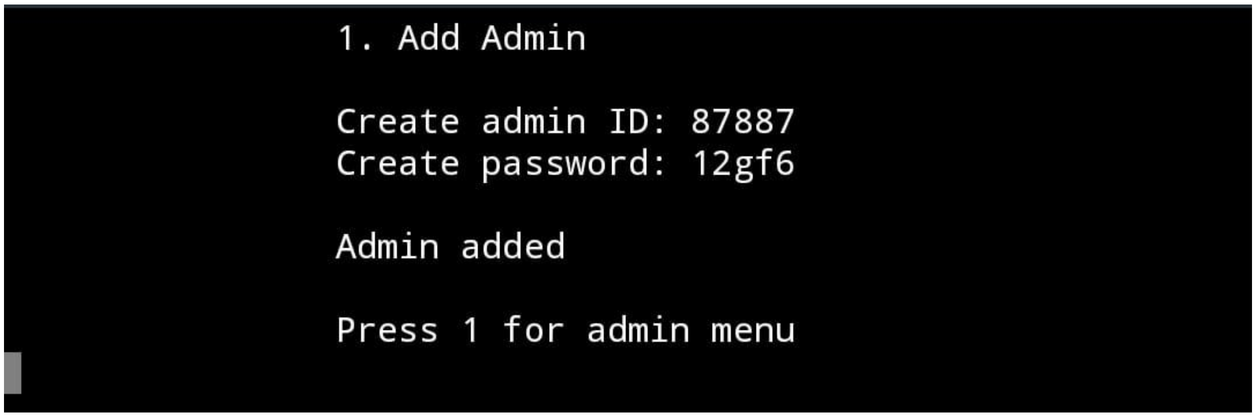
```
-----------------------------------------

        MANAGER    ID's   LIST

------------------------------------------

        ID's            password
------------------------------------------
        22330           dr67g
        22440           y78kh
        22550           1q2w3
        22660           awse4
        22770           de67y
        22880           fhd87



press 1 for admin menu
```

11. EMPLOYEE LIST



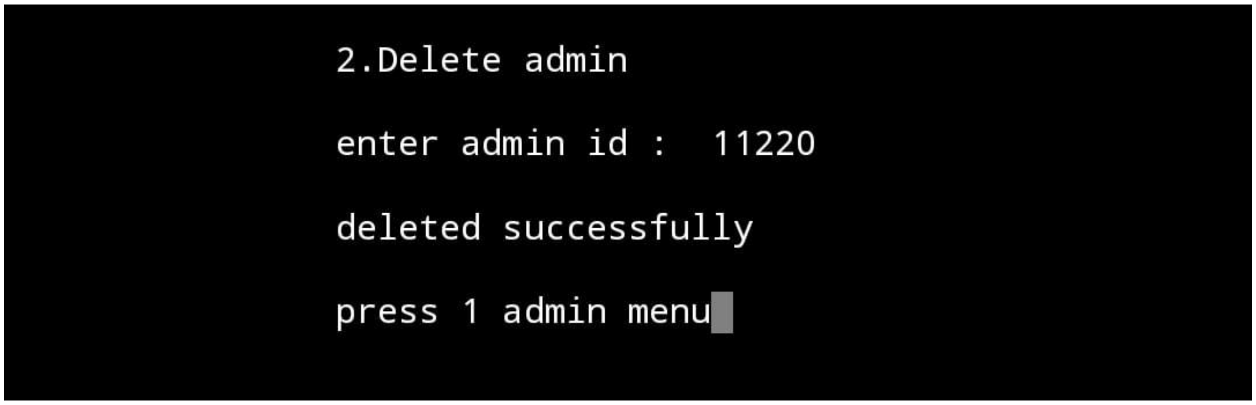
```
-----------------------------------------

        EMPLOYEE    ID's   LIST

------------------------------------------
        ID's            password
------------------------------------------
        33330           hg87j
        33440           12456
        33550           k86tr
        33660           pr675



press 1 for admin menu
```

12. ADD ADMIN



```
1. Add Admin

Create admin ID: 87887
Create password: 12gf6

Admin added

Press 1 for admin menu
```

13. DELETE ADMIN



```
2.Delete admin

enter admin id :  11220

deleted successfully

press 1 admin menu
```

DATA BASES

A data base is a structured collection of data that is stored electronically.
It can contain tiny type of data, such as words , numbers , images , videos , and files .

CREATING OF DATA BASE FOR OUR PROJECT :

1. Creating files in the hard disc in text mode
2. Create a file name for each of the admin ,manager and employee
3. Name the files as admin.txt ,manager.txt and employee.txt
4. Note the path of the file of each file and it should be written in file pointer of their respective file pointer names.
5. File paths will vary depending on the device and storage location.so be knowing your file path you need to edit the source code.

EXAMPLE :
Path of admin.txt:/ storage/emulated/0/EMS/admin.txt
File pointer : admin_p
Decleration of file pointer variable
admin_p = fopen("/storage/emulated/0/EMS/admin.txt","r");

STRUCTURE OF DATA BASE

1. Each line in text file will indicate details of each individual person in the company

EXAMPLE TEXT FILE :

Admin.txt

```
11330a55d7
11220ge64g
1111012323
24990pavan
88972foods
```

2. Since the text fie is admin so each line in the file will indicate the details of each individual admin in a company.

   READING FILE:

   1. File contains 5 digits and 5 characters as the details of an admin.
   2. 5 digits are to be written in the beginning which indicates id of the admin
   3. 5 characters are to be written after this,which indicates password of the admin.
      (caution: no gap should be in between id and password)
   4. Including of gap make string to read wrong value from the file

Manager.txt

```
22330dr67g
22440y78kh
225501q2w3
22660awse4
22770de67y
22880fhd87
```

Employee.txt

```
33330hg87j
3344012456
33550k86tr
33660pr675
```

FUNCTIONS USED

USER DEFINED FUNCTIONS :

1. **void admin()**	**:** for login of admin.
2. **void manager()**	**:**for login of manager
3. **void employee()**	**:**for login of employee
4. **void ainterface()**	**:**for printing admin menu
5. **void minterface()**	**:** for printing manager menu
6. void einterface()	**:** for printing employee menu
7. void A_D_admin()	**:** program for add/delete admin
8. void A_D_manager()	**:** program for add/delete manager
9. void A_D_employee()	**:** program for add/delete employee
10. void admin_List()	**:** program for printing list of admin IDs and passwords From the file on the moniter.
11. void manager_List()	**:** program for printing list of manager IDs and passwords From the file on the moniter
12. void employee_List ()	**:** program for printing the List of employee id's and password's from the file, on the monitor.
13. void main()	**:**for running the program**.**

HEADER FILE LIST:

1. stdio.h

2. conio.h

3. string.h

4. stdlib.h

5. stdbool.h

Functions Used :

1. stdio.h

printf( ) , scanf( ) , fopen( ) , fclose( ) , fprintf( ) , fscanf( ) , fputs( ) , fgets( ) , fseek( ) , feof( ) , rewind( ).

2. conio.h:

clrscr()

3. String.h:

strcmp(), strcspn()

4. stdlib.h:

system("cls"), remove(),rename().

5. stdbool.h:

Bool

Details of some functions:

1. strcspn( ):

* It calculates the length of the number of characters before the 1st occurrence of character present in both the strings.

SYNTAX :

strcspn(const char *str1, const char *str2);

2. system("cls");

* It clears the output screen and prints the next statement on the screen.
* It is the same as clrscr().
* But, the main difference between clrscr() and system("cls") is:
* clrscr() is available on Unix and Unix-like operating systems, while system("cls") is available on Windows.

3. remove() : To delete a given file**.**
4. rename() : To change the name of a file.
5. bool : bool is a keyword used to declare a boolean variable.

SCOPE:

By changing few lines in the program, the program could satisfy user requirement according to user

REFERENCE

1. **Problem solving and program design in c text book**
2. **“Let us c” book**