#include <stdio.h> ///for input output functions like printf, scanf

#include <stdlib.h>

#include <conio.h>

#include <windows.h> ///for windows related functions (not important)

#include <string.h> ///string operations

/\*\* List of Global Variable \*/

COORD coord = {0,0}; /// top-left corner of window

/\*\*

function : gotoxy

output: moves the cursor in specified position of console

\*/

void gotoxy(int x,int y)

{

coord.X = x;

coord.Y = y;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE),coord);

}

/\*\* Main function started \*/

void login()

{

int a=0,i=0;

char uname[10],c=' ';

char pword[10],code[10];

char user[10]="user";

char pass[10]="pass";

do

{

printf("\n :::::::::::::::::::::::::: LOGIN FORM :::::::::::::::::::::::::: ");

printf(" \n ENTER USERNAME:-");

scanf("%s", &uname);

printf(" \n ENTER PASSWORD:-");

while(i<10)

{

pword[i]=getch();

c=pword[i];

if(c==13) break;

else printf("\*");

i++;

}

pword[i]='\0';

i=0;

if(strcmp(uname,"user")==0 && strcmp(pword,"pass")==0)

{

printf(" \n\n\n WELCOME TO EMPLOYEE RECORD MANAGEMENT SYSTEM !!!! LOGIN IS SUCCESSFUL");

printf("\n LOADING PLEASE WAIT... \n");

for(i=0; i<3; i++)

{

printf(".");

Sleep(500);

}

printf("\n\n\n\t\t\t\tPress any key to continue...");

getch();

break;

}

else

{

printf("\n SORRY !!!! LOGIN IS UNSUCESSFUL");

a++;

getch();

}

}

while(a<=2);

if (a>2)

{

printf("\nSorry you have entered the wrong username and password for four times!!!");

getch();

}

system("cls");

}

int main()

{

int i=0;

login();

FILE \*fp, \*ft; /// file pointers

char another, choice;

/\*\* structure that represent a employee \*/

struct emp

{

char name[40];///name of employee

char id[40];

int age; /// age of employee

char address[20];

float bs; /// basic salary of employee

};

struct emp e;

char empid[40];

long int recsize;

fp = fopen("EMP.DAT","rb+");

if(fp == NULL)

{

fp = fopen("EMP.DAT","wb+");

if(fp == NULL)

{

printf("Cannot open file");

exit(1);

}

}

recsize = sizeof(e);

while(1)

{

system("cls");

printf(" \n :::::::::::::::::::::::::: |EMPLOYEES RECORD MANAGEMENT| :::::::::::::::::::::::::: \n");

gotoxy(30,05); /// move the cursor to postion 30, 10 from top-left corner

printf("1> Add Employee's Records");

gotoxy(30,07);

printf("2> Display Employee's Records");

gotoxy(30,9);

printf("3> Modify Employee's Records");

gotoxy(30,11);

printf("4> Delete Employee's Records");

gotoxy(30,13);

printf("5> Exit System");

gotoxy(30,15);

printf("Your Choice: ");

fflush(stdin);

choice = getche();

switch(choice)

{

case '1':

system("cls");

fseek(fp,0,SEEK\_END);

another = 'y';

while(another == 'y') /// if user want to add another record

{

printf("\nEnter Employee ID: ");

scanf("%s",e.id);

printf("\nEnter name: ");

scanf("%s",e.name);

printf("\nEnter age: ");

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

printf("\nEnter Address:");

scanf("%s",e.address);

printf("\nEnter basic salary: ");

scanf("%f", &e.bs);

fwrite(&e,recsize,1,fp); /// write the record in the file

printf("\nAdd another record(y/n) ");

fflush(stdin);

another = getche();

}

break;

case '2':

system("cls");

printf("EMPLOYEE's RECORD LIST (name, age, address, salary)");

rewind(fp); ///this moves file cursor to start of the file

printf("\n========================="

"==========================="

"======");

printf("\nID\t\tNAME\t\tAGE\t\tSALARY\t\tAddress");

printf("==========================="

"==========================="

"====\n");

while (fread(&e, recsize, 1, fp) == 1)

printf("\n%s\t\t%s\t\t%d\t\t%.2f\t\t\t%s",e.id,e.name, e.age,e.bs,e.address);

printf("\n\n\n\t");

system("pause");

break;

case '3': /// if user press 3 then do editing existing record

system("cls");

another = 'y';

while(another == 'y')

{

printf("\nEnter the employee ID to modify: ");

scanf("%s", empid);

rewind(fp);

while(fread(&e,recsize,1,fp)==1)

{

if(strcmp(e.id,empid) == 0) ///if entered name matches with that in file

{

printf("\nEnter new name: ");

scanf("%s",e.name);

printf("\nEnter new age: ");

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

printf("\nEnter new Address:");

scanf("%s",e.address);

printf("\nEnter new basic salary: ");

scanf("%f", &e.bs);

fseek(fp,-recsize,SEEK\_CUR);

fwrite(&e,recsize,1,fp); /// override the record

break;

}

}

printf("\nModify another record(y/n)");

fflush(stdin);

another = getche();

}

break;

case '4':

system("cls");

another = 'y';

while(another == 'y')

{

printf("\nEnter employee id to delete: ");

scanf("%s",empid);

ft = fopen("Temp.dat","wb");

rewind(fp);

while(fread(&e,recsize,1,fp) == 1) /// read all records from file

{

if(strcmp(e.id,empid) != 0) /// if the entered record match

{

fwrite(&e,recsize,1,ft); /// move all records except the one that is to be deleted to temp file

}

}

fclose(fp);

fclose(ft);

remove("EMP.DAT"); /// remove the orginal file

rename("Temp.dat","EMP.DAT"); /// rename the temp file to original file name

fp = fopen("EMP.DAT", "rb+");

printf("Delete another record(y/n)");

fflush(stdin);

another = getche();

}

break;

case '5':

fclose(fp); /// close the file

exit(0); /// exit from the program

}

}

return 0;

}#include <stdio.h> ///for input output functions like printf, scanf

#include <stdlib.h>

#include <conio.h>

#include <windows.h> ///for windows related functions (not important)

#include <string.h> ///string operations

/\*\* List of Global Variable \*/

COORD coord = {0,0}; /// top-left corner of window

/\*\*

function : gotoxy

output: moves the cursor in specified position of console

\*/

void gotoxy(int x,int y)

{

coord.X = x;

coord.Y = y;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE),coord);

}

/\*\* Main function started \*/

void login()

{

int a=0,i=0;

char uname[10],c=' ';

char pword[10],code[10];

char user[10]="user";

char pass[10]="pass";

do

{

printf("\n :::::::::::::::::::::::::: LOGIN FORM :::::::::::::::::::::::::: ");

printf(" \n ENTER USERNAME:-");

scanf("%s", &uname);

printf(" \n ENTER PASSWORD:-");

while(i<10)

{

pword[i]=getch();

c=pword[i];

if(c==13) break;

else printf("\*");

i++;

}

pword[i]='\0';

i=0;

if(strcmp(uname,"user")==0 && strcmp(pword,"pass")==0)

{

printf(" \n\n\n WELCOME TO EMPLOYEE RECORD MANAGEMENT SYSTEM !!!! LOGIN IS SUCCESSFUL");

printf("\n LOADING PLEASE WAIT... \n");

for(i=0; i<3; i++)

{

printf(".");

Sleep(500);

}

printf("\n\n\n\t\t\t\tPress any key to continue...");

getch();

break;

}

else

{

printf("\n SORRY !!!! LOGIN IS UNSUCESSFUL");

a++;

getch();

}

}

while(a<=2);

if (a>2)

{

printf("\nSorry you have entered the wrong username and password for four times!!!");

getch();

}

system("cls");

}

int main()

{

int i=0;

login();

FILE \*fp, \*ft; /// file pointers

char another, choice;

/\*\* structure that represent a employee \*/

struct emp

{

char name[40];///name of employee

char id[40];

int age; /// age of employee

char address[20];

float bs; /// basic salary of employee

};

struct emp e;

char empid[40];

long int recsize;

fp = fopen("EMP.DAT","rb+");

if(fp == NULL)

{

fp = fopen("EMP.DAT","wb+");

if(fp == NULL)

{

printf("Cannot open file");

exit(1);

}

}

recsize = sizeof(e);

while(1)

{

system("cls");

printf(" \n :::::::::::::::::::::::::: |EMPLOYEES RECORD MANAGEMENT| :::::::::::::::::::::::::: \n");

gotoxy(30,05); /// move the cursor to postion 30, 10 from top-left corner

printf("1> Add Employee's Records");

gotoxy(30,07);

printf("2> Display Employee's Records");

gotoxy(30,9);

printf("3> Modify Employee's Records");

gotoxy(30,11);

printf("4> Delete Employee's Records");

gotoxy(30,13);

printf("5> Exit System");

gotoxy(30,15);

printf("Your Choice: ");

fflush(stdin);

choice = getche();

switch(choice)

{

case '1':

system("cls");

fseek(fp,0,SEEK\_END);

another = 'y';

while(another == 'y') /// if user want to add another record

{

printf("\nEnter Employee ID: ");

scanf("%s",e.id);

printf("\nEnter name: ");

scanf("%s",e.name);

printf("\nEnter age: ");

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

printf("\nEnter Address:");

scanf("%s",e.address);

printf("\nEnter basic salary: ");

scanf("%f", &e.bs);

fwrite(&e,recsize,1,fp); /// write the record in the file

printf("\nAdd another record(y/n) ");

fflush(stdin);

another = getche();

}

break;

case '2':

system("cls");

printf("EMPLOYEE's RECORD LIST (name, age, address, salary)");

rewind(fp); ///this moves file cursor to start of the file

printf("\n========================="

"==========================="

"======");

printf("\nID\t\tNAME\t\tAGE\t\tSALARY\t\tAddress");

printf("==========================="

"==========================="

"====\n");

while (fread(&e, recsize, 1, fp) == 1)

printf("\n%s\t\t%s\t\t%d\t\t%.2f\t\t\t%s",e.id,e.name, e.age,e.bs,e.address);

printf("\n\n\n\t");

system("pause");

break;

case '3': /// if user press 3 then do editing existing record

system("cls");

another = 'y';

while(another == 'y')

{

printf("\nEnter the employee ID to modify: ");

scanf("%s", empid);

rewind(fp);

while(fread(&e,recsize,1,fp)==1)

{

if(strcmp(e.id,empid) == 0) ///if entered name matches with that in file

{

printf("\nEnter new name: ");

scanf("%s",e.name);

printf("\nEnter new age: ");

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

printf("\nEnter new Address:");

scanf("%s",e.address);

printf("\nEnter new basic salary: ");

scanf("%f", &e.bs);

fseek(fp,-recsize,SEEK\_CUR);

fwrite(&e,recsize,1,fp); /// override the record

break;

}

}

printf("\nModify another record(y/n)");

fflush(stdin);

another = getche();

}

break;

case '4':

system("cls");

another = 'y';

while(another == 'y')

{

printf("\nEnter employee id to delete: ");

scanf("%s",empid);

ft = fopen("Temp.dat","wb");

rewind(fp);

while(fread(&e,recsize,1,fp) == 1) /// read all records from file

{

if(strcmp(e.id,empid) != 0) /// if the entered record match

{

fwrite(&e,recsize,1,ft); /// move all records except the one that is to be deleted to temp file

}

}

fclose(fp);

fclose(ft);

remove("EMP.DAT"); /// remove the orginal file

rename("Temp.dat","EMP.DAT"); /// rename the temp file to original file name

fp = fopen("EMP.DAT", "rb+");

printf("Delete another record(y/n)");

fflush(stdin);

another = getche();

}

break;

case '5':

fclose(fp); /// close the file

exit(0); /// exit from the program

}

}

return 0;

}