Project Report:



I.T. DEPARTMENT

MANAGEMENT SYSTEM

IN

C PROGRAMMING

**Submitted to: Sir Shahroz Bakht**

**By:**

**Abdullah Muhammad Ibrahim (SP20-BSCS-0027)**

**Babar Hanif (SP20-BSCS-0020)**

**Basim Ahmad (SP20-BSCS-0026)**

TABLE OF CONTENTS

|  |  |
| --- | --- |
| **#** Topic | Page |
| 1. Introduction ………………………………………… 2. Source Code ………………………………………… 3. Algorithm ……………………………………………. 4. Results ………………………………………………… | 3  4  17  26 |

Introduction:

Managing any kind of member record is a big responsibility to perform. In complicated task like this one can have some errors which sometimes results in losses. As there are a large number of different members from this department from different faculties with their records, managing all this manually is not easy for a human. So we made a management system that is written in C-Language, which is the mother of all languages, to make data management easy for everyone. The program stores and use structure to store values and then add it, search it and print it from a file name “**record\_nxt.txt**” and can also modified and delete data by using another file “**test.txt**”. This file is also used to overwrite (edit) the data from the “**record\_nxt.txt**” when someone input the same data that already existed in the file. The use also had the choice to modify, replace or keep the values as it is.

Source Code:

#include<windows.h>

#include<stdio.h>

#include<conio.h>

#include <stdlib.h>

#include<string.h> //contains strcmp(),strcpy(),strlen(),etc

#include<ctype.h> //contains toupper(), tolower(),etc

#include<dos.h>

#include<time.h>

void password();

void mainmenu(void);

void addmember(void);

void deletemember(void);

void searchmember(void);

void viewmember(void);

void editmember(void);

void returnfunc(void);

int t(void);

FILE \*fp,\*ft; ///list of global files

int fm;

struct member{ //Every member info will be managed through this structure.

int ID;

char name[30];

char add[40];

char dept[20];

char Phone\_no[20];

};

struct member stu;

int main(){

password();

getch();

return 0;

}

void mainmenu(){

system("cls");

int choice;

//CREDITS:

printf("\n\t\t\t\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd[ IT DEPT. MANAGEMENT SYSTEM ]\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n\n");

printf("\t\t\t D E S I G N E D A N D D E V E L O P E D BY:\n\n");

printf("\t\t\t ABDULLAH M. IBRAHIM\t BASIM AHMAD\tBABAR HANIF\n\n");

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

//MAIN-MENU:

printf("\n\n\t\t\t\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd[ MAIN MENU ]\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n");

printf("\t\t\t1. Add Members\n\n");

printf("\t\t\t2. Delete Members\n\n");

printf("\t\t\t3. Search Members\n\n");

printf("\t\t\t4. View Members\n\n");

printf("\t\t\t5. Edit Members Information\n\n");

printf("\t\t\t6. Close application\n");

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

t();

printf("\t\t\tEnter Your Choice: ");

scanf("%d",&choice);

switch(choice){

case 1:

addmember();

break;

case 2:

deletemember();

break;

case 3:

searchmember();

break;

case 4:

viewmember();

break;

case 5:

editmember();

break;

case 6:{

system("cls");

printf("\n\n\t\t\t\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd[ IT DEPT. MANAGEMENT SYSTEM ]\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n");

printf("\n\t\t\t\t THANK YOU FOR USING OUR PROGRAM");

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

exit(0);

}

default:{

printf("\n\t\t\tInvalid Option!\a");

if (getch())

mainmenu();

}

}

//return 0;

}

void addmember(void){

system("cls");

printf("\n\n\t\t\t\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd[ ADD MEMBER INFO ]\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n\n");

FILE \*fp;

fp = fopen("record\_nxt.txt","ab+"); // creates/opens '.txt' extension file for adding member information.

if(fp == NULL){ //Warning if there is any error in opening the file

MessageBox(0,"\t\t\tError in Opening file\nMake sure your file is not write protected","Warning",0);

}

else{

///fflush(stdin);

//Taking input member information

printf("\t\t\tMember ID: ");scanf("%d",&stu.ID);

fflush(stdin);

printf("\n\t\t\tName: ");gets(stu.name);

printf("\n\t\t\tAddress: ");gets(stu.add);

printf("\n\t\t\tdept. name: ");gets(stu.dept);

printf("\n\t\t\tPhone Number: ");gets(stu.Phone\_no);

fwrite(&stu, sizeof(stu), 1, fp);

}

fclose(fp);

printf("\n\t\t\tThe record is sucessfully added\n");

printf("\t\t\tAdd another record? [Y/N]");

if(getch()=='n')

mainmenu();

else

system("cls");

addmember();

}

void deletemember( ){

system ("cls");

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

// printf("\xcd\xcd\xcd\xcd Delete Member \xcd\xcd\xcd\xcd\n\n");

int d;

char another='y';

while(another=='y') //loop re-iterates until user presses 'y' for yes in the dialogue box.

{

system("cls");

printf("\n\n\t\t\t\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd[ DELETE MEMBER ]\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n");

printf("\n\t\t\tEnter the Member ID to delete:");

scanf("%d",&d);

fp=fopen("record\_nxt.txt","rb+");

rewind(fp); ///reposition file pointer to stream's beggining so the member information (entire row) can be deleted from the start

while(fread(&stu,sizeof(stu),1,fp)==1){

if(stu.ID==d){

printf("\n\t\t\tMember name is %s",stu.name);

fm='Z';

}

}

if(fm!='Z'){ //checks if record is available or not.

printf("\n\t\t\tNo record is found in LIST");

if(getch())

mainmenu();

}

if(fm=='Z' ){

printf("\n\t\t\tThe Member is available.\n");

printf("\t\t\tDo you want to permanently delete this member? [Y/N]:");

if(getch()=='y'){

ft=fopen("test.txt","wb+"); //creates a temporary file for deleting

rewind(fp);

while(fread(&stu,sizeof(stu),1,fp)==1){

if(stu.ID!=d){

///fseek(ft,0,SEEK\_CUR);

fwrite(&stu,sizeof(stu),1,ft); //write all in tempory file except the record we want to delete

}

}

fclose(ft);

fclose(fp);

remove("record\_nxt.txt");

rename("test.txt","record\_nxt.txt"); //copies all item from temporary file to record file after the record of our choice is deleted

fp=fopen("record\_nxt.txt","rb+");

if(fm=='Z'){

printf("\n\t\t\tThe record has been sucessfully deleted.");

printf("\n\t\t\tDelete another record? [Y/N]: ");

}

}

else

mainmenu();

fflush(stdin);

another=getch();

}

}

mainmenu();

///returnfunc();

}

void searchmember(void){ //This function is used for searching any member in the record by his/her ID.

system("cls");

int s;

printf("\n\n\t\t\t\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd[ SEARCH MEMBER ]\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n\n");

printf("\t\t\tEnter the Member ID to search:");

scanf("%d",&s);

fp=fopen("record\_nxt.txt","rb+");

rewind(fp); ///reposition file pointer to stream's beggining then displays the member information if found.

while(fread(&stu,sizeof(stu),1,fp)==1){

if(stu.ID==s){

printf("\t\t\tThe Member is available.");

printf("\n\n\t\t\tID: %d",stu.ID);

printf("\n\n\t\t\tName: %s",stu.name);

printf("\n\n\t\t\tAddress: %s",stu.add);

printf("\n\n\t\t\tDept.: %s",stu.dept);

printf("\n\n\t\t\tPhone no: %s",stu.Phone\_no);

fm='Z';

}

}

if(fm!='Z'){

printf("\t\t\tNo record is found in LIST.");

if(getch())

mainmenu();

}

{

printf("");

fclose(fp);

returnfunc();

getch();

}

}

void viewmember(void){ //Views all the records of members in the form of tables in our database.

system("cls");

FILE \*fp;

int i=0,j;

system("cls");

printf("\n\n\t\t\t\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd[ MEMBERS LIST ]\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n\n");

printf("+------------+---------------------+---------------------------------+----------+-------------------+\n");

printf("| MEM\_ID | MEMBER NAME | ADDRESS | DEPT | PHONE\_NO |\n");

printf("+------------+---------------------+---------------------------------+----------+-------------------+\n");

j=5;

fp=fopen("record\_nxt.txt","rb");

while(fread(&stu,sizeof(stu),1,fp)==1){

printf(" %6d",stu.ID);

printf("\t%s",stu.name);

printf("\t\t%s",stu.add);

printf("\t\t%s",stu.dept);

printf("\t\t%s",stu.Phone\_no);

printf("\n");

j++;

i=i+1;

printf("-----------------------------------------------------------------------------------------------------\n");

}

printf("\n\t\t\tTotal Members = %d",i);

fclose(fp);

returnfunc();

getch();

}

void editmember(void){ //For editing the member information by verifying their ID

system("cls");

int s,print=37;

printf("\n\n\t\t\t\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd[ EDIT MEMBER INFO ]\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n");

printf("\n\t\t\tEnter the Member ID to edit: ");

scanf("%d",&s);

fp=fopen("record\_nxt.txt","rb+");

///rewind(fp); ///reposition file pointer to stream's beggining

while(fread(&stu,sizeof(stu),1,fp)==1) {

if(stu.ID==s){

printf("\n\t\t\tThe Member is available\n");

fflush(stdin);

printf("\n\t\t\tName: ");gets(stu.name);

printf("\n\t\t\tAddress: ");gets(stu.add);

printf("\n\t\t\tdept. name: ");gets(stu.dept);

printf("\n\t\t\tPhone Number: ");gets(stu.Phone\_no);

printf("\n\t\t\tThe record is modified");

fseek(fp,-sizeof(stu), SEEK\_CUR);

fwrite(&stu,sizeof(stu), 1, fp);

fm=1;

break;

}

}

if(fm!=1){

printf("\n\t\t\tNo record is found in List");

if(getch())

mainmenu();

}

fclose(fp);

returnfunc();

getch();

}

void returnfunc(void){ //This function is used to prompt the user to press Enter key in order to return to the mainmenu.

printf("\n\n\t\t\tPress [Enter] to return to the main menu.");

if(getch()==13)

mainmenu();

}

int t(void){ // used to display the current local-time (of mainmenu being displayed)

time\_t t;

time(&t);

printf("Date and time: %s\n",ctime(&t));

return 0;

}

void fordelay(int j){ // This loop doesnt display anything, but it will run a very huge number of times (it is used in the "Loading . . ." text)

int i,k;

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

k=i;

}

void password(void){

system("cls");

char pass[10],ch,password[10]="ABB"; //Password is entirely Uppercase.

int i=0;

printf("\n\n\t\t\t\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd[ IT DEPT. MANAGEMENT SYSTEM ]\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n\n");

printf("\t\t\tEnter the password to Login: ");

while(ch!=13){

ch=getch();

if(ch!=13 && ch!=8){ //13 = Enter key , 8 = backspace

putch('\*'); //characters will be replaced by starix '\*'

pass[i] = ch;

i++;

}

}

pass[i] = '\0';

if(strcmp(pass,password)==0){

printf("\n\n\t\t\tPassword Matched!!");

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

for(i=0;i<=6;i++){

fordelay(100000000); //after every 100000000th loop count, program will print a "." after every few miliseconds because the loop runs for a larger amount so it obviously will take time

printf(".");

}

system("cls");

mainmenu();

}

else

printf("\n\t\t\tPassword is invalid!!\a");

printf("\n\t\t\tTry again.");

getch();

main();

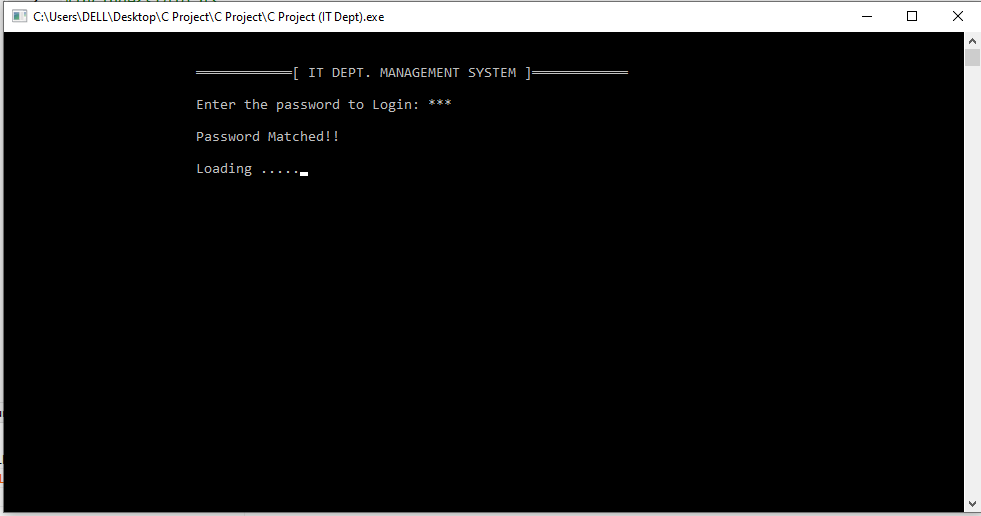
}

Algorithm:

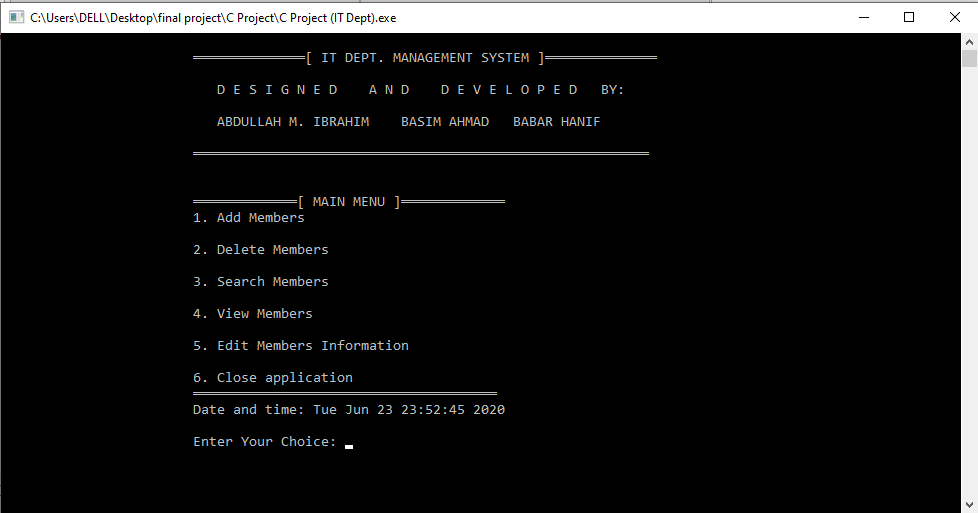
1. The program starts with a title screen in the main function which prompts the user for Password in order to access the program. If the password is wrong it asks the user for the password again after pressing any key.
2. If the password is matched, then it displays “**Loading . . .**” text which works through a loop and then mainmenu is displayed.
3. A Credit section is added on the main menu which shows the name of our project with the names of its developer’s.
4. A time function is also developed which displays on the bottom of the menu the current time and date of when the main menu is displayed.
5. In the menu the values are taken in switch-case condition and then according to the number bullets printed on the screen in the menu the program executes that specific case of the menu.
6. for Adding members function, inputs are taken in a structure and then values are added with file named **“record\_nxt.txt”** if there’s no such file then it creates a file with the same name and hence the record is addedand then asks the user whether he wants to add another member or not, if user says no, then user is returned to the mainmenu.
7. For Deleting members function, program asks user to input member ID, then it searches if there’s any record in the file existing with the given ID (same thing is done in the Search member function which displays info if the member is found), if it exists then program confirms if user wants to delete the record, when user confirms to delete, it creates a new temporary file called **“test.txt”** which copies all the records from the main record file except the record which is about to be deleted when the process is done, old record file is deleted and the **“test.txt”** is renamed to **“record\_nxt.txt”.** after the task is done, return function is called.
8. To edit members, its function confirms the ID of the member from user which he/she wants to edit, if the record is found then it asks for every information in the record (e.g. name, number etc). after the task is done, return function is called
9. The function which is used to view members will display all the members in the record file in the form of tables with the total number of records as-well.
10. The return function will work with a message asking user to return to the mainmenu by pressing Enter key. If the user presses enter, program directs user to the mainmenu.
11. If the user enter the wrong option in any function, the program will ask the user to enter your input again.
12. Exit function displays a box which thanks the user for using the program and performs exit function by pressing any key.

Result:

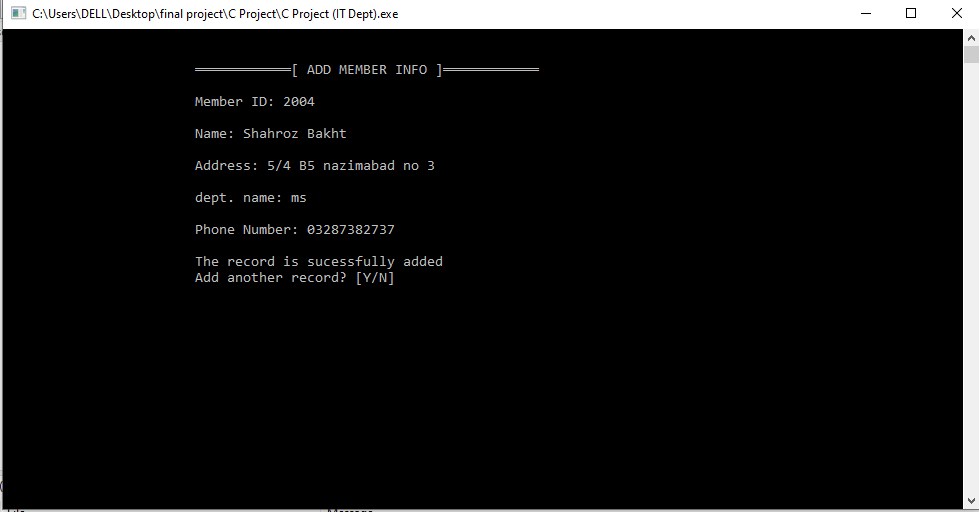
1. SS#1



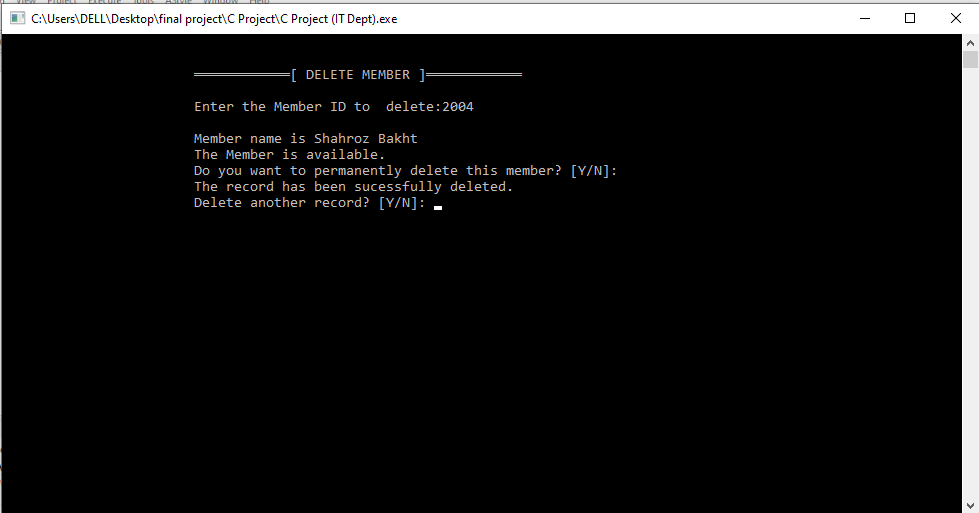
1. SS#2



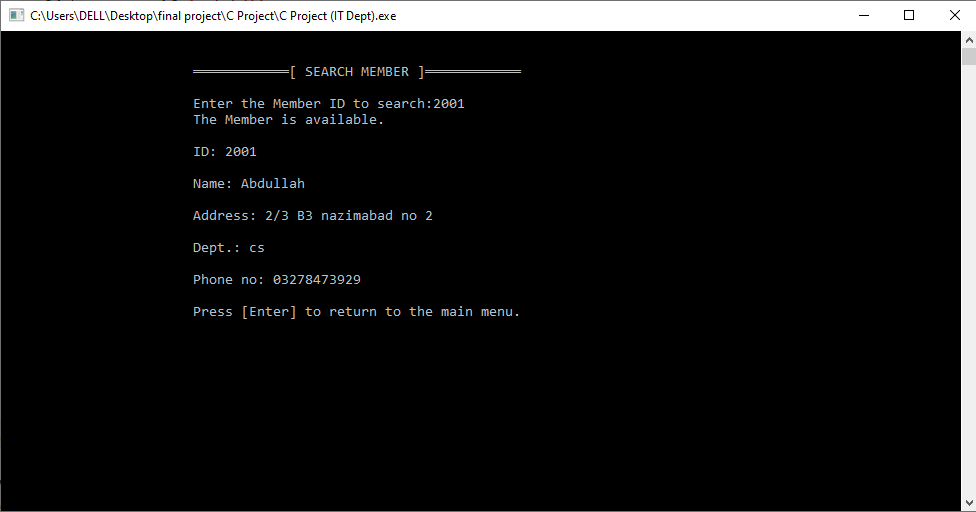
1. SS#3



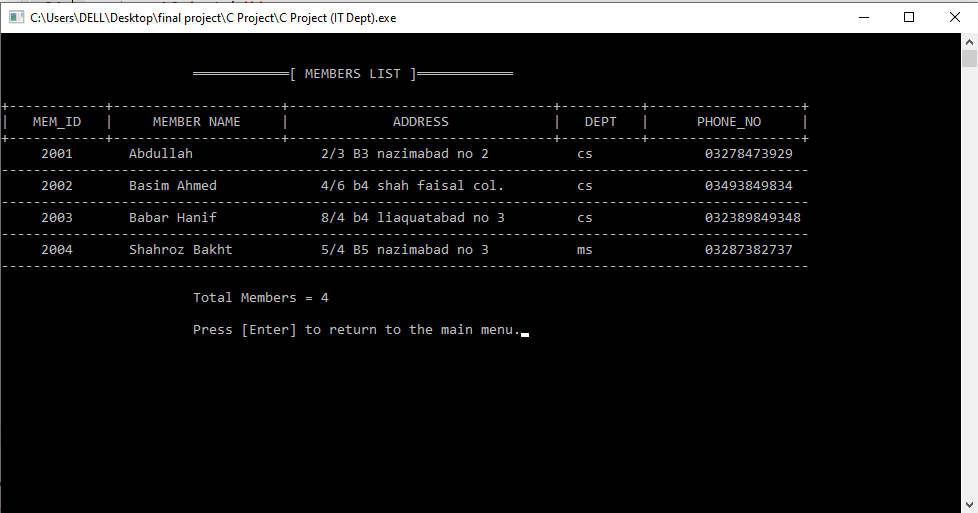
1. SS#4



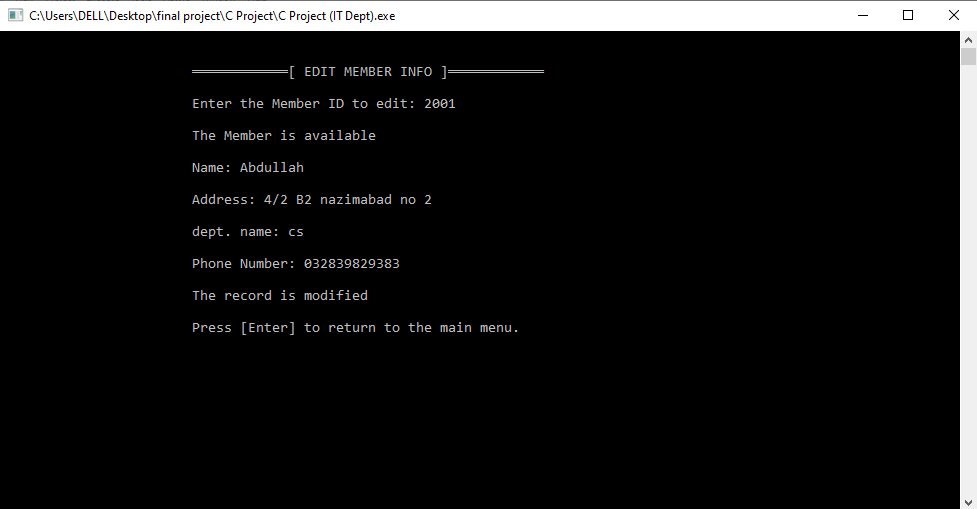
1. SS#5



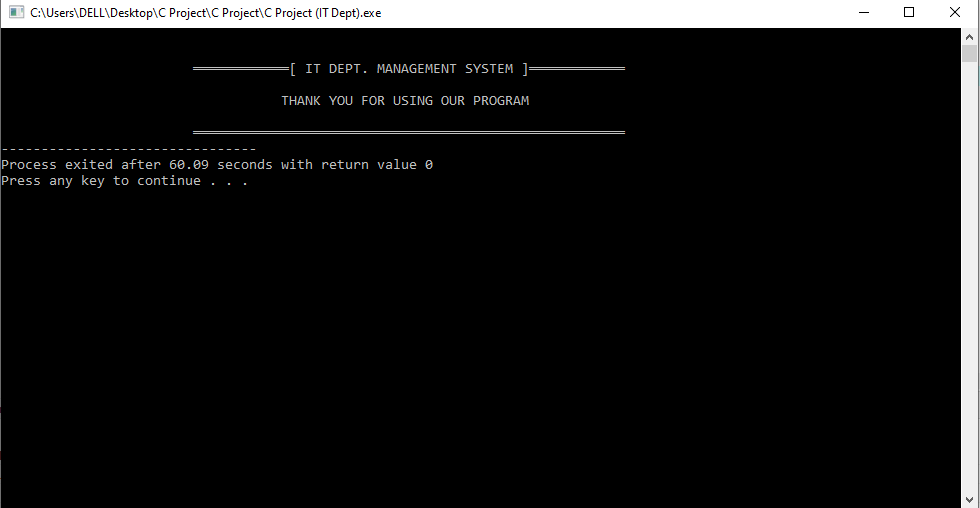
1. SS#6



1. SS#7



1. SS#8



**THANK YOU!**