**CONTACT MANAGEMENT SYSTEM:**

**NAME:** SAI CHARAN KASAVARAJU

**REG NO:** RA2111009010074

**NAME:** LAUREN SHAREL CHRISTOPHER

**REG NO:** RA2111009010075

**CODE:**

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<process.h>

#include<stdlib.h>

#include<dos.h>

struct contact

{

long ph;

char name[20],add[20],email[30];

}list;

char query[20],name[20];

FILE \*fp, \*ft;

int i,n,ch,l,found;

int main()

{

main:

system("cls"); /\* \*\*\*\*\*\*\*\*\*\*\*\*Main menu \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

printf("\n\t \*\*\*\* Welcome to contact Management System \*\*\*\*");

printf("\n\n\n\t\t\tMAIN MENU\n\t\t=====================\n\t\t[1] Add a new Contact\n\t\t[2] List all Contacts\n\t\t[3] Search for contact\n\t\t[4] Edita Contact\n\t\t[5] Delete a Contact\n\t\t[0] Exit\n\t\t=================\n\t\t");

printf("Enter the choice:");

scanf("%d",&ch);

switch(ch)

{

case 0:

printf("\n\n\t\tAre you sure u want to exit?");

break;

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*add new contacts\*\*\*\*\*\*\*\*\*\*\*\* \*/

case 1:

system("cls");

fp=fopen("contact.dll","a");

for (;;)

{ fflush(stdin);

printf("To exit enter blank space in the name input\nName (Use identical):");

scanf("%[^\n]",&list.name);

if(stricmp(list.name,"")==0 || stricmp(list.name," ")==0)

break;

fflush(stdin);

printf("Phone:");

scanf("%ld",&list.ph);

fflush(stdin);

printf("address:");

scanf("%[^\n]",&list.add);

fflush(stdin);

printf("email address:");

fgets(list.email,30,stdin);

printf("\n");

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

}

fclose(fp);

break;

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*list of contacts\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

case 2:

system("cls");

printf("\n\t\t================================\n\t\t\tLIST OF CONTACTS\n\t\t================================\n\nName\t\tPhone No\t Address\t\tE-mail ad.\n=================================================================\n\n");

for(i=97;i<=122;i=i+1)

{

fp=fopen("contact.dll","r");

fflush(stdin);

found=0;

while(fread(&list,sizeof(list),1,fp)==1)

{

if(list.name[0]==i || list.name[0]==i-32)

{

printf("\nName\t: %s\nPhone\t: %ld\nAddress\t: %s\nEmail\t: %s\n",list.name,

list.ph,list.add,list.email);

found++;

}

}

if(found!=0)

{

printf("=========================================================== [%c]-(%d)\n\n",i-32,found);

getch();

}

fclose(fp);

}

break;

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*search contacts\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

case 3:

system("cls");

do

{

found=0;

printf("\n\n\t..::CONTACT SEARCH\n\t===========================\n\t..::Name of contact to search: ");

fflush(stdin);

scanf("%[^\n]",&query);

l=strlen(query);

fp=fopen("contact.dll","r");

system("cls");

printf("\n\n..::Search result for '%s' \n===================================================\n",query);

while(fread(&list,sizeof(list),1,fp)==1)

{

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

name[i]=list.name[i];

name[l]='\0';

if(stricmp(name,query)==0)

{

printf("\n..::Name\t: %s\n..::Phone\t: %ld\n..::Address\t: %s\n..::Email\t:%s\n",list.name,list.ph,list.add,list.email);

found++;

if (found%4==0)

{

printf("..::Press any key to continue...");

getch();

}

}

}

if(found==0)

printf("\n..::No match found!");

else

printf("\n..::%d match(s) found!",found);

fclose(fp);

printf("\n ..::Try again?\n\n\t[1] Yes\t\t[0] No\n\t");

scanf("%d",&ch);

}while(ch==1);

break;

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*edit contacts\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

case 4:

system("cls");

fp=fopen("contact.dll","r");

ft=fopen("temp.dat","w");

fflush(stdin);

printf("..::Edit contact\n===============================\n\n\t..::Enter the name of contact to edit:");

scanf("%[^\n]",name);

while(fread(&list,sizeof(list),1,fp)==1)

{

if(stricmp(name,list.name)!=0)

fwrite(&list,sizeof(list),1,ft);

}

fflush(stdin);

printf("\n\n..::Editing '%s'\n\n",name);

printf("..::Name(Use identical):");

scanf("%[^\n]",&list.name);

fflush(stdin);

printf("..::Phone:");

scanf("%ld",&list.ph);

fflush(stdin);

printf("..::address:");

scanf("%[^\n]",&list.add);

fflush(stdin);

printf("..::email address:");

fgets(list.email,30,stdin);

printf("\n");

fwrite(&list,sizeof(list),1,ft);

fclose(fp);

fclose(ft);

remove("contact.dll");

rename("temp.dat","contact.dll");

break;

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*delete contacts\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

case 5:

system("cls");

fflush(stdin);

printf("\n\n\t..::DELETE A CONTACT\n\t==========================\n\t..::Enter the name of contact to delete:");

scanf("%[^\n]",&name);

fp=fopen("contact.dll","r");

ft=fopen("temp.dat","w");

while(fread(&list,sizeof(list),1,fp)!=0)

if (stricmp(name,list.name)!=0)

fwrite(&list,sizeof(list),1,ft);

fclose(fp);

fclose(ft);

remove("contact.dll");

rename("temp.dat","contact.dll");

break;

default:

printf("Invalid choice");

break;

}

printf("\n\n\n..::Enter the Choice:\n\n\t[1] Main Menu\t\t[0] Exit\n");

scanf("%d",&ch);

switch (ch)

{

case 1:

goto main;

case 0:

break;

default:

printf("Invalid choice");

break;

}

return 0;

}

**VISUAL STUDIO CODE:**

**Graphical user interface, text

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**OUTPUT:**

**FORMAT:**

1. Add a new Contact
2. List all Contacts
3. Search for Contact
4. Edit a Contact
5. Delete a Contact
6. Exit

Enter the choice:

Text

Description automatically generated

Text

Description automatically generated

After adding new contact, we can list out all contacts, search for contact, edit a contact, delete a contact by entering the respective cases.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Final list of contacts after editing

Text

Description automatically generated

Exiting(ending) the program via the exit code. (case 0)

Text

Description automatically generated