Computer Science Project File

The Matrimony APP

# Name: Rohil Kulshreshtha

# Class: XII-D

# Roll Number: 02

# Acad. Session: 2017-18

# Ahlcon Public School

# INDEX

1. Certificate
2. Acknowledgement
3. About the project
4. Hardware and software info
5. Classes and functions used
6. Binary file used
7. Header files used
8. Source code
9. Output screenshots
10. Bibliography

**Certificate**

The program described in this project file has been satisfactorily developed by **Rohil Kulshreshtha** studying in **Ahlcon Public School**, class **12th –D**, Roll No. **02** During the academic year **2017-18.**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Teacher’s Signature

**Acknowledgement**

In the accomplishment of the project successfully, I would like to express my special thanks of gratitude to my teacher Mr. Jagdish Devrani who gave me this golden opportunity to do this wonderful project on the topic “Database management application for Matrimony app”,

Which also helped me in doing a lot of research and I came to know about so many new things. Secondly, I would like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

**About The Project**

This program is essentially a simple database management system which can be used in on itself to create a database entry regarding matrimonial details of a person, or it can be utilized by other programs as a database access interface.

Each entry added to the database is uniquely password protected and the program is quite abstract in its working, talking full advantage of the Object oriented approach of the C++ language on which this program is built upon.

**Shortcomings of the program:**

* The program’s database file is easily available and although the program is password protected the file can be seen by anyone.
* The program might encounter error if 2 database entries are stored with the same username and password.(a unique code could be provided to entries to remove this discrepancy.)
* The program is very basic in features and it has the potential to become more vibrant.

**Hardware and Software Requirements**

The program requires minimal specification of essential hardwares like RAM and Processor for its functioning. A console interface is needed to display the program’s content and to communicate with it.

For Software, the console on which the program is run must have a compiler for C++ language it can be as basic as the one commonly included in the operating system of the console device.

**Resources used**

HEADER FILES

The following header files were used in the making of this program:

1. fstream.h
2. conio.h
3. stdio.h
4. string.h
5. process.h
6. dos.h

FUNCTIONS AND CLASSES

The main class used was matrimony whose declaration can be seen in the source code.

The self-declared functions involved are as follows:

Void input() -to take general input from console

Void output() -to display general output to console

Void addrecord()

Void displayrecord()

Void deleterecord()

Void updaterecord()

Abstract Functions:

char \*getname();

char \*getdob();

char \*getprof();

char \*getpassw();

char \*getgen();

char \*getreligion();

char \*getqualif();

char \*getlang();

char \*getstat();

char \*getpno();

char \*getid();

void setpassw(char []);

void setname(char []);

void setdob(char []);

void setprof(char []);

void setgen(char []);

void setreligion(char []);

void setqualif(char []);

void setlang(char []);

void setstat(char []);

void setpno(char []);

void setid(char []);

Binary file used as record is named as “matrimony.dat”

**Source code**

#include <fstream.h>

#include <conio.h>

#include <stdio.h>

#include <string.h>

#include <process.h>

#include <dos.h>

class matrimony

{

char passw[15],name[30],gen[20],prof[20],religion[20],lang[20],qualif[20],pno[20],stat[20],id[40],dob[20];

public:

void input();

void output();

char \*getname();

char \*getdob();

char \*getprof();

char \*getpassw();

char \*getgen();

char \*getreligion();

char \*getqualif();

char \*getlang();

char \*getstat();

char \*getpno();

char \*getid();

void setpassw(char []);

void setname(char []);

void setdob(char []);

void setprof(char []);

void setgen(char []);

void setreligion(char []);

void setqualif(char []);

void setlang(char []);

void setstat(char []);

void setpno(char []);

void setid(char []);

};

void matrimony::input()

{

cout<<"Enter Name"<<endl;

gets(name);

cout<<"Enter a suitable password"<<endl;

gets(passw);

cout<<"Enter Date of Birth as dd/mm/yyyy"<<endl;

gets(dob);

cout<<"Enter Gender(M/F)"<<endl;

gets(gen);

cout<<"Enter Profession"<<endl;

gets(prof);

cout<<"Enter Religion"<<endl;

gets(religion);

cout<<"Enter Mother Tongue"<<endl;

gets(lang);

cout<<"Enter Qualification"<<endl;

gets(qualif);

cout<<"Enter Marital Status"<<endl;

gets(stat);

cout<<"Enter Phone Number"<<endl;

gets(pno);

cout<<"Enter Email-ID"<<endl;

gets(id);

}

void matrimony::output()

{

cout<<"Name : "<<name<<endl;

cout<<"Date of Birth : "<<dob<<endl;

cout<<"Gender : "<<gen<<endl;

cout<<"Profession : "<<prof<<endl;

cout<<"Religion : "<<religion<<endl;

cout<<"Mother tongue : "<<lang<<endl;

cout<<"Qualification : "<<qualif<<endl;

cout<<"Marital status : "<<stat<<endl;

cout<<"Phone Number : "<<pno<<endl;

cout<<"Email ID : "<<id<<endl;

cout<<endl;

}

char \*matrimony::getpassw()

{

return passw;

}

char \*matrimony::getid()

{

return id;

}

char \*matrimony::getpno()

{

return pno;

}

char \*matrimony::getstat()

{

return stat;

}

char \*matrimony::getqualif()

{

return qualif;

}

char \*matrimony::getlang()

{

return lang;

}

char \*matrimony::getreligion()

{

return religion;

}

char \*matrimony::getgen()

{

return gen;

}

char \*matrimony::getprof()

{

return prof;

}

char \*matrimony::getdob()

{

return dob;

}

char \*matrimony::getname()

{

return name;

}

void matrimony::setid(char E[])

{

strcpy(id,E);

}

void matrimony::setpno(char P[])

{

strcpy(pno,P);

}

void matrimony::setstat(char S[])

{

strcpy(stat,S);

}

void matrimony::setqualif(char Q[])

{

strcpy(qualif,Q);

}

void matrimony::setlang(char L[])

{

strcpy(lang,L);

}

void matrimony::setreligion(char R[])

{

strcpy(religion,R);

}

void matrimony::setgen(char G[])

{

strcpy(gen,G);

}

void matrimony::setprof(char O[])

{

strcpy(prof,O);

}

void matrimony::setdob(char D[])

{

strcpy(dob,D);

}

void matrimony::setname(char N[])

{

strcpy(name,N);

}

void matrimony::setpassw(char pass[])

{

strcpy(passw,pass);

}

int i=0,f=0,choice,flag,pos;

long int getptr;

char c,pass[15],pass1[15],N[30],N1[30],O[20],O1[20],G[20],Gender[20],R[20],R1[20],L[20],L1[20],Q[20],Q1[20],S[20],S1[20],D[20],D1[20],E[40],E1[40],P[20],P1[20],ans;

fstream f1,f2;

matrimony s1,s2;

void addrecord()

{

clrscr();

cout<<endl<<endl;

f1.open("mat.dat",ios::out|ios::app|ios::binary);

cout<<"Enter the information of new user to be added"<<endl<<endl;

s1.input();

f1.write((char \*) &s1,sizeof(s1));

cout<<"Record added"<<endl;

f1.close();

void displayrecord()

{

clrscr();

cout<<endl<<endl;

s1.output();

}

void searchrecord()

{

f1.open("mat.dat", ios::in | ios::out | ios::binary | ios :: beg);

clrscr();

cout<<endl<<endl<<"Do you want to search in male category or female category?(M/F)"<<endl;

gets(Gender);

cout<<endl<<endl;

cout<<" ­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­ "<<endl;

cout<<" > <"<<endl;

cout<<" > Press 1 to search a record based on Name <"<<endl;

cout<<" > Press 2 to search a record based on Profession <"<<endl;

cout<<" > Press 3 to search a record based on Religion <"<<endl;

cout<<" > Press 4 to search a record based on Mother Tongue <"<<endl;

cout<<" > Press 5 to search a record based on Qualification <"<<endl;

cout<<" > Press 6 to search a record based on Marital status <"<<endl;

cout<<" > <"<<endl;

cout<<" ‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑"<<endl;

cout<<endl<<endl;

cout<<"Enter your choice"<<endl;

cin>>choice;

clrscr();

switch(choice)

{

case 1:

cout<<"Enter the Name of person whose record is to be searched"<<endl;

gets(N1);

cout<<endl;

flag=0;

f1.read((char \*) &s2,sizeof(s2));

while(!f1.eof())

{

if(strcmpi(s2.getname(),N1)==0 && strcmpi(s2.getgen(),Gender)==0)

{

flag=1;

s2.output();

}

f1.read((char \*) &s2,sizeof(s2));

}

if(flag==0)

{

cout<<"Record does not exists"<<endl;

}

break;

case 2:

cout<<"Enter the Profession of person whose record is to be searched"<<endl;

gets(O1);

cout<<endl;

flag=0;

f1.read((char \*) &s2,sizeof(s2));

while(!f1.eof())

{

if(strcmpi(s1.getprof(),O1)==0 && strcmpi(s1.getgen(),Gender)==0)

{

flag=1;

s2.output();

}

f1.read((char \*) &s2,sizeof(s2));

}

if(flag==0)

{

cout<<"Record does not exists"<<endl;

}

break;

case 3:

cout<<"Enter the Religion of person whose record is to be searched"<<endl;

gets(R1);

cout<<endl;

flag=0;

f1.read((char \*) &s2,sizeof(s2));

while(!f1.eof())

{

if(strcmpi(s1.getreligion(),R1)==0 && strcmpi(s1.getgen(),Gender)==0)

{

flag=1;

s2.output();

}

f1.read((char \*) &s2,sizeof(s2));

}

if(flag==0)

{

cout<<"Record does not exists"<<endl;

}

break;

case 4:

cout<<"Enter the Mother Tongue of person whose record is to be searched"<<endl;

gets(L1);

cout<<endl;

flag=0;

f1.read((char \*) &s2,sizeof(s2));

while(!f1.eof())

{

if(strcmpi(s1.getlang(),L1)==0 && strcmpi(s1.getgen(),Gender)==0)

{

flag=1;

s2.output();

}

f1.read((char \*) &s2,sizeof(s2));

}

if(flag==0)

{

cout<<"Record does not exists"<<endl;

}

break;

case 5:

cout<<"Enter the Qualification of person whose record is to be searched"<<endl;

gets(Q1);

cout<<endl;

flag=0;

f1.read((char \*) &s2,sizeof(s2));

while(!f1.eof())

{

if(strcmpi(s1.getqualif(),Q1)==0 && strcmpi(s1.getgen(),Gender)==0)

{

flag=1;

s2.output();

}

f1.read((char \*) &s2,sizeof(s2));

}

if(flag==0)

{

cout<<"Record does not exists"<<endl;

}

break;

case 6:

cout<<"Enter the Marital Status of person whose record is to be searched"<<endl;

gets(S1);

cout<<endl;

flag=0;

f1.read((char \*) &s2,sizeof(s2));

while(!f1.eof())

{

if(strcmpi(s1.getstat(),S1)==0 && strcmpi(s1.getgen(),Gender)==0)

{

flag=1;

s2.output();

}

f1.read((char \*) &s2,sizeof(s2));

}

if(flag==0)

{

cout<<"Record does not exists"<<endl;

}

break;

default:

cout<<"Invalid number entered"<<endl;

break;

}

}

void updaterecord()

{

clrscr();

f1.seekg(pos-sizeof(s1),ios::beg);

f1.read((char \*)&s1,sizeof(s1));

cout<<"Enter the information to be updated"<<endl;

cout<<"old Name is "<<s1.getname()<<endl;

cout<<"Do you wish to update it ? (Y/N)"<<endl;

cin>>ans;

if(ans=='y' || ans=='Y')

{

cout<<"Enter new Name"<<endl;

gets(N);

s1.setname(N);

}

cout<<"old Date of Birth is "<<s1.getdob()<<endl;

cout<<"Do you wish to update it ? (Y/N)"<<endl;

cin>>ans;

if(ans=='y' || ans=='Y')

{

cout<<"Enter new Date of Birth"<<endl;

gets(D);

s1.setdob(D);

}

cout<<"old Gender is "<<s1.getgen()<<endl;

cout<<"Do you wish to update it ? (Y/N)"<<endl;

cin>>ans;

if(ans=='y' || ans=='Y')

{

cout<<"Enter new Gender"<<endl;

gets(G);

s1.setgen(G);

}

cout<<"old Profession is "<<s1.getprof()<<endl;

cout<<"do you wish to update it ?"<<endl;

cin>>ans;

if(ans=='y' || ans=='Y')

{

cout<<"Enter new Profession"<<endl;

gets(O);

s1.setprof(O);

}

cout<<"old Religion is "<<s1.getreligion()<<endl;

cout<<"Do you wish to update it ? (Y/N)"<<endl;

cin>>ans;

if(ans=='y' || ans=='Y')

{

cout<<"Enter new Religion"<<endl;

gets(R);

s1.setreligion(R);

}

cout<<"old Mother Tongue is "<<s1.getlang()<<endl;

cout<<"Do you wish to update it ? (Y/N)"<<endl;

cin>>ans;

if(ans=='y' || ans=='Y')

{

cout<<"Enter new Mother Tongue"<<endl;

gets(L);

s1.setlang(L);

}

cout<<"old Qualification is "<<s1.getqualif()<<endl;

cout<<"Do you wish to update it ? (Y/N)"<<endl;

cin>>ans;

if(ans=='y')

{

cout<<"Enter new Qualification"<<endl;

gets(Q);

s1.setqualif(Q);

}

cout<<"old Marital Status is "<<s1.getstat()<<endl;

cout<<"Do you wish to update it ? (Y/N)"<<endl;

cin>>ans;

if(ans=='y' || ans=='Y')

{

cout<<"Enter new Marital status"<<endl;

gets(S);

s1.setstat(S);

}

cout<<"old Phone Number is "<<s1.getpno()<<endl;

cout<<"Do you wish to update it ? (Y/N)"<<endl;

cin>>ans;

if(ans=='y' || ans=='Y')

{

cout<<"Enter new Phone Number"<<endl;

gets(P);

s1.setpno(P);

}

cout<<"old Email ID is "<<s1.getid()<<endl;

cout<<"Do you wish to update it ? (Y/N)"<<endl;

cin>>ans;

if(ans=='y' || ans=='Y')

{

cout<<"Enter new Email ID"<<endl;

gets(E);

s1.setid(E);

}

f1.clear();

f1.seekg(pos-sizeof(s1), ios :: beg );

f1.write((char \*)&s1, sizeof(s1));

cout<<"Record updated"<<endl;

}

void deleterecord()

{

cout<<"Are you sure you want to delete your record?(Y/N)"<<endl;

cin>>ans;

if(ans=='Y' || ans=='y')

{

strcpy(N1,s1.getname());

f1.seekg(0, ios :: beg);

f2.open("temp.dat", ios :: beg | ios :: out);

while(!f1.eof())

{

f1.read((char\*)&s2, sizeof(s2));

if(strcmpi(s2.getname(),N1)!=0)

{

f2.write((char\*)&s2,sizeof(s2));

}

}

f1.close();

f2.close();

remove("mat.dat");

rename("temp.dat","mat.dat");

cout<<"Information deleted press any key to exit"<<endl;

getch();

exit(-1);

}

}

void main()

{

clrscr();

cout<<endl<<endl<<endl<<endl<<endl<<endl<<endl;

cout<<" ­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­"<<endl;

cout<<" > <"<<endl;

cout<<" > <"<<endl;

cout<<" > Welcome to <"<<endl;

cout<<" > The Matrimony App <"<<endl;

cout<<" > <"<<endl;

cout<<" > Press 1 to enter the app <"<<endl;

cout<<" > Press 2 exit <"<<endl;

cout<<" > <"<<endl;

cout<<" ‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑"<<endl;

cout<<endl<<endl;

cout<<"Enter your choice"<<endl<<endl;

cin>>choice;

switch(choice)

{

case 1:

clrscr();

do

{

clrscr();

cout<<endl<<endl;

cout<<" ­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­"<<endl;

cout<<" > <"<<endl;

cout<<" > Press 1 If you want to add Record <"<<endl;

cout<<" > Press 2 If you want to use Record <"<<endl;

cout<<" > <"<<endl;

cout<<" ‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑"<<endl;

cout<<endl<<endl;

cout<<"Enter your choice"<<endl<<endl;

cin>>choice;

switch(choice)

{

case 1:

clrscr();

cout<<endl;endl;

addrecord();

break;

case 2:

x:

i=0;

clrscr();

cout<<"Enter username"<<endl;

gets(N1);

cout<<"Enter password"<<endl;

while(i<20)

{

pass1[i]=getch();

if(pass1[i]==13)

{

pass1[i]='\0';

break;

}

else if(pass1[i]==8)

{

cout<<"\b \b";

i--;

}

else

{

cout<<"\*";

i++;

}

}

f1.open("mat.dat", ios::in | ios::out | ios::binary | ios :: beg);

if(!f1)

{

cout<<"file cannot be opened !!\n";

getch();

exit(-1);

}

else

while(!f1.eof())

{

f1.read((char\*)&s1, sizeof(s1));

if(strcmp(s1.getname(),N1)==0 && strcmp(s1.getpassw(),pass1)==0)

{

pos=f1.tellg();

do

{

clrscr();

f=1;

f1.seekg(pos-sizeof(s1),ios::beg); //sets it to users block

f1.read((char \*)&s1,sizeof(s1)); //reads users content and stores in s1

f1.seekg(0,ios::beg); //to the beginning of file

cout<<endl<<endl;

cout<<" ­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­"<<endl;

cout<<" > <"<<endl;

cout<<" > Press 1 to display your record <"<<endl;

cout<<" > Press 2 to search a record <"<<endl;

cout<<" > Press 3 to delete your record <"<<endl;

cout<<" > Press 4 to update your record <"<<endl;

cout<<" > Press 5 to exit <"<<endl;

cout<<" > <"<<endl;

cout<<" ‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑­‑"<<endl;

cout<<endl<<endl<<"Enter your choice"<<endl<<endl;

cin>>choice;

clrscr();

switch(choice)

{

case 1:

displayrecord();

break;

case 2:

f1.close();

searchrecord();

break;

case 3:

deleterecord();

break;

case 4:

updaterecord();

break;

case 5:

f1.close();

exit(-1);

break;

default:

cout<<"Invalid number entered"<<endl;

break;

}

cout<<"\n \n"; business

cout<<"Do you wish to continue(Y/N)"<<endl;

cin>>ans;

}while(ans=='y'||ans=='Y');

}

}

f1.close();

if(f==0)

{

cout<<endl;

cout<<"Wrong username or password combination."<<endl;

cout<<"Do you want to input again (Y/N)"<<endl;

cin>>ans;

if(ans=='y'|| ans=='Y')

{

goto x;

}

else

{

exit(-1);

}

}

f1.close();

break;

default:

cout<<"Invalid number entered"<<endl;

break;

}

cout<<endl<<endl;

cout<<"Do you wish to continue?(Y/N)"<<endl;

cin>>ans;

}while(ans=='y'|| ans=='Y');

break;

case 2:

clrscr();

exit(-1);

break;

default:

cout<<"Invalid number entered"<<endl;

break;

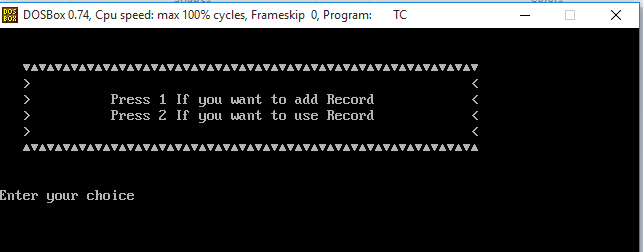
}

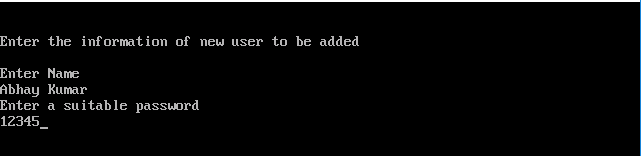
getch();

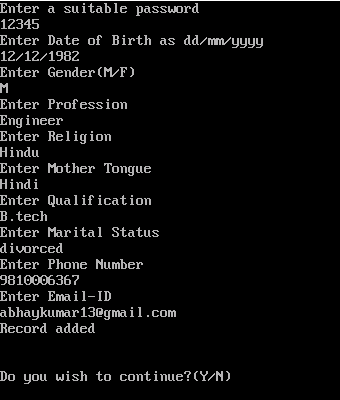
}

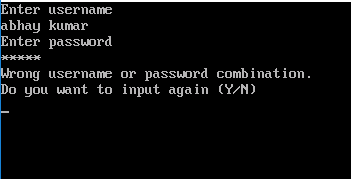
**OUTPUT SCREENSHOTS**

****

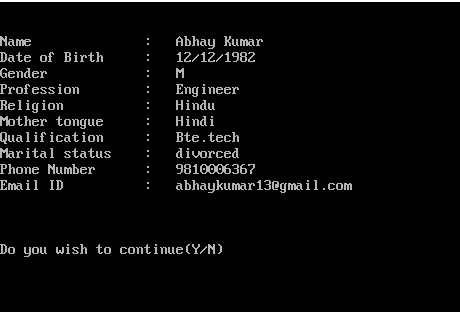


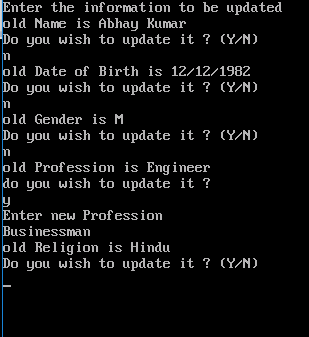


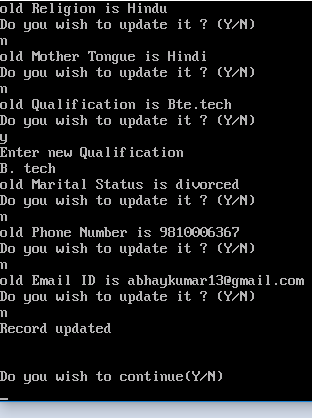


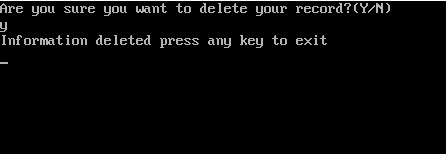


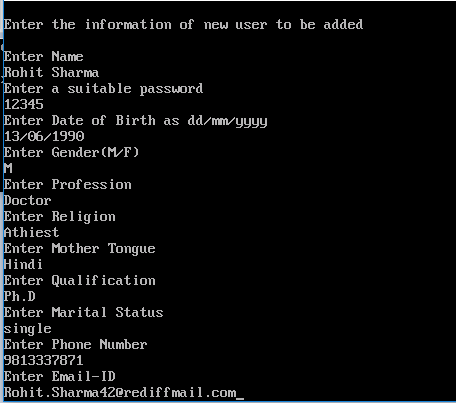


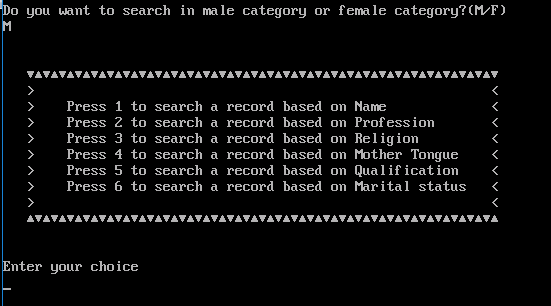


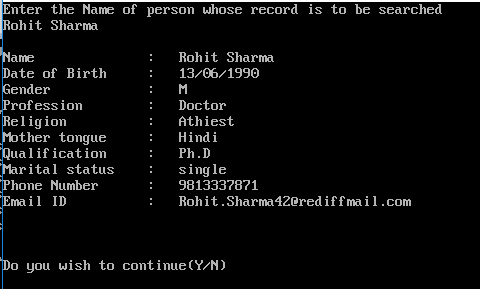


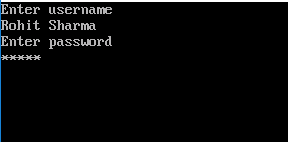












**Bibliography**

* Stackexchange.com
* Computer science with C++, by Sumita Arora
* Shaadi.com