NATIONAL INSTITUTE OF TECHNOLOGY,
TIRUCHIRAPPALLI-15

Department of Computer Applications

PHONEBOOK MANAGEMENT

PROJECT WORK

Submitted By

YASH KARIL -205119111

AND

CHETAN CHOUHAN-205119027

Under the guidance of

Dr.P.CHITRA

Submitted in fulfillment of the project in C++.

NATIONAL INSTITUTE OF TECHNOLOGY,
TIRUCHIRAPPALLI - 15

CERTIFICATE

*This is to certify that**, YASH KARIL & CHETAN CHOUHAN** student of 2 nd semester MCA (batch 2019-2022) of National Institute of Technology, Tiruchirappalli has successfully completed the project **PHONEBOOK MANAGEMENT** in C++ under the guidance of* **Dr.P.CHITRA.**

Signature

(Dr.P.CHITRA)

CONTENTS

ABSTRACT

Phonebook Management

The project on Phonebook management system is for maintaining all the records of the people present in a system. The main motive of this project is to maintain a database for all the people present in the phonebook management system and recover it when necessary, for our convenience.

This project will teach how to do basic operation that are ADD, DELETE,SEARCH,VIEW Using c++ language.

.

Feature of projects

The system shall be able to record so many people details : name,mobile,course.

The system shall be able to retrieve the details : name,mobile no. .

Then system shall be able to Edit and Delete the details : name,mobile no. .

Introduction to the project :-This is the project of phonebook, In which we are easily able to add or delete or search or we can see the all list of contacts. This will be the simplest project made up of class and objects and some c++ functions . In this project we can store many no. of contact no. . And any time we can edit them according to the user choice.

REQUIREMENTS

Software Requirement :-

Software	description
Windows	operating system
Turbo	c++ For Executing Program
MS Word	For Output Presentation

Hardware Requirement:-

Hardware	Description
Ram	256 MB
Hard Disk	20 GB
CD ROM	400 MB
PROCESSOR	PENTIUM III
MONITER	14.4”
KEYBOARD	104 KEYS

DEVELOPMENT LANGUAGE USED

I opted C++ as the development language for my project because C++ is a versatile language for handling very large programs. C++ follows all the standards given by OOPs.

Advantages of using C + +in the project :

Using C++ (an Object Oriented Programming language) allowed breaking complex large software programs to simpler,smaller and manageable components.

The following are some advantages of using OOPs:-

MODULAR DESIGN:-The software developed for the Railway Reservation System around OOPis modular, because this is built on objects and we know objects are entityinthemselves,whoseinternalworkingis hidden from other objects.

MODIFIABLE:-Because of its inherent properties of data abstraction and encapsulation the internal working of objects is hidden from other objects in the program. So any modification made in an object should not affect the rest of the system.

EXTENSIBLE:-If further enhancement in the existing Railway Reservation System is required for the its adaption in a new environment then it can easily be done by simply adding new features in old class types.

HEADERFILES USED

i.) #include<fstream.h> - For basic input, output and file handling functions

ii.) #include<conio.h> - For getch() and clrscr() functions

iii.)#include<stdio.h> - For gets() function

iv.)#include<string.h> - For string functions such as strcmpi()

v.)#include<process.h> - For exit(0) function

CODING

Phonebook project made by Yash Karil & Chetan Chouhan . We have given the code description in comments To understand the code you should read the comments .

```
#include <iostream>

#include <conio.h>

#include <string>

using namespace std;


//prototypes

void printline(char, int);

bool name_valid(string);

bool mob_valid(string);

//This is class contact to retrieve the contact from the user name, mob is the object in the class
```

Designing the class

The name of the class is “contact”. There are two data members – name and mob. Examine the class below to see the data members and member functions used in this class of the Phonebook project.

```
class contact

{

   string name;

   string mob;

    public:

          //Initialize the contact by a default value
```

```
contact(): name(""), mob("")
{}

// Shows all contacts of the phonebook
bool show()
{
    if(name != "")
    {
        cout << name << "\t" << mob << endl;
        return 1; //Indicates success
    }
    else
        return 0; //Indicates failure
}

//To Search a specific contact in the phonebook
bool show(string search_term)
{
    if(search_term == name)
    {
        cout << name << "\t" << mob << endl;
        return 1;
    }
    else
```

```
        return 0;
}

//Checks whether the name exists or not
bool name_exists(string tname)
{
    if(tname == name)
        return 1;
    else
        return 0;
}

//The contact to be add object is initialized by valid values
bool add(string new_name, string new_mob)
{
    if(name=="")
    {
        name = new_name;
        mob = new_mob;
        return 1; // Success
    }
    else
        return 0; // Failure
    }

    // Edits the contact details
    bool edit(string);

    //Sets the contact details to default values
    //That is, the contact details are thus erased
    bool erase(string new_name)
    {
        if(new_name==name)
        {
            name = "";
            mob = "";
            return 1;
        }
        else
            return 0;
    }
};

// Function for Edits the contact
bool contact :: edit(string new_name)
{
string new_mob;

if(new_name==name)

{

    cout << "\t\tEnter new name: "; cin >> new_name;

    cout << "\t\tEnter new mobile no: "; cin >> new_mob;


    name = new_name;

    mob = new_mob;

    return 1;

}

else

    return 0;

}
```

Creating an array of objects
An array of objects of the ‘contact’ class is created inside the main function. The following statement creates an array of objects.

```
contact person[100];
```

Person is the name of the object. A real-life meaningful name has been given to this object so that this C++ object looks like a real life objects. Thus it becomes easier to understand and write the code. This also increases the readability of the code.

100 objects are created. So you cannot add more than hundred contacts to the contacts-list of the Phonebook application. You may use dynamic memory allocation to creat objects as per your requirements. This also saves memory space. And the program becomes light.

```
//this is main function of the program

int main()

{
   contact person[100];


   string temp_name, temp_mob;

   int choice, i, counter;

   bool flag;

   bool cancel_flag;

       cout<<"\t\t";

       printline('-', 70);

       cout<<"\t\t";

       printline('-', 70);

       //cout<<"\t\t";

   cout << "\n\n\t\t**** PHONEBOOK MANAGAMENT SYSTEM ******" << endl;
```

Designing a menu-driven GUI

A console program is significantly less user-friendly than a GUI program. This why, a menu has been added to the program so that it becomes more user-friendly to the users of this application. The menu is as follows.

- 0. Show contacts.
- 1. Add contact.
- 2. Edit contact.
- 3. Delete contact.
- 4. Search contact.
- 5. Quit

Consider the following code.

```
   do

   {

       cout << "\n\n\n";

       cout<<"\t\t";
```

```
 printline('-', 70);

 cout<<"\t\t";

 printline('-', 70);

 //cout<<"\t\t";

 cout << "\n\t\t0. Show contacts" << endl

 << "\t\t1. Add Contact" << endl

 << "\t\t2. Edit Contact" << endl

 << "\t\t3. Delete Contact" << endl

 << "\t\t4. Search" << endl

 << "\t\t5. Quit" << endl << endl

 << "\t\tYour choice...";

 cin >> choice;


 system("cls");

cout<<"\t\t";

printline('-', 70);

cout<<"\t\t";

 cancel_flag = 0;

 flag = 0;

 counter = 0;


 switch(choice)

 {

    case 0:
```

Showing contacts

The following code shows all the contacts from the contacts-list. Examine the below code to understand how it works

```
cout << "\t\tShowing Contacts" << endl;

cout<<"\t\t";

printline('-', 70);

cout<<"\t\t";

for(i=0; i<100; i++)

   if(person[i].show())

      cout<<"\t\t";

      flag = 1;


if(!flag)

   cout << "\t\tNo contacts found!" << endl;

break;
```

Adding contacts

The following code adds a new contact to the contacts-list of the Phonebook application.

```
case 1:

   cout << "\t\tAdd New Contact\t\t\t\tpress $ to cancel" << endl;

   cout<<"\t\t";

   printline('-', 70);

   cout<<"\t\t";

   counter = 0;
```

```
//Loop until correct name and mobile number are entered
do
{
   flag = 0;
   if(counter)
      cout << "\t\tTry again\t\t\t\tpress $ to cancel"
                                              << endl;

   //counts how many times the do-while loop executes
                                    counter++;

   cout << "\t\tName: "; cin >> temp_name;

   //Cancel operation
   if(temp_name=="$")
   {
      cancel_flag = 1;
      break;
   }
   cout << "\t\tMobile No.: "; cin >> temp_mob;

   //Cancel operation
   if(temp_mob=="$")
   {
        cancel_flag = 1;

        break;

    }


    //Check whether name exists

    for(i=0; i<100; i++)

        if(person[i].name_exists(temp_name))

        {

            cout << "\t\tThe name you entered is already there"

                                                "in the phonebook, enter a different name."

                                                << endl;

            flag = 1;

            break;

        }


}while(!name_valid(temp_name) ||

                                                    flag ||

                                        !mob_valid(temp_mob));


if(cancel_flag)

{

    system("cls");

    break;

}
```

```
    //This code adds the contact to phonebook

    for(i=0; i<100; i++)

      if(person[i].add(temp_name, temp_mob))

      {

        cout << "\t\tContact added successfully!" << endl;

        flag = 1;

        break;

      }


    if(!flag)

      cout << "\t\tMemory full! Delete some contacts first."

                                << endl;

    break;
```

Editing a contact

The following code edits an existing contact. It edits both – name and mobile number.

```
  case 2:

    cout << "\t\tEnter a contact name to edit:"

                        "\t\t\t\tpress $ to cancel\n";

                         cin >> temp_name;


    //Cancel Operation

    if(temp_name=="$")
    {
        system("cls");
        break;
    }

    for(i=0; i<100; i++)
        if(person[i].edit(temp_name))
        {
            cout << "\t\tEdited Successfully!" << endl;
            flag = 1;
            break;
        }

    if(!flag)
        cout << "Contact name not found!" << endl;
    break;

case 3:
    do
    {
        if(counter)
            cout << "\t\tTry again" << endl;
        counter++;
        cout << "\t\tEnter a contact name to delete:"
```

```
                                   "\t\t\tpress $ to cancel\n";
                                   cin >> temp_name;


          //Cancel Operation
          if(temp_name=="$")
          {
             system("cls");
             break;
          }



          //Final Confirmation
          for(i=0; i<100; i++)
          if(person[i].name_exists(temp_name))
          {
             flag = 1;
             cout << "\t\tAre you sure you want to delete? (1/0)"
                                                 << endl;
             int yes;
             cin >> yes;
             if(!yes)
             {
                system("cls");
                cancel_flag = 1;
```

```
        }
        break;
    }

    if(!flag)
        cout << "Contact name not found!" << endl;

    if(cancel_flag)
        break;

    // This code deletes the contact
    if(flag)
    {
        for(i=0; i<100; i++)
            if(person[i].erase(temp_name))
            {
                cout << "\t\tDeleted successfully!" << endl;
                break;
            }
    }

}while(!flag);
break;
```

Searching for a contact

The following code searches for a contact.

```
case 4:
   do
   {
      if(counter)
         cout << "\t\tTry again" << endl;
      counter++;
      cout << "\t\tSearch a name: \t\t\t\tpress $ to cancel\n";
                                    cin >> temp_name;


      //Cancel Operation
      if(temp_name=="$")
      {
         system("cls");
         break;
      }


      for(i=0; i<100; i++)
         if(person[i].show(temp_name))
         {
            flag = 1;
            break;
         }
```

```
                if(!flag)
                    cout << "\t\tContact name not found" << endl;
            }while(!flag);

            break;

        case 5:
            return 0;
            break;

    }
} while(1);

getch();
return 0;
}

//prints a line
void printline(char ch, int size)
{
    for(int i=0; i<size; i++)
        cout << ch;
    cout << "\n";
```

```
}
```

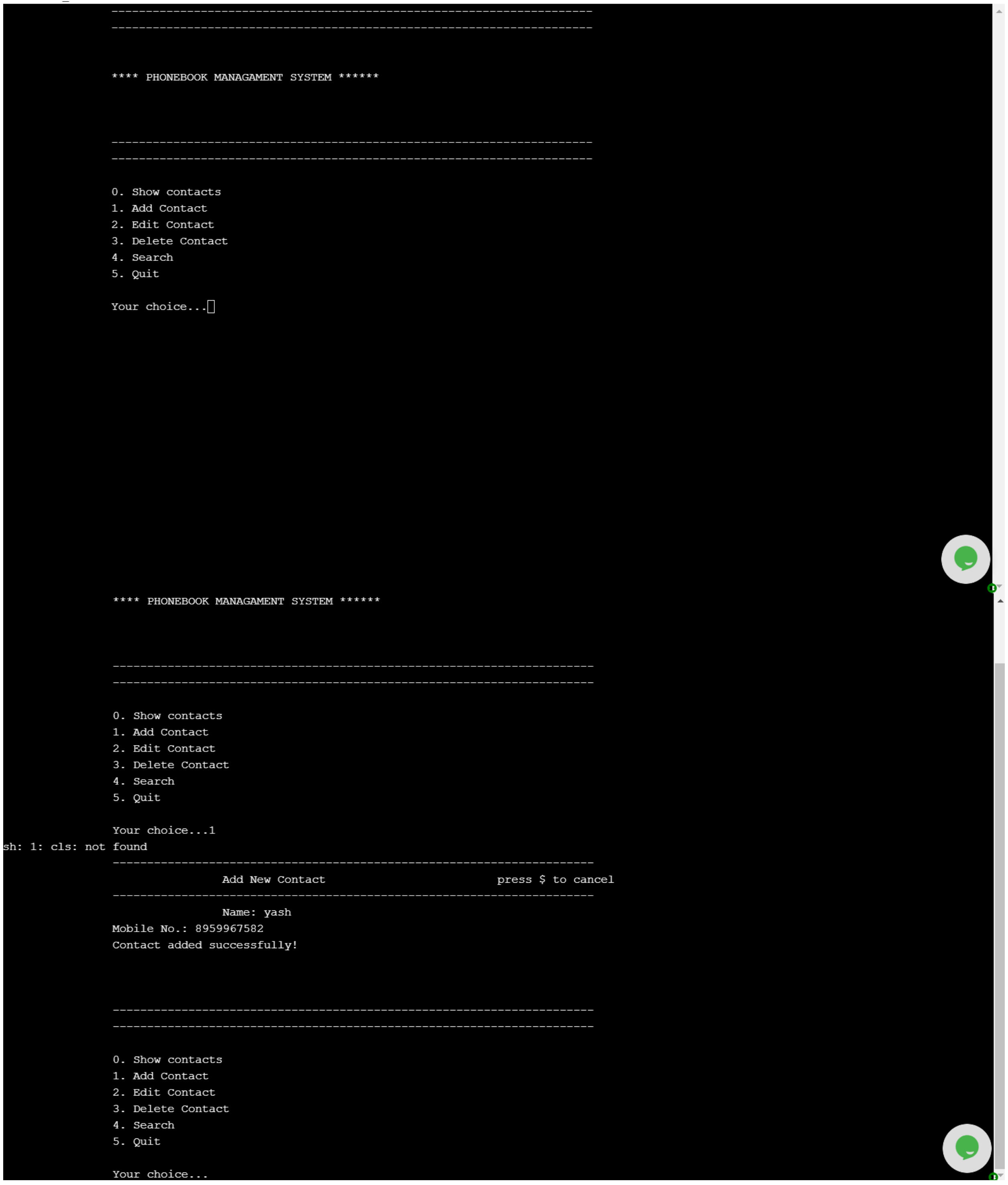
Validations

Two functions have been used for validations. One is name_valid(), another is mob_valid. The first one checks whether the name is valid, while the second function checks whether the mobile number is valid.

```
//Contact name validation

bool name_valid(string tname)

{



   if(tname.size()>20)

   {

      cout << "\t\tInvalid Name!\nEnter a name within 20 characters!"

                     << endl;

      return 0;

   }

   else if(tname == "")

   {

      cout << "\t\tInvalid Name!\nName cannot be blank!" << endl;

      return 0;

   }

   else

      return 1;

}
```

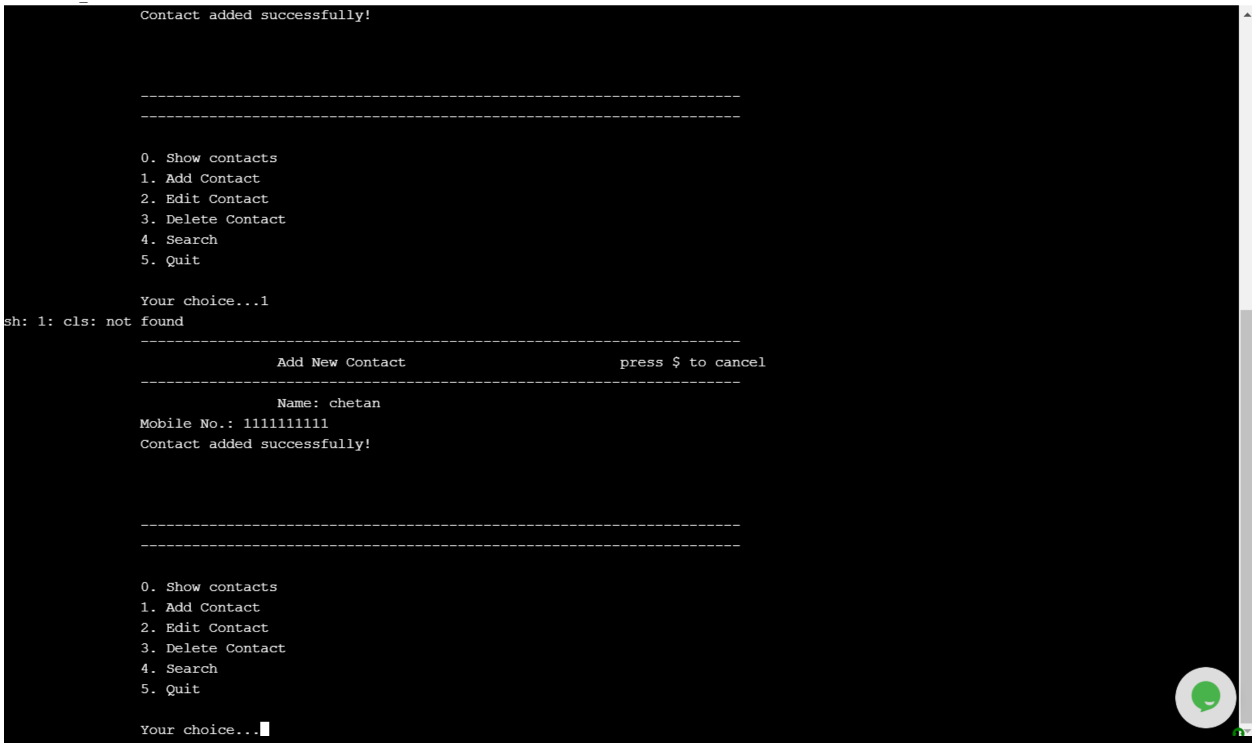
```
//mobile number validation

bool mob_valid(string tmob)

{

  if(tmob.size()>13 || tmob.size()<10)

  {

    cout << "\t\tInvalid mobile no.\nEnter a no."

              "between 10 and 13 digits" << endl;

    return 0;

  }

  else if(tmob == "")

  {

    cout << "\t\tInvalid mobile no.\nMobile"

              "no cannot be blank" << endl;

    return 0;

  }

  else

    return 1;

}
```

OUTPUT

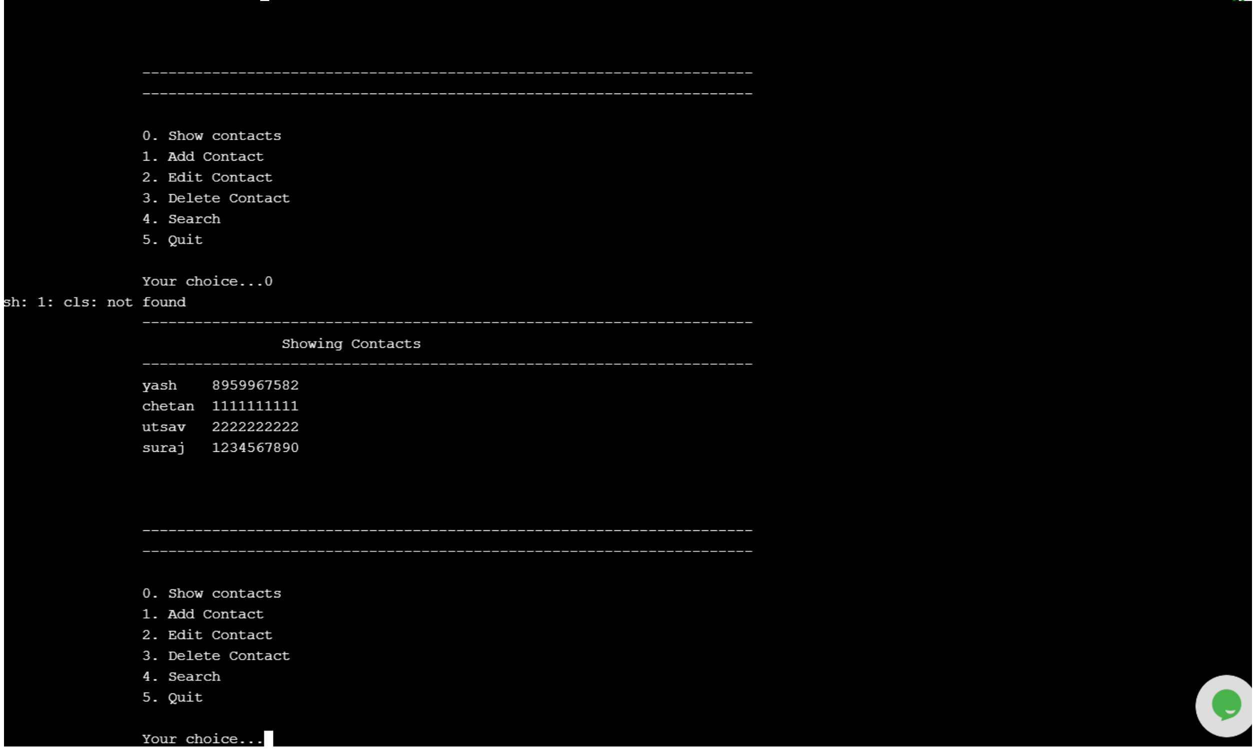


```
----------------------------------------------------------------------
----------------------------------------------------------------------


**** PHONEBOOK MANAGAMENT SYSTEM ******



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

0. Show contacts
1. Add Contact
2. Edit Contact
3. Delete Contact
4. Search
5. Quit

Your choice...
```

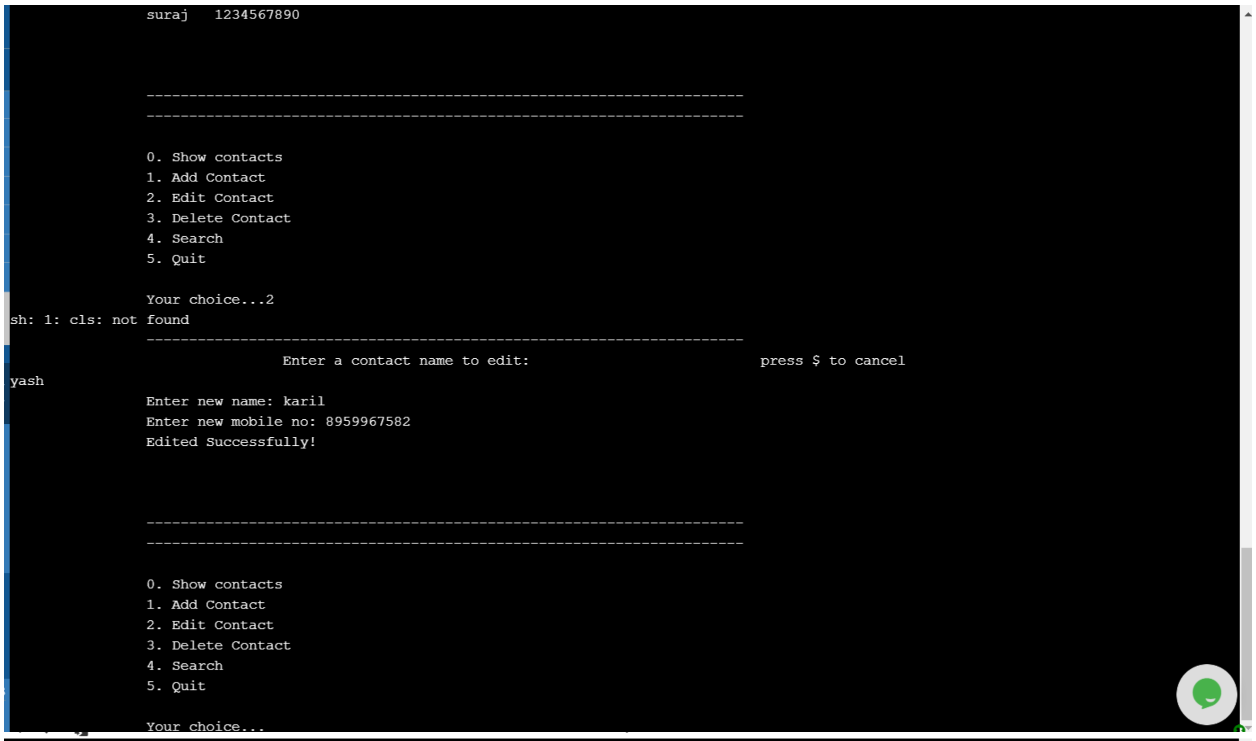
```
**** PHONEBOOK MANAGAMENT SYSTEM ******



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

0. Show contacts
1. Add Contact
2. Edit Contact
3. Delete Contact
4. Search
5. Quit

Your choice...1
sh: 1: cls: not found
----------------------------------------------------------------------
                Add New Contact                        press $ to cancel
----------------------------------------------------------------------
                Name: yash
Mobile No.: 8959967582
Contact added successfully!



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

0. Show contacts
1. Add Contact
2. Edit Contact
3. Delete Contact
4. Search
5. Quit

Your choice...
```



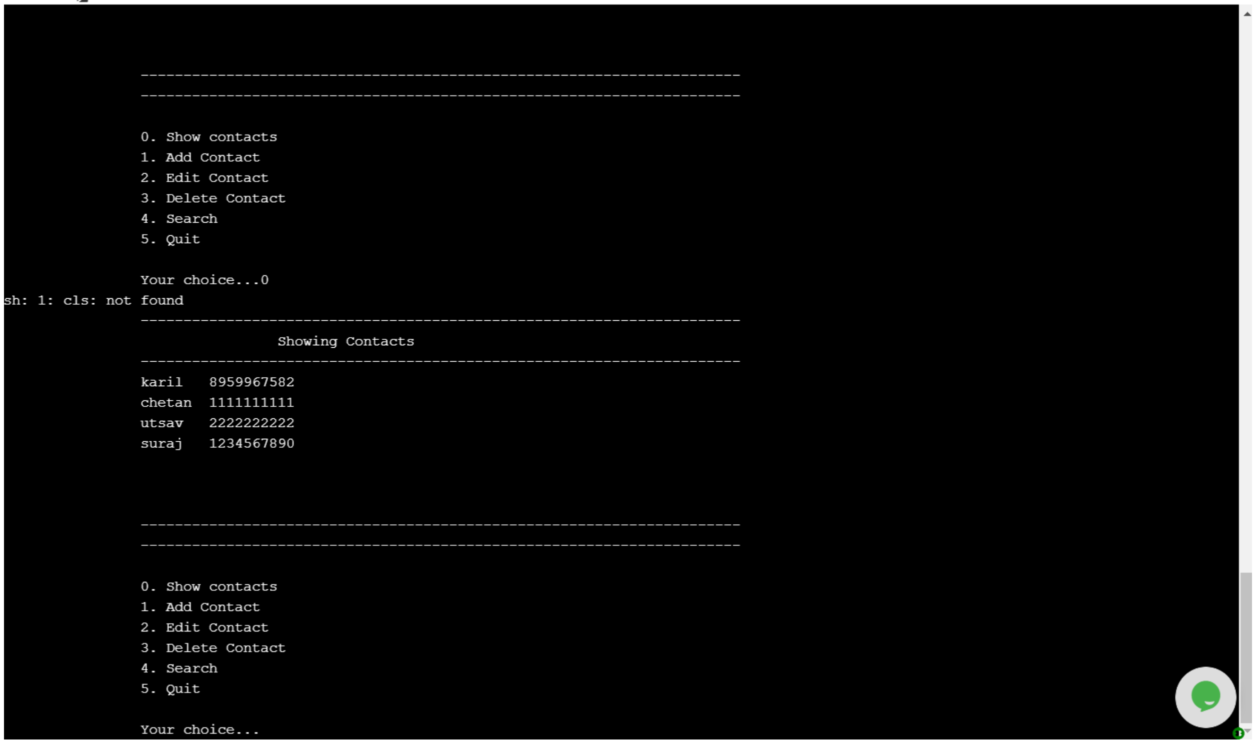
```
                Contact added successfully!



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


                0. Show contacts
                1. Add Contact
                2. Edit Contact
                3. Delete Contact
                4. Search
                5. Quit

                Your choice...1
sh: 1: cls: not found
                ----------------------------------------------------------------------
                                Add New Contact                         press $ to cancel
                ----------------------------------------------------------------------
                                Name: chetan
                Mobile No.: 1111111111
                Contact added successfully!



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


                0. Show contacts
                1. Add Contact
                2. Edit Contact
                3. Delete Contact
                4. Search
                5. Quit

                Your choice...
```



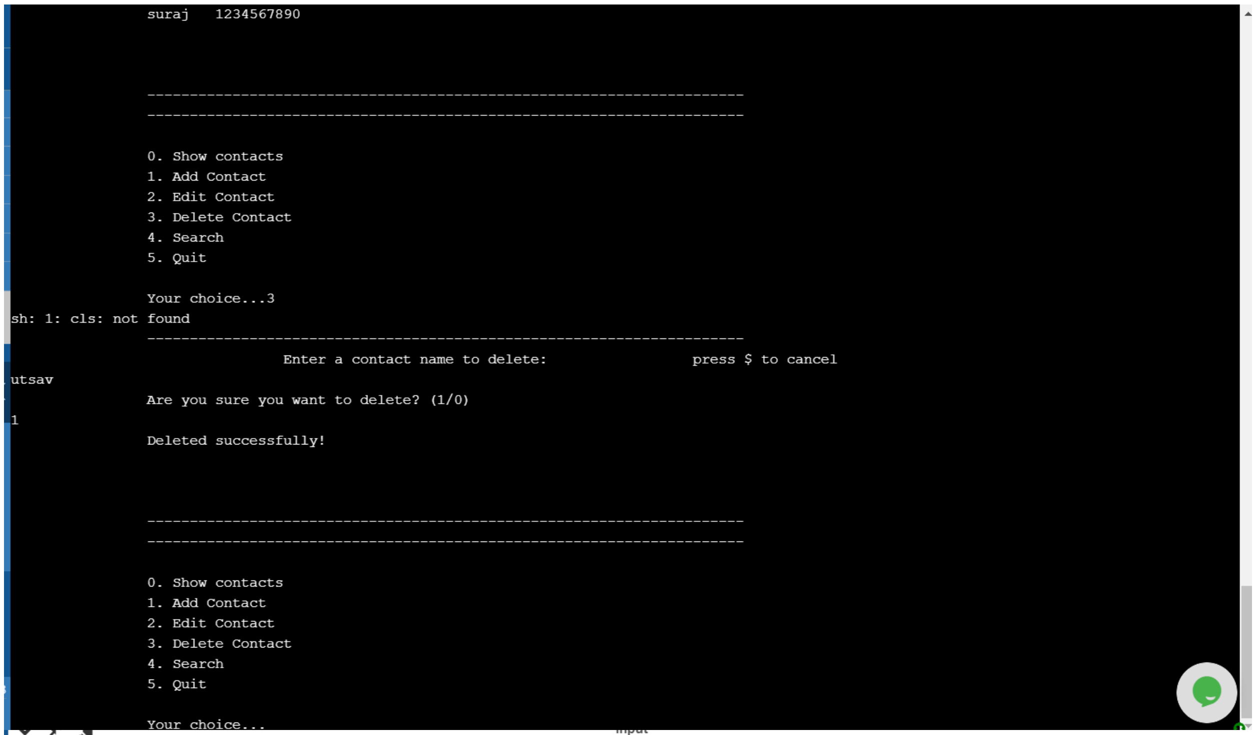
```
                ----------------------------------------------------------------------
                ----------------------------------------------------------------------


                0. Show contacts
                1. Add Contact
                2. Edit Contact
                3. Delete Contact
                4. Search
                5. Quit

                Your choice...0
sh: 1: cls: not found
                ----------------------------------------------------------------------
                                Showing Contacts
                ----------------------------------------------------------------------
                yash    8959967582
                chetan  1111111111
                utsav   2222222222
                suraj   1234567890



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


                0. Show contacts
                1. Add Contact
                2. Edit Contact
                3. Delete Contact
                4. Search
                5. Quit

                Your choice...
```



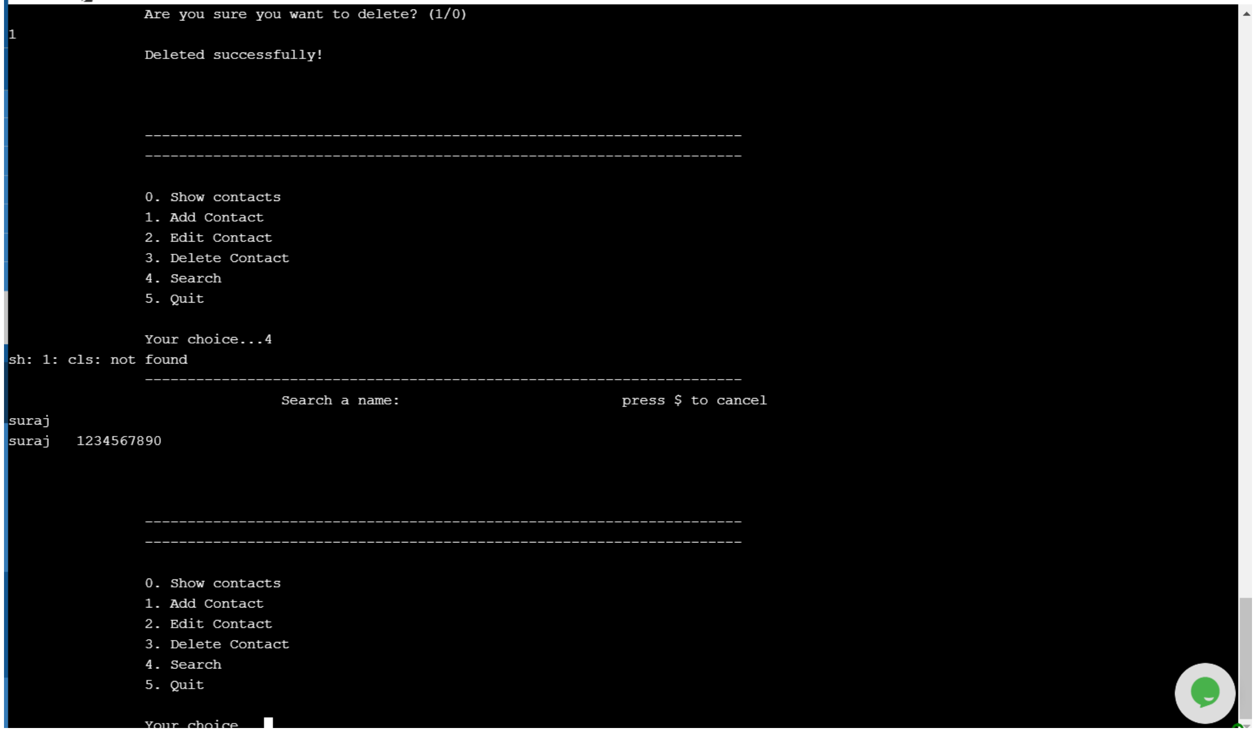
```
                suraj   1234567890



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


                0. Show contacts
                1. Add Contact
                2. Edit Contact
                3. Delete Contact
                4. Search
                5. Quit

                Your choice...2
sh: 1: cls: not found
                ----------------------------------------------------------------------
                                Enter a contact name to edit:                           press $ to cancel
yash
                Enter new name: karil
                Enter new mobile no: 8959967582
                Edited Successfully!



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


                0. Show contacts
                1. Add Contact
                2. Edit Contact
                3. Delete Contact
                4. Search
                5. Quit

                Your choice...
```



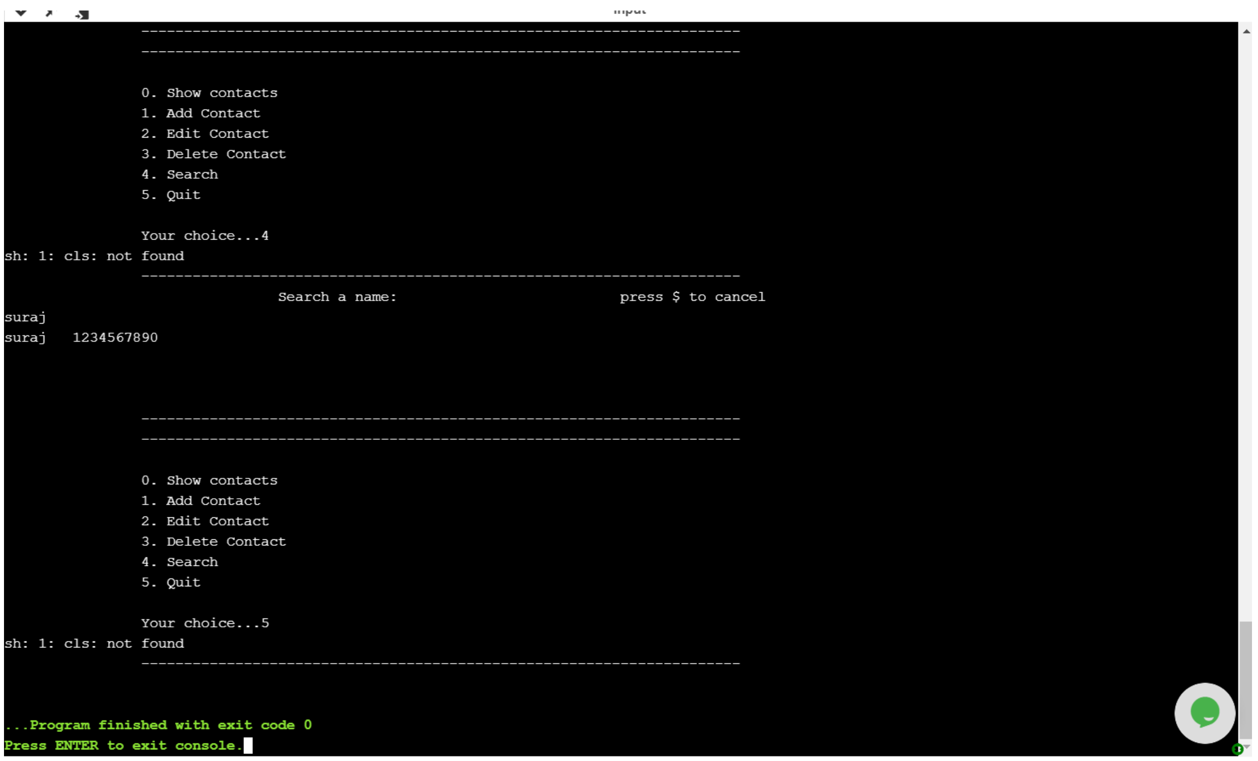
```


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


                0. Show contacts
                1. Add Contact
                2. Edit Contact
                3. Delete Contact
                4. Search
                5. Quit

                Your choice...0
sh: 1: cls: not found
                ----------------------------------------------------------------------
                                Showing Contacts
                ----------------------------------------------------------------------
                karil   8959967582
                chetan  1111111111
                utsav   2222222222
                suraj   1234567890



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


                0. Show contacts
                1. Add Contact
                2. Edit Contact
                3. Delete Contact
                4. Search
                5. Quit

                Your choice...
```



```
                suraj   1234567890



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

                0. Show contacts
                1. Add Contact
                2. Edit Contact
                3. Delete Contact
                4. Search
                5. Quit

                Your choice...3
sh: 1: cls: not found
                ----------------------------------------------------------------------
                                Enter a contact name to delete:                 press $ to cancel
utsav
                Are you sure you want to delete? (1/0)
1
                Deleted successfully!



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

                0. Show contacts
                1. Add Contact
                2. Edit Contact
                3. Delete Contact
                4. Search
                5. Quit

                Your choice...
```



```
                Are you sure you want to delete? (1/0)
1
                Deleted successfully!



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

                0. Show contacts
                1. Add Contact
                2. Edit Contact
                3. Delete Contact
                4. Search
                5. Quit

                Your choice...4
sh: 1: cls: not found
                ----------------------------------------------------------------------
                                Search a name:                          press $ to cancel
suraj
suraj   1234567890



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

                0. Show contacts
                1. Add Contact
                2. Edit Contact
                3. Delete Contact
                4. Search
                5. Quit

                Your choice...
```



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


                0. Show contacts
                1. Add Contact
                2. Edit Contact
                3. Delete Contact
                4. Search
                5. Quit

                Your choice...4
sh: 1: cls: not found
                ----------------------------------------------------------------------
                                Search a name:                          press $ to cancel
suraj
suraj   1234567890




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


                0. Show contacts
                1. Add Contact
                2. Edit Contact
                3. Delete Contact
                4. Search
                5. Quit

                Your choice...5
sh: 1: cls: not found
                ----------------------------------------------------------------------


...Program finished with exit code 0
Press ENTER to exit console.
```

THANK YOU