JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY,NOIDA-62
SEMESTER-3
DSA MINI PROJECT
B15

Phonebook Management System

- **29th Nov,2022**

OVERVIEW

The “Phonebook Management System” is a simple console application in C++. In this console application,users can add a new contact,delete an old record or update it. The Phonebook Management System undertaken as a project is based on relevant technologies.

GOALS

1. To allow users to add contacts in the phonebook.
2. To store those contacts into a file.
3. To update, delete or edit the contact in the phonebook.
4. To display the recent call history after call.
5. To provide greater speed and reduced time consumption.

SUBMITTED TO:

PROF. SIKHA JAIN

DR. APEKSHA AGGARWAL

SUBMITTED BY:

ADITYA KUMAR-21104042

STUTI GUPTA-21104043

AVIJOT SINGH-21104036

HIRDYARTH GUPTA-21104055

ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude to all those who have contributed significantly by sharing their knowledge and experiences. This project has been a truly educational experience for me as this helped me to grow by expanding my domain knowledge.This project has been very innovative as it reduced the gap between theoretical and practical knowledge.

I would like to express my gratitude to Prof. Sikha Jain ma'am, Dr. Apeksha Aggarwal ma'am, Prof. Arpita Jhadav ma,am and Prof. Dipty Tripathi ma,am for providing me this opportunity to learn and experience. I would also be thankful to them for helping me out at every single step.

Aditya Kumar

Stuti Gupta

Avijot Singh

Hirdyarth Gupta

Table of Contents

INTRODUCTION

The main aim of this project is to develop a Phonebook Management software. This software has been developed in C++ language. This will allow users to access various functions and processes available in the system. It has many features to maintain a contact list like adding a new contact,editing an existing contact,deleting a contact,storing them into a file and displaying them.It will work at a greater speed.

“Phonebook Management System” is a simple console application in C++. In this project, users can add a new phone record,delete an old record or update it. The code for this application will be optimized. We have used a doubly linked list to store the contact information in a node. Insertion, deletion or updation will be easier using a doubly linked list. The LRU(Least Frequently/Recently Used) algorithm maintains a linked list which has been called recently. LRU has been implemented using stack only. All the functions are optimized.

WORKING APPROACH:

- Creating a menu in the welcome page that will be accessed after running the program.
- There will be four options in the main menu, they are as follows:
    1. Add a contact
    2. Edit a contact
    3. Delete a contact
    4. Search a contact
    5. Display all contacts
    6. Delete all contacts
    7. History of recently called contacts

CONCEPTS USED:

- Basic C++ syntax.
- Features of Object Oriented Programming(OOPs).
- Stack.
- Doubly Linked LIsts.
- LRU algorithm.
- Searching and Sorting.
- File Handling.

SYNOPSIS

CLASS USED

“node”: This will create a node of doubly linked list which will contain two pointers which will keep records of previous and next nodes. A node will also contain four string variables.

“phoneBook”: It will contain name,phone number,email and address in a node.

Member functions of “phoneBook”:

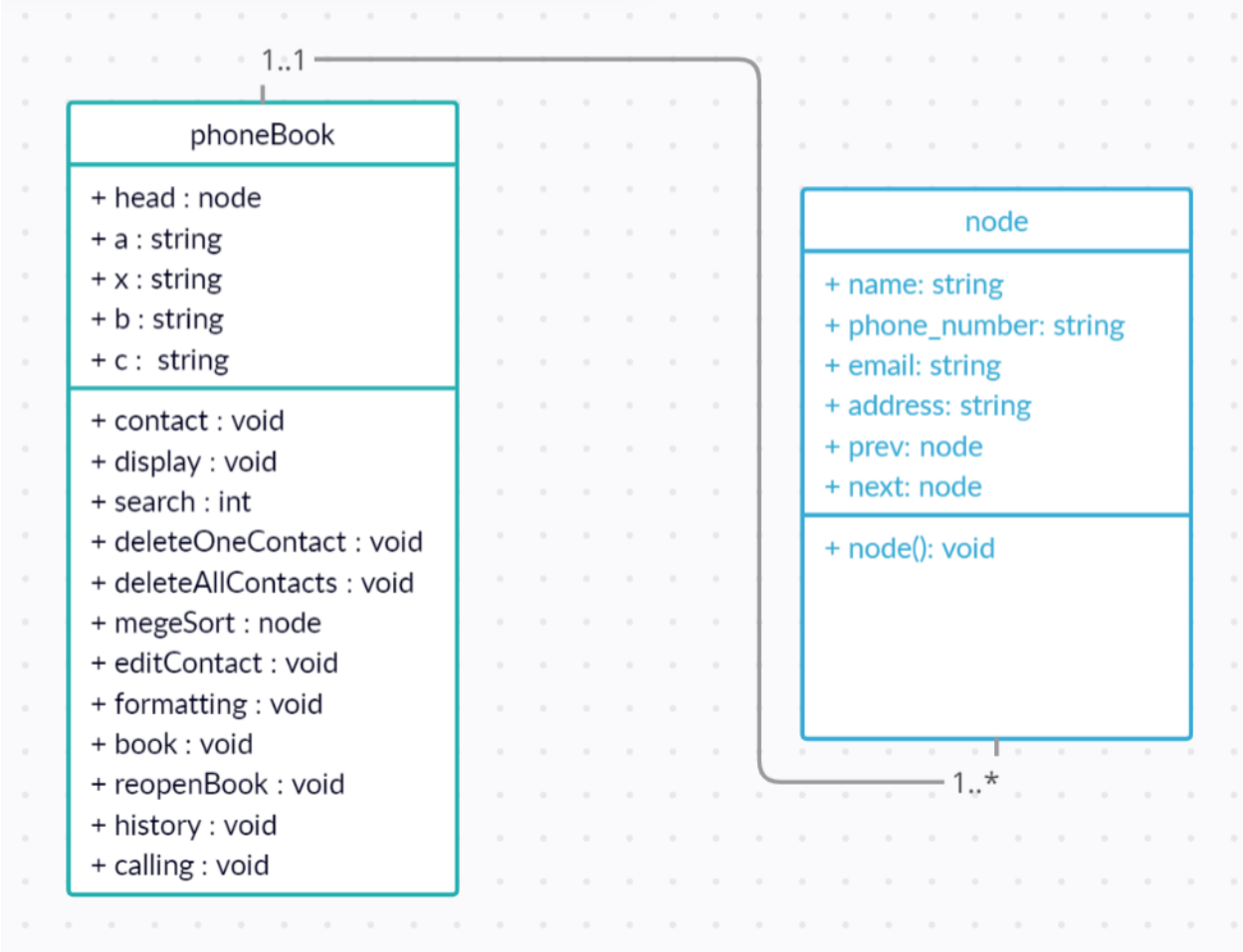
1. **addContact :** This will take input from the users and add the contact into the phonebook. It has been implemented using a doubly linked list.
2. **display :** This member function is used to display the contacts in the phonebook.
3. **search :** This function takes inputs from the users . Users can search a contact either by contact name or by phone number.
4. **deleteAllContacts :** This member function as the name suggests is used to delete all the contacts present in the contact list by deleting all the nodes.
5. **deleteOneContact :** This member function first searches the contact which has to be deleted and then deleted it from the contact list.
6. **midEle :** This member function will return the mid element of the doubly linked list.
7. **merge :** This member function is used to merge the nodes in sorted form and returns the merged doubly linked list.

8. **mergeSort :** This sorting method runs with a complexity of O(nlogn). Contact list has been sorted alphabetically on the basis of name.Sorting of doubly linked list has been used here.
9. **editContact :** This member function takes input from users and searches the given name or number into the contact list and edit that.
10. **formatting :** This member function is used to structure the output format on the console and takes command at every step.
11. **book :** This member function writes the linked lists' nodes data into a file.
12. **reopenBook :** This member function reads the file and stores the data into the doubly linked lists for keeping the records.
13. **history :** This member function prints the stack of nodes returned from the calling function. These printed records are a list of recently called persons.
14. **calling :** This member function provides the virtual facility to call a person and stores the information into the stack for keeping the records of recent call logs.

LIBRARY USED:

- #include<iostream>
- #include<fstream>
- #include<sstream>
- #include<stack>

UML DIAGRAM



IMPLEMENTATION

```
#include<iostream>
#include<fstream>
#include<sstream>
#include<stack>
using namespace std;
```

```
class node{
    public:
    string name;
    string phone_number;
    string email;
    string address;
    node*next;
    node*prev;
    node(){
        next=NULL;
        prev=NULL;
    }
};

stack<node*> st;


class phoneBook{
    public:
    node*head;
    string a;
    string x;
    string b;
    string c;

    phoneBook()
    {
        head=NULL;
        a="";
```

```
    x="";
    b="";
    c="";
}

void addContact()
{
    if(head==NULL)
    {
        node * n= new node;
        cout<<"Name of the Person: ";
        cin>>a;
        n->name=a;

        cout<<"Enter Contact Number: ";
        cin>>x;
        n->phone_number=x;

        cout<<"Enter email: ";
        cin>>b;
        n->email=b;

        cout<<"Enter address: ";
        cin>>c;
        n->address=c;

        n->next=NULL;
        n->prev=NULL;
        head=n;
        cout<<"Contact added successfully"<<endl;
```

```
    }
    else{
        node * n= new node;
        cout<<"Name of the Person: ";
        cin>>a;
        n->name=a;

        cout<<"Enter Contact Number: ";
        cin>>x;
        n->phone_number=x;

        cout<<"Enter email: ";
        cin>>b;
        n->email=b;

        cout<<"Enter address: ";
        cin>>c;
        n->address=c;

        n->next=NULL;
        node*temp=head;
        while(temp->next!=NULL)
        {
            temp=temp->next;
        }
        temp->next=n;
        n->prev=temp;
        cout<<"Contact added successfully"<<endl;
    }
}
```

```
  void display()
  {
     if(head==NULL)
     {
        cout<<"Welcome to the PhoneBook"<<endl;
        cout<<"PhoneBook is empty,Please add some contacts"<<endl;
        cout<<"ThankYou"<<endl;
        return;
     }

     int total=0;
     head=mergeSort(head);
     node*temp=head;
     cout<<" Name"<<"   Number"<<"  Email"<<"   Address\n";
     while(temp!=NULL)
     {
        total++;
        cout<<temp->name;
        cout<<"   "<<temp->phone_number;
        cout<<"   "<<temp->email;
        cout<<"  "<<temp->address;
        cout<<endl;
        temp=temp->next;
     }
     cout<<endl<<"Total Number of Contacts in the Phonebook:
"<<total<<endl;

  }
```

```
int search()
{
    bool flag=false;
    node*temp=head;
    cout<<"******************"<<endl;
    cout<<"Press 1 if you want to search by Name"<<endl;
    cout<<"Press 2 if you want to search by Phone Number"<<endl;
    int command;
    cout<<"Enter the command:"<<endl;
    cin>>command;

    if(command==1 and temp!=NULL)
    {
        cout<<"Enter the name to be searched:"<<endl;
        cin>>a;
        while(temp!=NULL)
        {
            if(temp->name==a)
            {
                cout<<"***********"<<endl;
                cout<<"Name :"<<temp->name<<endl;
                cout<<"Phone NUmber: "<<temp->phone_number<<endl;
                cout<<"Email: "<<temp->email<<endl;
                cout<<"Address: "<<temp->address<<endl;
                cout<<"***********"<<endl;
                flag=true;
                break;
            }
            temp=temp->next;
        }
```

```
        if(flag==false)
        {
            cout<<"Contact doesn't exist by this name"<<endl;
        }
    }
    else if(command==2 and temp!=NULL)
    {
        cout<<"Enter the Phone Number you want to search:"<<endl;
        cin>>x;
        while(temp!=NULL)
        {
            if(temp->phone_number==x)
            {
                cout<<"***********"<<endl;
                cout<<"Name :"<<temp->name<<endl;
                cout<<"Phone NUmber: "<<temp->phone_number<<endl;
                cout<<"Email: "<<temp->email<<endl;
                cout<<"Address: "<<temp->address<<endl;
                cout<<"***********"<<endl;
                flag=true;
                break;
            }
            temp=temp->next;
        }
        if(flag==false)
        {
            cout<<"Contact doesn't exist by this number"<<endl;
        }
    }
}
```

```
void deleteAllContacts()
{
    node*temp=head;
    node*temp2;
    if(head==NULL)
    {
        cout<<"PhoneBook is already empty"<<endl;
        return;
    }
    while(temp!=NULL)
    {
        temp2=temp;
        temp=temp->next;
        delete temp2;
    }
    head=NULL;
    cout<<"All contacts had been deleted successfully"<<endl;

}

void deleteOneContact()
{
    node*temp=head;
    cout<<"****************"<<endl;
    cout<<"Press 1 if you want to search by Name"<<endl;
    cout<<"Press 2 if you want to search by Phone Number"<<endl;
    int command;
    cout<<"Enter the command:"<<endl;
    cin>>command;
```

```
if(command==1 and temp!=NULL)
{
    bool flag=false;
    cout<<"Enter the name to be searched:"<<endl;
    cin>>a;
    while(temp!=NULL)
    {
        if(temp->name==a)
        {
            cout<<"***********"<<endl;
            cout<<"Name :"<<temp->name<<endl;
            cout<<"Phone NUmber: "<<temp->phone_number<<endl;
            cout<<"Email: "<<temp->email<<endl;
            cout<<"Address: "<<temp->address<<endl;
            cout<<"***********"<<endl;
            flag=true;
            break;
        }
        temp=temp->next;
    }
    if(flag==true)
    {
        int command;
        cout<<"Press 1 to delete the contact"<<endl;
        cin>>command;
        if(command==1 and temp==head)
        {
            node*t1=temp;
            temp=temp->next;
```

```
            delete t1;
            temp->prev=NULL;
            head=temp;
            cout<<"Contact deleted successfully"<<endl;
        }
        else if(command==1 and temp->next==NULL)
        {
            temp->prev->next=NULL;
            delete temp;
            cout<<"Contact deleted successfully"<<endl;
        }
        else if(command==1)
        {
            node*t1=temp;
            temp->prev->next=t1->next;
            temp->next->prev=t1->prev;
            delete t1;
            cout<<"Contact deleted successfully"<<endl;
        }
        else{
            cout<<"Please enter valid command"<<endl;
        }
    }
    else if(flag==false)
    {
        cout<<"Contact by this name doesn't exist in phoneBook"<<endl;
    }
}
else if(command==2 and temp!=NULL)
{
```

```
bool flag=false;
cout<<"Enter the number to be searched:"<<endl;
cin>>x;
while(temp!=NULL)
{
    if(temp->phone_number==x)
    {
        cout<<"***********"<<endl;
        cout<<"Name :"<<temp->name<<endl;
        cout<<"Phone NUmber: "<<temp->phone_number<<endl;
        cout<<"Email: "<<temp->email<<endl;
        cout<<"Address: "<<temp->address<<endl;
        cout<<"***********"<<endl;
        flag=true;
        break;
    }
    temp=temp->next;
}
if(flag==true)
{
    int command;
    cout<<"Press 1 to delete the contact"<<endl;
    cin>>command;
    if(command==1 and temp==head)
    {
        node*t1=temp;
        temp=temp->next;
        delete t1;
        temp->prev=NULL;
        head=temp;
```

```
            cout<<"Contact deleted successfully"<<endl;
        }
        else if(command==1 and temp->next==NULL)
        {
            temp->prev->next=NULL;
            delete temp;
            cout<<"Contact deleted successfully"<<endl;
        }
        else if(command==1)
        {
            node*t1=temp;
            temp->prev->next=t1->next;
            temp->next->prev=t1->prev;
            delete t1;
            cout<<"Contact deleted successfully"<<endl;
        }
        else{
            cout<<"Please enter valid command"<<endl;
        }
    }
    else if(flag==false)
    {
        cout<<"Contact by this number doesn't exist in phoneBook"<<endl;
    }
  }
}

//slow will point to the middle element of the Linked List
node*midEle(node*head)
{
```

```
    node*slow=head;
    node*fast=slow->next;
    while(fast!=NULL and fast->next!=NULL)
    {
        slow=slow->next;
        fast=fast->next->next;
    }
    return slow;
}

node*merge(node*a,node*b)
{
    node*dummy=new node;
    node*curr=dummy;
    while(a!=NULL and b!=NULL)
    {
        if(a->name<b->name)
        {
            curr->next=a;
            a=a->next;
            curr=curr->next;
        }
        else{
            curr->next=b;
            b=b->next;
            curr=curr->next;
        }
    }
    while(a!=NULL)
    {
```

```
        curr->next=a;
        a=a->next;
        curr=curr->next;
    }
    while(b!=NULL)
    {
        curr->next=b;
        b=b->next;
        curr=curr->next;
    }
    return dummy->next;
}

node*mergeSort(node*head)
{
    if(head==NULL or head->next==NULL)
    {
        return head;
    }
    node*mid=midEle(head);
    node*rightStart=mid->next;
    mid->next=NULL;
    node*a=mergeSort(head);
    node*b=mergeSort(rightStart);
    node*newHead=merge(a,b);
    return newHead;
}

int editContact()
{
```

```
node*temp=head;
cout<<"******************"<<endl;
cout<<"Press 1 if you want to search by name"<<endl;
cout<<"Press 2 if you want to search by number"<<endl;
int command;
cout<<"Enter the command :"<<endl;
cin>>command;

if(command==1 and temp!=NULL)
{
    bool flag=false;
    cout<<"Enter the name to be edited:"<<endl;
    cin>>a;
    while(temp!=NULL)
    {
        if(temp->name==a)
        {
            cout<<"****************"<<endl;
            cout<<"Name :"<<temp->name<<endl;
            cout<<"Contact Number :"<<temp->phone_number<<endl;
            cout<<"Email :"<<temp->email<<endl;
            cout<<"Address :"<<temp->address<<endl;
            flag=1;
            break;
        }
        temp=temp->next;
    }
    if(flag==1)
    {
        int command;
```

```
        cout<<"Press 1 to edit the contact : "<<endl;
        cin>>command;
        if(command==1)
        {
            cout<<"ENter the new name :"<<endl;
            cin>>a;
            cout<<"Enter the updated number :"<<endl;
            cin>>x;
            cout<<"Enter the updated email :"<<endl;
            cin>>b;
            cout<<"Enter the updated address :"<<endl;
            cin>>c;

            temp->name=a;
            temp->phone_number=x;
            temp->email=b;
            temp->address=c;

            cout<<"Contact updated successfully"<<endl;
        }
        else{
            cout<<"Enter the right command"<<endl;
        }
    }
}
else if(command==2 and temp!=NULL)
{
    bool flag=false;
    cout<<"Enter the contact number too be edited :";
    cin>>x;
```

```
while(temp!=NULL)
{
    if(temp->phone_number==x){
        cout<<"**************"<<endl;
        cout<<"Name :"<<temp->name<<endl;
        cout<<"Contact Number :"<<temp->phone_number<<endl;
        cout<<"Email :"<<temp->email<<endl;
        cout<<"Address :"<<temp->address<<endl;
        flag=true;
        break;
    }
    temp=temp->next;
}
if(flag==true){
    int command;
    cout<<"Press 1 to edit the contact:"<<endl;
    cin>>command;
    if(command==1){
        cout<<"ENter the new name :"<<endl;
        cin>>a;
        cout<<"Enter the updated number :"<<endl;
        cin>>x;
        cout<<"Enter the updated email :"<<endl;
        cin>>b;
        cout<<"Enter the updated address :"<<endl;
        cin>>c;

        temp->name=a;
        temp->phone_number=x;
        temp->email=b;
```

```
            temp->address=c;

            cout<<"Contact updated successfully"<<endl;
          }
          else{
            cout<<"Please enter the right command"<<endl;
          }
        }
    }
    else{
        cout<<"Please enter the right command "<<endl;
    }
}
void formatting(){
        cout<<"**************"<<endl;
            cout<<"  1. Add a Contact"<<endl;
            cout<<"  2. Edit an existing Contact"<<endl;
            cout<<"  3. Delete a Contact"<<endl;
            cout<<"  4. Search a Contact"<<endl;
            cout<<"  5. Display All the Contacts"<<endl;
            cout<<"  6. Delete All Contacts"<<endl;
        cout<<"  7. Recent Call History List"<<endl;
            cout<<"**************"<<endl;

            int command;
            cout<<"  Enter the Command: ";
            cin>>command;
            try
            {
                    if(command>=1&&command<=7)
```

```
    {
            if(command==1)
    {
            addContact();
            book();
    formatting();
    }
    else if(command==2)
    {
            editContact();
 book();
            formatting();
           }
    else if(command==3)
    {
    deleteOneContact();
book();
    formatting();
    }
    else if(command==4)
    {
            search();
            formatting();
    }
    else if(command==5)
    {
            display();
            book();
            formatting();
    }
```

```
                else if(command==6)
                {
                        deleteAllContacts();
                        book();
                        formatting();
                }
            else if(command==7)
            {
                calling();
                formatting();
            }
                        }
                        else
                        {
                                throw(command);
                        }
                }
                catch(int command)
                {
                        cout<<"  You Entered wrong Command... Run the
Code Again"<<endl;
                        formatting();
                }
        }
  void book()
  {
     node*temp=head;
     ofstream myfile("Phonebook.txt");
     if(myfile.is_open()){
        while(temp!=NULL)
```

```
        {
            myfile<<temp->name<<"\n";
            myfile<<temp->phone_number<<"\n";
            myfile<<temp->email<<"\n";
            myfile<<temp->address<<"\n";

            temp=temp->next;
        }
        myfile.close();

    }
    else{
        cout<<"File is Empty."<<endl;
    }
}
void reopenbook()
{
    string str="";
    fflush(stdin);
    fflush(stdout);
    ifstream myfile("Phonebook.txt");
    fflush(stdin);
    fflush(stdout);
    if(myfile.is_open() && myfile.peek()!=EOF){
        int i=0;
        node*n=new node;
        while(myfile>>str){
            if(i%4==0){
                if(head==NULL){
                    n->name=str;
```

```
            n->next=NULL;
            n->prev=NULL;
            head=n;
        }else{
            n=new node();
            n->name=str;
            n->next=NULL;
            node*temp=head;
            while(temp->next!=NULL){
                temp=temp->next;
            }
            temp->next=n;
            n->prev=temp;
        }
    }else if(i%4==1){
        if(head==NULL){
            n->phone_number=str;
            n->next=NULL;
            n->prev=NULL;
            head=n;
        }else{
            n->phone_number=str;
        }
    }
    else if(i%4==2){
        if(head==NULL){
            n->email=str;
            n->next=NULL;
            n->prev=NULL;
            head=n;
```

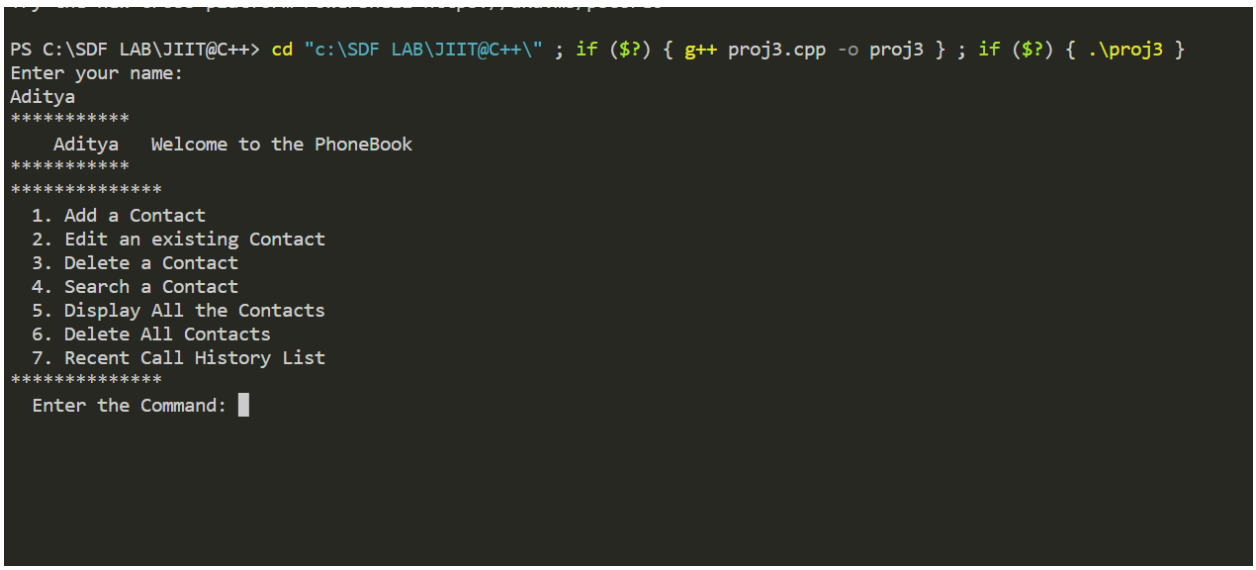
```
            }else{
                n->email=str;
            }
        }
        else if(i%4==3){
            if(head==NULL){
                n->address=str;
                n->next=NULL;
                n->prev=NULL;
                head=n;
            }else{
                n->address=str;
            }
        }
        i++;
    }
    myfile.close();
  }
  else{
    cout<<"Empty file"<<endl;
  }
}
void history(stack<node *> a)
{
  while(!a.empty())
  {
    node * t =a.top();
    cout<<t->name<<"\t\t";
    cout<<t->phone_number<<"\t\t";
    cout<<t->email<<"\t\t";
```

```
        cout<<t->address<<"\t\t";
        cout<<endl;
        a.pop();
    }
}
void calling()
{

    cout<<"Enter the name of person you want to call:"<<endl;
    cin>>a;
    bool flag=false;
    node*temp=head;
    while(temp!=NULL)
    {
        if(temp->name==a)
        {
            st.push(temp);
            flag=true;
            break;
        }
        temp=temp->next;
    }
    if(flag==false)
    {
        cout<<"This contact does not exist,please add to contact first"<<endl;
    }
    else{
        history(st);
    }
}
```

```
    };
int main()
{
    phoneBook pb;
    pb.reopenbook();
    string n;
    cout<<"Enter your name: "<<endl;
    cin>>n;
    cout<<"***********"<<endl;
    cout<<"    "<<n<<"   Welcome to the PhoneBook  "<<endl;
    cout<<"***********"<<endl;
    pb.formatting();
    return 0;
}
```

OUTPUT



```
**********
    Aditya   Welcome to the PhoneBook
**********
**********
  1. Add a Contact
  2. Edit an existing Contact
  3. Delete a Contact
  4. Search a Contact
  5. Display All the Contacts
  6. Delete All Contacts
  7. Recent Call History List
**********
  Enter the Command: 5
 Name   Number  Email   Address
akshat   777   @ak  jhgbc
atharv   875   @at  rajasthan
marvelous   480   @youtuber  lucknow
scooty   564   @scooty  agra
vipul   8756   @w  jhgd
yash   123   @y  kjgaef

Total Number of Contacts in the Phonebook: 6
**********
  1. Add a Contact
  2. Edit an existing Contact
  3. Delete a Contact
  4. Search a Contact
  5. Display All the Contacts
  6. Delete All Contacts
  7. Recent Call History List
```

```
  Enter the Command: 1
Name of the Person: shubhay
Enter Contact Number: 876
Enter email: @shu
Enter address: kota
Contact added successfully
**********
  1. Add a Contact
  2. Edit an existing Contact
  3. Delete a Contact
  4. Search a Contact
  5. Display All the Contacts
  6. Delete All Contacts
  7. Recent Call History List
**********
  Enter the Command: 5
 Name   Number  Email   Address
akshat   777   @ak  jhgbc
atharv   875   @at  rajasthan
marvelous   480   @youtuber  lucknow
scooty   564   @scooty  agra
shubhay   876   @shu  kota
vipul   8756   @w  jhgd
yash   123   @y  kjgaef

Total Number of Contacts in the Phonebook: 7
**********
```

```
  Enter the Command: 2
******************
Press 1 if you want to search by name
Press 2 if you want to search by number
Enter the command :
1
Enter the name to be edited:
scooty
****************
Name :scooty
Contact Number :564
Email :@scooty
Address :agra
Press 1 to edit the contact :
1
ENter the new name :
stuti
Enter the updated number :
564
Enter the updated email :
@stuti
Enter the updated address :
agra
Contact updated successfully
**************
```

```
agra
Contact updated successfully
**************
  1. Add a Contact
  2. Edit an existing Contact
  3. Delete a Contact
  4. Search a Contact
  5. Display All the Contacts
  6. Delete All Contacts
  7. Recent Call History List
**************
  Enter the Command: 5
 Name   Number  Email   Address
akshat   777   @ak  jhgbc
atharv   875   @at  rajasthan
marvelous   480   @youtuber  lucknow
shubhay   876   @shu  kota
stuti   564   @stuti  agra
vipul   8756   @w  jhgd
yash   123   @y  kjgaef

Total Number of Contacts in the Phonebook: 7
**************
  1. Add a Contact
```

```
Total Number of Contacts in the Phonebook: 7
**************
  1. Add a Contact
  2. Edit an existing Contact
  3. Delete a Contact
  4. Search a Contact
  5. Display All the Contacts
  6. Delete All Contacts
  7. Recent Call History List
**************
  Enter the Command: 3
****************
Press 1 if you want to search by Name
Press 2 if you want to search by Phone Number
Enter the command:
2
Enter the number to be searched:
876
***********
Name :shubhay
Phone NUmber: 876
Email: @shu
Address: kota
***********
Press 1 to delete the contact
1
Contact deleted successfully
```

```
**************
  1. Add a Contact
  2. Edit an existing Contact
  3. Delete a Contact
  4. Search a Contact
  5. Display All the Contacts
  6. Delete All Contacts
  7. Recent Call History List
**************
  Enter the Command: 5
 Name   Number  Email   Address
akshat   777   @ak  jhgbc
atharv   875   @at  rajasthan
marvelous   480   @youtuber  lucknow
stuti   564   @stuti  agra
vipul   8756   @w  jhgd
yash   123   @y  kjgaef

Total Number of Contacts in the Phonebook: 6
**************
```

```
Enter the name of person you want to call:
vipul
vipul           8756            @w              jhgd
stuti           564             @stuti          agra
**************
  1. Add a Contact
  2. Edit an existing Contact
  3. Delete a Contact
  4. Search a Contact
  5. Display All the Contacts
  6. Delete All Contacts
  7. Recent Call History List
**************
  Enter the Command: 7
Enter the name of person you want to call:
atharv
atharv          875             @at             rajasthan
vipul           8756            @w              jhgd
stuti           564             @stuti          agra
**************
  1. Add a Contact
```

```
**************
  1. Add a Contact
  2. Edit an existing Contact
  3. Delete a Contact
  4. Search a Contact
  5. Display All the Contacts
  6. Delete All Contacts
  7. Recent Call History List
**************
  Enter the Command: 6
All contacts had been deleted successfully
**************
  1. Add a Contact
  2. Edit an existing Contact
  3. Delete a Contact
  4. Search a Contact
  5. Display All the Contacts
  6. Delete All Contacts
  7. Recent Call History List
**************
  Enter the Command: 5
Welcome to the PhoneBook
PhoneBook is empty,Please add some contacts
ThankYou
**************
  1. Add a Contact
  2. Edit an existing Contact
  3. Delete a Contact
  4. Search a Contact
  5. Display All the Contacts
  6. Delete All Contacts
  7. Recent Call History List
**************
  Enter the Command:
```

CONCLUSION

We learned a lot of new things in the process of project making. We have successfully developed a phonebook management software which is very fast and efficient. The project has been developed using C++ language as per our course description.We used various libraries. We provided many options in our software as per the demands of users.

Thanks to everyone!!!