**VISVESVARAYA TECHNOLOGICAL UNIVERSITY JNANASANGAMA, BELAGAVI – 590018**



# Mini Project Report Phone Book

**Submitted in partial fulfilment for the award of degree of**

**Bachelor of Engineering**

**In**

**Information Science and Engineering**

Submitted by

### Ashutosh-1RF20IS011 R-Dhanesh1RF20IS040 Viptha-1RF20IS062 Vishal-1RF20IS063



RV Institute of Technology and Management®

(Affiliated to VTU, Belagavi)

# JP Nagar, Bengaluru - 560076

**Department of Information Science and Engineering**

FINDING CELEBRITY Data Structure and Algorithm

# RV Institute of Technology and Management®

(Affiliated to VTU, Belagavi)

**JP Nagar, Bengaluru - 560076**

**Department of Information Science and Engineering**



**CERTIFICATE**

Certified that the Mini project entitled “**Phone Book**” carried out by

### Ashutosh-1RF20IS011 R-Dhanesh1RF20IS040 Viptha-1RF20IS062 Vishal-1RF20IS063

are bonafide students of 2nd Semester B.E, **RV Institute of Technology and Management** in partial fulfilment for the Bachelor of Engineering in INFORMATION SCIENCE AND ENGINEERING, of the **Visvesvaraya Technological University,** Belagavi, during the academic year 2021 - 2022. The Mini project report has been approved as it satisfies the academic requirements in respect of C Programming for Problem Solving.

Dr.Ajina

Associate Professor Dept. CSE RVITM,

Bengaluru - 560076

# ABSTRACT

The system of Phonebook has been developed successfully with the immense contribution of many people. We would start by thanking our Honorable Principal Dr Jayapal R whose continued cooperation and support to the Department of Information Science Engineering was a pivotal force in making this project a success.

We would like to take the opportunity to thank and express our deep sense of gratitude to our mentor Dr.Ajina for giving us the opportunity to work on data structures and file handling in C. We would like to appreciate her guidance, encouragement and cooperation throughout the project.

We would also like to thank our Professor Dr.Ajina who has imparted the knowledge of Data Structures & Applications and made us thorough with the course, her valuable contribution and guidance have been certainly indispensable for our project work.

We hope that we can build upon the experience and knowledge that we have gained and make a valuable contribution in coming future.

# Table of Contents

Contents

[Mini Project Report 1](#_heading=h.gjdgxs)

[PhoneBook 1](#_heading=h.gjdgxs)

[ABSTRACT 3](#_heading=h.30j0zll)

[Table of Contents 4](#_heading=h.1fob9te)

[Chapter 1 INTRODUCTION 5](#_heading=h.3znysh7)

* 1. [: Introduction to the project 5](#_heading=h.2et92p0)
  2. [: Introduction to the application 5](#_heading=h.35nkun2)

[CHAPTER 2 6](#_heading=h.3dy6vkm)

[Algorithm 6](#_heading=h.1t3h5sf)

[CHAPTER 3 7](#_heading=h.4d34og8)

* 1. [: Implementation details & Code 7](#_heading=h.2s8eyo1)
  2. [: Experimental Results/Snapshots 13](#_heading=h.17dp8vu)

[Conclusion 18](#_heading=h.3rdcrjn)

[Future extension 18](#_heading=h.26in1rg)

[References 18](#_heading=h.lnxbz9)

4

# Chapter 1 INTRODUCTION

## : Introduction to the project

#### PhoneBook:

* This mini project in C Phonebook allows you to perform simple Phonebook operations like in your mobile. You can add, list, modify, search and delete Phonebook-related records. File handling and data structure concepts has been extensively used for almost all functions in this mini project.
* Phonebook is a very simple mini project in C that can help you understand the basic concepts of functions, file handling and data structure. This application will teach you how to add, list, modify or edit, search and delete data to/from the file.
* Adding new records, listing them, modifying them and updating, search for contacts saved, and deleting the phonebook records are the basic functions which make up the main menu of this Phonebook application (as shown in the main menu screenshot below).
* Phone book application is primarily meant for keeping the records of the persons. Phone book application will provide the basic set of features of adding a new contact, searching, updating, deleting a contact.

**1.2 : Relevance of the application**

* This program is made on accordance to make searching the phone book in an easier manner
* This saves us time
* Keeps us more organized
* will provide the basic set of features of adding a new contact, searching, updating, deleting a contact
* it allows a particular contact to be identified by name and address.
* This mini project in C Phonebook allows to perform simple Phonebook operations like in the mobile. One can add, list, modify, search and delete Phonebook-related records. File handling and data structure concepts has been extensively used for almost all functions in this mini project. Phonebook in C is a console application without graphics. The source code is complete and totally error-free. It is compiled in Code Blocks with GCC compiler. Functions, file handling and data structure are used. This application contains how to add, list, modify or edit, search and delete data to/from the file. Adding new records, listing them, modifying them and updating, search for contacts saved, and deleting the phonebook records are the basic functions which make up the main menu of this Phonebook application. Personal information such as name, gender, father’s name, phone number, citizenship number, email and address are asked while adding a record into the Phonebook. These records can then be modified, listed, searched for and removed.

# CHAPTER 2

**Algorithm**

1. Start
2. Create structure for contact  
   \\we create a structure that defines the contact based on Name, address, father name, mother name, mobile number, sex, email id, Aadhar number using different data types such as char/long/int.
3. Create the required functions: -  
   We create the menu  
   Start/back/add record/list record/modify record/delete record  
   these functions can have a function of void or int as necessary.
4. We create record function
5. We create add record.  
   We ask the user to ask the user to add their personal address and hence save the record.
6. List record  
   here we organize different contact in a structural manner in a memory block called file which is dynamically allocated with the help of pointers.
7. Search record  
   here we traverse through the information provided by using pointers
8. Modify record  
   Helps to edit the details of the contact if needed
9. Delete Entry  
   it deletes the particular contact’s name and its details

# CHAPTER 3

## : Implementation details & Code

### Ashutosh-1RF20IS011 R-Dhanesh1RF20IS040 Viptha-1RF20IS062 Vishal-1RF20IS063

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<stdlib.h>

struct person

{

char name[35];

char address[50];

char father\_name[35];

char mother\_name[30];

long int mble\_no;

char sex[8];

char mail[100];

char citision\_no[20];

};

void menu();

void got();

void start();

void back();

void addrecord();

void listrecord();

void modifyrecord();

void deleterecord();

void searchrecord();

int main()

{

start();

return 0;

}

void back()

{

start();

}

void start()

{

menu();

}

void menu()

{

printf("\n\t\t\*\*\*\*\*WELCOME TO PHONEBOOK\*\*\*\*");

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

printf("\t1.Add New \t2.List \t3.Exit \n\t4.Modify \t5.Search\t6.Delete\n");

switch( getch() )

{

case '1': addrecord();

break;

case '2': listrecord();

break;

case '3': exit(0);

break;

case '4': modifyrecord();

break;

case '5': searchrecord();

break;

case '6': deleterecord();

break;

default:

printf("\nEnter 1 to 6 only");

printf("\nEnter any key\n");

getch();

menu();

}

}

void addrecord()

{

FILE \*f;

struct person p;

f=fopen("project","ab+");

printf("\nEnter name: ");

got(p.name);

printf("\nEnter the address: ");

got(p.address);

printf("\nEnter father name: ");

got(p.father\_name);

printf("\nEnter mother name: ");

got(p.mother\_name);

printf("\nEnter phone no. : ");

scanf("%ld",&p.mble\_no);

printf("Enter sex:");

got(p.sex);

printf("\nEnter e-mail: ");

got(p.mail);

printf("\nEnter Aadhar no: ");

got(p.citision\_no);

fwrite(&p,sizeof(p),1,f);

fflush(stdin);

printf("\nrecord saved");

fclose(f);

printf("\n\nEnter any key");

getch();

menu();

}

void listrecord()

{

struct person p;

FILE \*f;

f=fopen("project","rb");

if(f==NULL)

{

printf("\nfile opening error in listing :");

exit(1);

}

while(fread(&p,sizeof(p),1,f)==1)

{

printf("\n\n\n YOUR RECORD IS\n\n ");

printf("\nName=%s\nAdress=%s\nFather name=%s\nMother name=%s\nMobile no=%ld\nSex=%s\nE-mail=%s\nAadhar no=%s",p.name,p.address,p.father\_name,p.mother\_name,p.mble\_no,p.sex,p.mail,p.citision\_no);

getch();

}

fclose(f);

printf("\n Enter any key");

getch();

menu();

}

void searchrecord()

{

struct person p;

FILE \*f;

char name[100];

f=fopen("project","rb");

if(f==NULL)

{

printf("\n error in opening\a\a\a\a");

exit(1);

}

printf("\nEnter name of person to search\n");

got(name);

while(fread(&p,sizeof(p),1,f)==1)

{

if(strcmp(p.name,name)==0)

{

printf("\n\tDetail Information About %s",name);

printf("\nName:%s\naddress:%s\nFather name:%s\nMother name:%s\nMobile no:%ld\nsex:%s\nE-mail:%s\nAadhar no:%s",p.name,p.address,p.father\_name,p.mother\_name,p.mble\_no,p.sex,p.mail,p.citision\_no);

}

else

printf("file not found");

}

fclose(f);

printf("\n Enter any key");

getch();

menu();

}

void deleterecord()

{

struct person p;

FILE \*f,\*ft;

int flag;

char name[100];

f=fopen("project","rb");

if(f==NULL)

{

printf("CONTACT'S DATA NOT ADDED YET.");

}

else

{

ft=fopen("temp","wb+");

if(ft==NULL)

printf("file opening error");

else

{

printf("Enter CONTACT'S NAME:");

got(name);

fflush(stdin);

while(fread(&p,sizeof(p),1,f)==1)

{

if(strcmp(p.name,name)!=0)

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

if(strcmp(p.name,name)==0)

flag=1;

}

fclose(f);

fclose(ft);

if(flag!=1)

{

printf("NO CONACT'S RECORD TO DELETE.");

remove("temp.txt");

}

else

{

remove("project");

rename("temp.txt","project");

printf("\nRECORD DELETED SUCCESSFULLY.");

}

}

}

printf("\nEnter any key\n ");

getch();

menu();

}

void modifyrecord()

{

int c;

FILE \*f;

int flag=0;

struct person p,s;

char name[50];

f=fopen("project","rb+");

if(f==NULL)

{

printf("CONTACT'S DATA NOT ADDED YET.");

exit(1);

}

else

{

printf("\nEnter CONTACT'S NAME TO MODIFY: \n");

got(name);

while(fread(&p,sizeof(p),1,f)==1)

{

if(strcmp(name,p.name)==0)

{

printf("\nEnter name: ");

got(s.name);

printf("\nEnter the address: ");

got(s.address);

printf("\nEnter father name: ");

got(s.father\_name);

printf("\nEnter mother name: ");

got(s.mother\_name);

printf("\nEnter phone no: ");

scanf("%ld",&s.mble\_no);

printf("\nEnter sex: ");

got(s.sex);

printf("\nEnter e-mail: ");

got(s.mail);

printf("\nEnter Aadhar no \n");

got(s.citision\_no);

fseek(f,-sizeof(p),SEEK\_CUR);

fwrite(&s,sizeof(p),1,f);

flag=1;

break;

}

fflush(stdin);

}

if(flag==1)

{

printf("\n Your data id modified");

}

else

{

printf(" \n data is not found");

}

fclose(f);

}

printf("\n Enter any key");

getch();

menu();

}

void got(char \*name)

{

int i=0,j;

char c,ch;

do

{

c=getch();

if(c!=8&&c!=13)

{

\*(name+i)=c;

putch(c);

i++;

}

if(c==8)

{

if(i>0)

{

i--;

}

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

{

ch=\*(name+j);

putch(ch);

}

}

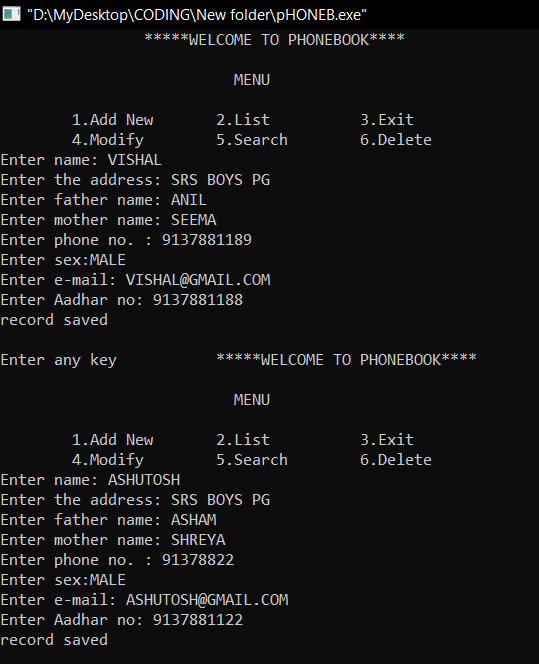
}while(c!=13);

\*(name+i)='\0';

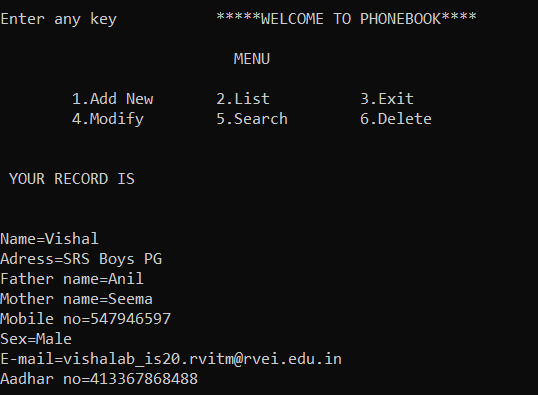
}

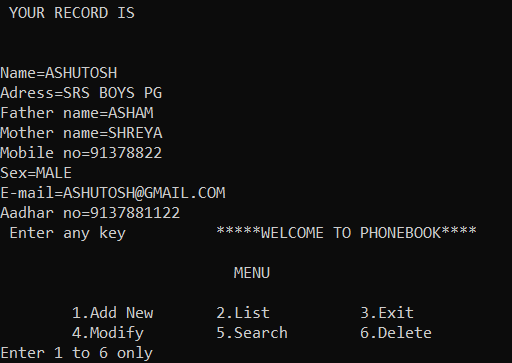
## : Experimental Results/Snapshots

## Taking data entry 1 (press 1)

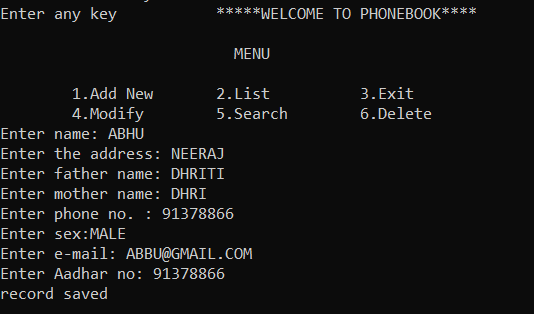


1. Checking the data List (press 2)

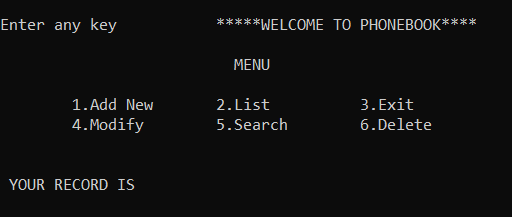


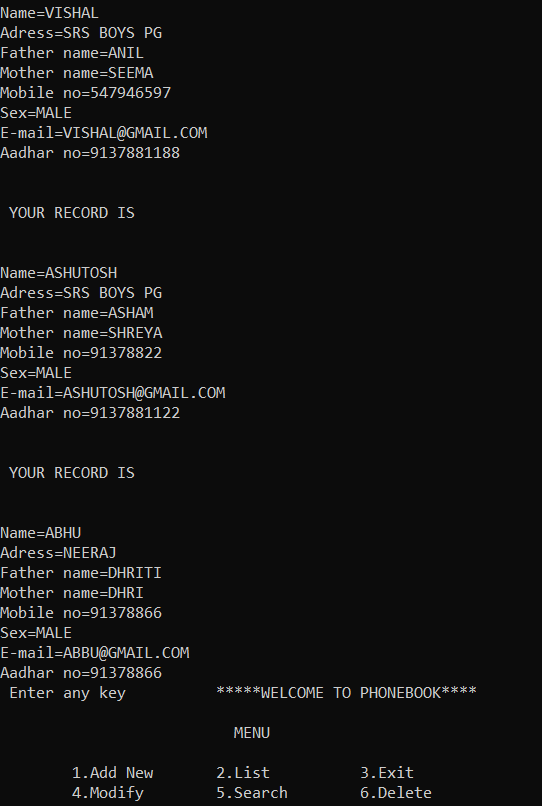


1. Taking third entry (press 1)

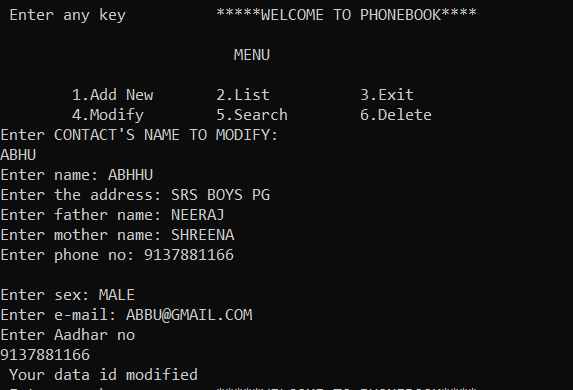


1. We check the list again (press 2)

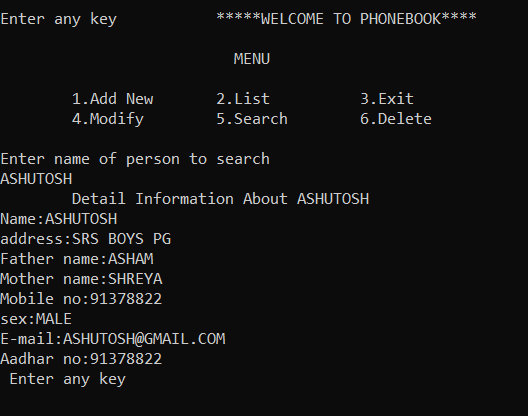




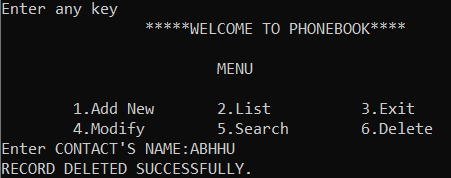
1. We use MODIFY function to modify the third entry (press 4)



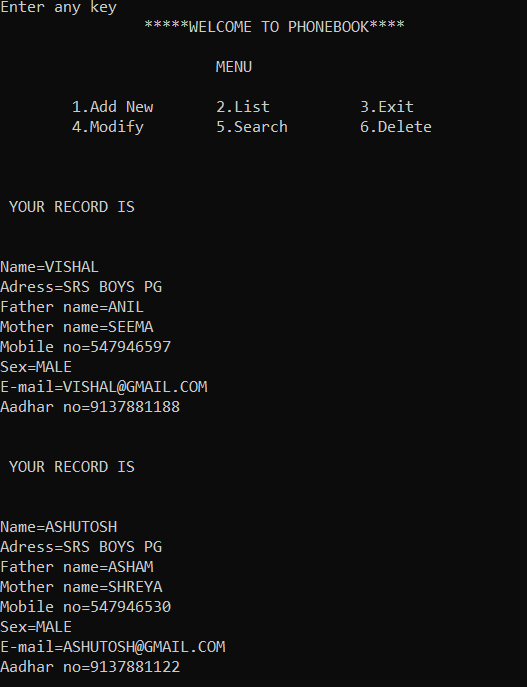
1. We use search function to search for ASHUTOSH data (press 5)



1. Using delete function (press 6)



1. Checking the list again (press 2)



1. EXIT

# Conclusion

Phone Book is a project that is provide by technical assignment help to us in that we get a simple SQL based solution to store our contacts. We can use it to replace our hard phonebook or even use it as an office wide phone directory. This will help user to easily search and manage contacts using this system. Phonebook is the one, which contain details of an individual along with their landline numbers. Apart from the telephone number of individuals, it also contains address and number of important relatives of individual. It not only contains local codes but also ISD codes.

# Future extension

* Android Applications
* Online Aadhar card update
* Juniors projects >\_>

# References

1. Wikipedia
2. Geeks for geeks
3. You tube
4. Let us C++ by Yashwant kanetkar