****

**Green University of Bangladesh**

**Department of Computer Science and Engineering (CSE)**

**Faculty of Sciences and Engineering**

**Semester: (Spring, Year:2022), B.Sc. in CSE (Day)**

**Course Title: Data Structures and Algorithm**

**Course Code: CSE-106 Section:DC**

**Lab Project Name: Phonebook Management System**

**Student Details**

| **Name** | **ID** |
| --- | --- |
| Md Kowsar Ul Alom | 213902096 |

**Submission Date : 11/09/2022**

**Course Teacher’s Name : Md. Sultanul Islam Ovi**

**[For Teachers use only: Don’t Write Anything inside this box]**

| **Lab Project Status**  **Marks: ………………………………… Signature: .....................**  **Comments: .............................................. Date: ..............................** |
| --- |

Table of Contents

**Chapter 1 Introduction 3**

1.1 Introduction 3

1.2 Design Goals/Objective 3

**Chapter 2 Design/Development/Implementation of the Project 4**

2.1 Section (Choose the name of this section as appropriate with your project) 4

2.2 Section (Choose the name of this section as appropriate with your project) 4

2.2.1 Subsection 4

**Chapter 3 Performance Evaluation 5**

3.1 Simulation Environment/ Simulation Procedure 5

3.2 Results and Discussions 5

**Chapter 4 Conclusion 6**

4.1 Introduction 6

4.1 Practical Implications 6

4.2 Scope of Future Work 6

**References 7**

# Chapter 1 Introduction

## Introduction

The phonebook application works specifically for tracking people. The Phonebook application contains a set of basic functions for adding, searching, updating, and deleting new contacts. This mini-C phonebook design allows you to perform simple tasks in your phonebook, such as mobile phones. You can add text to the phonebook, find, edit, search, and delete. The concept of file management and data structure is often used in almost all functions in this project.  It uses functions, file management, and data structure. This application provides information on adding, viewing, modifying, receiving, and deleting data from/to files. Adding new entries, browsing them, editing and updating, searching for saved contacts, and deleting contacts in the phonebook is one of the most important services that become the main menu in the phonebook application. When you add anything to your phone book, you will be asked for personal information such as name, gender, first name, phone number, nationality, email address, and address. You can then edit, view, search, and delete this text.

## Design Goals/Objective

Adding new items, viewing them by logging in, editing and updating, searching for saved contacts and deleting data in the phonebook is one of the main features of the main phonebook application (shown in the main menu below). ).  
 Personal information, such as name, type, identity, phone number, nationality, email address and address, is required to add a login to the phonebook. You can then edit, view, search and delete this text. It is estimated that there are more than 600 million mobile phone users in the world and the number is also increasing. The success of mobile phones is simply described: they are always there, everywhere. When a trader moves from one place to another, he is doing business without business. If the boy is late returning home, he can tell his parents. If you have a problem with your device, you can seek help along the way. Mobile phones are used for communication and communication between people. Informal meetings (such as going to the bar) are usually arranged on an anonymous and timely map over a mobile phone. In general, there are many ways to use a mobile phone. But there are some problems. Technology decides to reach potential everywhere, anywhere, but its existence does not mean that it can reach all possible respondents! Sharing information about the real world of customers is the way out of this problem. But now mobile user interactions depend on older versions. In particular, the connection between the user and the general user is not sensitive to the context, making it difficult to know when to call and under what circumstances.

The main idea of the program is to increase awareness of computer systems and the use of computer resources. The main purpose of this package is to reduce the pressure on users to learn more about computers and software. This helps maintain a direct connection between the computer and the user. This “My Phonebook” feature allows users with other logs to add and easily search for buttons and search options. There are also other services, for example: – Evaluate data in and out of the database. Our project is as follows: – “ADD” button on the login form, which allows you to add data to the database. Press the “Delete” button to delete transaction data. Press “UPDATE” to update the data. Press the “SEARCH” button to search for information in the database.Press the “SORT” option for sorting the contacts.

# Chapter 2

# Design/Development/Implementation of the Project

## (Phonebook management system)

In this project I have added linked list ,binary search ,quick sort. So in simple I have add the quick sort function to sort according to the phone number . So, if you add some extra feature then you can sort the contact list according to the 1st character of the persons name.You can use file to store the contact list.

**PROGRAM OF PHONEBOOK APPLICATION USING C**

**#include<stdlib.h>**

**#include<string.h>**

**#include<stdio.h>**

**struct person**

**{**

**char name[30];**

**char country\_code[4];**

**long int mble\_no;**

**char sex[8];**

**char mail[100];**

**struct person \*next;**

**}\*head;**

**struct person\* last\_node(struct person\* head)**

**{**

**struct person\* temp = head;**

**while (temp != NULL && temp->next != NULL)**

**{**

**temp = temp->next;**

**}**

**return temp;**

**}**

**struct person\* parition(struct person\* first, struct person\* last)**

**{**

**struct person\* pivot = first;**

**struct person\* front = first;**

**int temp = 0;**

**while (front != NULL && front != last)**

**{**

**if (front->mble\_no < last->mble\_no)**

**{**

**pivot = first;**

**temp = first->mble\_no;**

**first->mble\_no = front->mble\_no;**

**front->mble\_no = temp;**

**first = first->next;**

**}**

**front = front->next;**

**}**

**temp = first->mble\_no;**

**first->mble\_no = last->mble\_no;**

**last->mble\_no = temp;**

**return pivot;**

**}**

**struct person \*middle(struct person\* start,struct person\* last)**

**{**

**if (start == NULL)**

**return NULL;**

**struct person\* slow = start;**

**struct person\* fast = start -> next;**

**while (fast != last)**

**{**

**fast = fast -> next;**

**if (fast != last)**

**{**

**slow = slow -> next;**

**fast = fast -> next;**

**}**

**}**

**return slow;**

**}**

**//Search**

**struct person\* search(struct person \*head, int value)**

**{**

**struct person\* start = head;**

**struct person\* last = NULL;**

**do**

**{**

**struct person \*mid = middle(start, last);**

**if (mid == NULL)**

**return NULL;**

**if (mid -> mble\_no == value)**

**return mid;**

**else if (mid -> mble\_no < value)**

**start = mid -> next;**

**else**

**last = mid;**

**}**

**while (last == NULL ||**

**last != start);**

**return NULL;**

**}**

**void insert(char\* name,char \* country\_code,long int phone,char \*sex, char\* mail)**

**{**

**struct person \* p1 = (struct person \*) malloc(sizeof(struct person));**

**p1->mble\_no= phone;**

**strcpy(p1->name, name);**

**strcpy(p1->country\_code, country\_code);**

**strcpy(p1->sex,sex);**

**strcpy(p1->mail,mail);**

**p1->next = NULL;**

**if(head==NULL)**

**{**

**head = p1;**

**}**

**else**

**{**

**p1->next = head;**

**head = p1;**

**}**

**}**

**void update(int phone)**

**{**

**struct person \* temp = head;**

**while(temp!=NULL)**

**{**

**if(temp->mble\_no==phone)**

**{**

**printf("\n\t\t\tEnter new name: ");**

**fflush(stdin);**

**scanf("%s", temp->name);**

**printf("\t\t\tEnter new phone number: ");**

**fflush(stdin);**

**scanf("%ld", &temp->mble\_no);**

**printf("\t\t\tEnter new country code : ");**

**fflush(stdin);**

**scanf("%s",temp->country\_code);**

**printf("\t\t\tEnter gender : ");**

**fflush(stdin);**

**scanf("%s",temp->sex);**

**printf("\t\t\tEnter your email : ");**

**fflush(stdin);**

**scanf("%s",temp->mail);**

**printf("\t\t\tUpdation Successful!!!\n");**

**return;**

**}**

**temp = temp->next;**

**}**

**printf("\n\t\t\tSearched mobile number is not found \n");**

**}**

**void Delete(long int phone)**

**{**

**struct person \* temp1 = head;**

**struct person\* temp2 = head;**

**while(temp1!=NULL)**

**{**

**if(temp1->mble\_no==phone)**

**{**

**if(temp1==temp2)**

**{**

**head = head->next;**

**free(temp1);**

**}**

**else**

**{**

**temp2->next = temp1->next;**

**free(temp1);**

**}**

**printf("\n\t\t\tRecord Successfully Deleted !!!\n");**

**return;**

**}**

**temp2 = temp1;**

**temp1 = temp1->next;**

**}**

**printf("\n\t\t\tSearched number is not found\n");**

**}**

**int display()**

**{**

**int count=0;**

**int i;**

**struct person \* temp = head;**

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

**printf("\t\t\t\t\* Here is all records listed in phonebook \*\n");**

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

**printf(" NAME\t\t\t\t COUNTRY CODE\t\t PHONE NO\t\t GENDER\t\t EMAIL\n");**

**printf("---------------------------------------------------------------------------------------------------------------------------------------------\n");**

**while(temp!=NULL)**

**{**

**int i;**

**int len1 = 40 - strlen(temp->name);**

**int len2 = 19 - strlen(temp->country\_code);**

**int len3 = 15;**

**int len4 = 21 - strlen(temp->sex);**

**printf("%s",temp->name);**

**for(i=0; i<len1; i++)**

**printf(" ");**

**printf("%s",temp->country\_code);**

**for(i=0; i<len2; i++)**

**printf(" ");**

**printf("%ld",temp->mble\_no);**

**for(i=0; i<len3; i++)**

**printf(" ");**

**printf("%s",temp->sex);**

**for(i=0; i<len4; i++)**

**printf(" ");**

**printf("%s",temp->mail);**

**printf("\n");**

**fflush(stdin);**

**temp=temp->next;**

**}**

**}**

**//Quick Sort**

**void quick\_sort(struct person\* first, struct person\* last)**

**{**

**if (first == last)**

**{**

**return;**

**}**

**struct person\* pivot = parition(first, last);**

**if (pivot != NULL && pivot->next != NULL)**

**{**

**quick\_sort(pivot->next, last);**

**}**

**if (pivot != NULL && first != pivot)**

**{**

**quick\_sort(first, pivot);**

**}**

**}**

**void menu()**

**{**

**system("cls");**

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

**printf("\t\t\* Welcome TO My phone book \*\n");**

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

**printf("\t\t\t1) Add Person\n\n");**

**printf("\t\t\t2) Search Person\n\n");**

**printf("\t\t\t3) Remove person\n\n");**

**printf("\t\t\t4) Update person\n\n");**

**printf("\t\t\t5) Print Contacts\n\n");**

**printf("\t\t\t6) Sort the contacts \n\n");**

**};**

**int main()**

**{**

**int size;**

**head = NULL;**

**int choice;**

**char name[100];**

**long int phone;**

**char ccode[7];**

**char sex[10];**

**char mail[500];**

**struct person \*temp1;**

**menu();**

**while(choice!=0)**

**{**

**printf("\n\t\t\tEnter your choice : ");**

**scanf("%d", &choice);**

**switch (choice)**

**{**

**case 1:**

**printf("\n\t\t\tEnter name: ");**

**fflush(stdin);**

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

**printf("\t\t\tEnter country code : ");**

**fflush(stdin);**

**scanf("%s", ccode);**

**printf("\t\t\tEnter phone number: ");**

**scanf("%ld",&phone);**

**printf("\t\t\tEnter gender : ");**

**fflush(stdin);**

**scanf("%s",sex);**

**printf("\t\t\tEnter your email : ");**

**fflush(stdin);**

**scanf("%s",mail);**

**insert(name, ccode, phone, sex,mail);**

**break;**

**case 2:**

**printf("\n\t\t\tEnter phone number to search: ");**

**scanf("%ld", &phone);**

**if (search(head,phone) == NULL)**

**printf("Value not present\n");**

**else**

**{**

**struct person \* s=search(head,phone);**

**struct person \* temp = head;**

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

**printf("\t\t\t\t\* \t Searched Person Found !!! \t \*\n");**

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

**printf(" NAME\t\t\t\t COUNTRY CODE\t\t PHONE NO\t\t GENDER\t\t EMAIL\n");**

**printf("---------------------------------------------------------------------------------------------------------------------------------------------\n");**

**while(temp!=NULL)**

**{**

**int i;**

**int len1 = 40 - strlen(temp->name);**

**int len2 = 19 - strlen(temp->country\_code);**

**int len3 = 15;**

**int len4 = 21 - strlen(temp->sex);**

**printf("%s",temp->name);**

**for(i=0; i<len1; i++)**

**printf(" ");**

**printf("%s",temp->country\_code);**

**for(i=0; i<len2; i++)**

**printf(" ");**

**printf("%ld",temp->mble\_no);**

**for(i=0; i<len3; i++)**

**printf(" ");**

**printf("%s",temp->sex);**

**for(i=0; i<len4; i++)**

**printf(" ");**

**printf("%s",temp->mail);**

**printf("\n");**

**fflush(stdin);**

**temp=temp->next;**

**}**

**break;**

**case 3:**

**printf("\n\t\t\tEnter phone number to delete: ");**

**scanf("%ld", &phone);**

**Delete(phone);**

**break;**

**case 4:**

**printf("\n\t\t\tEnter phone number to update: ");**

**scanf("%ld", &phone);**

**update(phone);**

**break;**

**case 5:**

**display();**

**break;**

**case 6:**

**quick\_sort(head,last\_node(head));**

**display();**

**break;**

**}**

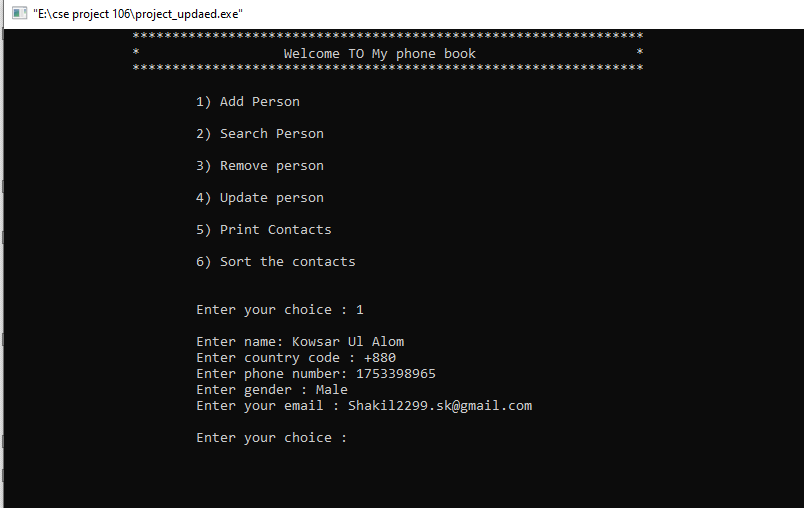
**}**

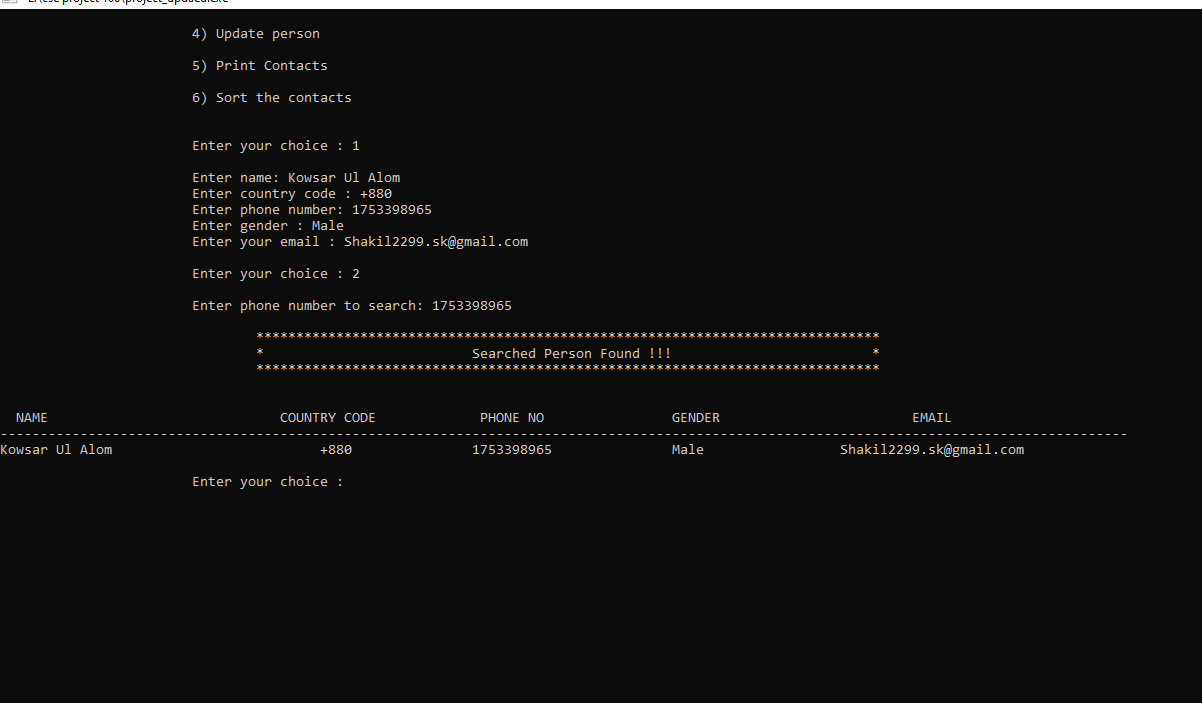
**}**

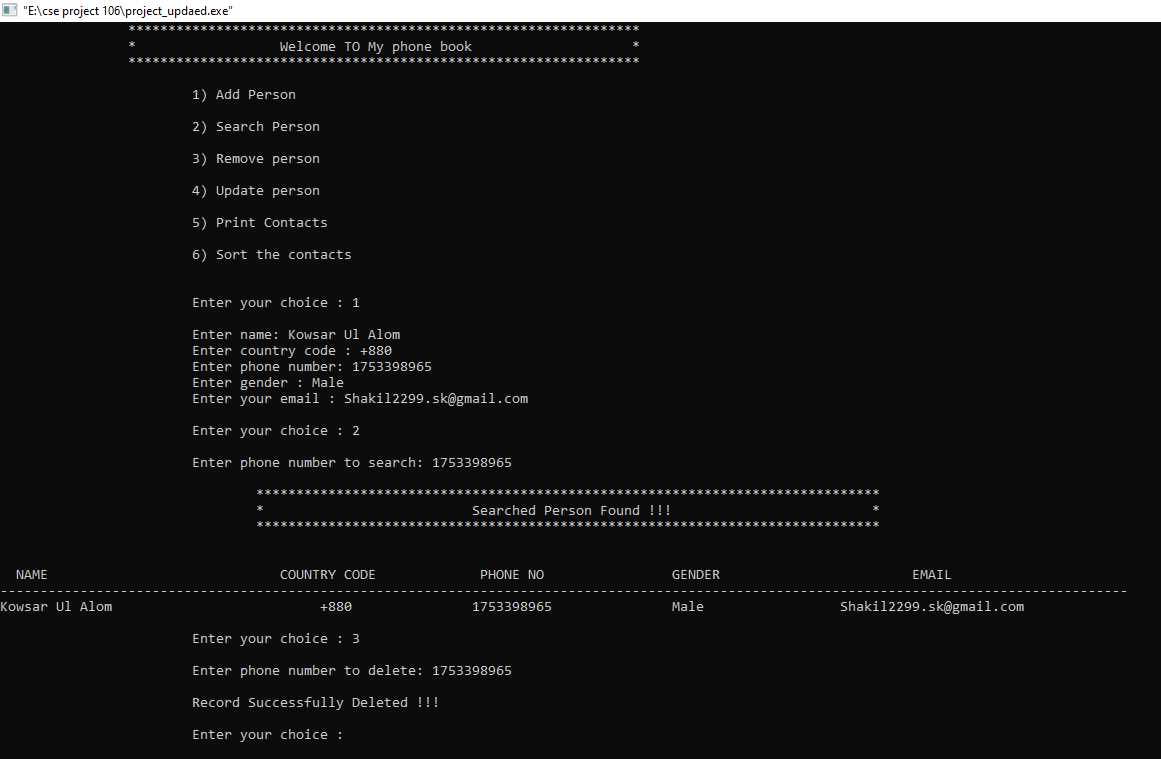
**}**

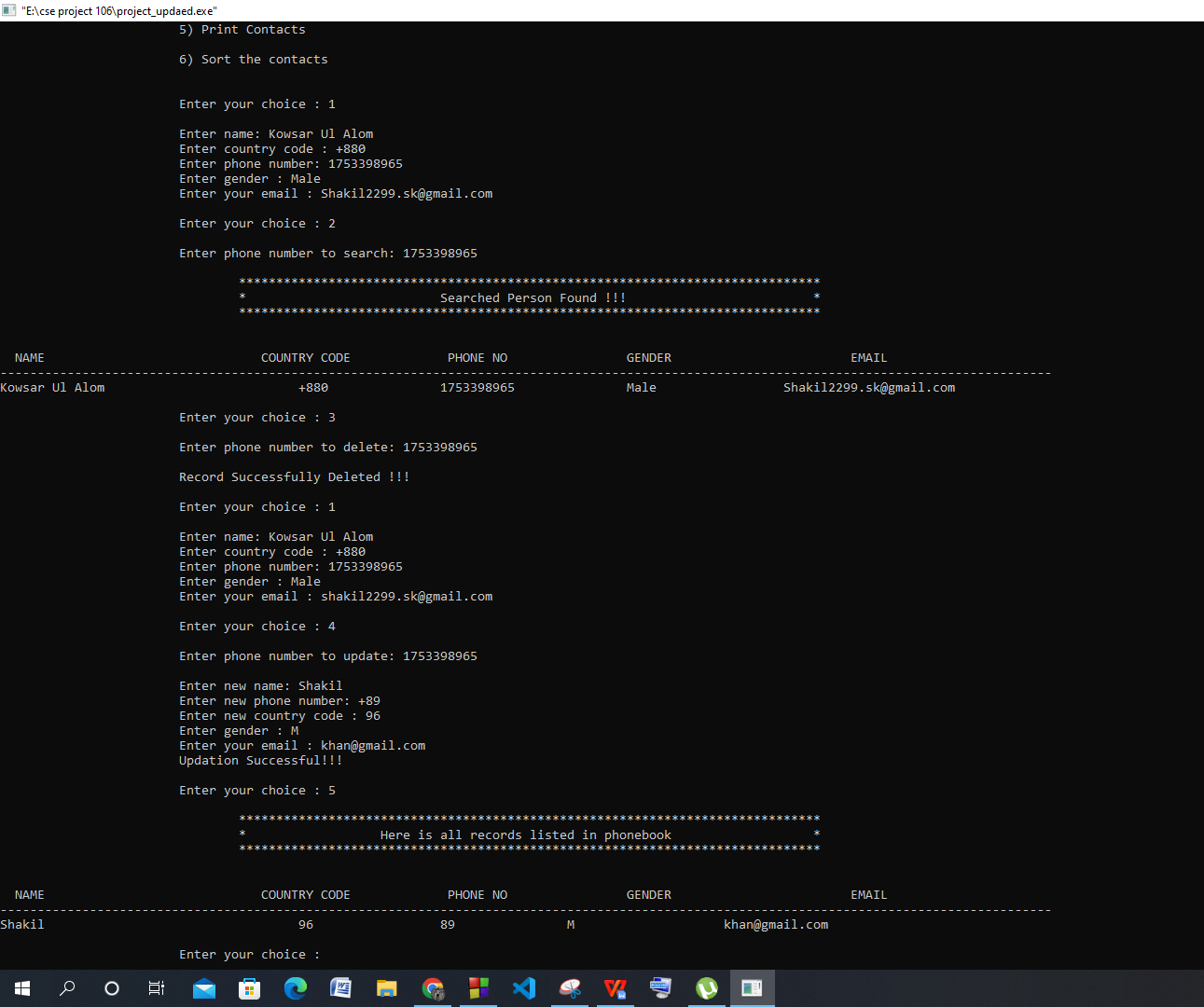
# Chapter 3 Performance Evaluation

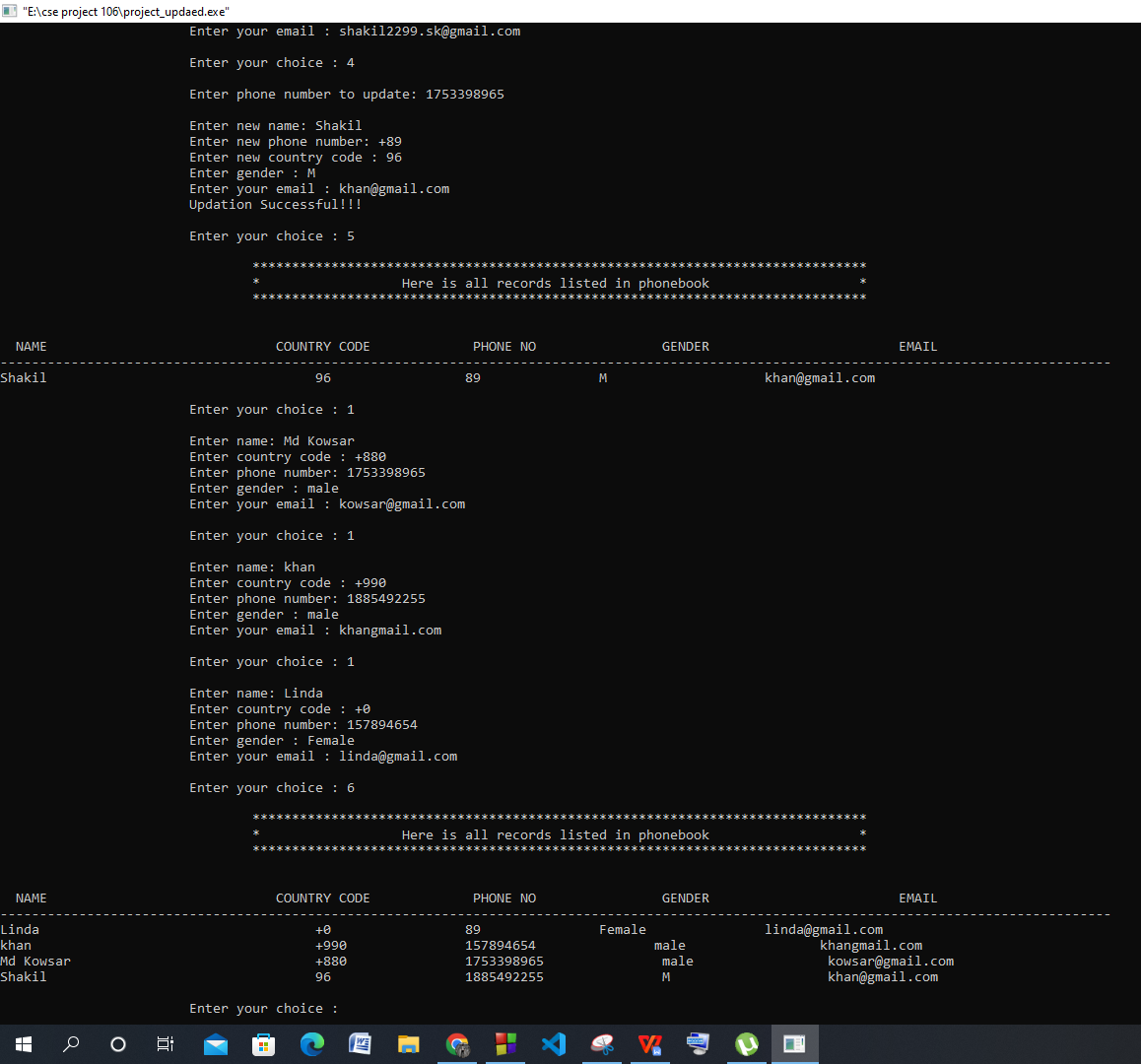
## Results and Discussions











# 

# Chapter 4 Conclusion

The application software has been implemented successfully by using test cases .And the language used is C language. This application is used to search, delete, modify and some functions which is used to remember our friends details more easily.

# References

1. Author Initial. Author Surname, Title. City: Publisher, Year Published, p. Pages Used.
2. A. Rezi and M. Allam, ”Techniques in array processing by means of transforma- tions, ” in Control and Dynamic Systems, Vol. 69, Multidemsional Systems, C. T. Leondes, Ed. San Diego: Academic Press, 1995, pp. 133-180.
3. O. B. R. Strimpel, ”Computer graphics,” in McGraw-Hill Encyclopedia of Science and Technology, 8th ed., Vol. 4. New York: McGraw-Hill, 1997, pp. 279-283.
4. K. Schwalbe, Information Technology Project Management, 3rd ed. Boston: Course Technology, 2004.
5. M. N. DeMers, Fundamentals of Geographic Information Systems, 3rd ed. New York: John Wiley, 2005.
6. M. Bell, et al., Universities Online: A survey of online education and services in Australia, Occasional Paper Series 02-A. Canberra: Department of Education, Science and Training, 2002.
7. Bdren, http://vsession.bdren.net.bd/faculty/classSchedules. [Accessed date: 13.09.21]