# PHONEBOOK MANAGEMENT SYSTEM

A

DATA STRUCTURES AND ALGORITHMS

**Bachelor of Technology**

in

**Computer Science & Engineering**

**By**


| | |
|---|---|
| K. SUHANI | 2103A51269 |
| S. SAHITHI | 2103A51291 |
| S. HRISHITHA | 2103A51244 |
| S. SRIJA | 2103A51296 |


Under the guidance of

JAGADISH SRIPELLI

Designation s

**Submitted to**

**School of Computer Science and Artificial Intelligence**


**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CERTIFICATE**

This is to certify that the **DSA Course Project** Report entitled **PHONEBOOK MANAGEMENT SYSTEM** is a record of bonafide work carried out by the students **K. SUHANI, S. HRISHITHA, S. SAHITHI, S.SRIJA** bearing Roll No(s) **2103A51269, 2103A51244, 2103A51291, 2103A51296** during the academic year 2022-2023

**Lab In-charge** **Head of the Department**

# INDEX/CONTENTS


| SL.NO | TITLE | PAGE NO. |
|---|---|---|
| 1. | ABSTRACT | 4 |
| 2. | INTRODUCTION | 5-6 |
| 3. | OBJECTIVES | 6 |
| 4. | MODULES DESCRIPTION | 7 |
| 5. | KNOWLEDGE REQUIRED | 8 |
| 6. | REQUIREMENTS | 8 |
| 7. | SOURCE CODE | 9-16 |
| 8. | RESULT | 17-20 |

# ABSTRACT:

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. 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 structures concepts has been extensively used for almost all functions in this mini project. Phonebook in C isa console application without graphics. The source code is complete and totally error-free. It is compiled in Code Blocks with DEV++ 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.

# INTRODUCTION:

The Phone Book project is developing a program which deals with the combination of structures, arrays, File pointers and other functions. This program could do some operations on arrays such as insertion, deletion, sorting, searching, update, retrieve, merging, append, exit. By implementing this program, we can execute the inserted contact data, deletion of the data, searching, updating, append, exit with numbers by using arrays and file pointers. This program is implemented for only numbers that can enter into an array. To do this analysis manually it takes a lot of time and patience but by implementing this program using a high-level language like C it becomes much easier. But before going to make final solution for the problem, the problem must be analysed. First of all, the basic information regarding the program which consists of complex numbers. This program is solved by using several methods like one can solve this program using user defined functions concept, loops conditions, go to statements. In this abstract we used the concept of functions, while loop, for loop, switch case and if condition's which helps to execute the problem much easier. The following steps are followed while implementing the given program using if and while loop. • The input is entered i.e., the value of choice (the menu no) select the particular menu • Next it goes to particular menu and then go to the particular function. It prints the resultant value which came from the execution. • The outcome of the work is one can get the required changes like inserting or deleting or searching or append or update or exit for a given array. Adding new records, displaying/showing them, search for contacts and deleting the phonebook records are the basic functions which make up the main menu of this phonebook application. Personal information

such as name, contact number, Gmail address, Facebook account are asked while adding a record into the phonebook. The record can be then be created, displayed, searched and removed. I have used many functions in this project.

# OBJECTIVES:

To store basic information of a person under a single contact number. If we're trying to find details of a person we need to search email id and Facebook separately with different applications. Here, by this Phone Book management we can get the details of email id and Facebook directly when we type the phone number.

# MODULES DESCRIPTION:

In this module four modules are used.

## 1.Inserting/Adding

In this module , we can perform insertion operation. So the user can add or insert an contact in the phonebook. While adding the contact , it asks us to enter the Contact name , Contact number , Gmail address and Facebook account.

## 2. Deleting

In this module , we can perform delete operation . So if the user wants to delete any contact which is no more useful for them then they can delete that contact while deleting the contact we need to enter the contact name and the contact will be deleted.

## 3. Displaying

In this module, all the contacts which are added to the phonebook will be displayed, Soo the user all see all the contacts in the contact list.

## 4. Searching

In this module, We can perform search operation. So if the user wants to search a particular contact then it asks us to enter the contact name and the remaining details will be viewed like the contact number, Gmail address and facebook account.

# KNOWLEDGE REQUIRED:

- ❖ Control statements (if, switch)
- ❖ Jump Statement-break
- ❖ Loops statements
- ❖ Arrays
- ❖ strings and its functions
- ❖ Pointers (pointer to strings and pointer to structures)
- ❖ (structures)
- ❖ Functions (Any type of user defined functions)
- ❖ Linked Lists: Single linked list

# REQUIREMENTS:

**Software requirements:**

- Edition Windows 11 Home Single Language
- Version 21H2
- Installed on 28-03-2022
- OS build 22000.1219
- Experience Windows Feature Experience Pack 1000.22000.1219.0
- Dev C++

**Hardware requirements:**

- This Application size is 33Kb and the size of the code is 5Kb so such amount of memory is required from hard disk.
- RAM: minimum 256MB.
- Mouse, keyboard.

## SOURCE CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <ctype.h>
typedef struct list add;
struct list
{
char name[40];
char num[20];
char gmail[40];
char fb[40];
add* next;
};
int i=1,size=0;
void Insert(add** head)
{
add* temp=(add*)malloc(sizeof(add));
printf("\n\n\n\t\tContact name : ");
gets(temp->name);
system("cls");
printf("\n\n\n\t\tContact Name : ");
gets(temp->name);
system("cls");
printf("\n\n\n\t\t\tMobile Number : ");
gets(temp->num);
```

```
system("cls");
printf("\n\n\n\t\t\tGmail id : ");
gets(temp->gmail);
system("cls");
printf("\n\n\n\t\t\tFacebook Account : ");
gets(temp->fb);
system("cls");
temp->next = NULL;
size++;
if(size == 100)
{
system("cls");
printf("\n\n\n\t\t\tPhone Memory is full!!!");
printf("\n\t\tIf you want to add more contact,You have to delete some contact from your list...\n\n\n");
return;
}
else
{
if(*head == NULL)
{
*head = temp;
system("cls");
printf("\n\n\t\t\t\t\tDone!!!\n\n");
return;
}
else
{
```

```
add* p = *head;
while(p->next != NULL)
{
p = p->next;
}
p->next = temp;
system("cls");
printf("\n\t\t\t\t\tDone!!!\n\n");
}
return;
}
}
void Search(add*head)
{
char ch[40];
printf("\n\n\n\t\tContact name : ");
gets(ch);
system("cls");
printf("\n\n\n\t\tContact name : ");
gets(ch);
system("cls");
if(head == NULL)
{
system("cls");
printf("\n\n\n\t\tNo Contact exists in this Phone Book List!!!\n\n");
return;
}
else
```

```
{
add*p = head;
while(p != NULL)
{
if(strcmp((p->name),ch) == 0)
{
system("cls");
printf("\n\t\t\t%s",p->name);
printf("\n\t\t------------------------");
printf("\n\t\tNumber : %s",p->num);
printf("\n\t\tGmail ID : %s",p->gmail);
printf("\n\t\tFacebook Account : %s\n\n",p->fb);
return;
}
p = p->next;
}
system("cls");
printf("\n\n\n\t\tThis Contact is not exists in the list! ");
}
}
void Delete(add** head)
{
char ch[40];
printf("\n\t\tContact name : ");
gets(ch);
system("cls");
printf("\n\t\tContact name : ");
gets(ch);
```

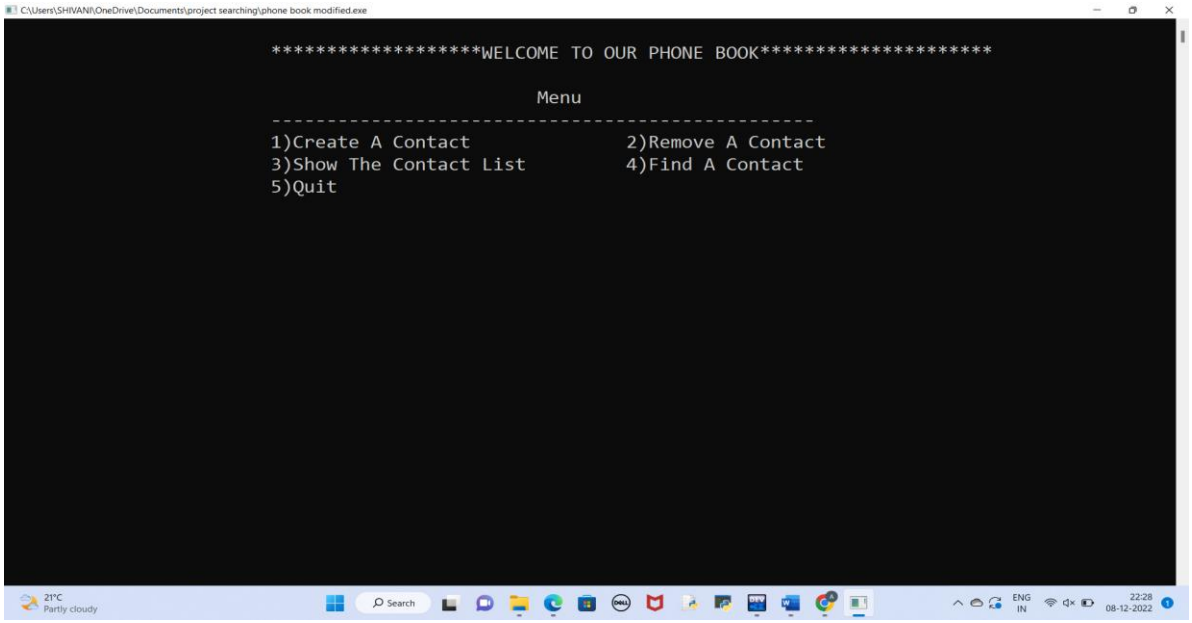
```
system("cls");
if(*head == NULL)
{
system("cls");
printf("\n\t\t\t\t\tNo Contact exists in this Phone Book List!\n\n");
return;
}
else
{
if(strcmp(((*head)->name),ch) == 0)
{
add*p=*head;
*head = (*head)->next;
free(p);
printf("\n\t\t\t\t\tDone!!!\n\n\n\n");
return;
}
else
{
add*p = *head;
while(p->next != NULL)
{
if(strcmp((p->next->name),ch) == 0)
{
p->next = p->next->next;
size--;
return;
}
```

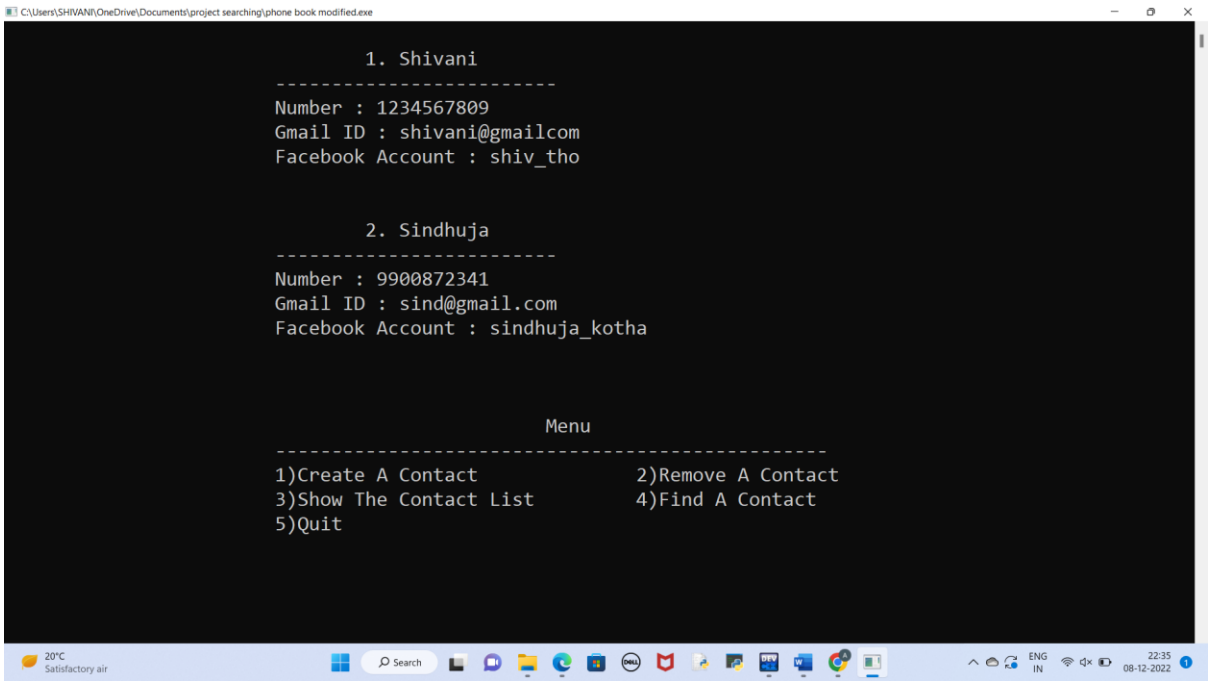
```
p = p->next;
}
system("cls");
printf("\n\t\t\t\t\tInvalid Contact!!!\n\n");
}
}
}
void Display(add* head)
{
if(head == NULL)
{
system("cls");
printf("\n\n\n\t\tNo Contact exists in this Phone Book List!");
return;
}
else
{
add*p = head;
while(p != NULL)
{
printf("\n\t\t\t\t%d.%c%s",i,32,p->name);
printf("\n\t\t\t------------------------");
printf("\n\t\t\tNumber : %s",p->num);
printf("\n\t\t\tGmail ID : %s",p->gmail);
printf("\n\t\t\tFacebook Account : %s\n\n",p->fb);
p = p->next;
i++;
}
```

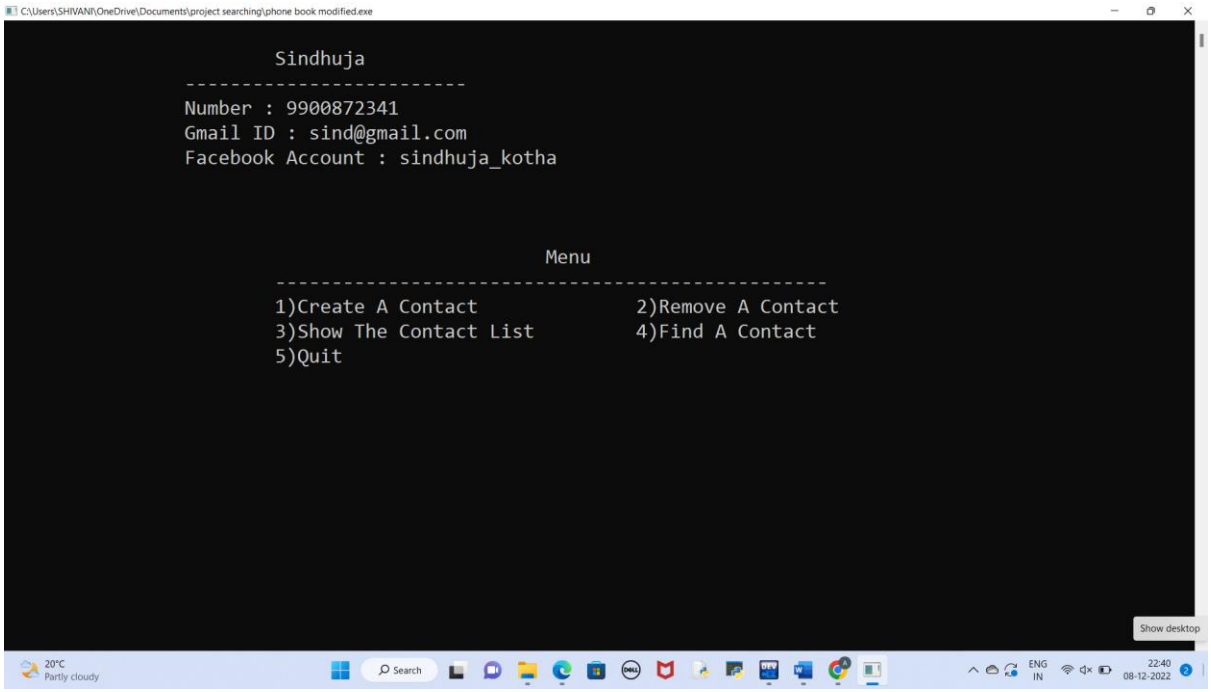
```
i=1;
return;
}
}
void main()
{
add* head=NULL;
char c[40];
mainhome:
system("cls");

printf("\n\t\t\t**********WELCOME TO OUR PHONE
BOOK**********");

while(1)
{
printf("\n\n\t\t\t\t\t\tMenu");
printf("\n\t\t\t-----------------------------------------------");
printf("\n\t\t\t1)Create A Contact\t\t2)Remove A Contact");
printf("\n\t\t\t3)Show The Contact List\t\t4)Find A Contact");
printf("\n\t\t\t5)Quit");
int ch;
scanf("%d",&ch);
if(ch == -1)
{
break;
}
else
```

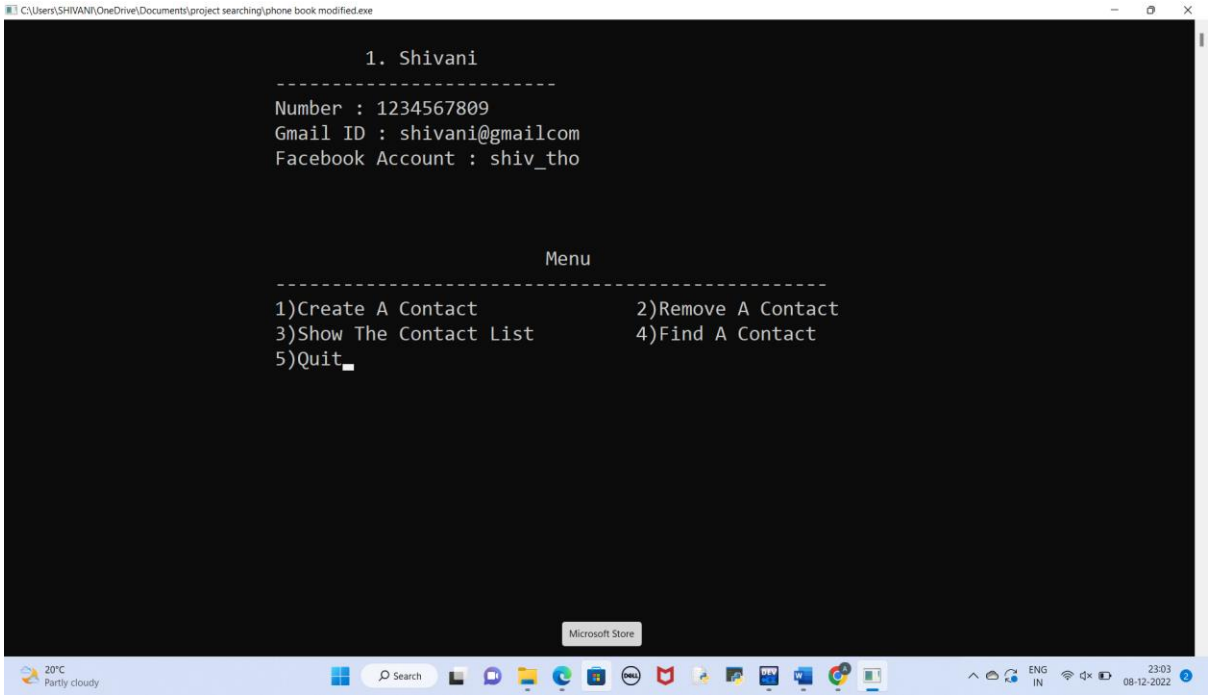
```
{
switch(ch)
{
case 1 :system("cls");
Insert(&head);
break;
case 2 :system("cls");
Delete(&head);
break;
case 3 :system("cls");
Display(head);
break;
case 4 :system("cls");
Search(head);
break;
case 5:
        exit(0);
default :printf("\n\t\tInvalid Choice!Try again!!!");
break;
}
}
}
}
```

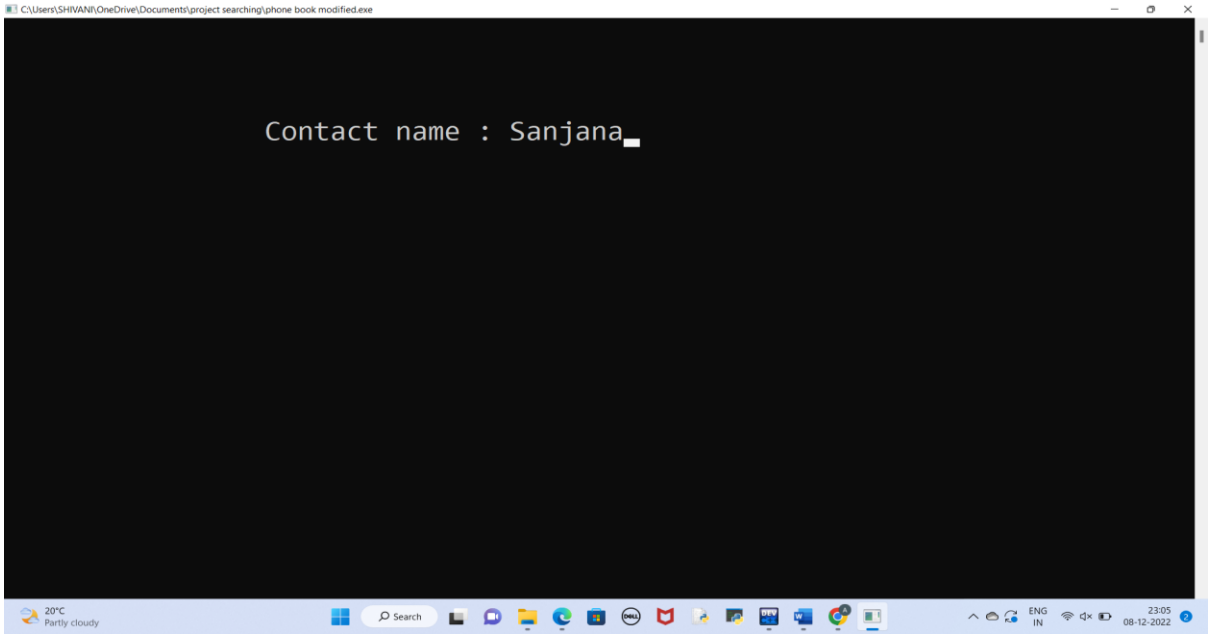
# OUTPUT:

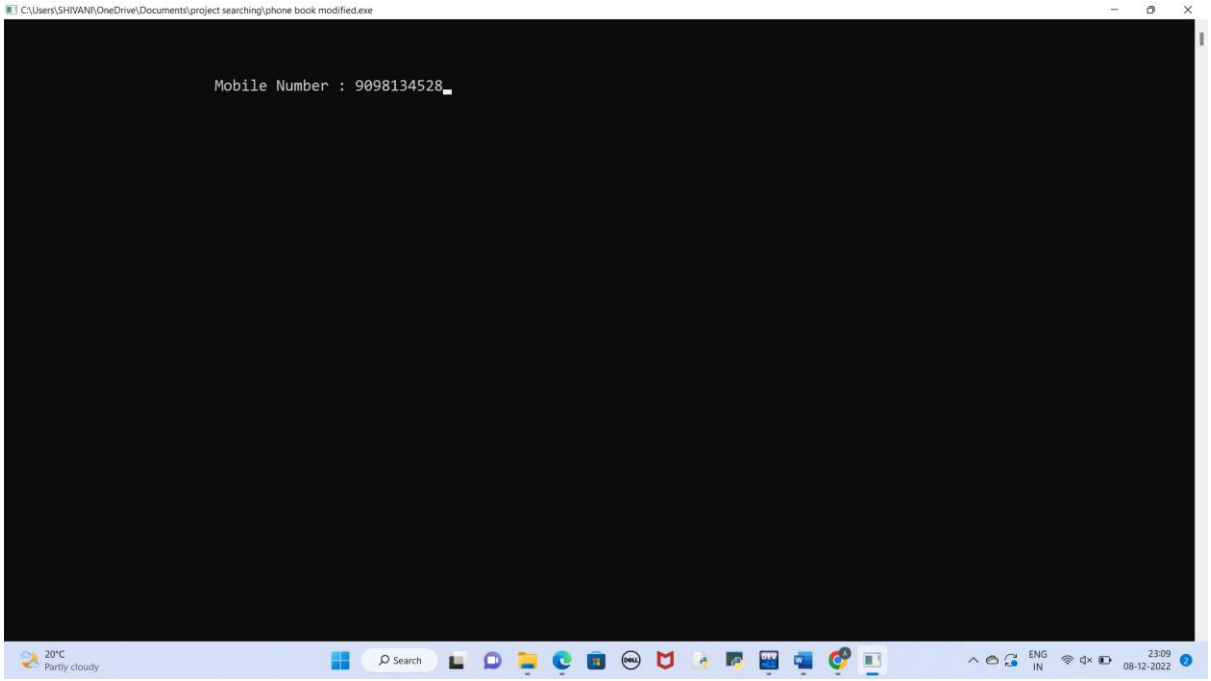

C:\Users\SHIVANI\OneDrive\Documents\project searching\phone book modified.exe
**********WELCOME TO OUR PHONE BOOK**********
Menu
1)Create A Contact
2)Remove A Contact
3)Show The Contact List
4)Find A Contact
5)Quit

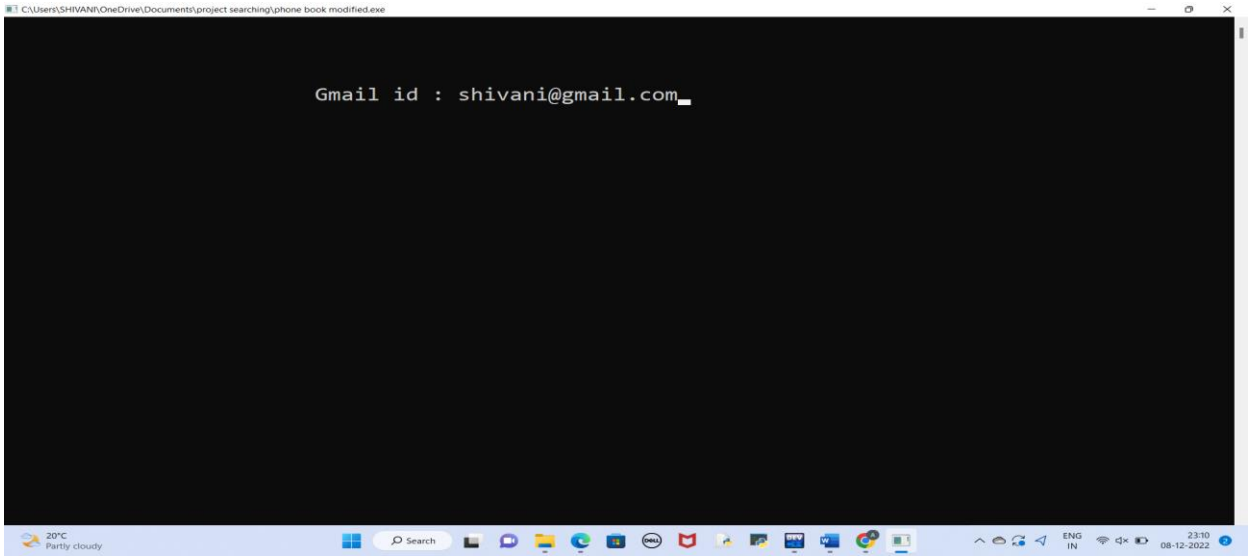

C:\Users\SHIVANI\OneDrive\Documents\project searching\phone book modified.exe
1. Shivani
Number : 1234567809
Gmail ID : shivani@gmailcom
Facebook Account : shiv_tho
2. Sindhuja
Number : 9900872341
Gmail ID : sind@gmail.com
Facebook Account : sindhuja_kotha
Menu
1)Create A Contact
2)Remove A Contact
3)Show The Contact List
4)Find A Contact
5)Quit

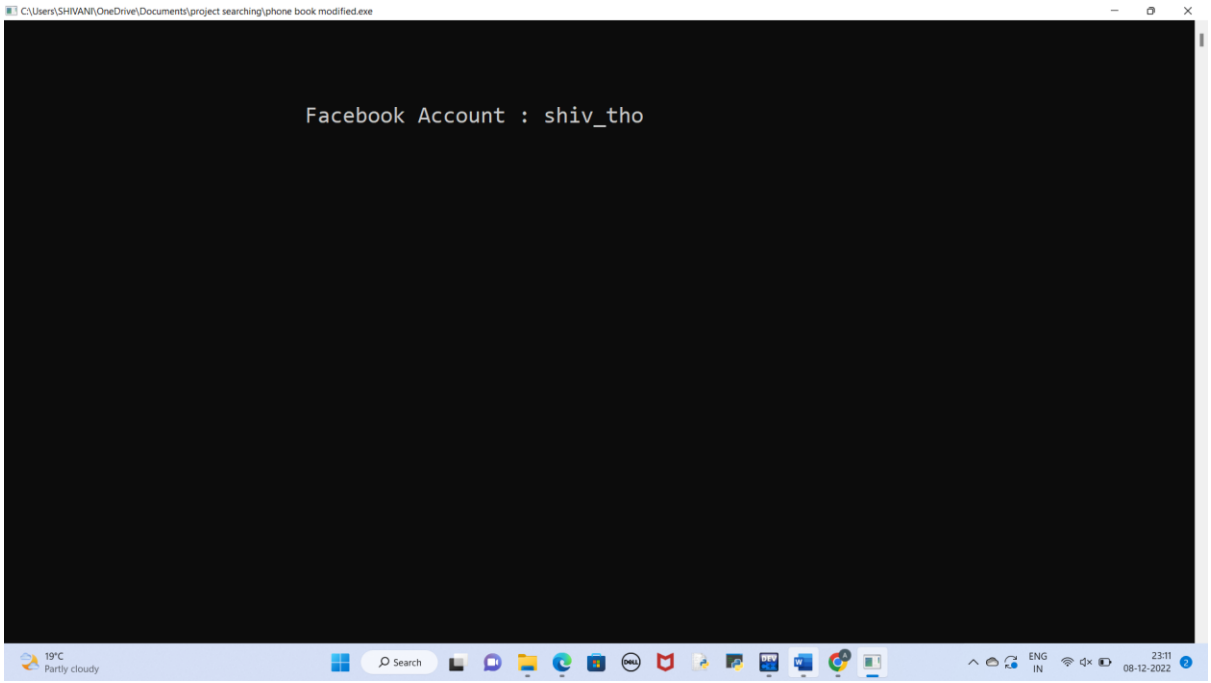

C:\Users\SHIVANI\OneDrive\Documents\project searching\phone book modified.exe
Sindhuja
Number : 9900872341
Gmail ID : sind@gmail.com
Facebook Account : sindhuja_kotha
Menu
1)Create A Contact
2)Remove A Contact
3)Show The Contact List
4)Find A Contact
5)Quit
Show desktop


C:\Users\SHIVANI\OneDrive\Documents\project searching\phone book modified.exe
1. Shivani
Number : 1234567809
Gmail ID : shivani@gmailcom
Facebook Account : shiv_tho
Menu
1)Create A Contact
2)Remove A Contact
3)Show The Contact List
4)Find A Contact
5)Quit
Microsoft Store


C:\Users\SHIVANI\OneDrive\Documents\project searching\phone book modified.exe
Contact name : Sanjana
20°C
Partly cloudy
Search
ENG
IN
23:05
08-12-2022


C:\Users\SHIVANI\OneDrive\Documents\project searching\phone book modified.exe
Mobile Number : 9098134528
20°C
Partly cloudy
Search
ENG
IN
23:09
08-12-2022


C:\Users\SHIVANI\OneDrive\Documents\project searching\phone book modified.exe
Gmail id : shivani@gmail.com
20°C
Partly cloudy
Search
ENG
IN
23:10
08-12-2022


C:\Users\SHIVANI\OneDrive\Documents\project searching\phone book modified.exe
Facebook Account : shiv_tho
19°C
Partly cloudy
Search
ENG
IN
23:11
08-12-2022