KATHMANDU UNIVERSITY

Department of Computer Science and Engineering

Dhulikhel, Kavre

Project Report of Phone Book System

Submitted By:	**Submitted To:**
Bhupendra Kumar Sah 3rd Semester	Dr.Prakash Poudyal Department of Computer Science and Engineering

Abstract

Phone Book System is a small web application developed for mini project. The phonebook is a very simple C++ (OOP) mini-project that can help us to understand the basic concepts of functions, Linked List, and data structure. In Olden days we stored all our important contact details in books and papers. Here we proposed a new system, by using this application we can store all the details in a central repository.

In manual method if we forget information book then it is very difficult to get the contact details. By using this application, we can see our contacts anywhere in the world. In this Project we can save out contacts and we can search them by name and we can see all of them at a time.

Table of Contents

Chapter 1: Introduction

The Phone Book System is the simple program. It can be Implement in various ways. This is Implemented by using Single Linked List. The Program will teach us how to add, delete, display the contact from the list. Phone Book is a project that is provide by Computer department for technical assignment help to us in that we get a simple Linked List 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.

Background:-

This system is developed using the general need required by the user while using the phone directory book. In order to keep updated the phone book directory, the admin will have the authority to add and delete as well as modify the existing records within the phone book directory. The users of the directory will only have the authority to search any particular record and listing details of the available records.

To Provide the search result within short interval of time we can Implemented this by using hashing. Hashing will provided the search results within the second and it’s take the time of O(1).

In Mobile, Phone Book System is required. It helps us to save out contact list. It’s still exits in our digital life. In modern days all the records are managed in digitally so, It’s the solution to manage the contact in digital ways.

Literature Survey:-

- Phone Book is a project that is provide by technical assignment help to us in that we get a simple Linked List/Hashing based solution to store our contacts.
- This will help user to easily search and manage contact using this system.
- The names and contacts numbers are stored in the memory by the user. So that user also can easily find the required person along with their address and telephone numbers and user search option.
- For searching operation, users will able to get any particular record using their contact or phone number or name but the only condition is that, customers records must be available within the file system.

Chapter 2: Objectives

This system is developed using the general need required by the user while using the phone directory book and provided a lot of facility to their user. The objectives of my Project Phone Book is to record the details various activities of user. It will simplify the task and reduce the paper word. The system is very user friendly and it is anticipated that administrations, academics, students and applicants will easily access functions of the system.

Most Objectives of this Project are following:-

- Large data can be stored.
- To record the details various activities of user.
- It will simplify the task and reduce the paper work.
- The system is very user friendly.
- Easily accessed by administrators, academics, students and applicants

Chapter 3: Proposed System

Under this Phonebook Project, all the functionality has been added to meet the user’s requirements in just few seconds. To provide the desired result on time, it is using with Linked List. User can stored large number of contact in the list for this the main idea is used is dynamic memory allocation. Which helps us to store the larger number of contact dynamically.

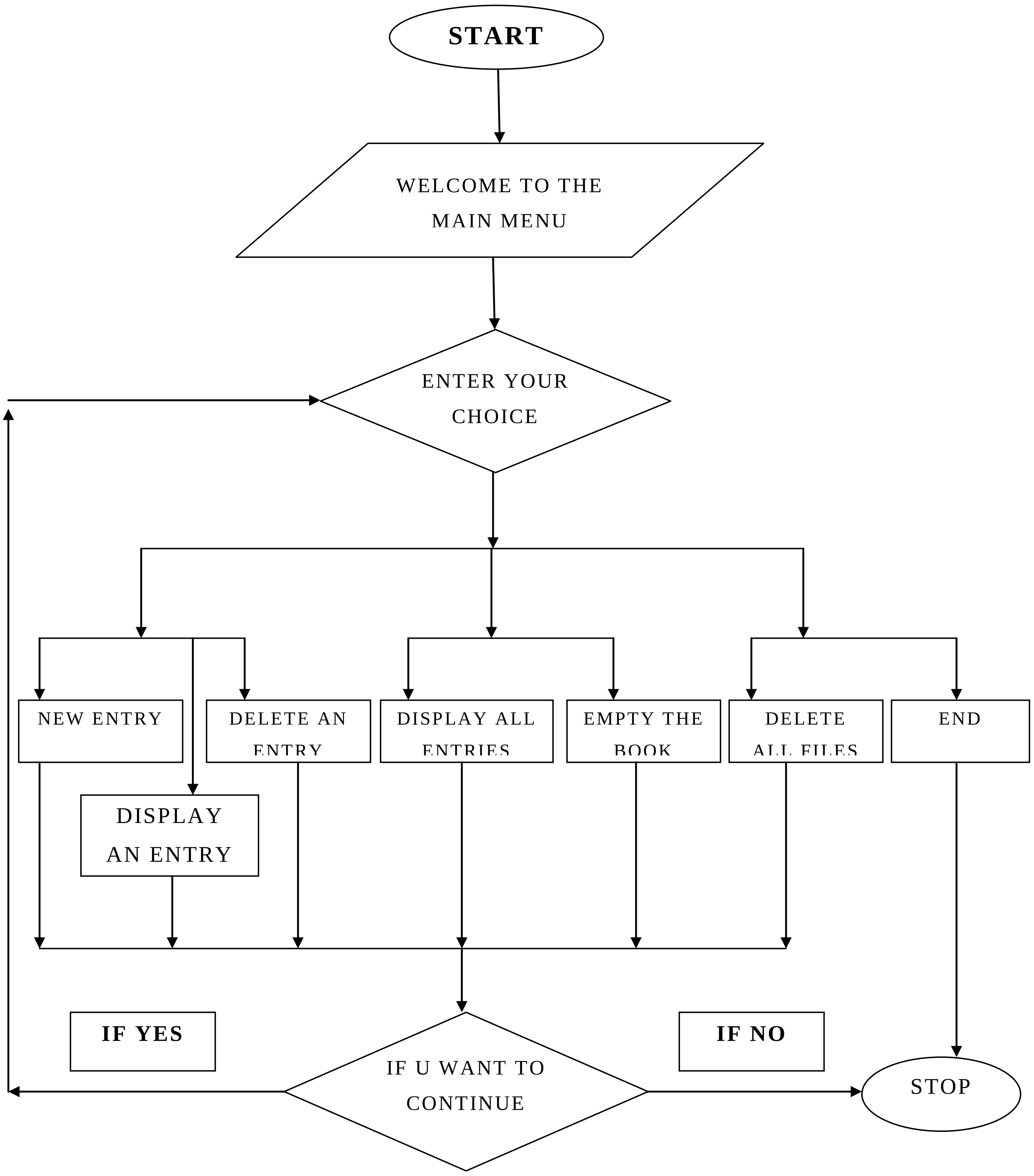
The names are present in user-sated formats like first list is added in first. When user wants to search the contact/name in the list it can show . All the contact are linked together so the searching is done in time of O(n). Which is only the disadvantages of this Phonebook System.

Chapter 4: Problem Statements

Phone Book is a project that is provide by technical assignment help to us in that we get a simple Linked List and Hashing 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. This system is developed using the general need required by the user while using the phone directory book. In order to keep updated the phone book directory, the admin will have the authority to add and delete as will as modify the existing records within the phone book directory.

The users of the directory will only have the authority to search any particular record and listing details of all available records. Admin will have the authority to perform various operations such as add customer records, search any particular record, delete record, delete all the existing records. To make all operations as easier as possible, user-friendly approach has been take into account by which users have to only give thei anser during the final confirmation to make their operations successful. The background processing system will take care of all processing task and maintain data integrity in order to reduce the redundancy of data. For searching operation, users will able to get any particular record using their contact number or name but the only condition is that, customers record must be available within the List. If no such record will be available, proper error message will be displayed as per user input provided to the system.

Chapter 5: System Design Flow Chart



Chapter 6: Implementation

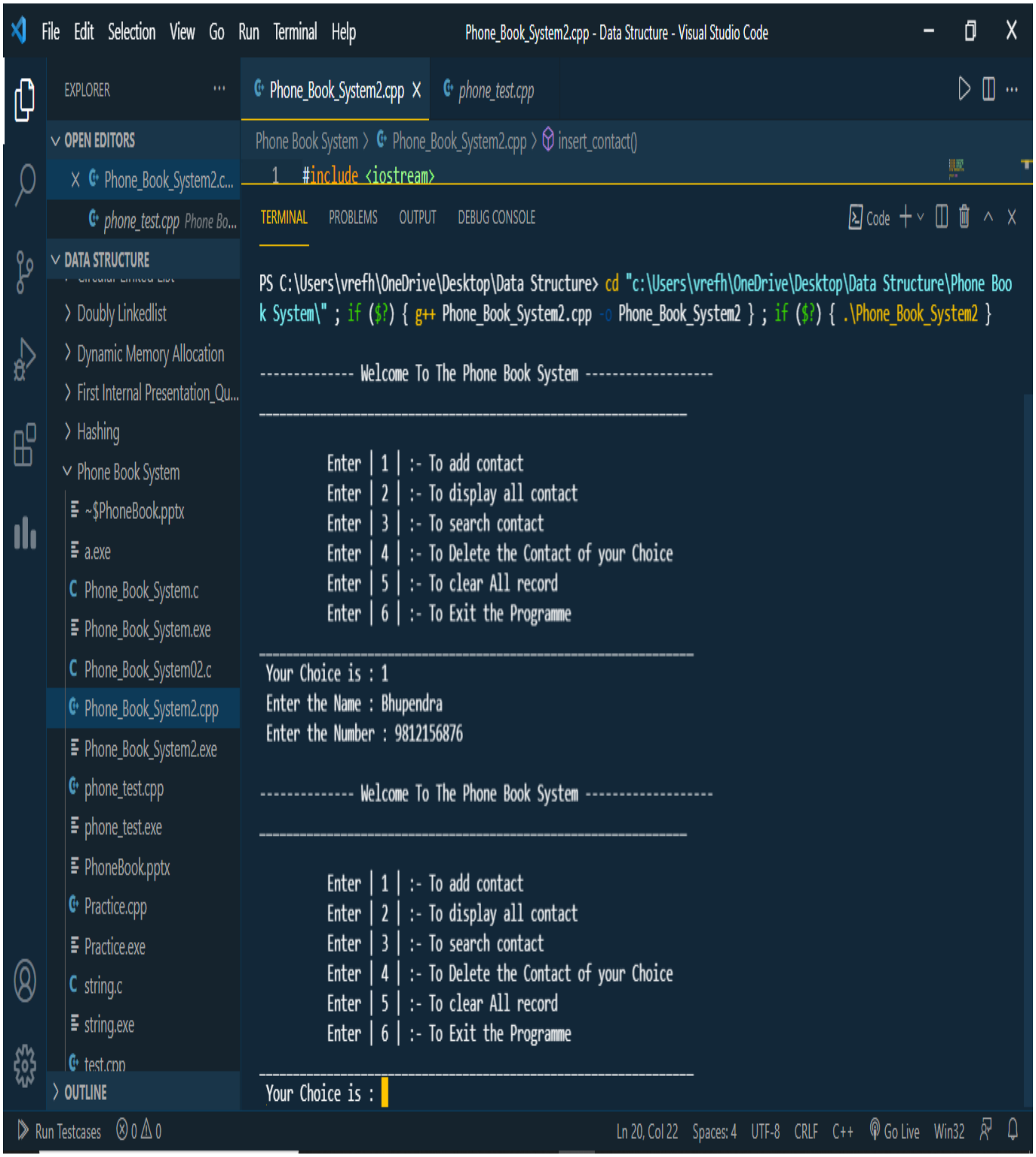
Code for Phone Book System:-

```
#include <iostream>
#include <stdlib.h>
#include <string>
using namespace std;

struct node
{
    string name, number;
    struct node *next;
};
struct node *temp;
struct node *head = NULL;
struct node *Create_node()
{
    struct node *newnode;
    newnode = new node;
    return (newnode);
}

void insert_contact()
{
    struct node *temp, *ptr;
    temp = Create_node();
    cout << " Enter the Name : ";
    cin >> temp->name;
    cout << " Enter the Number : ";
    cin >> temp->number;
    temp->next = NULL;
    if (head == NULL)
    {
        head = temp;
    }
    else
    {
        ptr = head;
        while (ptr->next != NULL)
        {
            ptr = ptr->next;
        }
        ptr->next = temp;
    }
}
void display_contact()
{
    struct node *travel;
    if (head == NULL)
    {
        cout << "\n\t Contact List is Empty  !!!  " << endl
             << endl;
    }
    travel = head;
    cout << " ----------Displying The Contact List --------------
-----" << endl
         << endl;
    cout << "*****************************************************
***" << endl
         << endl;
    while (travel != NULL)
    {
        cout << "| Name : " << travel->name << endl;
        cout << "| Number : " << travel->number << endl;
        cout << endl;
        travel = travel->next;
    }
    cout << "*****************************************************
***" << endl;
}
```

```
void search_contact()
{
    node *search_node = head;
    string srch;
    cout << "\n Enter your desired contact you want to search  :
 ";
    cin >> srch;
    bool found = false;
    if (head == NULL)
    {
        cout << "\n \t Sorry ! List is Empty !!!! " << endl;
    }
    else
    {
        while (search_node != NULL)
        {
            if (srch == search_node->name || srch == search_node-
>number)
            {
                cout << "\t \t___________________________________
_______" << endl;
                cout << "\n \t \t \t | Full name: " << search_nod
e->name << endl;
                cout << "\n \t \t \t | Phone number: " << search_
node->number << endl;
                cout << "\t \t___________________________________
_______" << endl;
                found = true;
                break;
            }
            search_node = search_node->next;
        }
    }
    if (found == true)
    {
        cout << "\t\t Your Contact is Sucessfully Found ! " << en
dl;
    }
    else
    {
        cout << "\t\t Desired contact not fount " << endl;
    }
}

void delete_Contact()
{
    node *search_node = head;
    node *temp = head;
    string Desrch;
    bool found = false;
    cout << "\n Enter your desired contact you want to Delete :
";
    cin >> Desrch;
    if (head == NULL)
    {
        cout << "\nList is Empty " << endl;
    }
    else
    {
        while (search_node != NULL)
        {
            if (Desrch == search_node-
>name || Desrch == search_node->number)
            {
                if (head->next == NULL)
                {
                    delete head;
                    head = NULL;
                    cout << " \n\t Contact has been Deleted Suces
sfully ! " << endl;
                    found = true;
                    break;
                }
                else if (search_node == head)
                {
                    search_node = head;
                    head = head->next;
                    delete search_node;
                    cout << " \n\t Contact has been Deleted Suces
sfully ! " << endl;
                    found = true;
                    break;
                }
                else if (search_node->next == NULL)
                {
                    temp->next = NULL;
                    delete search_node;
                     cout << " \n\t Contact has been Deleted Suce
ssfully ! " << endl;
                    found = true;
                    break;
                }
            }
            temp = search_node;
            search_node = search_node->next;
        }
    }
    if (found == false)
    {
        cout << "\n\n \t SORRY!!! Your Contact is not Found in th
e List ! \n"
             << endl;
    }
}

void clear_all()
{
    if (head == NULL)
    {
        cout << " \n \t Sorry !!! List is Empty . " << endl;
    }
    else
    {

        while (head != NULL)
        {
            temp = head;
            head = head->next;
            delete temp;
        }
        cout << "\n\t\tALL contact list has been deleted " << end
l;
    }
}
void menu()
{
    cout << endl;
    cout << "-------------- Welcome To The Phone Book System ----
---------------" << endl;
    cout << "____________________________________________________
___________" << endl
         << endl;
    ;
    cout << "\t  Enter | 1 | :- To add contact " << endl;
    cout << "\t  Enter | 2 | :- To display all contact " << endl;
    cout << "\t  Enter | 3 | :- To search contact " << endl;
    cout << "\t  Enter | 4 | :-
 To Delete the Contact of your Choice " << endl;
    cout << "\t  Enter | 5 | :- To clear All record " << endl;
    cout << "\t  Enter | 6 | :- To Exit the Programme " << endl;
    cout << "____________________________________________________
____________" << endl;
}
```

```cpp
int main()
{
    int op;
    while (true)
    {
        menu();
        cout << " Your Choice is : ";
        cin >> op;
        switch (op)
        {
        case 1:
            insert_contact();
            break;
        case 2:

            display_contact();
            break;
        case 3:
            search_contact();
            break;
        case 4:
            delete_Contact();
            break;
        case 5:
            clear_all();
            break;
        case 6:
            exit(0);
        default:
            cout << "SORRY !!! Invalid Inputs " << endl;
        }
    }
    return 0;
}
```

Chapter 7: Results (Input and Output Screen Snapshots)



```
#include <iostream>
```

```
PS C:\Users\vrefh\OneDrive\Desktop\Data Structure> cd "c:\Users\vrefh\OneDrive\Desktop\Data Structure\Phone Boo
k System\" ; if ($?) { g++ Phone_Book_System2.cpp -o Phone_Book_System2 } ; if ($?) { .\Phone_Book_System2 }

------------- Welcome To The Phone Book System ------------------

_____________________________________________________________


          Enter | 1 | :- To add contact
          Enter | 2 | :- To display all contact
          Enter | 3 | :- To search contact
          Enter | 4 | :- To Delete the Contact of your Choice
          Enter | 5 | :- To clear All record
          Enter | 6 | :- To Exit the Programme

______________________________________________________________
 Your Choice is : 1
 Enter the Name : Bhupendra
 Enter the Number : 9812156876

------------- Welcome To The Phone Book System ------------------

_____________________________________________________________


          Enter | 1 | :- To add contact
          Enter | 2 | :- To display all contact
          Enter | 3 | :- To search contact
          Enter | 4 | :- To Delete the Contact of your Choice
          Enter | 5 | :- To clear All record
          Enter | 6 | :- To Exit the Programme

______________________________________________________________
 Your Choice is :
```

Chapter 8: Conclusion

This application software has been computed successfully and was also tested successfully. It is user friendly, and has required options, which can be utilized by the user to perform the desired operations.

The software is developed using Linked List. The goals that are achieved by the software are:-

- ✓ Optimum utilization of resources.
- ✓ Efficient management of records.
- ✓ Simplification of the operations.
- ✓ Less processing time and getting required information.
- ✓ User friendly.
- ✓ Portable and flexible for further enhancement.

Reference/Bibliography

1. Code With Harry (https://www.youtube.com/watch?v=5_5oE5lgrhw&list=PLu0W_9lII9ahIappRPN0MCAgtOu3lQjQi)
2. Saurabh Shukla (https://www.youtube.com/watch?v=5uTQz43k4gg&list=PLe_7x5eaUqtXn30ILctb28rooeNkmdAib)
3. Telephone Directory (https://stackoverflow.com/questions/4579967/telephone-directory-using-linked-list)