

(Documentation)

Data Structure

Project name : Phone Directory System

(The main interface of the program)



Compiler used : (*Visual Studio 2019*)

- Before starting to run the code, you must install (**C++/CLI support for v142 build tools (latest)**)

Modifying — Visual Studio Community 2019 — 16.10.3

Workloads Individual components Language packs Installation locations

Search components (Ctrl+Q)

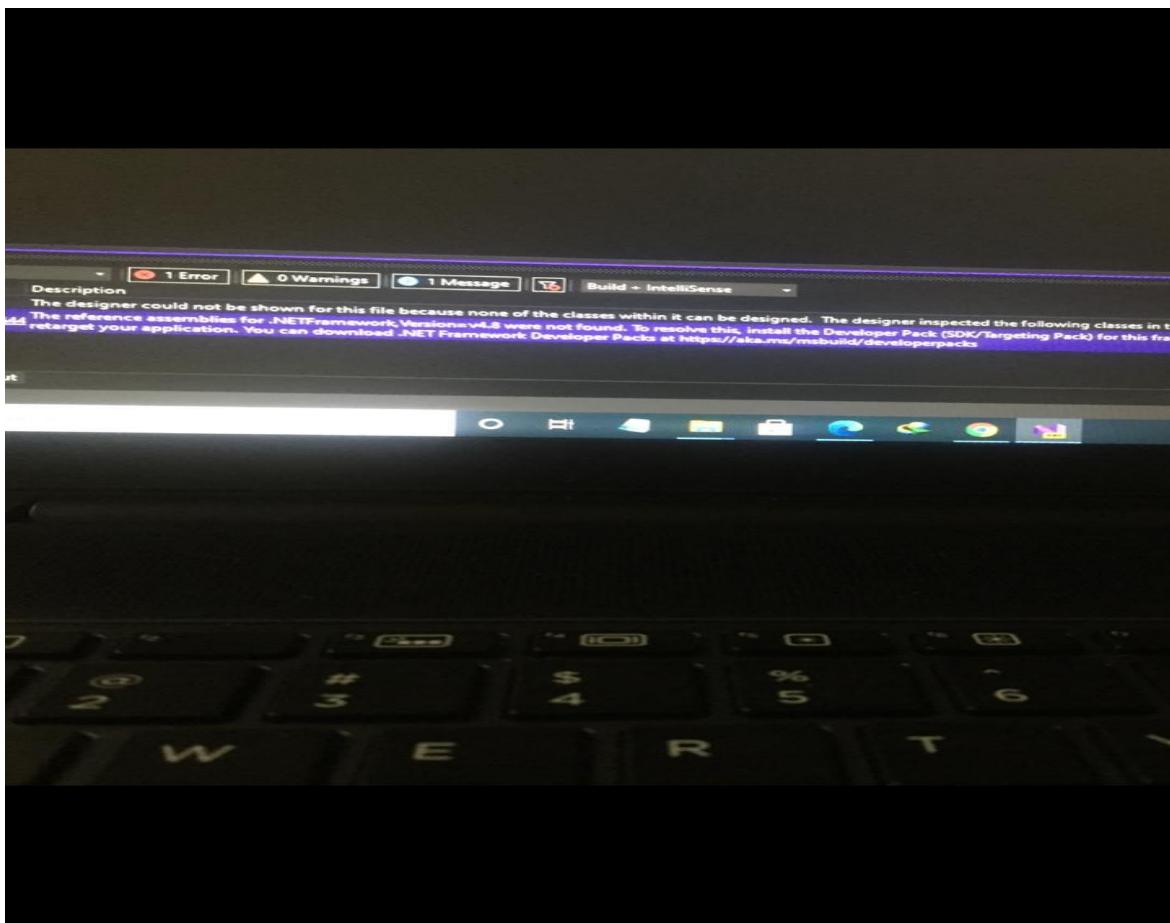
- ☐ C++ Windows XP Support for VS 2017 / (v141) tools [Deprecated]
- ☐ C++/CLI support for v141 build tools (14.16)
- ☐ C++/CLI support for v142 build tools (14.20)
- ☐ C++/CLI support for v142 build tools (14.21)
- ☐ C++/CLI support for v142 build tools (14.22)
- ☐ C++/CLI support for v142 build tools (14.23)
- ☐ C++/CLI support for v142 build tools (14.24)
- ☐ C++/CLI support for v142 build tools (14.25)
- ☐ C++/CLI support for v142 build tools (14.26)
- ☐ C++/CLI support for v142 build tools (14.27)
- ☒ C++/CLI support for v142 build tools (14.28-16.8)
- ☐ C++/CLI support for v142 build tools (14.28-16.9)
- ☐ C++/CLI support for v142 build tools (14.29-16.10)
- ☒ C++/CLI support for v142 build tools (Latest)
- ☐ Incredibuild - Build Acceleration
- ☒ MSBuild
- ☐ MSVC v140 - VS 2015 C++ build tools (v14.00)
- ☐ MSVC v141 - VS 2017 C++ ARM build tools (v14.16)
- ☐ MSVC v141 - VS 2017 C++ ARM Spectre-mitigated libs (v14.16)

Installation details

- Visual Studio core editor
- ▾ Desktop development with C++
 - ▾ Included
 - ✓ C++ core desktop features
 - ▾ Optional
 - ✓ MSVC v142 - VS 2019 C++ x64/x86 build t...
 - ✓ Windows 10 SDK (10.0.19041.0)
 - ✓ Just-In-Time debugger
 - ✓ C++ profiling tools
 - ✓ C++ CMake tools for Windows
 - ✓ C++ ATL for latest v142 build tools (x86 &...
 - ✓ Test Adapter for Boost.Test
 - ✓ Test Adapter for Google Test
 - ✓ Live Share
 - ✓ IntelliCode
 - ✓ C++ AddressSanitizer
 - ☐ MSVC v142 - VS 2019 C++ ARM64 build t...
 - ☐ C++ MFC for latest v142 build tools (x86...
 - ✓ C++/CLI support for v142 build tools (Late...
 - ☐ C++ Modules for v142 build tools (x64/x8...

After installing (C++/CLI support for v142 build tools (latest)), you should check whether you are facing a problem like this

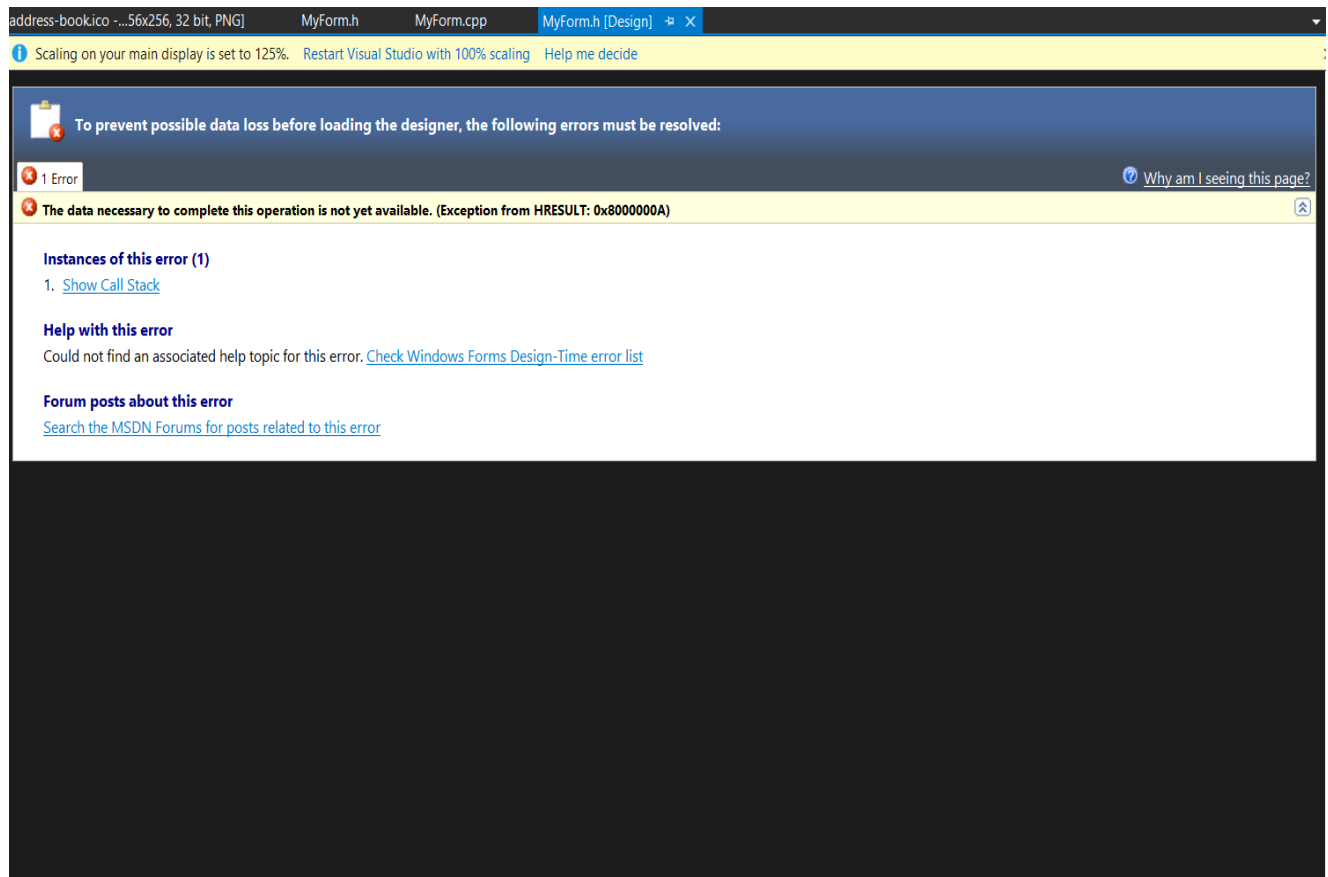
→(The reference assemblies for .NETFramework,version v4.8 were not found)



If you encounter this problem, you should install (ndp48-devpack-enu).

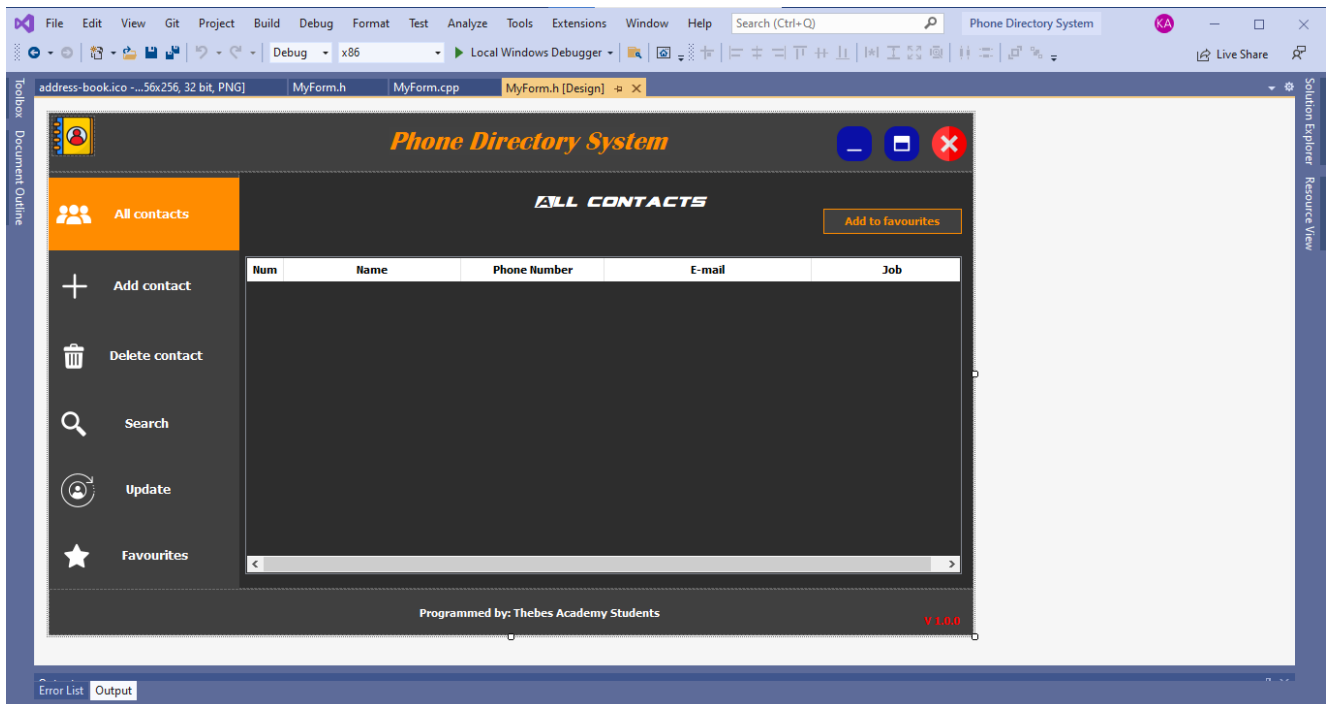
We have uploaded this program with the source file, it is located in a folder called (**ndp48-devpack-enu**).

Name	Date modified	Type	Size
.vs	1/1/2022 2:07 AM	File folder	
data	1/1/2022 7:49 AM	File folder	
Debug	1/1/2022 9:47 AM	File folder	
icons	1/1/2022 7:50 AM	File folder	
ndp48-devpack-enu	1/1/2022 10:40 AM	File folder	
icon.aps	12/14/2021 6:39 AM	APS File	2 KB
MyForm	1/1/2022 7:50 AM	C++ Source	1 KB
MyForm	1/1/2022 7:50 AM	C/C++ Header	140 KB
MyForm	1/1/2022 6:07 AM	.NET Managed Re...	313 KB
Phone Directory System	12/14/2021 12:58 AM	Microsoft Visual S...	2 KB
Phone Directory System	1/1/2022 4:56 AM	VC++ Project	7 KB
Phone Directory System.vcxproj	1/1/2022 4:56 AM	VC++ Project Filte...	2 KB
Phone Directory System.vcxproj	12/14/2021 12:58 AM	Visual Studio Proj...	1 KB
PhoneDirectorySystem	1/1/2022 7:47 AM	Application	560 KB
Resource.aps	1/1/2022 4:56 AM	APS File	130 KB
resource	1/1/2022 4:56 AM	C/C++ Header	1 KB
Resource	1/1/2022 4:56 AM	Resource Script	2 KB

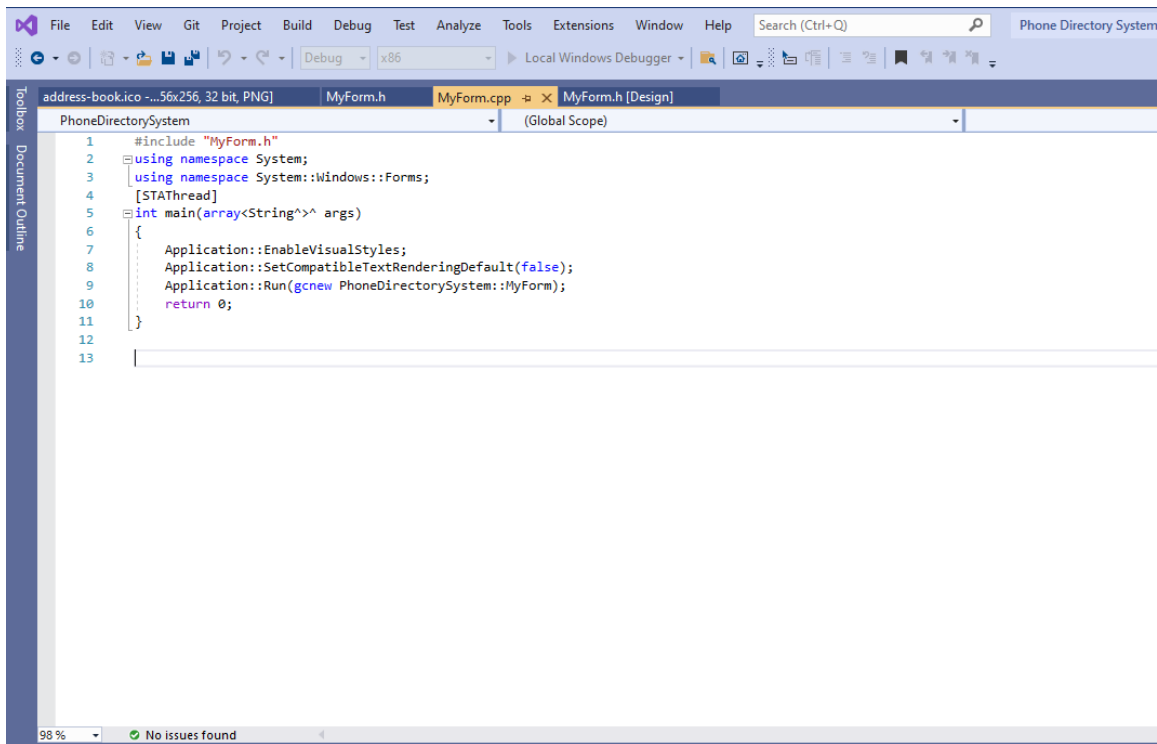


If this problem appears when opening the project file (**run the code then close the project file and reopen it again**).

MyForm.h[Design]



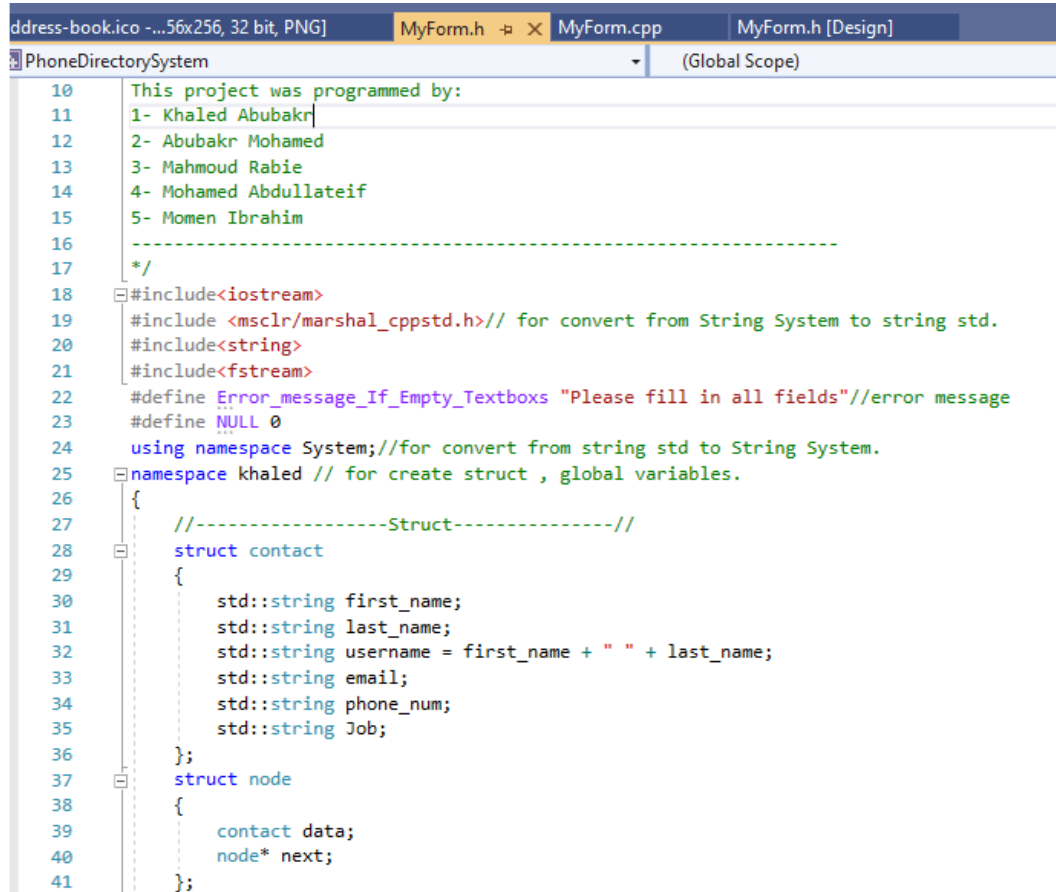
MyForm.cpp



```
1 #include "MyForm.h"
2 using namespace System;
3 using namespace System::Windows::Forms;
4 [STAThread]
5 int main(array<String^>^ args)
6 {
7     Application::EnableVisualStyles();
8     Application::SetCompatibleTextRenderingDefault(false);
9     Application::Run(gcnew PhoneDirectorySystem::MyForm);
10    return 0;
11 }
12
13
```

This contains the main function and initialization of the project.

MyForm.h



```
10 This project was programmed by:
11 1- Khaled Abubakr
12 2- Abubakr Mohamed
13 3- Mahmoud Rabie
14 4- Mohamed Abdullateif
15 5- Momen Ibrahim
16 -----
17 */
18 #include<iostream>
19 #include <msclr/marshal_cppstd.h> // for convert from String System to string std.
20 #include<string>
21 #include<fstream>
22 #define Error_message_Empty_Textboxs "Please fill in all fields"//error message
23 #define NULL 0
24 using namespace System; //for convert from string std to String System.
25 namespace khaled // for create struct , global variables.
26 {
27     //-----Struct-----//
28     struct contact
29     {
30         std::string first_name;
31         std::string last_name;
32         std::string username = first_name + " " + last_name;
33         std::string email;
34         std::string phone_num;
35         std::string Job;
36     };
37     struct node
38     {
39         contact data;
40         node* next;
41     };
42 }
```

we used `#include<msclr/marshal_cppstd.h>`

{

To convert from (String^) System to (string) std.

}

We used #include<fstream>

{

To write and read from file.

}

We used #define (Error_message_If_Empty_Textboxs**) "Please fill in all fields"**

{

We have defined an error message.

}

We used struct node

{

To use in Stack based linked list.

}

```

40     node* next;
41 };
42 //-----global pointers-----//
43 node* top = NULL;
44 //-----global variable-----//
45 int count_click_btn = 0;
46 int count_click_btn_delete = 0;
47 int count_click_search_btn = 0;
48 int count_click_btn_update = 0;
49 int count_click_btn_favourite = 0;
50 int count_click_search1_btn = 0;
51 //-----Functions Interfaces-----//

```

- We Declare pointer called top to used in stack based linked list.
- We have declared 6 counters so that we can display the contact data in the listView only once when the button is pressed .

Functions InterFaces

```
//-----Functions Interfaces-----//  
void erase();  
void save();  
int check_phonenum();  
int check_email();  
int check_email_in_edit();  
int read_count();  
int check_phonenum2();  
int check_phonenum_in_edit();  
int check_phonenum2_in_edit();  
void push(contact);  
void Bubble_sort();  
void count_contacts(int);  
void count_FVcontacts(int);  
int read_count_Favourite();  
void push_data_from_file();  
void ClearItemsFromListview();  
void display_in_AllContact_page();  
void display_in_DeleteContact_page();  
void display_in_SearchContact_page();  
void display_in_UpdateContact_page();  
void Merge(std::string* , std::string* , std::string* , std::string* , int , int , int );  
void MergeSort(std::string* , std::string* , std::string* , std::string* , int , int);  
void display_in_FavouritesContact_page();  
int check_Phonenum_in_Favourite();  
void save_edit();  
void liner_search1();  
//-----//
```

```

1
2 namespace PhoneDirectorySystem {
3
4     using namespace System;
5     using namespace System::ComponentModel;
6     using namespace System::Collections;
7     using namespace System::Windows::Forms;
8     using namespace System::Data;
9     using namespace System::Drawing;
10    //-----//
11    using namespace khaled; // for use struct and global variables.
12    using namespace System::IO; // This To write in file and read from file .->#include<fstream>.
13    //-----//

```

Using namespace Khaled

```
{
```

we can call the struct and global variable in the system.

```
}
```

Using namespace System::IO

```
{
```

To write and read from file

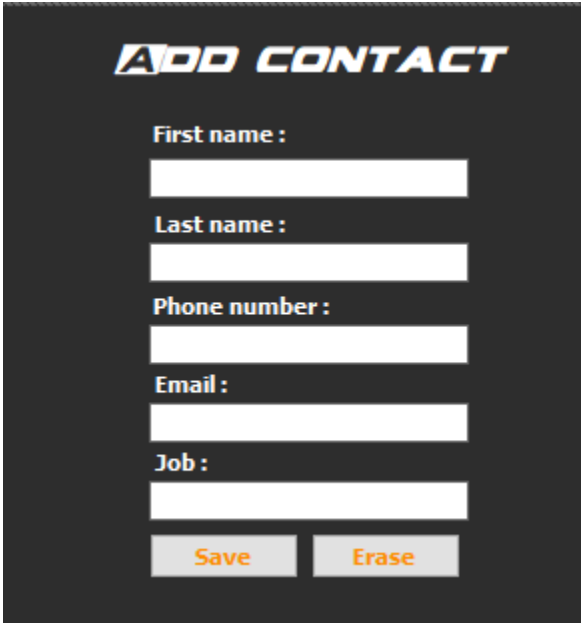
```
}
```

```

void erase()//This function is used to delete all text boxes used to add a new contact
{
    FNtextbox->Text = "\\0";
    LNtextbox->Text = "\\0";
    PNtextbox->Text = "\\0";
    Etextbox->Text = "\\0";
    Jtextbox->Text = "\\0";
}

```

This function is used to delete all text boxes used to add a new contact



The image shows a dark-themed user interface for adding a contact. At the top, the title "ADD CONTACT" is displayed in a stylized, italicized font. Below the title, there are five labeled text input fields stacked vertically: "First name :", "Last name :", "Phone number :", "Email :", and "Job :". Each label is followed by a white rectangular input box. At the bottom of the form, there are two buttons: "Save" and "Erase", both with orange text on a dark background.

```

int check_phonenum()
{
    String^ str = PNtextbox->Text;
    std::string x = msclr::interop::marshal_as< std
    int c = 0;
    for (int i = 0; i < x.length(); i++)//for check
    {
        if (x.at(i) >= 48 && x.at(i) <= 64)
        {
            c++;
        }
        else
        {
            break;
        }
    }
    if (x.length() != 11)// check if phone numbers
    {
        return 0;
    }
    else if (c == x.length())//for check if Are the
    {
        return 1;
    }
    else
    {
        return 2;
    }
}

```

Int Check_phonenum()

{

This function checks whether the phone number contains 11 digits or not, and this function checks whether there are characters or not.

}

This function is called when a new contact is added.

ADD CONTACT

First name :

Last name :

Phone number :

Email :

Job :

Check_phonenum() is called here.


```

int check_email()
{
    String^ str = Etextbox->Text;
    std::string x = msclr::interop::marshal_as< std::string >(str); //convert from String System to string st
    if (x.find("@") <= x.length())
    {
        if (x.find("@") == x.rfind("@"))
        {
            if (x.length() - x.find(".com", x.find("@")) == 4)
            {
                return 0;
            }
            else
            {
                return 1;
            }
        }
        else
        {
            return 1;
        }
    }
    else
    {
        return 1;
    }
}

```

int Check_email()

{

This function checks whether email contains (@) or not, and this function checks whether email contains (.com) or not

}

This function is called when a new contact is added.

ADD CONTACT

First name :

Last name :

Phone number :

Email :

Job :

Save

Erase

Check_email() is called here.

```

int check_email_in_edit()
{
    String^ str = Emailtextbox->Text;
    std::string x = msclr::interop::marshal_as< std::string >(str); //convert from String System to string
    if (x.find("@") <= x.length())
    {
        if (x.find("@") == x.rfind("@"))
        {
            if (x.length() - x.find(".com", x.find("@")) == 4)
            {
                return 0;
            }
            else
            {
                return 1;
            }
        }
        else
        {
            return 1;
        }
    }
    else
    {
        return 1;
    }
}
}

```

Int check_email_in_edit()

{

This function checks whether email contains (@) or not, and this function checks whether email contains (.com) or not

}

This function is called when a new contact is edited.

check_email_in_edit()

EDIT CONTACT

First name :

Last name :

Phone number :

E-mail :

Job

Save

Close

check_email_in_edit() is called here.

```

int check_phonenum2()
{
    std::ifstream file;
    std::string* PHONE_NUM = new std::string[read_cou
    int c = 1;
    file.open("data/PhoneNum.txt");
    for (int i = 0; i < read_count(); i++)
    {
        getline(file, PHONE_NUM[i]);
    }
    file.close();
    for (int i = 0; i < read_count(); i++)
    {
        String^ str = gcnew String(PHONE_NUM[i].data(
        if (str == PNTtextbox->Text)
        {
            c++;
        }
    }
    if (c > 1)
    {
        return 0;
    }
    else
    {
        return 1;
    }
    delete[]PHONE_NUM;
}

```

int check_phonenum2()

{

This function checks if the phone number already exists or no.

}

This function is called when a new contact is added.

```

int check_phonenum_in_edit()
{
    String^ str = PhoneNtextbox->Text;
    std::string x = msclr::interop::marshal_as< std::string >(str); //convert from String System to string std.
    int c = 0;
    for (int i = 0; i < x.length(); i++) //for check if Are there letters?.
    {
        if (x.at(i) >= 48 && x.at(i) <= 64)
        {
            c++;
        }
        else
        {
            break;
        }
    }
    if (x.length() != 11) // check if phone numbers field containing 11 digits.
    {
        return 0;
    }
    else if (c == x.length()) //for check if Are there letters?.
    {
        return 1;
    }
    else
    {
        return 2;
    }
}

```

Int Check_phonenum_in_edit()

{

This function checks whether the phone number contains 11 digits or not, and this function checks whether there are characters or not.

}

This function is called when a new contact is edited.

```

int check_phonenum2_in_edit()
{
    std::ifstream file;
    std::string* PHONE_NUM = new std::string[read_count()];
    int c = 1;
    file.open("data/PhoneNum.txt");
    for (int i = 0; i < read_count(); i++)
    {
        getline(file, PHONE_NUM[i]);
    }
    file.close();
    for (int i = 0; i < read_count(); i++)
    {
        String^ str = gcnew String(PHONE_NUM[i].data());
        if (Update_listView->SelectedItem[0]->SubItems[2]->Text == PhoneNtextbox->Text)
        {
            c = 1;
            break;
        }
        else if (str == PhoneNtextbox->Text)
        {
            c++;
        }
    }
    if (c > 1)
    {
        return 0;
    }
    else
    {
        return 1;
    }
    delete[]PHONE_NUM;
}

```

int check_phonenum2_in_edit()

{

This function checks if the phone number already exists or no.

}

This function is called when a new contact is edited.

```
void push(contact c)//stack based linked list
{
    node* new_node = new node;
    new_node->data = c;
    new_node->next = NULL;
    if (top != NULL)
    {
        new_node->next = top;
        top = new_node;
    }
    else
    {
        top = new_node;
    }
}
```

Void push(contact c)

{

This function is used to add an element to the stack

}


```

void Bubble_sort() {
    if (top != NULL)
    {
        node* i, * j;
        contact tmp;
        for (i = top; i->next != NULL; i = i->next) {
            for (j = top; j->next != NULL; j = j->next) {
                if (j->data.username > j->next->data.username) {
                    tmp = j->data;
                    j->data = j->next->data;
                    j->next->data = tmp;
                }
            }
        }
    }
}

```

Void Bubble_sort()

{

This function is used to sort contact before display in listview.

}



The bubble sort function is used in (display all contacts) in alphabetical order relative to the name

```
void count_contacts(int c) // how many Favourite contacts.  
{  
    String^ str = Convert::ToString(c);  
    StreamWriter file("data/count.txt");  
    file.WriteLine(str);  
    file.Close();  
}
```

Void count_contacts(int c)

{

This function used to save the count of contacts and when you add one contact it increases by one and when you delete one contact it decreases by one

}

```
void count_FVcontacts(int c)// how many contacts..  
{  
    String^ str = Convert::ToString(c);  
    StreamWriter file("data/FVcount.txt");  
    file.WriteLine(str);  
    file.Close();  
}
```

Void count_FVcontacts(int c)

{

This function is used to save the count of contacts in the favorites and when you add one contact it increases by one and when you delete one contact it decreases by one

}

```

int read_count()
{
    StreamReader file("data/count.txt");
    String^ str = file.ReadLine();
    file.Close();
    return Convert::ToInt32(str);
}
int read_count_Favourite()
{
    StreamReader file("data/FVcount.txt");
    String^ str = file.ReadLine();
    file.Close();
    return Convert::ToInt32(str);
}

```

Int read_count() and int read_count_Favourite()

{

This function is used to read the contact count and this function is used to read the count and use as size to array

}

```

void push_data_form_file()// push data from file .here used stack based linked list.
{
    std::ifstream file, file1, file2, file3, file4, file5;
    file.open("data/FirstName.txt");
    file1.open("data/LastName.txt");
    file2.open("data/PhoneNum.txt");
    file3.open("data/Email.txt");
    file4.open("data/Job.txt");
    file5.open("data/username.txt");
    for (int i = 0; i < read_count(); i++)
    {
        contact c;
        getline(file, c.first_name);
        getline(file1, c.last_name);
        getline(file2, c.phone_num);
        getline(file3, c.email);
        getline(file4, c.Job);
        getline(file5, c.username, '\n');
        push(c);
    }
    file.close();
    file1.close();
    file2.close();
    file3.close();
    file4.close();
    file5.close();
}

```

Push_data_form_file()

```
{
```

This function is used to read the contact data from the file and then add it to the stack

```
}
```

```

void ClearItemsFromListview()//Clear All Items from listview.
{
    listViewItem = gcnew Windows::Forms::ListViewItem("\\0");
    All_listView->Items->Clear();
    Delete_listView->Items->Clear();
    Update_listView->Items->Clear();
}

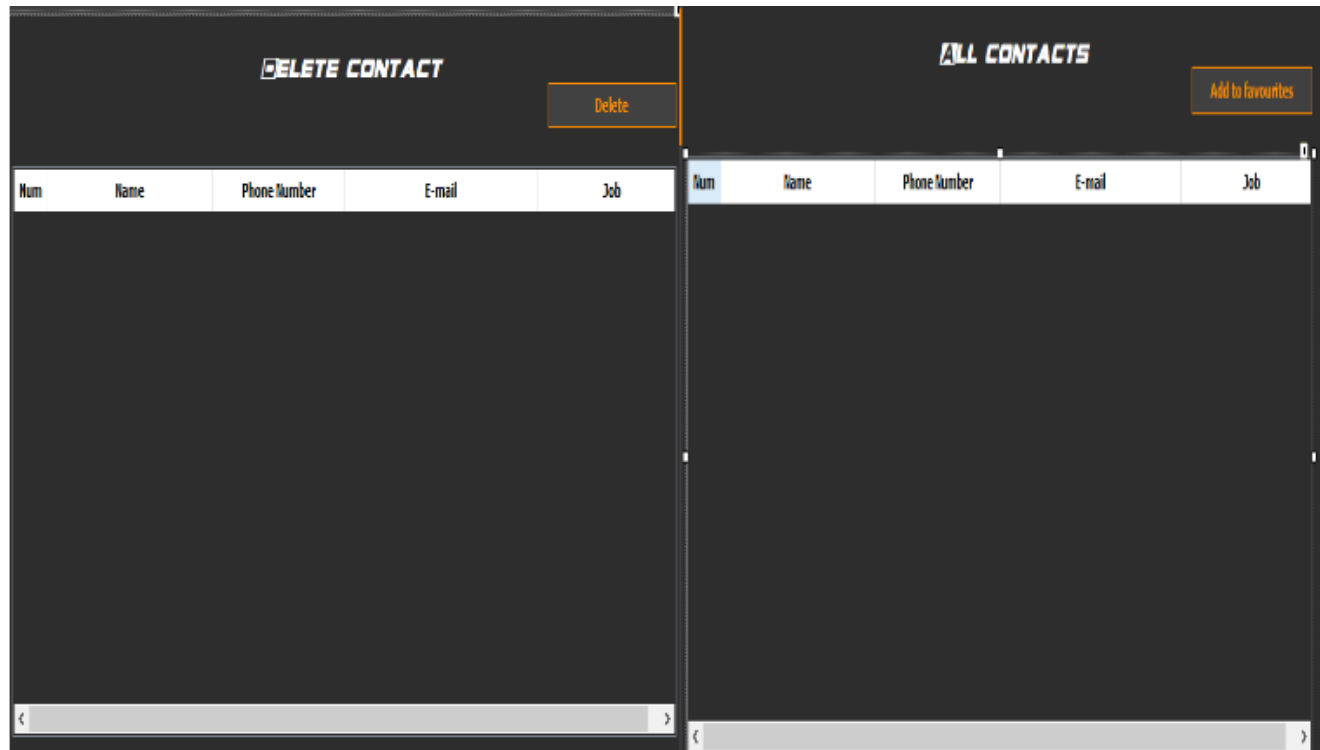
```

Void ClearItemsFormListView()

```
{
```

This Function used to delete items from (all_listview ,delete_listview and update_listview)

```
}
```



```

void display_in_AllContact_page()// display list from stack in viewlist.
{
    push_data_form_file();
    Bubble_sort();
    std::string un, pn, e, j;
    int count = 1;
    while (top != NULL)
    {
        String^ c;
        c = Convert::ToString(count);
        un = top->data.username;
        pn = top->data.phone_num;
        e = top->data.email;
        j = top->data.Job;
        top = top->next;
        //-----//
        String^ str1 = gcnew String(un.data());
        String^ str2 = gcnew String(pn.data());
        String^ str3 = gcnew String(e.data());
        String^ str4 = gcnew String(j.data());
        listViewItem = gcnew Windows::Forms::ListViewItem(c);
        listViewItem->SubItems->Add(str1);
        listViewItem->SubItems->Add(str2);
        listViewItem->SubItems->Add(str3);
        listViewItem->SubItems->Add(str4);
        All_listView->Items->Add(listViewItem);
        count++;
    }
}

```

Void display_in_All_Contact_page()

{

This function displays contacts from stack in All_listView.

}


```

void display_in_DeleteContact_page()
{
    {
        push_data_form_file();
        Bubble_sort();
        std::string un, pn, e, j;
        int count = 1;
        while (top != NULL)
        {
            String^ c;
            c = Convert::ToString(count);
            un = top->data.username;
            pn = top->data.phone_num;
            e = top->data.email;
            j = top->data.Job;
            top = top->next;
            //-----//
            String^ str1 = gcnew String(un.data());
            String^ str2 = gcnew String(pn.data());
            String^ str3 = gcnew String(e.data());
            String^ str4 = gcnew String(j.data());
            listViewItem = gcnew Windows::Forms::ListViewItem(c);
            listViewItem->SubItems->Add(str1);
            listViewItem->SubItems->Add(str2);
            listViewItem->SubItems->Add(str3);
            listViewItem->SubItems->Add(str4);
            Delete_listView->Items->Add(listViewItem);
            count++;
        }
    }
}

```

Void display_in_DeleteContact_page()

{

This function displays contacts from stack in Delete_listView.

}

```

void display_in_SearchContact_page()
{
    push_data_form_file();
    Bubble_sort();
    std::string un, pn, e, j;
    int count = 1;
    while (top != NULL)
    {
        String^ c;
        c = Convert::ToString(count);
        un = top->data.username;
        pn = top->data.phone_num;
        e = top->data.email;
        j = top->data.Job;
        top = top->next;
        //-----//
        String^ str1 = gcnew String(un.data());
        String^ str2 = gcnew String(pn.data());
        String^ str3 = gcnew String(e.data());
        String^ str4 = gcnew String(j.data());
        listViewItem = gcnew Windows::Forms::ListViewItem(c);
        listViewItem->SubItems->Add(str1);
        listViewItem->SubItems->Add(str2);
        listViewItem->SubItems->Add(str3);
        listViewItem->SubItems->Add(str4);
        searchlistView->Items->Add(listViewItem);
        count++;
    }
}

```

Void display_in_SearchContact_page()

{

This function displays contacts from stack in searchlistView.

}

```

void display_in_UpdateContact_page()
{
    push_data_form_file();
    Bubble_sort();
    std::string un, pn, e, j;
    int count = 1;
    while (top != NULL)
    {
        String^ c;
        c = Convert::ToString(count);
        un = top->data.username;
        pn = top->data.phone_num;
        e = top->data.email;
        j = top->data.Job;
        top = top->next;
        //-----//
        String^ str1 = gcnew String(un.data());
        String^ str2 = gcnew String(pn.data());
        String^ str3 = gcnew String(e.data());
        String^ str4 = gcnew String(j.data());
        listViewItem = gcnew Windows::Forms::ListViewItem(c);
        listViewItem->SubItems->Add(str1);
        listViewItem->SubItems->Add(str2);
        listViewItem->SubItems->Add(str3);
        listViewItem->SubItems->Add(str4);
        Update_listView->Items->Add(listViewItem);
        count++;
    }
}

```

Void display_in_UpdateContact_page()

{

This function displays contacts from stack in Update_listView.

}

```
void Merge(std::string* a, std::string* b, std::string* c, std::string* d, int low, int high, int mid) { ... }  
void MergeSort(std::string* a, std::string* b, std::string* c, std::string* d, int low, int high) { ... }
```

Void merge() and mergesort()

{

This functions Alphabetizes your current favuorite contacts By use recursion

}

```

void display_in_FavouritesContact_page()
{
    std::ifstream file, file1, file2, file3, file4;
    std::string* USERNAME = new std::string[read_count()];
    std::string* MOBILENUM = new std::string[read_count()];
    std::string* EMAIL = new std::string[read_count()];
    std::string* JOB = new std::string[read_count()];
    //-----//
    file.open("data/username.txt");
    file2.open("data/PhoneNum.txt");
    file3.open("data/Email.txt");
    file4.open("data/Job.txt");
    //-----//
    for (int i = 0; i < read_count(); i++)
    {
        getline(file, USERNAME[i]);
        getline(file2, MOBILENUM[i]);
        getline(file3, EMAIL[i]);
        getline(file4, JOB[i]);
    }
    //-----//
    file.close();
    file1.close();
    file2.close();
    file3.close();
    file4.close();
    //-----//
    std::ifstream file6;
    std::string* CHECKPHONENUM = new std::string[read_count_Favourite()];
    file6.open("data/Favourite.txt");
    for (int i = 0; i < read_count_Favourite(); i++)
    {
        getline(file6, CHECKPHONENUM[i]);
    }
    file6.close();
    //-----//
    int count = 1;
    //-----//
    MergeSort(USERNAME, MOBILENUM, EMAIL, JOB, 0, (read_count() - 1));
    //-----//
    for (int i = 0; i < read_count(); i++)

```

Void display_in_FavouritesContact_page()

{

This function use in display contacts in FVlistview.

}

```

void liner_search1()
{
    push_data_form_file();
    Bubble_Sort();
    std::string fn, ln, un, pn, e, j;
    int count = 1;
    while (top != NULL)
    {
        String* c;
        c = Convert::ToString(count);
        fn = top->data.first_name;
        ln = top->data.last_name;
        un = top->data.username;
        pn = top->data.phone_num;
        e = top->data.email;
        j = top->data.job;
        top = top->next;
        //-----//
        String* str = gcnew String(fn.data());
        String* str0 = gcnew String(ln.data());
        String* str1 = gcnew String(un.data());
        String* str2 = gcnew String(pn.data());
        String* str3 = gcnew String(e.data());
        String* str4 = gcnew String(j.data());
        if (searchtextbox->Text == str || searchtextbox->Text == str0 || searchtextbox->Text == str1 || searchtextbox->Text == str2 || searchtextbox->Text == str3 || searchtextbox->Text == str4)
        {
            ListViewItem = gcnew Windows::Forms::ListViewItem(c);
            ListViewItem->SubItems->Add(str1);
            ListViewItem->SubItems->Add(str2);
            ListViewItem->SubItems->Add(str3);
            ListViewItem->SubItems->Add(str4);
            searchlistviewdemo->Items->Add(ListViewItem);
        }
        count++;
    }
}
//-----//

```

Void liner_search1()

{


This Function searches for a contact using the first name, second name, first name + second name, phone number, email, or job.

}

```

private: System::Void pn_all_Paint(System::Object^ sender, System::Windows::Forms::PaintEventArgs^ e)
{
    if (All_listView->Items->Count > 0)
    {
        label19->Text = "\\0";
    }
    else
    {
        label19->Text = "Not found contacts,please add contacts for display";
    }
    if (count_click_btn == 0)
    {
        display_in_AllContact_page();
        count_click_btn += 2;
    }
    if (All_listView->Items->Count > 0)
    {
        label122->Text = "Please click on the contact you want to add to the favourites, then press the button (Add to favourites)";
    }
    else
    {
        label122->Text = "\\0";
    }
}

```

 **ref class** System::Windows::Forms::PaintEventArgs
 (The documentation cache is still being constructed. Please [Search Online](#))

Void Pn_all_paint()

```
{
```

This checks whether all listview-> items are empty or not, if empty make label19 display("Not found contacts, please add contacts for display"), If it is not empty, make label19 clear("Not found contacts, please add contacts for display")

```
}
```

ALL CONTACTS

Add to favourites

Num	Name	Phone Number	E-mail	Job
Not found contacts,please add contacts for display				


```
private: System::Void pn_delete_Paint(System::Object^ sender, System::Windows::Forms::PaintEventArgs^ e)
{
    if (Delete_listView->Items->Count > 0)
    {
        label18->Text = "\\0";
        deletelabel->Text = "Please click on the contact you want to delete and then click on the (Delete) button";
    }
    else
    {
        deletelabel->Text = "\\0";
        label18->Text = "Not found contacts to delete";
    }
}
```

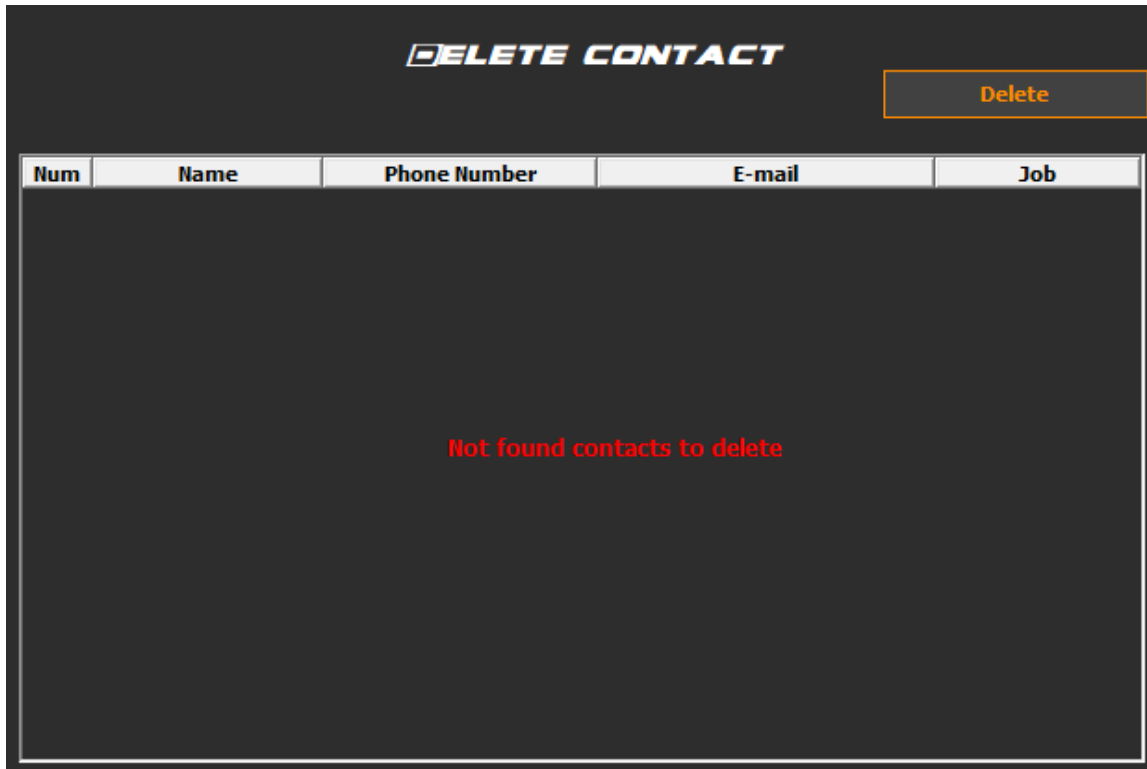
Void pn_delete_paint()

{

This check delete_listview if empty or no if empty display("Not Found contacts to delete").

If not empty display("Please click on the contact you want to delete and then click on the (Delete) button")

}



Delete_listview is empty

DELETE CONTACT

Delete

Please click on the contact you want to delete and then click on the (Delete) button

Num	Name	Phone Number	E-mail	Job
1	khaled abubakr	12345678910	khaledabobakr15@gmail.com	student

Delete_listview not empty

```
private: System::Void pn_fav_Paint_1(System::Object^ sender, System::Windows::Forms::PaintEventArgs^ e)
{
    if (FVlistview->Items->Count > 0)
    {
        label21->Text = "\\0";
        RemoveLb->Text = "Please click on the contact you want to remove, then press the button (Remove from favorites)";
    }
    else
    {
        RemoveLb->Text = "\\0";
        label21->Text = "Not found contacts to remove";
    }
}
```

Void pn_fav_paint_1()

{

This check if FVlistview empty or no ,if empty display in label21 ("Not found contacts to remove")

If not empty label21 ("Please click on the contact you want to remove, then press the button (Remove from favorites))

}



FVlistview not empty

FAVOURITES CONTACT				
Remove from favourites				
Num	Name	Phone Number	E-mail	Job
Not found contacts to remove				

FVlistview is empty

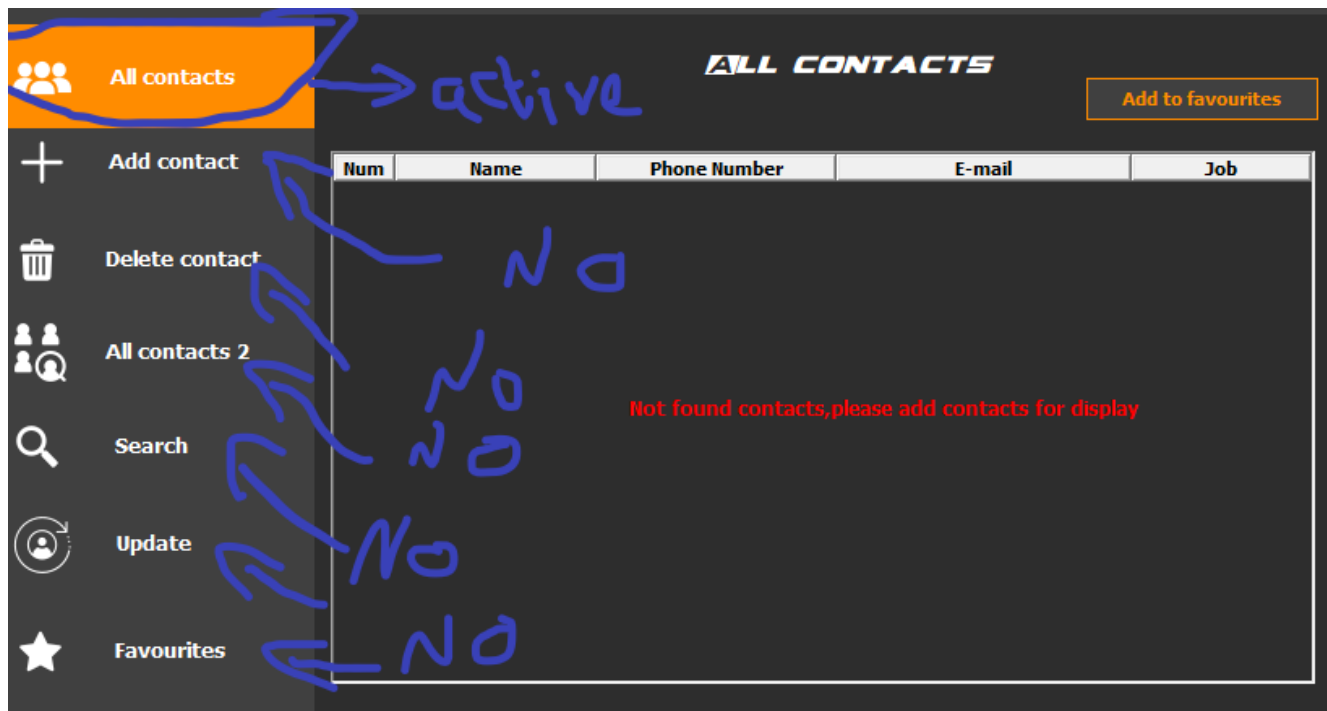
```
private: System::Void button7_Click(System::Object^ sender, System::EventArgs^ e) //add contact button.
{
    this->button7->BackColor = System::Drawing::Color::DarkOrange;
    this->button8->BackColor = System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_cast<System::Byte>(64)), static_cast<System::Int32>(static_cast<System::Byte>(64)),
        static_cast<System::Int32>(static_cast<System::Byte>(64)));
    this->button9->BackColor = System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_cast<System::Byte>(64)), static_cast<System::Int32>(static_cast<System::Byte>(64)),
        static_cast<System::Int32>(static_cast<System::Byte>(64)));
    this->button10->BackColor = System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_cast<System::Byte>(64)), static_cast<System::Int32>(static_cast<System::Byte>(64)),
        static_cast<System::Int32>(static_cast<System::Byte>(64)));
    this->button4->BackColor = System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_cast<System::Byte>(64)), static_cast<System::Int32>(static_cast<System::Byte>(64)),
        static_cast<System::Int32>(static_cast<System::Byte>(64)));
    this->button5->BackColor = System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_cast<System::Byte>(64)), static_cast<System::Int32>(static_cast<System::Byte>(64)),
        static_cast<System::Int32>(static_cast<System::Byte>(64)));
    pn_add->BringToFront();
}
private: System::Void button8_Click(System::Object^ sender, System::EventArgs^ e) { ... }
private: System::Void button5_Click(System::Object^ sender, System::EventArgs^ e) { ... }
private: System::Void button9_Click(System::Object^ sender, System::EventArgs^ e) { ... }
private: System::Void button10_Click(System::Object^ sender, System::EventArgs^ e) { ... }
private: System::Void button4_Click(System::Object^ sender, System::EventArgs^ e) { ... }
```

From button7_click to button4_click

{

Move from one panel to another panel with the color of the active button changing to orange

}



```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) //close button.  
{  
    Environment::Exit(0);  
}
```



Void button_Click()

{

This Button Close Program

}

```
private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e) //Minimized button.  
{  
    this->WindowState = FormWindowState::Minimized;  
}
```

Void button3_Click()



{

This Button Minimized

}


```
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e)//Maximized button
{
    if (this->WindowState == FormWindowState::Normal)
    {
        this->WindowState = FormWindowState::Maximized;
    }
    else
    {
        this->WindowState = FormWindowState::Normal;
    }
}
```

Void Button2_click()

{



This Button Maximized.

}

```
private: System::Void button13_Click(System::Object^ sender, System::EventArgs^ e)//erase button
{
    erase();
}
```

Void button13_Click()

{

This button use in erase all TextBoxs in(add_contacts_page)

}

ADD CONTACT

First name :

Last name :

Phone number :

Email :

Job :

Save

Erase

```

private: System::Void button12_Click(System::Object^ sender, System::EventArgs^ e)//save_button
{
    //-----check if textboxes empty or no-----//
    if (FNtextbox->Text != "" && LNtextbox->Text != "" && PNtextbox->Text != "" && Etextbox->Text != "" && Jtextbox->Text != "")
    {
        if (check_phonenum() == 0)
        {
            MessageBox::Show("Please enter an 11-digit phone number", "Error Message");
        }
        else if (check_phonenum() == 2)
        {
            MessageBox::Show("Please enter numbers only in the Phone number field", "Error Message");
        }
        else if (check_phonenum2() == 0)
        {
            MessageBox::Show("This phone number is already exists,please enter another phone number", "Error Message");
        }
        else if (check_email() == 1)
        {
            MessageBox::Show("The email must contain only one(@)sign and must contain(.com)at the end of the sentence. for example(example@exam
        }
        else
    }
}

```

Void button12_click()

{

This button is save button in (add contacts page)

}

ADD CONTACT

First name :

Last name :

Phone number :

Email :

Job :

Save

Erase

```

private: System::Void delete_btn_Click(System::Object^ sender, System::EventArgs^ e) {
    if (Delete_listView->SelectedItems->Count > 0)
    {
        if (MessageBox::Show("Are you sure you want to delete the contact?", "Confirmation Message", MessageBoxButtons::YesNo, MessageBoxIconInformation) == DialogResult::Yes)
        {
            //-----//
            String^ lb = Delete_listView->SelectedItems[0]->SubItems[2]->Text;
            std::string x = msclr::interop::marshal_as< std::string >(lb); //convert from String System to string std.
            //-----//
            std::ifstream file1, file2, file3, file4, file5, file0, file15;
            file1.open("data/FirstName.txt");
            file2.open("data/LastName.txt");
            file3.open("data/PhoneNum.txt");
            file4.open("data/Email.txt");
            file5.open("data/Job.txt");
            file0.open("data/Favourite.txt");
            file15.open("data/username.txt");
            //-----//
            std::string* FN = new std::string[read_count()];
            std::string* LN = new std::string[read_count()];
            std::string* PN = new std::string[read_count()];
            std::string* E = new std::string[read_count()];
            std::string* J = new std::string[read_count()];
            std::string* un = new std::string[read_count()];
            std::string* P = new std::string[read_count_Favourite()];
        }
    }
}

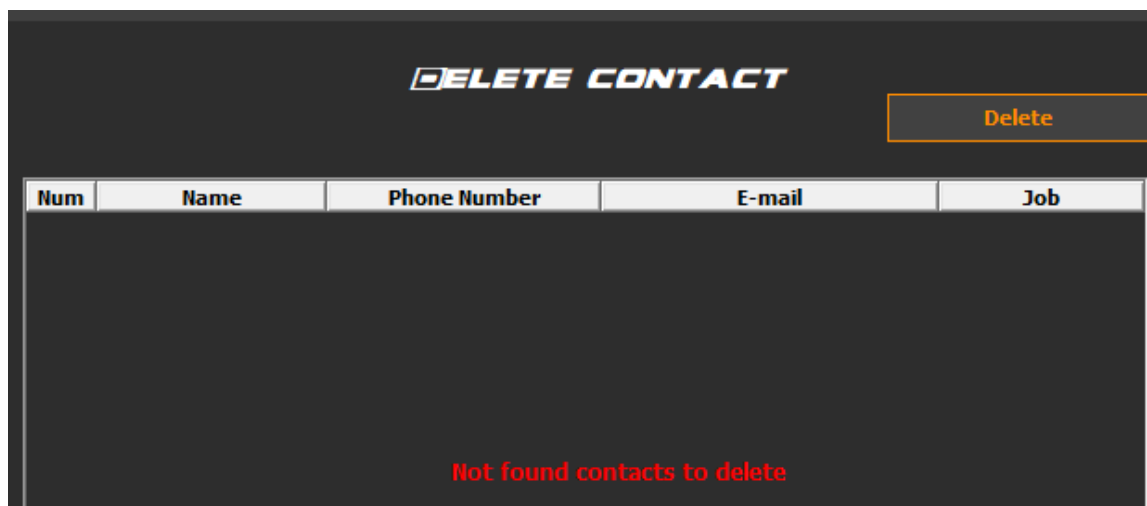
```

Void delete_btn_Click()

{

This button is delete button in (delete page)

}



```

private: System::Void button14_Click(System::Object^ sender, System::EventArgs^ e)//Edit button
{
    if (Update_listView->Items->Count == 0)
    {
        MessageBox::Show("There are no contacts to edit", "Error Message");
    }
    else
    {
        if (Update_listView->SelectedItems->Count > 0)
        {
            pn_edit->Show();
            String^ un = (Update_listView->SelectedItems[0]->SubItems[1]->Text);
            std::string UserName = msclr::interop::marshal_as<std::string>(un);
            int c = UserName.length();
            int c2 = UserName.find(" ");
            int c4 = UserName.find(" ") + 1;
            std::string h;
            std::string b;
            int d = 0;
            for (int i = c4; i < c; i++)
            {
                h = UserName.at(i);
                b = b.append(h);
            }
            std::string y = b;
            std::string x = UserName.erase(c2);

```

Void button14_click()

{

This button is edit button in (Update page)

}

UPDATE

Edit contact

Num	Name	Phone Number	E-mail	Job
Not found contacts to edit				

```

private: System::Void addfavouritebtn_Click(System::Object^ sender, System::EventArgs^ e) {
    if (All_listView->Items->Count == 0)
    {
        MessageBox::Show("There are no contacts to add to favourites", "Error Message");
    }
    else
    {
        if (All_listView->SelectedItem->Count == 0)
        {
            MessageBox::Show("Please click on the contact you want to add to favourites", "Error Message");
        }
        else
        {
            if (check_Phonenum_in_Favourite() == 1)
            {
                if (MessageBox::Show("Do you want to add this contact to favourites?", "Confirmation Message", MessageBoxButtons::YesNo, Messa
                {
                    StreamWriter file("data/Favourite.txt", true);
                    file.WriteLine(All_listView->SelectedItem[0]->SubItems[2]->Text);
                    count_FVcontacts(read_count_Favourite() + 1);
                    file.Close();
                    FVlistView->Items->Clear();
                    count_click_btn_favourite = 0;
                    MessageBox::Show("Successfully added to favourites", "Success Message");
                }
            }
        }
    }
}

```

Void addfavouritebtn_click()

{

This button is (add to favourite)in all contacts page

}

ALL CONTACTS

Add to favourites

Num	Name	Phone Number	E-mail	Job
Not found contacts,please add contacts for display				

```

private: System::Void Removebtn_Click(System::Object^ sender, System::EventArgs^ e) {
    if (FVListView->Items->Count == 0)
    {
        MessageBox::Show("There are no contacts to remove", "Error Message");
    }
    else
    {
        if (FVListView->SelectedItem->Count == 0)
        {
            MessageBox::Show("Please click on the contact you want to remove", "Error Message");
        }
        else
        {
            if (MessageBox::Show("Are you sure you removed a contact?", "Confirmation Message", MessageBoxButtons::YesNo, MessageBoxIcon::Question) == System::Window:
            {
                String^ lb = FVListView->SelectedItem->SubItems[2]->Text;
                std::string PHONE = mscclr::interop::marshal_as<std::string>(lb); //convert from String System to string std.
                std::string* PHONEnum = new std::string[read_count_Favourite()];
                std::ifstream file;
                std::ofstream file1;
                //-----//
                file.open("data/Favourite.txt");
                for (int i = 0; i < read_count_Favourite(); i++)
                {
                    getline(file, PHONEnum[i]);
                }
                file.close();
                //-----//
                for (int i = 0; i < read_count_Favourite(); i++)
                {

```

Void Removebtn_Click()

{

This button is Remove from favourite in favourite page

}

FAVOURITES CONTACT

Remove from favourites

Num	Name	Phone Number	E-mail	Job
Not found contacts to remove				

```

private: System::Void search1_Click(System::Object^ sender, System::EventArgs^ e) {
1   if (search1textbox->Text == "")
    {
        MessageBox::Show("Please type in the search field", "Error Message");
    }
2   else
    {
        if (count_click_search1_btn == 0)
        {
            liner_search1();
            search_pn2->Show();
            search1->Hide();
            searchdemo_btn->Show();
            if (searchlistviewdemo->Items->Count == 0)
            {
                label27->Text = "This contact is not found";
            }
            count_click_search1_btn++;
        }
        else if (count_click_search1_btn == 1)
        {
            search1textbox->Text = "";
        }
    }
}

```

void Search1_click()

{

This button is search button in search page.

}

