**Linked In**



**Submitted by:**

Hussain Iftikhar 2021-CS-126

**Supervised by:**

Mam Maida Shahid

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

Contents

* Short Description ----------------------------------------------------------- 3
* Project Features ------------------------------------------------------------- 3
* Cases Of Project ------------------------------------------------------------- 3
* Class Diagram ---------------------------------------------------------------- 4
* User Interface ----------------------------------------------------------------- 5
* Technology and Resources -------------------------------------------------- 10
* Full Code Of Project ---------------------------------------------------------- 10
* **Description Of Project:**

This is Desktop Application using C# .NET. This is the social networking application like Linked In. In this application you can perform different features like see connection add connection search connection chatting with them. Database used for this project is Microsoft SQL Database which store the request Sign up user, request and friend.

* **Project Features:**

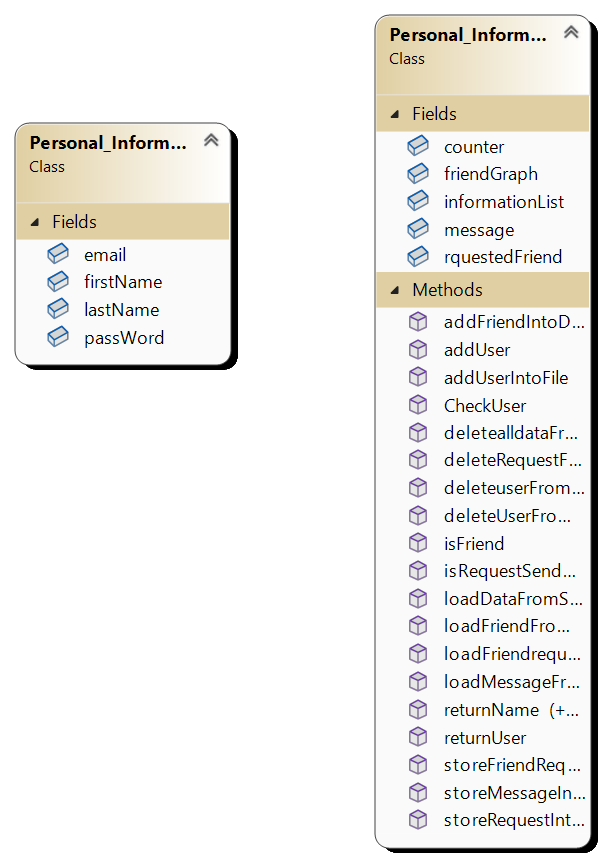
1. Add Connections
2. Search Connection
3. Send Request
4. Show All Connection
5. Chat With Friend
6. Delete Request
7. See Connection Of Connection
8. Profile Update
9. Mutual Connection

* **Cases Of Project:**

There is only one user in this application and user can perform different features.

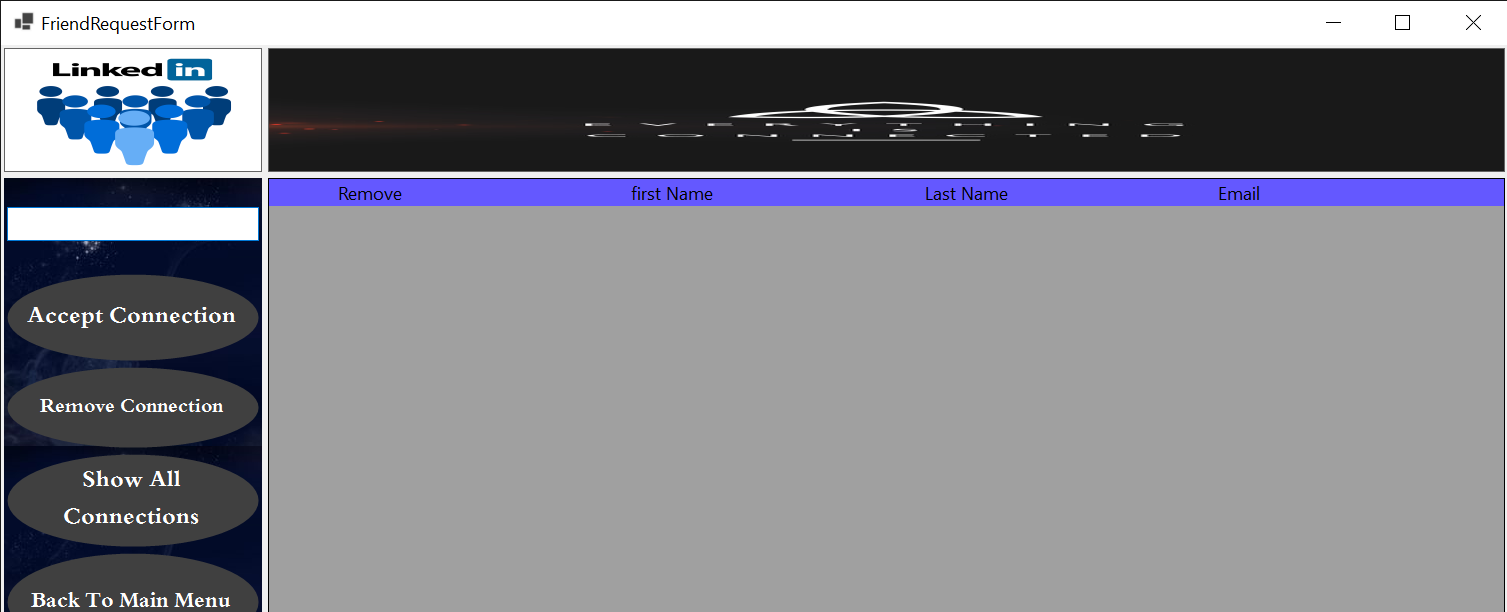
1. If user has no account it will first create and account with unique email and then perform the different features.
2. First user can update his personal information and cannot update his email.
3. Second User can search user by simply enter the name of the user and all the user with this name is appear. Copy the email of user with which you want to add the connection.
4. Enter the email of your friend for sending the request if you already send the request or you are already friend then you would not be able to send the request again for this user.
5. User can also check the request which he received from different user. By entering the email of user he/ she can delete or accept the request of this user and then the connection is added between these two users by using graph.
6. User can see his/her connection by simply click on show connection button.
7. User can also do chat with his/ her connection/friend by simply entering hi email id to start the chat between them.
8. User can also see the connection of connection of his/her friend by entering the unique email id.
9. User can also see the mutual connection between his/her friend by click on the mutual connection Button.

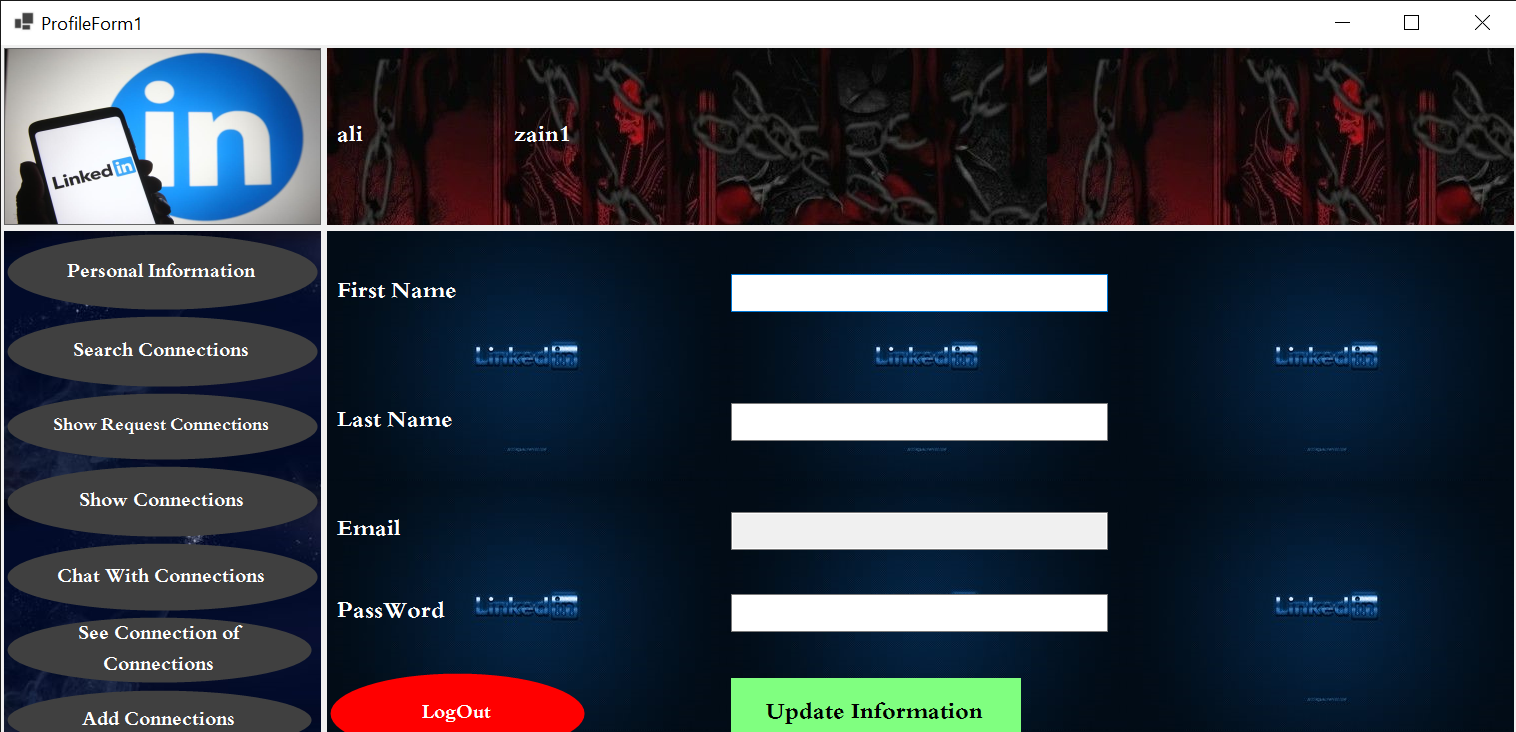
* **Class Diagram:**

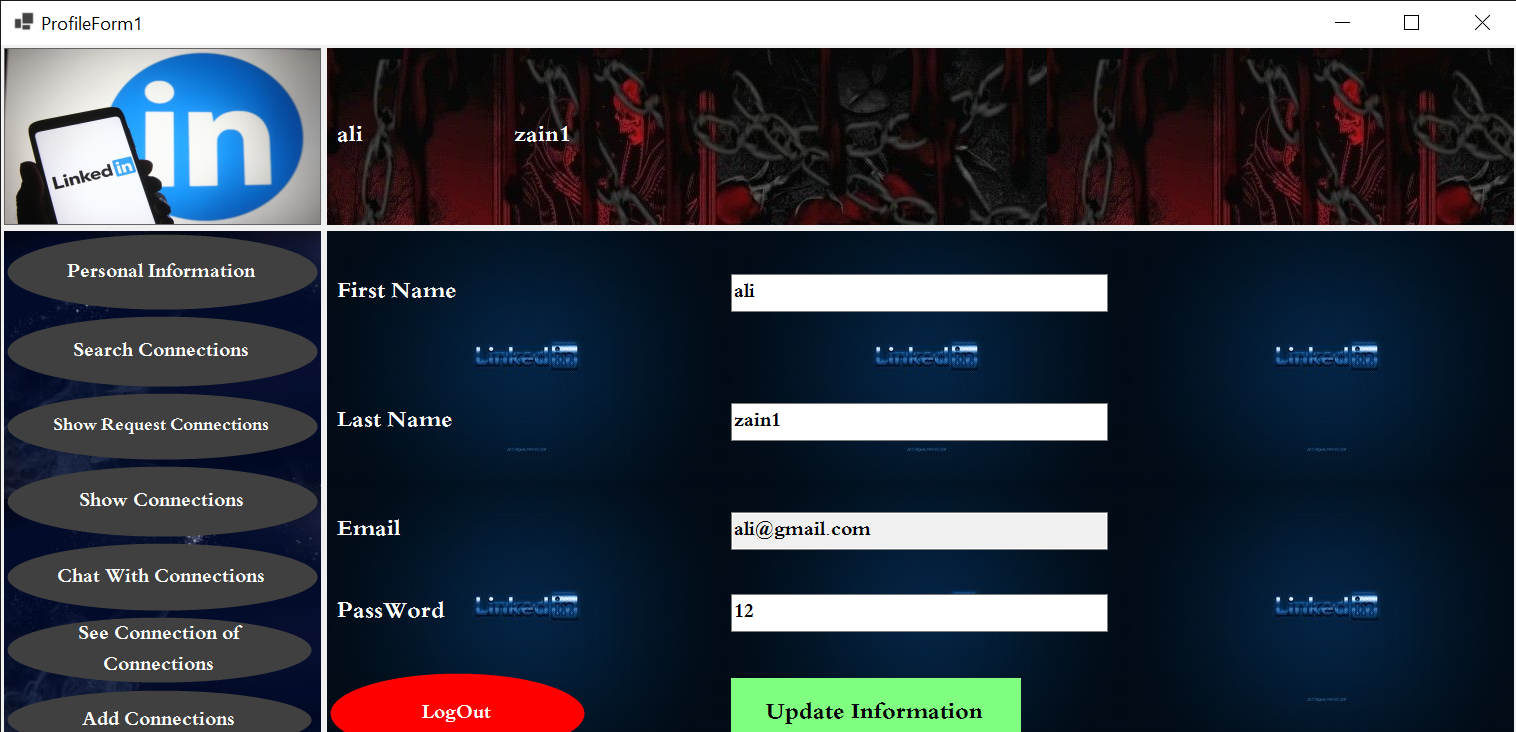


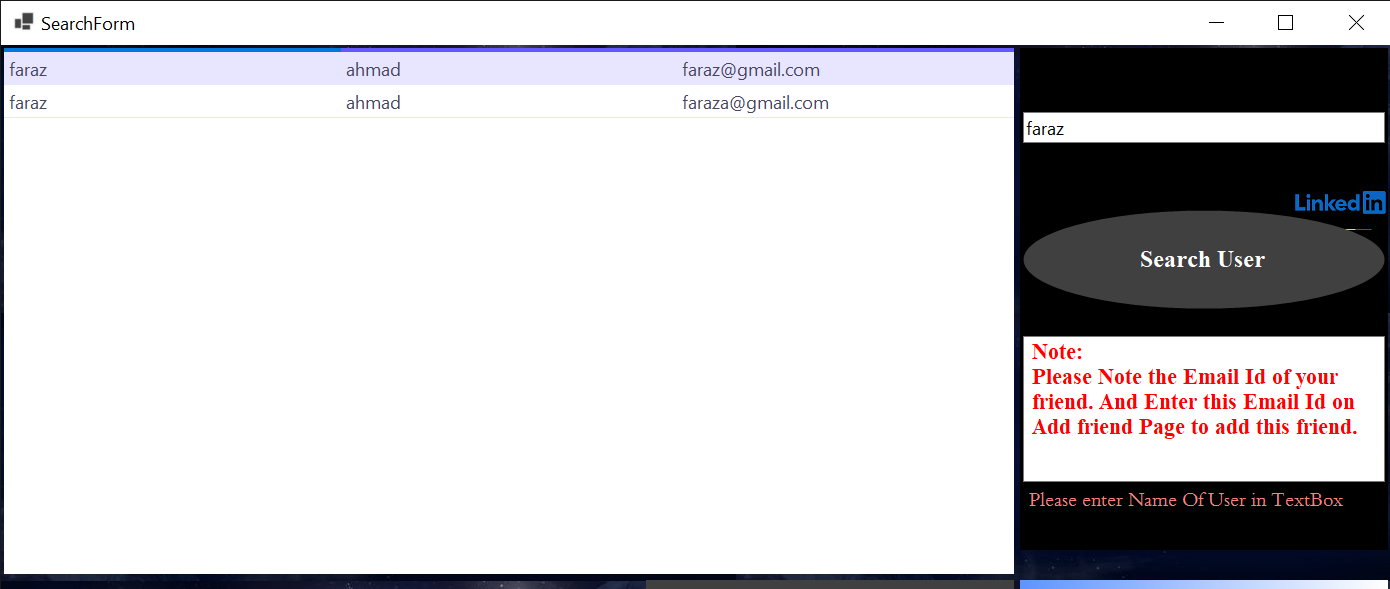
* **User Interface Design:**

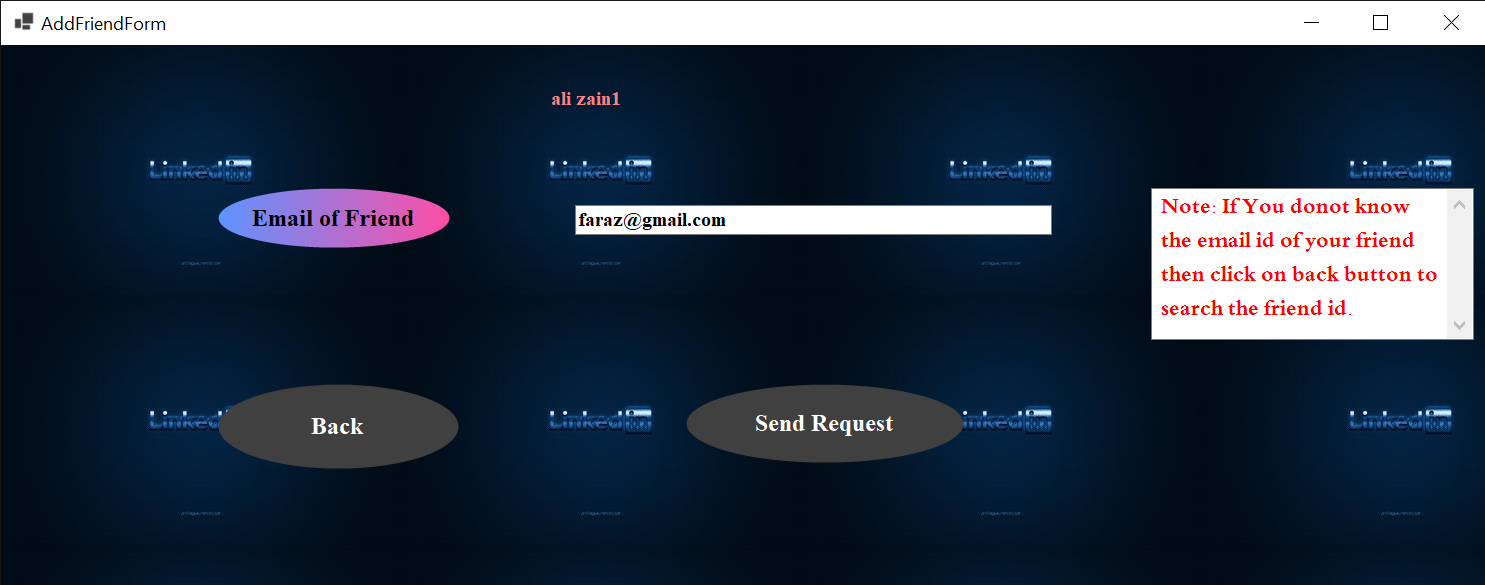


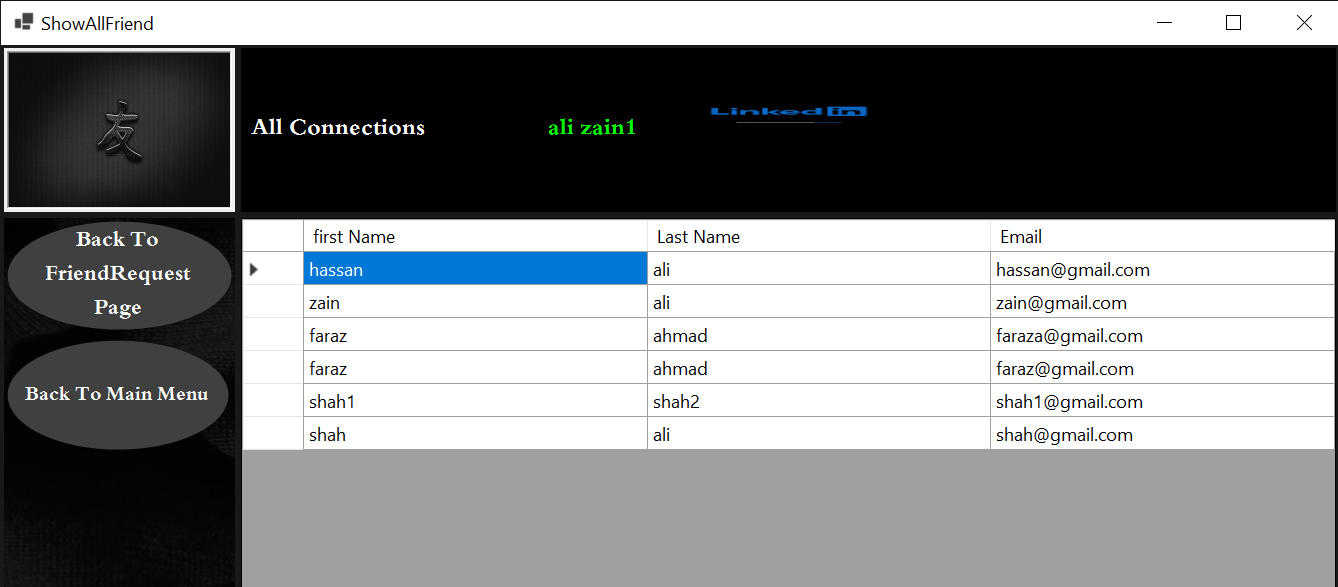


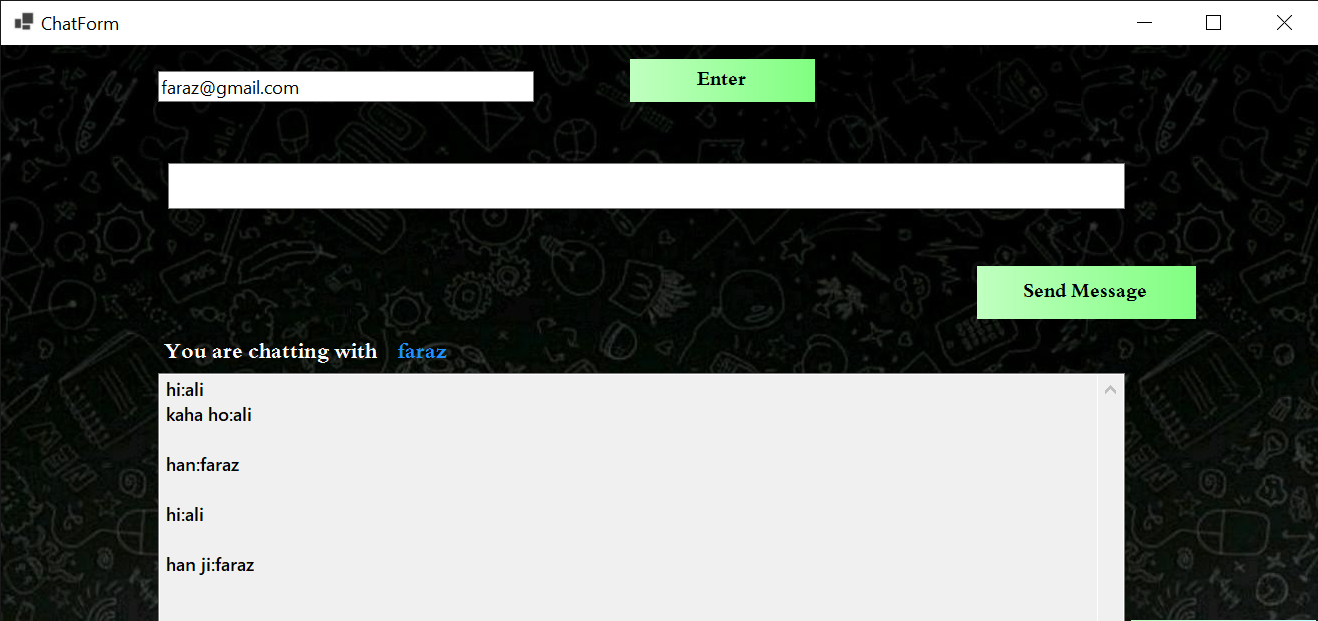


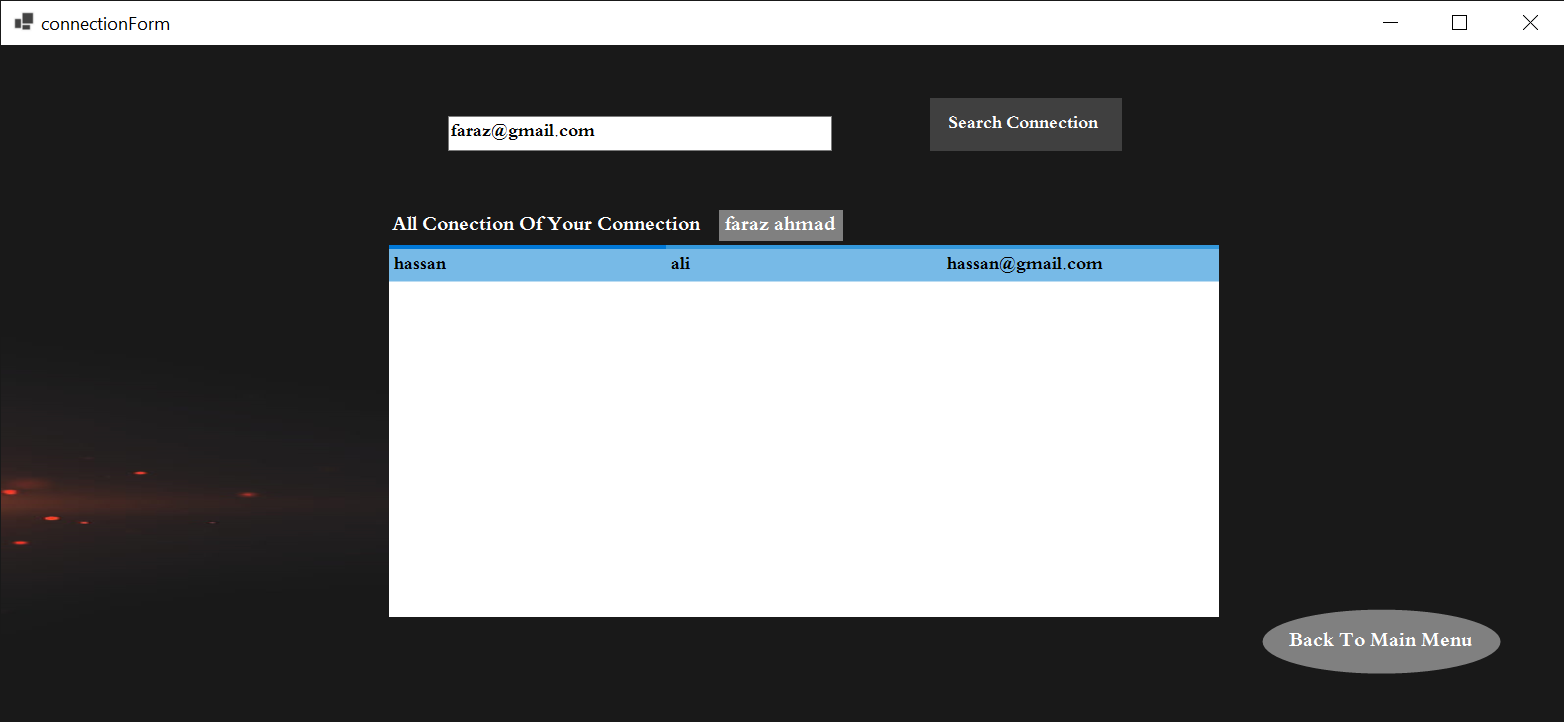


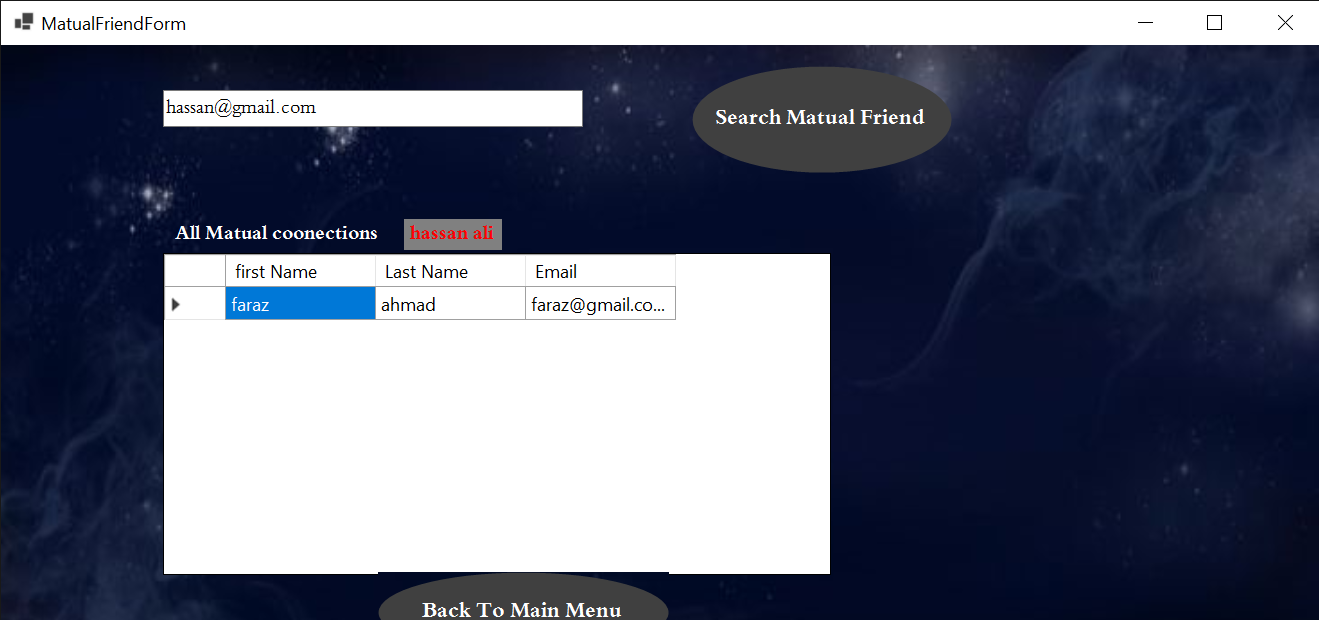


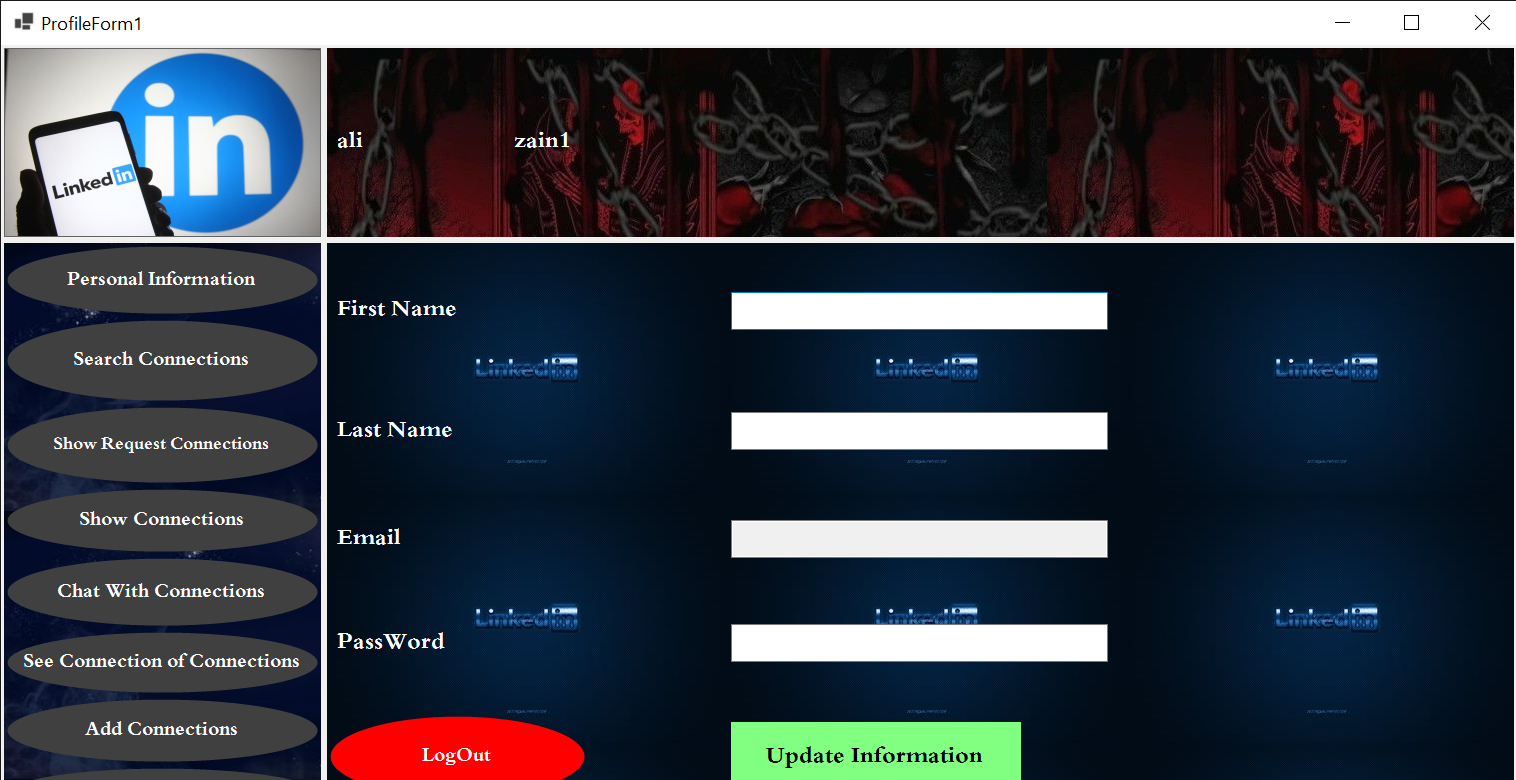












* **Technology And Resources:**

1. **For Frontend:**

I used the windows forms to develop the front end of the application guna is used for coloring features in form.

1. **For Backend:**

For backend developed is used the list and dictionary to store the sign up user and friend respectively. All the data (sign up user, requests and friends) is store in SQL database.

And for chatting purpose I used the text file to store the chat between two users.

* **Full Code Of Project:**

1. **Personal\_Information BL Class:**

public class Personal\_Information

{ public string email;

public string passWord;

public string firstName;

public string lastName;

}

1. **Personal\_Information DL:**

public class Personal\_InformationDL

{

public static string message = "";

public static Dictionary<Personal\_Information,List<Personal\_Information>> rquestedFriend=new Dictionary<Personal\_Information, List<Personal\_Information>>();

public static List<Personal\_Information> informationList=new List<Personal\_Information>();

public static Dictionary<Personal\_Information,List<Personal\_Information>> friendGraph=new Dictionary<Personal\_Information, List<Personal\_Information>>();

public static int counter=0;

public static void addUser(string firstName,string lastName,string email,string password)

{

Personal\_Information temp= new Personal\_Information();

temp.email = email;

temp.firstName=firstName;

temp.lastName=lastName;

temp.passWord = password;

informationList.Add(temp);

counter++;

}

public static void addUserIntoFile(string firstName, string lastName, string email,string passWord)

{

StreamWriter temp = new StreamWriter("personal.txt",true);

temp.WriteLine(firstName + "," + lastName + "," + email + "," + passWord);

temp.Flush();

temp.Close();

}

public static Personal\_Information returnName(string email,string password)

{

foreach (Personal\_Information temp in informationList)

{

if (temp.email == email && temp.passWord == password)

{

return temp;

}

}

return null;

}

public static void deletealldataFromDataBase()

{

SqlConnection con1 = new SqlConnection("Data Source=DESKTOP-D4TTB52;Initial Catalog=msdb;Integrated Security=True");

con1.Open();

SqlCommand command = new SqlCommand("DELETE FROM [FriendTable]", con1);

command.ExecuteNonQuery();

con1.Close();

MessageBox.Show("Request Accept SuccessFully And Deleted From DataBase");

}

public static void deleteuserFromdatabase()

{

SqlConnection con1 = new SqlConnection("Data Source=DESKTOP-D4TTB52;Initial Catalog=msdb;Integrated Security=True");

con1.Open();

SqlCommand command = new SqlCommand("DELETE FROM [Tabel]", con1);

command.ExecuteNonQuery();

con1.Close();

MessageBox.Show("Request Accept SuccessFully And Deleted From DataBase");

}

public static void loadDataFromSQL\_DataBase()

{

SqlConnection con = new SqlConnection("Data Source=DESKTOP-D4TTB52;Initial Catalog=msdb;Integrated Security=True");

con.Open();

SqlCommand command = new SqlCommand("select \* from Tabel", con);

SqlDataAdapter data = new SqlDataAdapter(command);

DataTable table = new DataTable();

data.Fill(table);

for (int i = 0; i < table.Rows.Count; i++)

{

DataRow row = table.Rows[i];

string name = row["First\_Name"].ToString();

string lastName = row["Last\_Name"].ToString();

string email = row["Email"].ToString();

email = email.Trim();

string passWord = row["Password"].ToString();

addUser(name, lastName, email, passWord);

}

}

public static void deleteUserFromRequestList(Personal\_Information user,Personal\_Information deleteduser)

{

foreach(KeyValuePair<Personal\_Information,List<Personal\_Information>> friend in rquestedFriend)

{

if(friend.Key.email == user.email)

{

for(int i=0;i<friend.Value.Count();i++)

{

if (friend.Value[i].email == deleteduser.email)

{

friend.Value.RemoveAt(i);

}

}

}

}

foreach (KeyValuePair<Personal\_Information, List<Personal\_Information>> friend in rquestedFriend)

{

if (friend.Key.email == deleteduser.email)

{

for (int i = 0; i < friend.Value.Count(); i++)

{

if (friend.Value[i].email == user.email)

{

friend.Value.RemoveAt(i);

}

}

}

}

}

public static void deleteRequestFromDataBase()

{

SqlConnection con1 = new SqlConnection("Data Source=DESKTOP-D4TTB52;Initial Catalog=msdb;Integrated Security=True");

con1.Open();

SqlCommand command = new SqlCommand("DELETE FROM [RequestTable]", con1);

command.ExecuteNonQuery();

con1.Close();

MessageBox.Show("Request Accept SuccessFully And Deleted From DataBase");

}

public static void storeRequestIntoDataBase()

{

foreach(KeyValuePair<Personal\_Information,List<Personal\_Information>> againrequest in rquestedFriend)

{

Personal\_Information t1 = new Personal\_Information();

t1 = againrequest.Key;

foreach(Personal\_Information var in againrequest.Value)

{

Personal\_Information t2 = new Personal\_Information();

t2 = var;

storeFriendRequestIntoDataBase(t1, t2);

}

}

}

public static bool isFriend(Personal\_Information check,Personal\_Information frind)

{

foreach(KeyValuePair<Personal\_Information , List<Personal\_Information>> t in friendGraph)

{

if(check.email == t.Key.email)

{

foreach(Personal\_Information var in t.Value)

{

if(var.email == frind.email)

{

return false;

}

}

}

}

return true;

}

public static bool isRequestSendAlready(Personal\_Information check, Personal\_Information frind)

{

foreach (KeyValuePair<Personal\_Information, List<Personal\_Information>> t in rquestedFriend)

{

if (check.email == t.Key.email)

{

foreach (Personal\_Information var in t.Value)

{

if (var.email == frind.email)

{

return false;

}

}

}

}

return true;

}

public static void storeFriendRequestIntoDataBase(Personal\_Information user,Personal\_Information request)

{

SqlConnection con1 = new SqlConnection("Data Source=DESKTOP-D4TTB52;Initial Catalog=msdb;Integrated Security=True");

con1.Open();

SqlCommand command = new SqlCommand("insert into [RequestTable] values (@kName,@kLastName,@kEmail,@kPassword,@vName,@vLastName,@vEmail,@vPassword) ", con1);

command.Parameters.AddWithValue("@kName", user.firstName);

command.Parameters.AddWithValue("@kLastName", user.lastName);

command.Parameters.AddWithValue("@kEmail", user.email);

command.Parameters.AddWithValue("@kPassword", user.passWord);

command.Parameters.AddWithValue("@vName", request.firstName);

command.Parameters.AddWithValue("@vLastName", request.lastName);

command.Parameters.AddWithValue("@vEmail", request.email);

command.Parameters.AddWithValue("@vPassword", request.passWord);

command.ExecuteNonQuery();

con1.Close();

}

public static bool CheckUser(string Name, string Password)

{

foreach (Personal\_Information temp in informationList)

{

if (temp.email == Name && temp.passWord == Password)

{

return true;

}

}

return false;

}

public static void addFriendIntoDataBase(Personal\_Information user1,Personal\_Information user2)

{

SqlConnection con1 = new SqlConnection("Data Source=DESKTOP-D4TTB52;Initial Catalog=msdb;Integrated Security=True");

con1.Open();

SqlCommand command = new SqlCommand("insert into [FriendTable] values (@FName,@FLastName,@FEmail,@FPassword,@RName,@RLastName,@REmail,@RPassword) ", con1);

command.Parameters.AddWithValue("@FName", user1.firstName);

command.Parameters.AddWithValue("@FLastName", user1.lastName);

command.Parameters.AddWithValue("@FEmail", user1.email);

command.Parameters.AddWithValue("@FPassword", user1.passWord);

command.Parameters.AddWithValue("@RName", user2.firstName);

command.Parameters.AddWithValue("@RLastName", user2.lastName);

command.Parameters.AddWithValue("@REmail", user2.email);

command.Parameters.AddWithValue("@RPassword", user2.passWord);

command.ExecuteNonQuery();

con1.Close();

}

public static void loadFriendFromDataBase()

{

SqlConnection con = new SqlConnection("Data Source=DESKTOP-D4TTB52;Initial Catalog=msdb;Integrated Security=True");

con.Open();

SqlCommand command = new SqlCommand("select \* from FriendTable", con);

SqlDataAdapter data = new SqlDataAdapter(command);

DataTable table = new DataTable();

data.Fill(table);

for (int i = 0; i < table.Rows.Count; i++)

{

DataRow row = table.Rows[i];

string fname = row["FName"].ToString();

string flastName = row["FLastName"].ToString();

string femail = row["FEmail"].ToString();

femail = femail.Trim();

string passWord = row["FPassword"].ToString();

string rname = row["RName"].ToString();

string rlastName = row["RLastName"].ToString();

string remail = row["REmail"].ToString();

remail = remail.Trim();

string rpassWord = row["RPassword"].ToString();

Personal\_Information temp=new Personal\_Information();

temp.firstName = fname;

temp.lastName = flastName;

temp.email=femail;

temp.passWord = passWord;

Personal\_Information temp1 = new Personal\_Information();

temp1.firstName = rname;

temp1.lastName = rlastName;

temp1.email = remail;

temp1.passWord = rpassWord;

if (!friendGraph.ContainsKey(temp))

friendGraph.Add(temp, new List<Personal\_Information>());

if (!friendGraph.ContainsKey(temp1))

friendGraph.Add(temp1, new List<Personal\_Information>());

friendGraph[temp].Add(temp1);

}

}

public static void loadFriendrequestFromDataBase()

{

SqlConnection con = new SqlConnection("Data Source=DESKTOP-D4TTB52;Initial Catalog=msdb;Integrated Security=True");

con.Open();

SqlCommand command = new SqlCommand("select \* from RequestTable", con);

SqlDataAdapter data = new SqlDataAdapter(command);

DataTable table1 = new DataTable();

data.Fill(table1);

for (int i = 0; i < table1.Rows.Count; i++)

{

DataRow row = table1.Rows[i];

string fname = row["kName"].ToString();

string flastName = row["kLastName"].ToString();

string femail = row["kEmail"].ToString();

femail = femail.Trim();

string passWord = row["kPassword"].ToString();

string rname = row["vName"].ToString();

string rlastName = row["vLastName"].ToString();

string remail = row["vEmail"].ToString();

remail = remail.Trim();

string rpassWord = row["vPassword"].ToString();

Personal\_Information temp = new Personal\_Information();

temp.firstName = fname;

temp.lastName = flastName;

temp.email = femail;

temp.passWord = passWord;

Personal\_Information temp1 = new Personal\_Information();

temp1.firstName = rname;

temp1.lastName = rlastName;

temp1.email = remail;

temp1.passWord = rpassWord;

if (!rquestedFriend.ContainsKey(temp))

rquestedFriend.Add(temp, new List<Personal\_Information>());

if (!rquestedFriend.ContainsKey(temp1))

rquestedFriend.Add(temp1, new List<Personal\_Information>());

rquestedFriend[temp1].Add(temp);

//rquestedFriend[temp].Add(temp1);

}

}

public static void storeMessageIntoFile(string path, string Message)

{

StreamWriter temp = new StreamWriter(path);

temp.WriteLine(Message);

temp.Close();

}

public static bool loadMessageFromFile(string path)

{

message = "";

if (File.Exists(path))

{

StreamReader fileVariable = new StreamReader(path);

string record;

while ((record = fileVariable.ReadLine()) != null)

{

message = message + record + Environment.NewLine;

}

fileVariable.Close();

return true;

}

return false;

}

public static string returnName(string email)

{

foreach (Personal\_Information temp in informationList)

{

if (temp.email == email )

{

return temp.firstName;

}

}

return null;

}

public static Personal\_Information returnUser(string email)

{

foreach (Personal\_Information temp in informationList)

{

if (temp.email == email)

{

return temp;

}

}

return null;

}

}

1. **Form 1 Code:**

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void signInButton\_Click(object sender, EventArgs e)

{

bool checker = Personal\_InformationDL.CheckUser(emailBox.Text, passwordBox.Text);

Personal\_Information temp = Personal\_InformationDL.returnName(emailBox.Text, passwordBox.Text);

if (checker == true)

{

ProfileForm1 profileform = new ProfileForm1(temp);

this.Hide();

profileform.Show();

}

else

{

MessageBox.Show("User Not Prsent");

}

}

private void SignUpButton\_Click(object sender, EventArgs e)

{

this.Hide();

JoinForm newForm = new JoinForm();

newForm.Show();

}

}

1. **Profile Form Code:**

public partial class ProfileForm1 : Form

{

Personal\_Information information = new Personal\_Information();

public ProfileForm1(Personal\_Information temp)

{

information = temp;

InitializeComponent();

firstName.Text = information.firstName;

lastName.Text = information.lastName;

}

private void personalButton\_Click(object sender, EventArgs e)

{

firstNameBox.Text = information.firstName;

lastNameBox.Text = information.lastName;

emailBox.Text = information.email;

passWordBox.Text = information.passWord;

}

private void updateButton\_Click(object sender, EventArgs e)

{

information.firstName = firstNameBox.Text;

information.lastName = lastNameBox.Text;

information.email = emailBox.Text;

information.passWord = passWordBox.Text;

SqlConnection con = new SqlConnection("Data Source=DESKTOP-D4TTB52;Initial Catalog=msdb;Integrated Security=True");

con.Open();

SqlCommand command = new SqlCommand("Update Tabel set First\_name=@First\_name,Last\_Name=@Last\_Name,Password=@Password where Email = @Email", con);

command.Parameters.AddWithValue("@Email", information.email);

command.Parameters.AddWithValue("@First\_name", firstNameBox.Text);

command.Parameters.AddWithValue("@Last\_Name", lastNameBox.Text);

command.Parameters.AddWithValue("@Password", passWordBox.Text);

command.ExecuteNonQuery();

con.Close();

MessageBox.Show("Data Updated SuccessFully");

}

private void searchButton\_Click(object sender, EventArgs e)

{

this.Close();

SearchForm newForm = new SearchForm(information);

newForm.Show();

}

private void requestButton\_Click(object sender, EventArgs e)

{

this.Hide();

FriendRequestForm friendForm = new FriendRequestForm(information.firstName, information.lastName, information.email, information.passWord);

friendForm.Show();

}

private void connectionsButton\_Click(object sender, EventArgs e)

{

this.Hide();

ShowAllFriend form = new ShowAllFriend(information);

form.Show();

}

private void chatButton\_Click(object sender, EventArgs e)

{

this.Hide();

ChatForm chat = new ChatForm(information);

chat.Show();

}

private void connectionButton\_Click(object sender, EventArgs e)

{

this.Hide();

connectionForm connection = new connectionForm(information);

connection.Show();

}

private void logoutButton\_Click\_1(object sender, EventArgs e)

{

this.Close();

Form1 form = new Form1();

form.Show();

}

private void addbutton\_Click(object sender, EventArgs e)

{

this.Close();

AddFriendForm form = new AddFriendForm(information);

form.Show();

}

}

1. **Search Form**

public partial class SearchForm : Form

{

Personal\_Information temp=new Personal\_Information();

public SearchForm(Personal\_Information request)

{

InitializeComponent();

temp = request;

}

private void searchButton\_Click(object sender, EventArgs e)

{

string name = searchBox.Text;

bool check = false;

DataTable table = new DataTable();

table.Columns.Add("first Name", typeof(string));

table.Columns.Add("Last Name", typeof(string));

table.Columns.Add("Email", typeof(string));

for(int i=0;i<Personal\_InformationDL.informationList.Count;i++)

{

if (name == Personal\_InformationDL.informationList[i].firstName)

{

table.Rows.Add(Personal\_InformationDL.informationList[i].firstName, Personal\_InformationDL.informationList[i].lastName, Personal\_InformationDL.informationList[i].email);

check = true;

}

}

dataGV.DataSource = table;

if(check == false)

{

MessageBox.Show("No user Found");

}

}

private void addButton\_Click(object sender, EventArgs e)

{

this.Close();

AddFriendForm frindForm = new AddFriendForm(temp);

frindForm.Show();

}

private void backButton\_Click(object sender, EventArgs e)

{

this.Close();

ProfileForm1 form = new ProfileForm1(temp);

form.Show();

}

1. **Friend Request Form:**

public partial class FriendRequestForm : Form

{

Personal\_Information temp=new Personal\_Information();

Personal\_Information temp2=new Personal\_Information();

DataTable table;

public FriendRequestForm(string name,string lastName,string email,string passWord)

{

InitializeComponent();

temp.firstName = name;

temp.lastName = lastName;

temp.email = email;

temp.passWord = passWord;

}

private void FriendRequestForm\_Load(object sender, EventArgs e)

{

table = new DataTable();

table.Columns.Add("first Name", typeof(string));

table.Columns.Add("Last Name", typeof(string));

table.Columns.Add("Email", typeof(string));

foreach(KeyValuePair<Personal\_Information,List<Personal\_Information>> user in Personal\_InformationDL.rquestedFriend)

{

if(user.Key.email == temp.email)

{

foreach(Personal\_Information val in user.Value)

{

table.Rows.Add(val.firstName, val.lastName, val.email);

}

}

}

dataGV.DataSource = table;

}

private void addButton\_Click(object sender, EventArgs e)

{

foreach(Personal\_Information user in Personal\_InformationDL.informationList)

{

if( user.email == emailBox.Text )

{

if (!Personal\_InformationDL.friendGraph.ContainsKey(temp))

Personal\_InformationDL.friendGraph.Add(temp, new List<Personal\_Information>());

if (Personal\_InformationDL.friendGraph.ContainsKey(temp))

{

if(Personal\_InformationDL.isFriend(temp,user) == true)

{

Personal\_InformationDL.friendGraph[temp].Add(user);

MessageBox.Show("Friend Added Successfully");

}

else

{

MessageBox.Show("There are already friend ");

}

}

if (!Personal\_InformationDL.friendGraph.ContainsKey(user))

Personal\_InformationDL.friendGraph.Add(user, new List<Personal\_Information>());

if (Personal\_InformationDL.friendGraph.ContainsKey(user))

{

if (Personal\_InformationDL.isFriend(user, temp) == true)

{

Personal\_InformationDL.friendGraph[user].Add(temp);

MessageBox.Show("Friend Added Successfully");

}

else

{

MessageBox.Show("There are already friend ");

}

}

Personal\_InformationDL.addFriendIntoDataBase(temp, user);

Personal\_InformationDL.addFriendIntoDataBase(user, temp);

Personal\_InformationDL.deleteUserFromRequestList(temp, user);

Personal\_InformationDL.deleteRequestFromDataBase();

Personal\_InformationDL.storeRequestIntoDataBase();

}

}

this.Close();

FriendRequestForm form = new FriendRequestForm(temp.firstName, temp.lastName, temp.email, temp.passWord);

form.Show();

}

private void showButton\_Click(object sender, EventArgs e)

{

this.Hide();

ShowAllFriend friend = new ShowAllFriend(temp);

friend.Show();

}

private void removeButton\_Click(object sender, EventArgs e)

{

foreach(Personal\_Information user in Personal\_InformationDL.informationList)

{

if(emailBox.Text == user.email)

{

Personal\_InformationDL.deleteUserFromRequestList(temp, user);

Personal\_InformationDL.deleteRequestFromDataBase();

Personal\_InformationDL.storeRequestIntoDataBase();

}

}

this.Hide();

FriendRequestForm again = new FriendRequestForm(temp.firstName,temp.lastName,temp.email,temp.passWord);

again.Show();

}

private void backButton\_Click(object sender, EventArgs e)

{

this.Close();

ProfileForm1 form = new ProfileForm1(temp);

form.Show();

}

}

1. **Connection Form:**

public partial class connectionForm : Form

{

Personal\_Information user1 = new Personal\_Information();

public connectionForm(Personal\_Information temp)

{

InitializeComponent();

user1 = temp;

}

private void searchConnection\_Click(object sender, EventArgs e)

{

string email = emailBox.Text;

Personal\_Information key = new Personal\_Information();

key=Personal\_InformationDL.returnUser(email);

DataTable table = new DataTable();

table.Columns.Add("first Name", typeof(string));

table.Columns.Add("Last Name", typeof(string));

table.Columns.Add("Email", typeof(string));

foreach (KeyValuePair<Personal\_Information, List<Personal\_Information>> user in Personal\_InformationDL.friendGraph)

{

if (user.Key.email == key.email)

{

namelabel.BackColor = Color.Gray;

namelabel.Text=user.Key.firstName+" "+user.Key.lastName;

for (int i = 0; i < user.Value.Count(); i++)

{

if (user.Value[i].email != user1.email)

table.Rows.Add(user.Value[i].firstName, user.Value[i].lastName, user.Value[i].email);

}

}

}

connectionGV.DataSource = table;

}

private void backButton\_Click(object sender, EventArgs e)

{

this.Close();

ProfileForm1 form = new ProfileForm1(user1);

form.Show();

}

}

1. **Chat Form:**

public partial class ChatForm : Form

{

string temp = "";

string path="";

string path1 = "";

Personal\_Information name1=new Personal\_Information();

public ChatForm(Personal\_Information name)

{

InitializeComponent();

name1 = name;

}

private void sendButton\_Click(object sender, EventArgs e)

{

string message = messageWriteBox.Text;

temp = temp + message + ":" + name1.firstName + Environment.NewLine;

messageShowBox.Text = temp;

Personal\_InformationDL.storeMessageIntoFile(path, temp);

Personal\_InformationDL.storeMessageIntoFile(path1, temp);

}

private void enterButton\_Click(object sender, EventArgs e)

{

string email = emailBox.Text;

string name=Personal\_InformationDL.returnName(email);

chatlabel1.Text = name;

path = name + name1.firstName + ".txt";

path1 = name1.firstName + name + ".txt";

if(Personal\_InformationDL.loadMessageFromFile(path) == true)

{

}

else

{

Personal\_InformationDL.loadMessageFromFile(path1);

}

temp = Personal\_InformationDL.message;

messageShowBox.Text = temp;

}

private void backButton\_Click(object sender, EventArgs e)

{

this.Close();

ProfileForm1 profile = new ProfileForm1(name1);

profile.Show();

}

}

1. **Add Friend Form:**

public partial class AddFriendForm : Form

{

Personal\_Information information = new Personal\_Information();

public AddFriendForm(Personal\_Information temp)

{

InitializeComponent();

information = temp;

nameLabel.Text = information.firstName+" "+information.lastName;

}

private void requestButton\_Click(object sender, EventArgs e)

{

foreach(Personal\_Information personal in Personal\_InformationDL.informationList)

{

if(personal.email == emailBox.Text)

{

if (!Personal\_InformationDL.rquestedFriend.ContainsKey(personal))

Personal\_InformationDL.rquestedFriend.Add(personal, new List<Personal\_Information>());

if(Personal\_InformationDL.isFriend(personal,information) == true || Personal\_InformationDL.isFriend(information, personal)==true)

{

if (Personal\_InformationDL.isRequestSendAlready(personal, information) == true)

{

Personal\_InformationDL.rquestedFriend[personal].Add(information);

Personal\_InformationDL.storeFriendRequestIntoDataBase(information, personal);

MessageBox.Show("Request Send Successfully ");

}

else

{

MessageBox.Show("You have already send request to this user so you are no longer available to send request again thanks");

}

}

else

{

MessageBox.Show("This user is already your friend you are no longer available to send request again thanks");

}

}

}

}

private void backButton\_Click(object sender, EventArgs e)

{

this.Close();

SearchForm form = new SearchForm(information);

form.Show();

}

}

1. **Join Form:**

public partial class JoinForm : Form

{

public JoinForm()

{

InitializeComponent();

MessageBox.Show("Please Enter a unique Email TO join Up OtherWise you will not be able to join");

}

private void guna2GradientCircleButton3\_Click(object sender, EventArgs e)

{

string firstName=FirstNameBox.Text;

firstName=firstName.Trim();

string lastName=lastNameBox.Text;

lastName=lastName.Trim();

string email=emailBox.Text;

email = email.Trim();

string passWord=passWordBox.Text;

passWord=passWord.Trim();

SqlConnection con = new SqlConnection("Data Source=DESKTOP-D4TTB52;Initial Catalog=msdb;Integrated Security=True");

con.Open();

SqlCommand command = new SqlCommand("insert into Tabel values (@First\_name,@Last\_Name,@Email,@Password) ", con);

command.Parameters.AddWithValue("@First\_Name", FirstNameBox.Text);

command.Parameters.AddWithValue("@Last\_Name", lastNameBox.Text);

try

{

command.Parameters.AddWithValue("@Email", emailBox.Text);

email = emailBox.Text;

}

catch

{

MessageBox.Show("PLease Enter a Unique Email");

}

command.Parameters.AddWithValue("@Password", passWordBox.Text);

command.ExecuteNonQuery();

con.Close();

Personal\_InformationDL.addUser(firstName, lastName, email, passWord);

MessageBox.Show("User Added SuccessFully");

this.Close();

Form1 exit = new Form1();

exit.Show();

}

private void showButton\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection("Data Source=DESKTOP-D4TTB52;Initial Catalog=msdb;Integrated Security=True");

con.Open();

SqlCommand command = new SqlCommand("select \* from Tabel", con);

SqlDataAdapter data=new SqlDataAdapter(command);

DataTable table= new DataTable();

data.Fill(table);

UserDG.DataSource= table;

for(int i=0;i<table.Rows.Count;i++)

{

DataRow row = table.Rows[i];

string name = row["First\_Name"].ToString();

string lastName = row["Last\_Name"].ToString();

string email = row["Email"].ToString();

string passWord = row["Password"].ToString();

Personal\_InformationDL.addUser(name,lastName, email, passWord);

}

}

}

1. **Show All Friend Form:**

public partial class ShowAllFriend : Form

{

Personal\_Information friend = new Personal\_Information();

public ShowAllFriend(Personal\_Information temp)

{

InitializeComponent();

friend = temp;

namelabel.Text = friend.firstName+" " + friend.lastName;

}

private void ShowAllFriend\_Load(object sender, EventArgs e)

{

DataTable table = new DataTable();

table.Columns.Add("first Name", typeof(string));

table.Columns.Add("Last Name", typeof(string));

table.Columns.Add("Email", typeof(string));

for(int i=0;i<Personal\_InformationDL.friendGraph.Count();i++)

{

var item=Personal\_InformationDL.friendGraph.ElementAt(i);

var itemkey=item.Key;

var itemvalue=item.Value;

if(itemkey.email == friend.email)

{

for(int x=0;x<itemvalue.Count;x++)

{

table.Rows.Add(itemvalue[x].firstName, itemvalue[x].lastName, itemvalue[x].email);

}

}

}

dataGV.DataSource = table;

}

private void BackButton\_Click\_1(object sender, EventArgs e)

{

this.Close();

FriendRequestForm form = new FriendRequestForm(friend.firstName, friend.lastName, friend.email, friend.passWord);

form.Show();

}

private void menuButton\_Click(object sender, EventArgs e)

{

this.Close();

ProfileForm1 form = new ProfileForm1(friend);

form.Show();

}

}

1. **Mutual FriendForm:**

Personal\_Information information = new Personal\_Information();

public MatualFriendForm(Personal\_Information temp)

{

InitializeComponent();

information = temp;

}

private void backButton\_Click(object sender, EventArgs e)

{

this.Close();

ProfileForm1 form = new ProfileForm1(information);

form.Show();

}

private void guna2CircleButton1\_Click(object sender, EventArgs e)

{

string email = emailBox.Text;

Personal\_Information key = new Personal\_Information();

key = Personal\_InformationDL.returnUser(email);

DataTable table = new DataTable();

table.Columns.Add("first Name", typeof(string));

table.Columns.Add("Last Name", typeof(string));

table.Columns.Add("Email", typeof(string));

foreach (KeyValuePair<Personal\_Information, List<Personal\_Information>> user in Personal\_InformationDL.friendGraph)

{

if (user.Key.email == key.email)

{

nameLabel.BackColor = Color.Gray;

nameLabel.Text = user.Key.firstName + " " + user.Key.lastName;

for (int i = 0; i < user.Value.Count(); i++)

{

foreach(KeyValuePair<Personal\_Information,List<Personal\_Information>> matual in Personal\_InformationDL.friendGraph)

{

if(information.email == matual.Key.email)

{

for(int x=0;x<matual.Value.Count;x++)

{

if (matual.Value[x].email == user.Value[i].email)

{

table.Rows.Add(user.Value[i].firstName, user.Value[i].lastName, user.Value[i].email);

}

}

}

}

}

}

}

matualGV.DataSource = table;

}