LINKEDINE CLONE

Database Project

TEAM MEMEBERS: IMRAN YOUNAS
TAMIA WAHEED
Hammad Irfan

PROJECT REPORT

The Projects Implements a LinkedIn App using Windows Form App (.NET Framework). In this Project we have basically implemented a network of Forms linked with one another. The Project include Signup For LinkedIn accounts, Log In accounts, Adding deleting Posts, Manage Comments, User Profile, Posting Job Advertisements, Applying for Jobs, Search Function and creating connections with Users.

Project Requirements:

- ERD/EERD
- ERD to Schema Mapping
- Login/Signup
- Manage Comments (Add, View)
- Manage Posts (Add, delete, view, update)
- Post job advertisements
- Apply for job advertisements
- Report of Job Advertisements Applications
- Profile Management
- Search other's Profile
- Send Connection Request
- Logout
- Submit assignments
- Manage own comments
- Overall flow of the application
- GUI Design

Software's used:

- Draw.io

  By using online ERD/EERD tool we have designed the Enhanced Entity-Relationship Diagram of LinkedIn App according to the requirements of Project.

- Microsoft Visual Studio

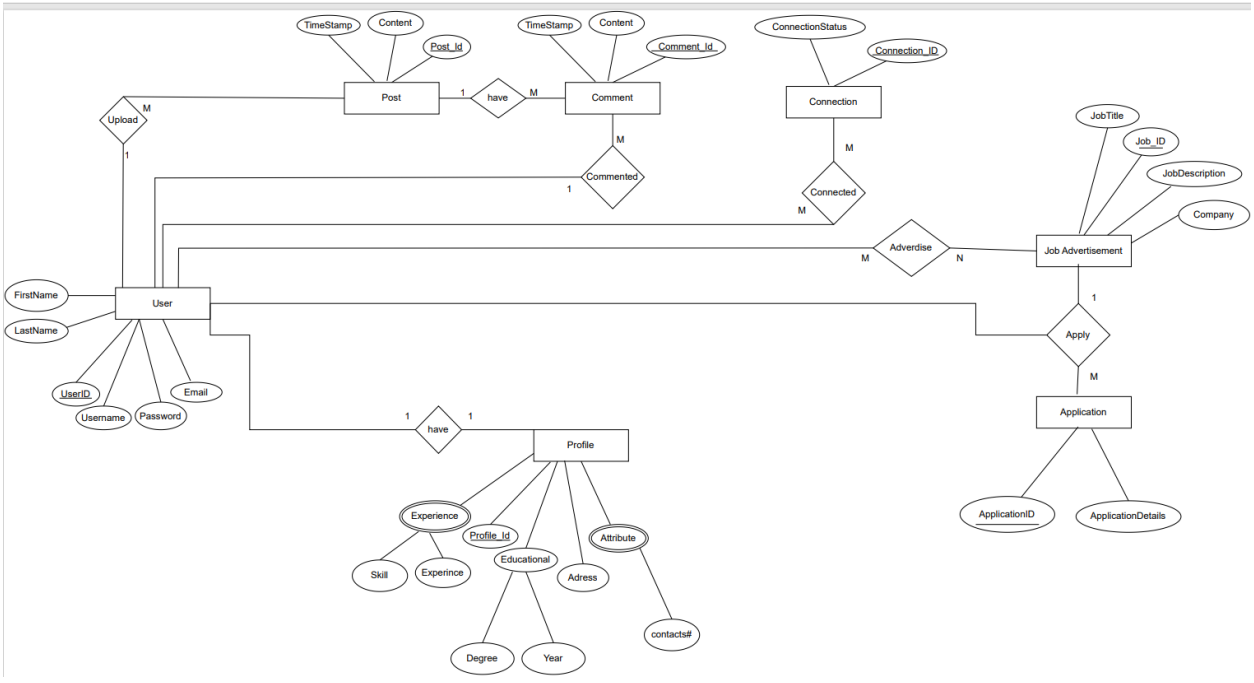
  Microsoft Visual Studio is used to generate the GUI of Project using Windows Form. C# .Net is used to build connections with database and SQL queries are used to implement the functionality of Windows Form.

- Microsoft SQL management Studio

  We created the Relational Database using SQL Server Management Studio. This includes Create Tables in Database, determine keys (Primary, Foreign) and adding constraints in Database.

The Project Contains Following Steps:

EERD Diagram:



Relational Schema:

```
CREATE TABLE Users (
       UserID INT IDENTITY(1,1) PRIMARY KEY ,
       FirstName VARCHAR (50) NOT NULL ,
       LastName VARCHAR (50) NOT NULL ,
       Email VARCHAR (80) UNIQUE NOT NULL ,
       Password VARCHAR (50) NOT NULL ,
       Status BIT DEFAULT (0),
       CHECK (DATALENGTH(password) >= 8));


CREATE TABLE Post (
       Post_id INT IDENTITY(1,1) PRIMARY KEY ,
       Post_Title VARCHAR(150),
       Post_Content NVARCHAR(MAX) NOT NULL,
       Timestamp DATETIME DEFAULT(GETDATE()),
       UserID INT NOT NULL,
       NoOfLikes INT DEFAULT(0) ,
       NoOfComments INT  DEFAULT (0) ,
       FOREIGN KEY (UserID) REFERENCES Users(UserID));


CREATE TABLE User_Profile (
       Profile_id INT IDENTITY (1,1) PRIMARY KEY,
    First_Name VARCHAR(25) NOT NULL,
       Last_Name VARCHAR(25) NOT NULL,
       UserID INT NOT NULL,
       User_Location VARCHAR(100),
       PhoneNo VARCHAR (11) NOT NULL,
       FOREIGN KEY (UserID) REFERENCES Users(UserID));


CREATE TABLE User_Education (
       Education_id INT PRIMARY KEY,
       Profile_id INT NOT NULL,
    Institute_Name VARCHAR(100) NOT NULL,
       Degree VARCHAR(70) NOT NULL,
       Starting_date DATE NOT NULL,
       Ending_date DATE,
       FOREIGN KEY (Profile_id) REFERENCES User_Profile(Profile_id));


CREATE TABLE User_ProfessionalSkills (
       Profession_id INT PRIMARY KEY,
       Profile_id INT NOT NULL,
    Company_Name VARCHAR(100) NOT NULL,
       JobPosition VARCHAR(70) NOT NULL,
       JobDescription VARCHAR(250) NOT NULL,
       Starting_date DATE NOT NULL,
       Ending_date DATE,
       FOREIGN KEY (Profile_id) REFERENCES User_Profile(Profile_id));
```

```sql
CREATE TABLE Likes (
	Likeid INT IDENTITY (1,1) PRIMARY KEY,
	Post_id INT NOT NULL,
	UserID INT NOT NULL,
	FOREIGN KEY (UserID) REFERENCES Users(UserID),
	FOREIGN KEY (Post_id) REFERENCES Post(Post_id));


CREATE TABLE Comments (
	Cmntid INT IDENTITY (1,1) PRIMARY KEY,
	Post_id INT NOT NULL,
	UserID INT NOT NULL,
	CmntTxt NVARCHAR(MAX) NOT NULL,
	timestamp DATE NOT NULL DEFAULT (GETDATE()),
	FOREIGN KEY (UserID) REFERENCES Users(UserID),
	FOREIGN KEY (Post_id) REFERENCES Post(Post_id));


CREATE TABLE JobAdds (
	JOBID INT PRIMARY KEY,
	UserID INT NOT NULL,
	Profile_id INT NOT NULL,
	JobPosition VARCHAR(75) NOT NULL,
	JobDescription VARCHAR(250) NOT NULL,
	JobRequirements VARCHAR(250) NOT NULL,
	Deadline DATE NOT NULL,
	FOREIGN KEY (UserID) REFERENCES Users(UserID),
	FOREIGN KEY (Profile_id) REFERENCES User_Profile(Profile_id));


CREATE TABLE JobApplication (
	App_id INT IDENTITY (1,1) PRIMARY KEY,
	JOBID INT NOT NULL,
	UserID INT NOT NULL,
	Profile_id INT NOT NULL,
	Resume_CV NVARCHAR(MAX) NOT NULL,
	CoverLetter NVARCHAR(MAX) NOT NULL,
	ApplyDate DATE NOT NULL,
	FOREIGN KEY (UserID) REFERENCES Users(UserID),
	FOREIGN KEY (Profile_id) REFERENCES User_Profile(Profile_id),
	FOREIGN KEY (JOBID) REFERENCES JobAdds(JOBID));


CREATE TABLE User_Connection (
	Conn_ID INT PRIMARY KEY,
	UserID INT NOT NULL,
	ConnUserID INT NOT NULL,
	FOREIGN KEY (UserID) REFERENCES Users(UserID),
	FOREIGN KEY (ConnUserID) REFERENCES Users(UserID));
```

The SQL server Database is connected to Microsoft Visual Studio. We generate a file for Database connections and add connection string in it. In every form we initialize a new connection using Database connections file.

Login:

Users Can Login their accounts if they already exists on database using email and Password.

Login page is connected with Users Table to help verify whether the account exists or not.

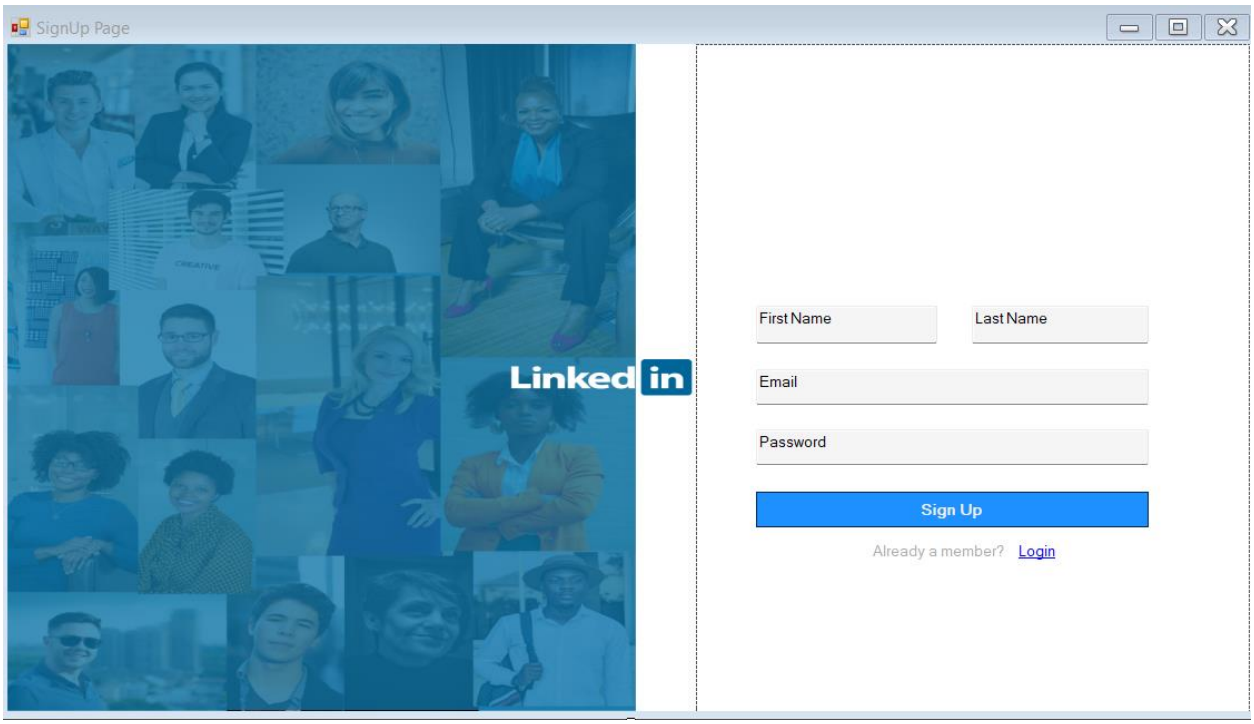
SQL Queries:

Select (To check the entered email and Password from the database)

We implemented a check if the entered email exists in database user Logged In if not generate error message.

Signup:

Users can enter new accounts using Signup Form. Signup form is also connected with Users table to help register Users. In Sign up form user inserts the required information to register the account.



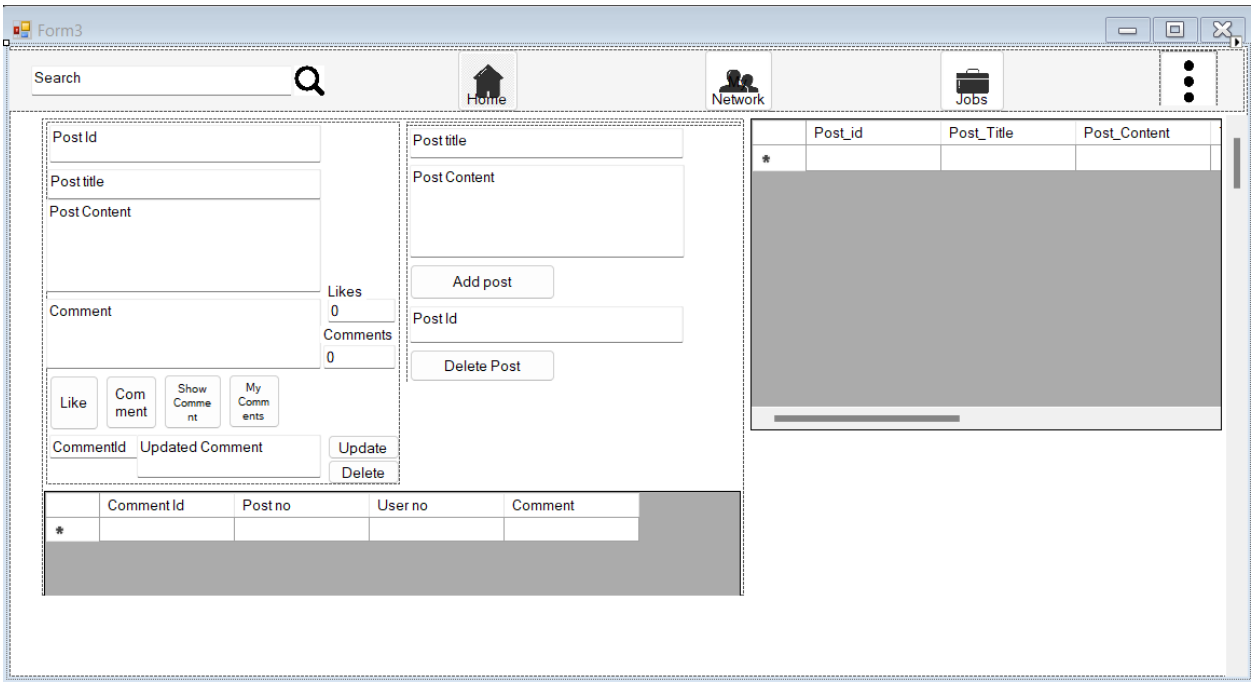
SQL Queries:

Insert (To insert the entered Firstname, LastName, email and Password in the database)

We implemented a check if the entered email exists in database user already exists if not create Account for user (Successfully Signed up).

Homepage:

Homepage is displayed after the user logged In the account. Homepage manage posts, Jobs, Networks, Search Function, Profile and Logout the account.



SQL Queries:

Select (To show posts on homepage using DataGridView , to show comments by pressing Show Comments, to show My comments by pressing My comments button. Select query retrieve data from tables in database)

Insert (To insert the Posts in the database in Posts Table, to insert Like in Likes Table by pressing Like Button, to insert Comments in Comment Table by using Comment button)

Update (To update comments in comment table in database)

Delete (To delete Posts in Post table in database, to delete comments in Comments Table in database)

Like:

User can like the Post displayed on Homepage by clicking on ‘Like’ button. The tables Likes will store the user response along with its ID.

Comment:

User can comment on the Post displayed on Homepage by clicking on ‘Comment’ button. The tables Comments will store the user response along with its ID. The User can also see comments on the Post by clicking ‘Show Comments’ button on the post. It will display the comments on the Post. User can also see his own Comments by clicking on ‘My comments’ button. User can Update and Delete his own comment by entering comment ID and content. The buttons Update and Delete are available for this purpose.

Add Posts:

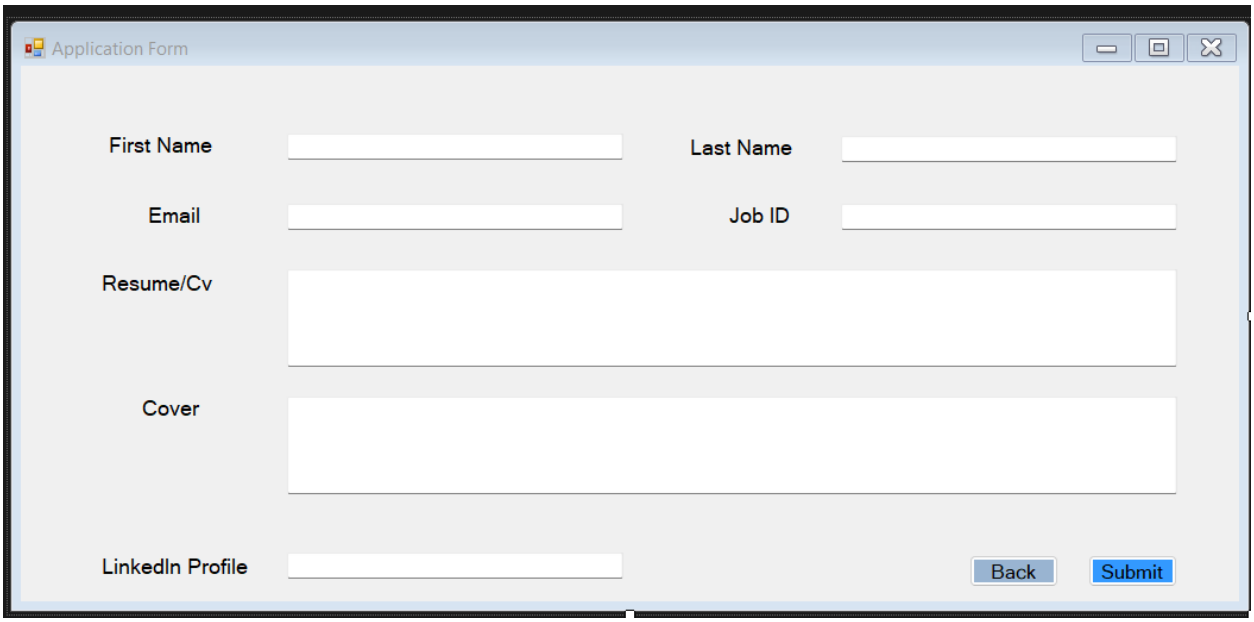
User can Add Posts in the Homepage and can also delete posts. The Posts entered in database are displayed on another Form.

Job Advertisements:

Job Advertisements Form display jobs that are in JobAdd Table . User can apply in these jobs by pressing Apply button that takes the user to Application Form.

Job Applications:

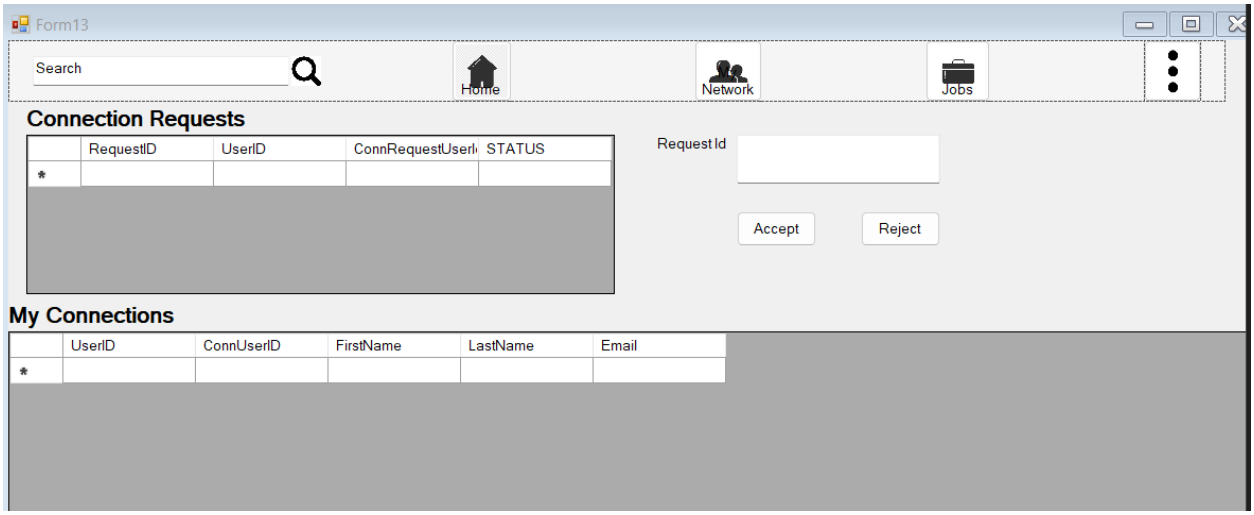
User submit applications of Job in Application Form against the JobId. Job Application form is used to Insert Data in the JobApplicantion Table in database.



SQL queries:

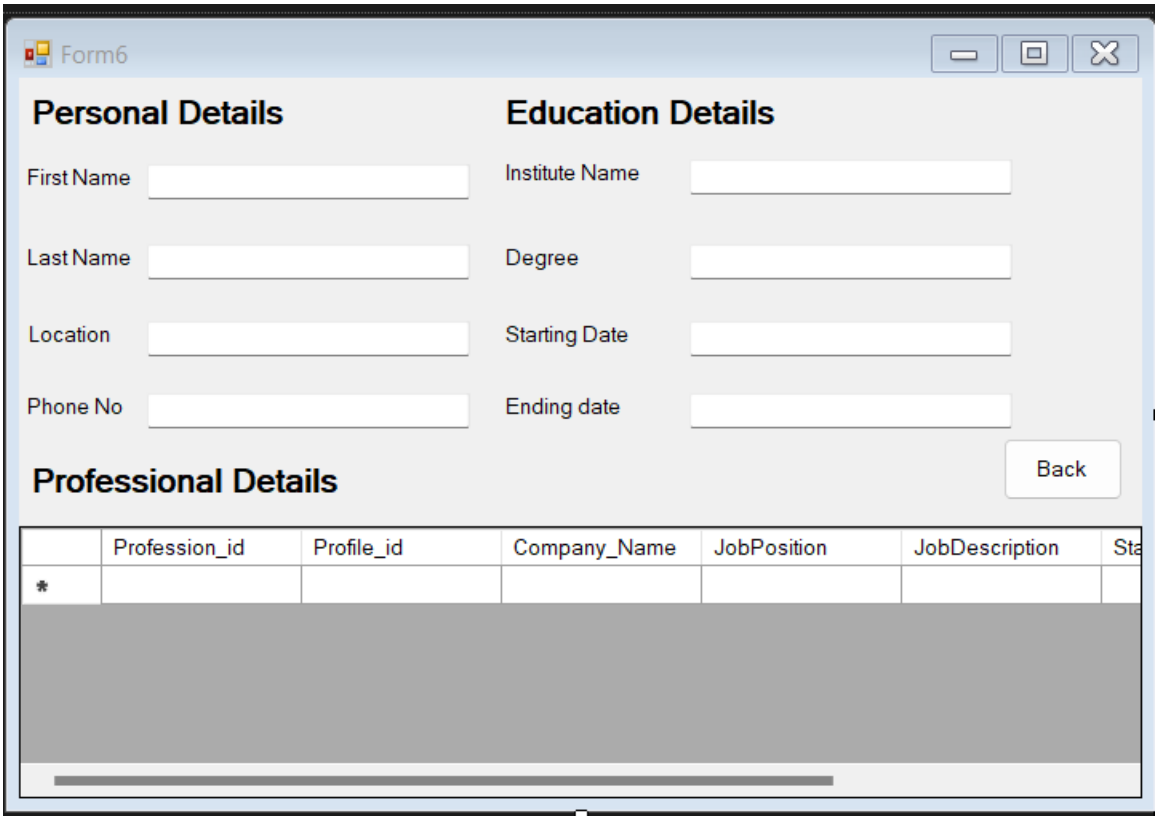
Insert (To Insert data in database Application Table)

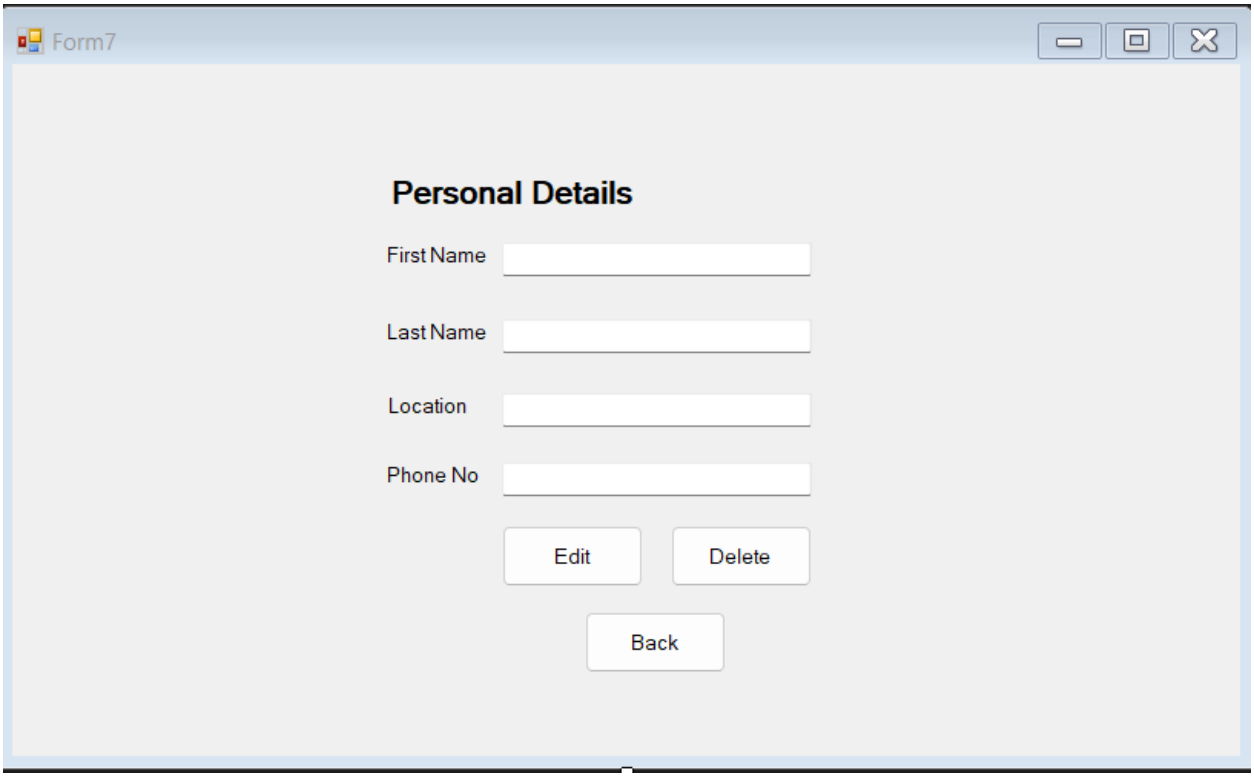
Network:



Profile:

Users can Create, View, Edit and Delete Profile. Profile displays the User Personal details. User fills the data in Tables User_Profile, User_Education and User_ProfessionalSkills by Inserting data in Forms.




Form7
Personal Details
First Name
Last Name
Location
Phone No
Edit
Delete
Back

END