

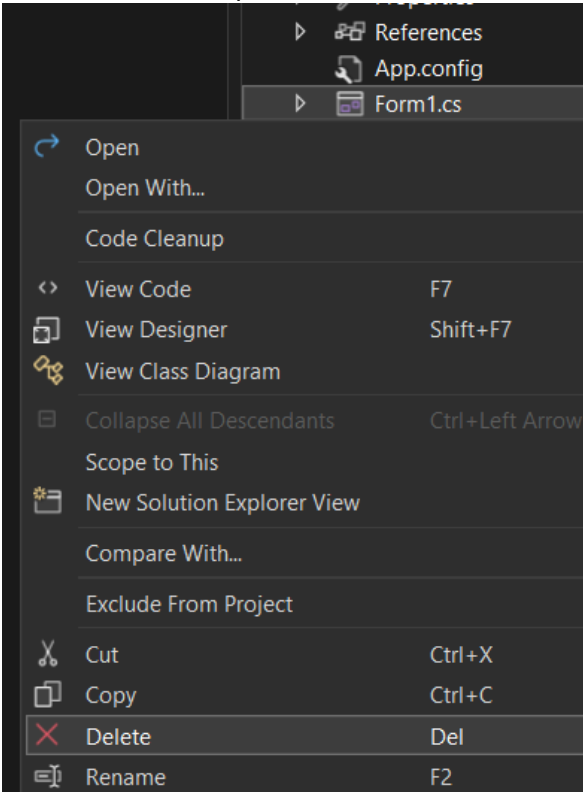
Directions: Follow the steps bellow to accomplish this task.

Reminders:

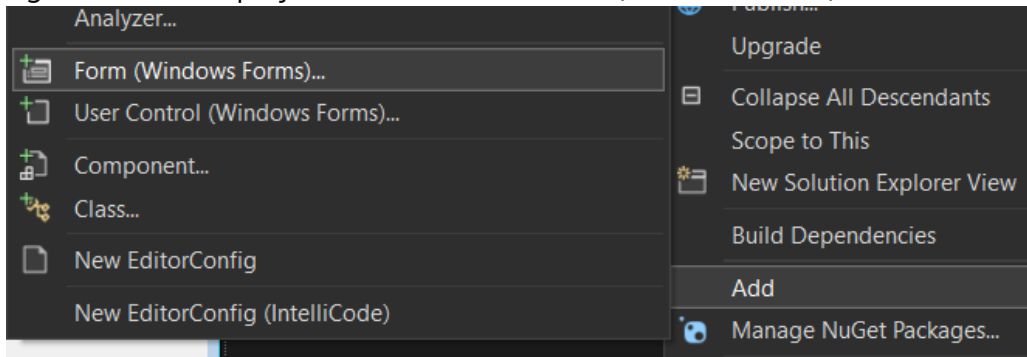
- 1. DO NOT SKIP ANY STEP GIVEN.
- 2. Read and analyze any errors encountered.
- 3. Try to resolve the errors first before asking for help.

PART 1 – Form Setup

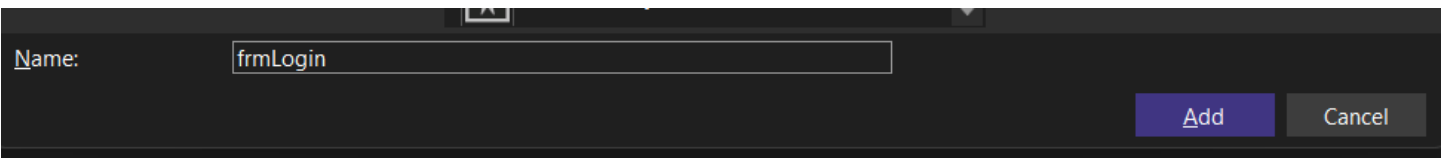
- 1. Create a new project and name it as Code_ACT3_Lname_Fname.
- 2. In the solutions explorer, delete Form1.



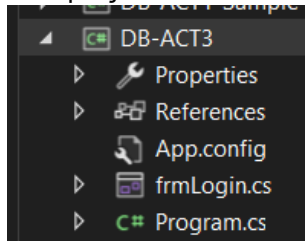
- 3. Right-click on the project and click Add > Form (Windows Forms)...



- 4. In the dialog box, type the name of the form as frmLogin and click Add.

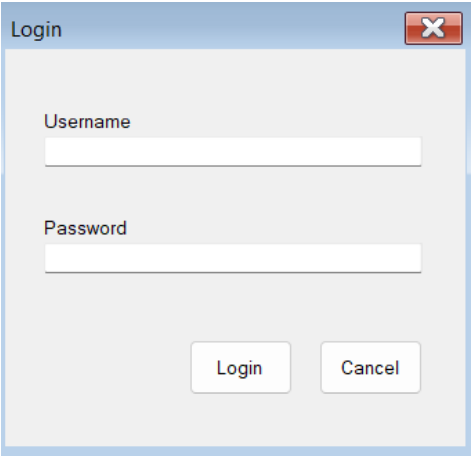


Your project files should look like the image below.

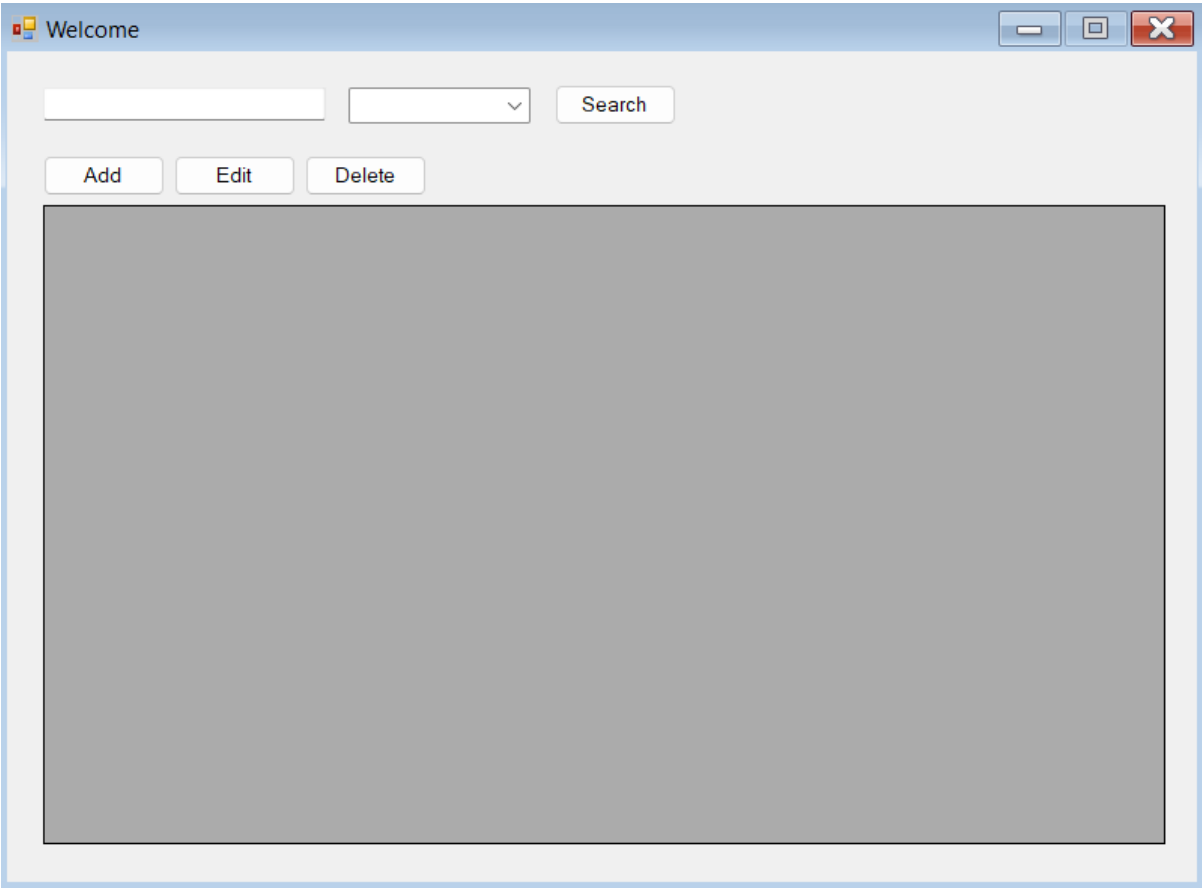


- 5. Change the following properties for the frmLogin.
 - a. StartPosition = CenterScreen
 - b. Text = Login
 - c. MaximizeBox = False
 - d. MinimizeBox = False
 - e. FormBorderStyle = FixedDialog

6. Add the controls shown in the figure below. Apply proper naming in each control. You may also apply design to your form.

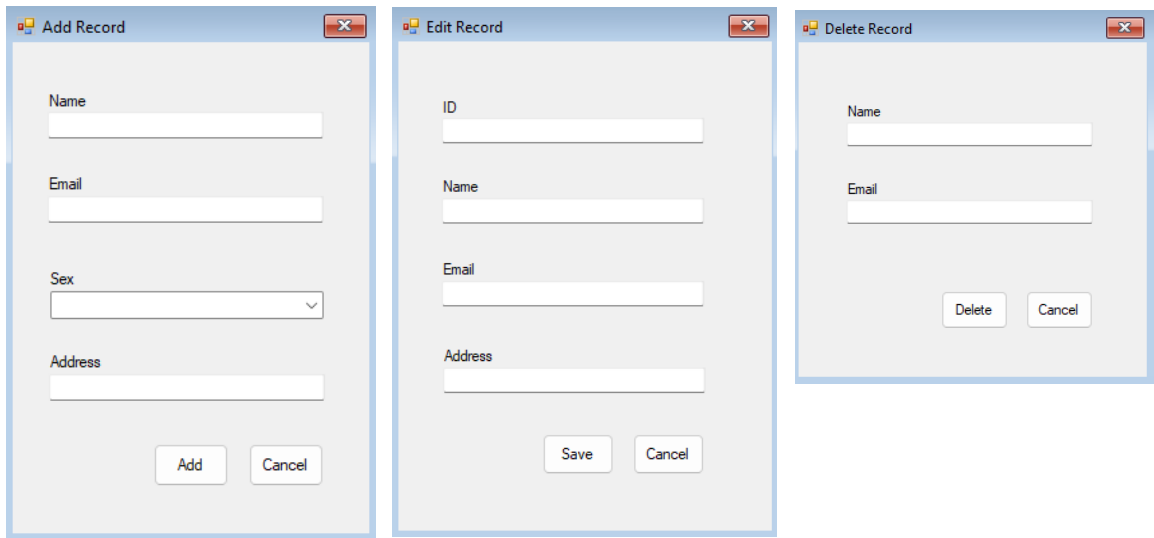


7. For the password textbox, look for the PasswordChar property and type *.
8. Add a new form and name it as frmMain.
9. Change the following form properties
- a. Text = Welcome
 - b. StartPosition = CenterScreen
 - c. MaximizeBox = False
 - d. FormBorderStyle = FixedDialog
10. Add the controls shown in the figure below. Apply proper naming in each control. You may also apply design to your form.



11. The combo box in the form should contain the following:
- a. All
 - b. Male
 - c. Female
12. Add 3 other forms – frmAdd, frmEdit, frmDelete.
13. Change the following form properties for each form
- a. Text = Add New Record(for frmAdd), Edit Record(for frmEdit), Delete Record(for frmDelete)
 - b. StartPosition = CenterScreen
 - c. MaximizeBox = False
 - d. FormBorderStyle = FixedDialog

14. Add the controls for each form as shown in the figure below. Apply proper naming in each control. You may also apply design to your form.



PART 2 – Coding

1. Open frmLogin and go to the text editor by pressing F7.
2. Prepare the following: (if you do not know the given items, you can always review the previous activities)
 - a. Connection string
 - b. System.Data.OleDb
 - c. OleDbConnection
3. Double-click the Login button and declare a string variable named query. Type the string query value of the variable as shown in the image below.

```
1 reference
private void btnLogin_Click(object sender, EventArgs e)
{
    string query = "select username, password from account where username=@uname and password=@pword";
}
```

4. Type the instance of the OleDbConnection and pass the connection string.

```
1 reference
private void btnLogin_Click(object sender, EventArgs e)
{
    string query = "select username, password from account where username=@uname and password=@pword";
    conn = new OleDbConnection(connStr);
}
```

5. Open the connection and create a new instance of the OleDbCommand and pass the arguments query and connection.

```
1 reference
private void btnLogin_Click(object sender, EventArgs e)
{
    string query = "select username, password from account where username=@uname and password=@pword";
    conn = new OleDbConnection(connStr);
    conn.Open();
    OleDbCommand cmd = new OleDbCommand(query, conn);
}
```

6. After the Open method, we need to use the OleDbCommand.Parameters.AddWithValue() method. This will be used to assign the values that corresponds to the required parameters in our query. In this case, the required parameters are @uname and @pword found in the WHERE clause. The values that correspond to each are txtUname.Text and txtPword.Text respectively.

```
1 reference
private void btnLogin_Click(object sender, EventArgs e)
{
    string query = "select username, password from account where username=@uname and password=@pword";
    conn = new OleDbConnection(connStr);
    conn.Open();
    OleDbCommand cmd = new OleDbCommand(query, conn);
    cmd.Parameters.AddWithValue("@uname", txtUname.Text);
    cmd.Parameters.AddWithValue("@pword", txtPword.Text);
}
```

7. Initiate the OleDbDataReader and assign the command ExecuteReader() method to run the query the we passed in the OleDbCommand.

```
1 reference
private void btnLogin_Click(object sender, EventArgs e)
{
    string query = "select username, password from account where username=@uname and password=@pword";
    conn = new OleDbConnection(connStr);
    conn.Open();
    OleDbCommand cmd = new OleDbCommand(query, conn);
    cmd.Parameters.AddWithValue("@uname", txtUname.Text);
    cmd.Parameters.AddWithValue("@pword", txtPword.Text);
    OleDbDataReader rdr = cmd.ExecuteReader();
}
```

8. Create an IF-Else statement.

```
OleDbDataReader rdr = cmd.ExecuteReader();

if()
{
}
else
{
}
```

9. In the IF condition, we can use the OleDbDataReader to check whether the query that was executed returned any result. For this, we'll use the OleDbDataReader.HasRows property.

```
if(rdr.HasRows)
{
}
else
{
}
```

10. If the condition returns TRUE, we need to read the result using the OleDbDataReader.Read() method. This method is optional. We just need it in this activity.

```
if(rdr.HasRows)
{
    rdr.Read();
}
```

11. Open frmMain and open the text editor. In the form main method, add a string parameter.

```
0 references
public frmMain(string username)
{
    InitializeComponent();
}
```

12. Initiate a global string variable.

```
string _username;

1 reference
public frmMain(string username)
{
    InitializeComponent();
}
```

13. Assign the value of the parameter to the global variable. This way, you'll be able to use the variable and its value in any method within the class.

```
string _username;

1 reference
public frmMain(string username)
{
    InitializeComponent();
    _username = username;
}
```

14. Go back to the design view in the same form and double click the form to create the Form_Load event.
15. In the event, type the code below. The *this* keyword pertains to the current form. In the case of the code below, it means that you are assigning "Welcome " + _username to the Text property of frmMain.

```
1 reference
private void frmMain_Load(object sender, EventArgs e)
{
    this.Text = "Welcome " + _username;
}
```

16. Go back to the IF statement in the frmLogin.
17. Create an instance of frmMain and below it, type the ShowDialog() method. Leave the error.

```
if(rdr.HasRows)
{
    rdr.Read();
    frmMain frm = new frmMain();
    frm.ShowDialog();
}
```

18. The error in step 17 occurred because we included a parameter in the frmMain. Therefore, we need to add argument with the same datatype as the parameter.

```
if(rdr.HasRows)
{
    rdr.Read();
    frmMain frm = new frmMain();
    frm.ShowDialog();
}
```

```
public frmMain(string username)
{
    InitializeComponent();
}
```

19. As a requirement in this activity, we'll pass the value *username* taken from our query. Using the OleDbDataReader.Read() method, we can retrieve the value by using the code below.

```
rdr["username"].ToString();
```

The "username" (including the quotation marks) pertains to the columns indicated in the select query. Take note that you cannot call a column unless it is indicated in your query or when using the asterisk (*) symbol.

```
1 reference
private void btnLogin_Click(object sender, EventArgs e)
{
    string query = "select username password from account where usernam
    conn = new OleDbConnection(connstr);
}
```

20. The code in step 19 can now be assigned in a string variable then and can be placed as the argument (A) or simply pass it as the argument (B).

A

```
if(rdr.HasRows)
{
    rdr.Read();
    string username = rdr["username"].ToString();
    frmMain frm = new frmMain(username);
    frm.ShowDialog();
}
```

B

```
if(rdr.HasRows)
{
    rdr.Read();
    frmMain frm = new frmMain(rdr["username"].ToString());
    frm.ShowDialog();
}
```

21. In the ELSE clause, simply use a message box as shown below.

```
else
{
    MessageBox.Show("Credentials are incorrect", "Credentials");
}
```

22. After the IF-ELSE statement, close the connection.

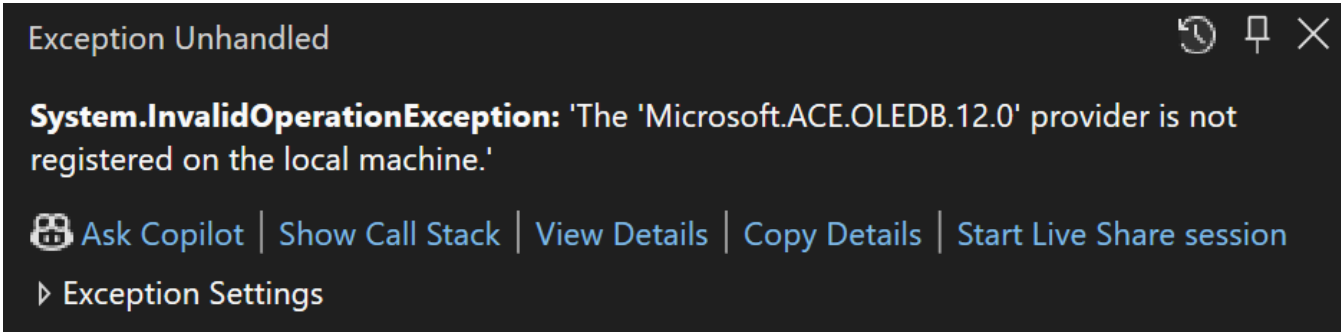
```
frm.ShowDialog();
}
else
{
    MessageBox.Show("Credentials are incorrect", "Credentials");
}
conn.Close();
```

23. Below is the code for frmLogin.

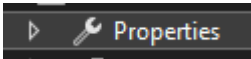
```
1 reference
private void btnLogin_Click(object sender, EventArgs e)
{
    string query = "select username, password from account where username=@uname and password=@pword";
    conn = new OleDbConnection(connStr);
    conn.Open();
    OleDbCommand cmd = new OleDbCommand(query, conn);
    cmd.Parameters.AddWithValue("@uname", txtUname.Text);
    cmd.Parameters.AddWithValue("@pword", txtPword.Text);
    OleDbDataReader rdr = cmd.ExecuteReader();

    if(rdr.HasRows)
    {
        rdr.Read();
        frmMain frm = new frmMain(rdr["username"].ToString());
        frm.ShowDialog();
    }
    else
    {
        MessageBox.Show("Credentials are incorrect", "Credentials");
    }
    conn.Close();
}
```

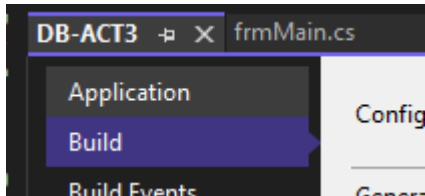
24. Try your program. If you encounter a problem similar to what is shown below, follow the steps given.



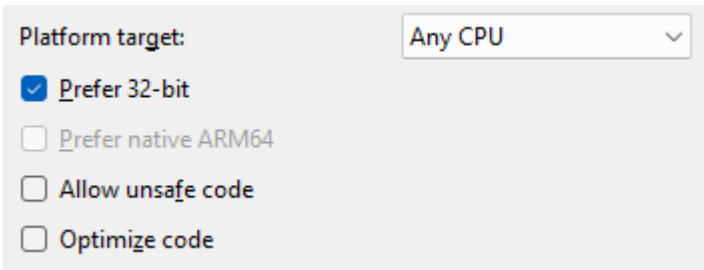
- a. Go to the Solution Explorer.
- b. Double click Properties.



- c. In the properties tab, click Build.



- d. In the Platform Target, check or uncheck the Prefer 32-bit checkbox.



PART 3 – Challenge

- 1. frmMain
 - a. The frmMain must display results of search queries.
 - b. The search query must require the name and gender of the employee.
 - c. Displaying records should be real-time where results should be displayed even when the user is still typing.
 - d. When the ALL item is selected in the combobox, all employee results regardless of gender should be displayed.
 - e. When the MALE item is selected in the combobox, all male employees should be displayed.
 - f. When the FEMALE item is selected in the combobox, all female employees should be displayed.
- 2. frmAdd
 - a. This form should be displayed when the Add button is clicked.
 - b. Add new employee records.
 - c. All fields are required.
 - d. The id is not included because it is an AutoNumber.
- 3. frmEdit
 - a. This form should be displayed when the Edit button is clicked.
 - b. Only the name, email, address can be edited.
 - c. Use the employee id as reference.
- 4. frmDelete
 - a. This form should be displayed when the Delete button is clicked.
 - b. Use employee name and email as reference for the deletion.