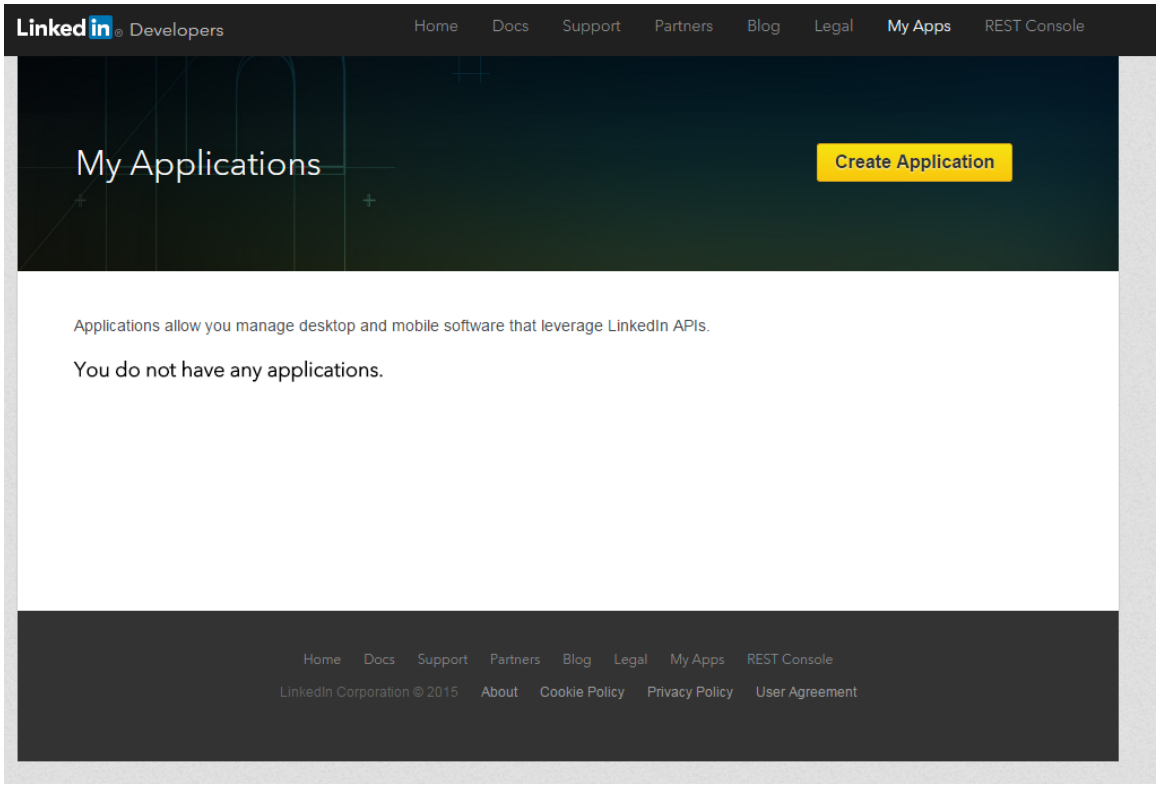
Sign in using LinkedIn and MVC

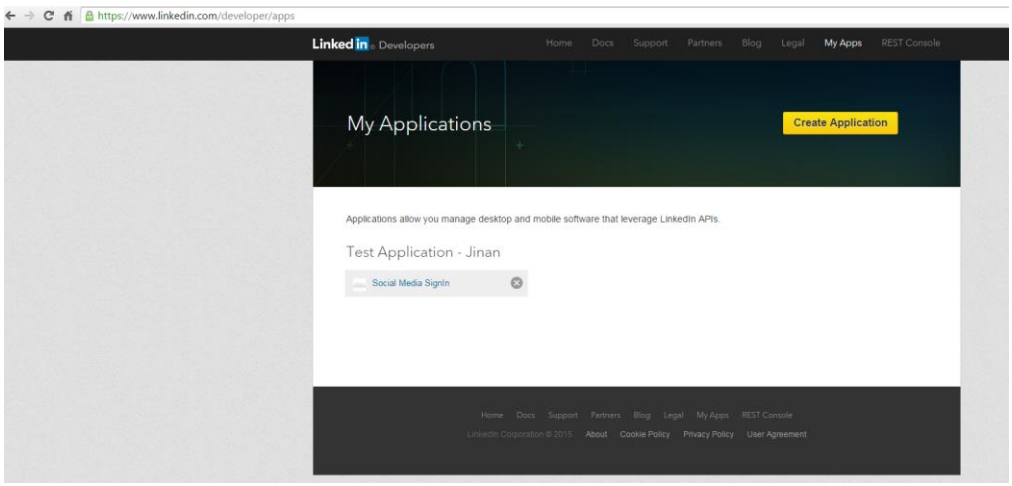
This article shows and highlights how to sign in using LinkedIn button.

First of all, you need to register your application with LinkedIn, as with other global sign-in providers. The screenshot below shows how:



After you create your application on LinkedIn site, it is as if you have registered it with LinkedIn sot allow users to sign in using LinkedIn button. Linked in Uses openAuth protocol.

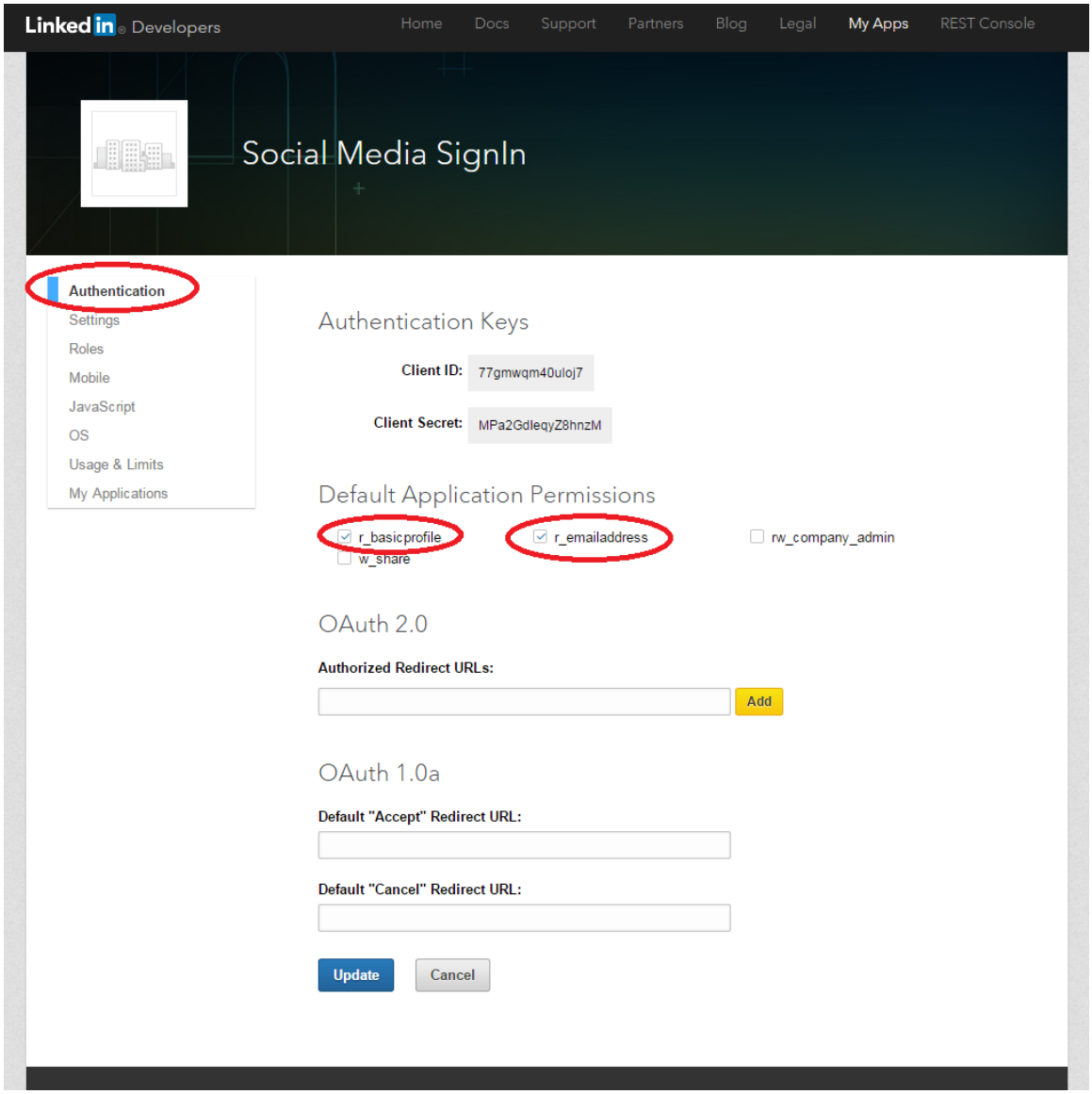
Below is a scrrenshot at my test app that I have just created:



There are several settings you must configure before jumping to implementing the actual sign-in:

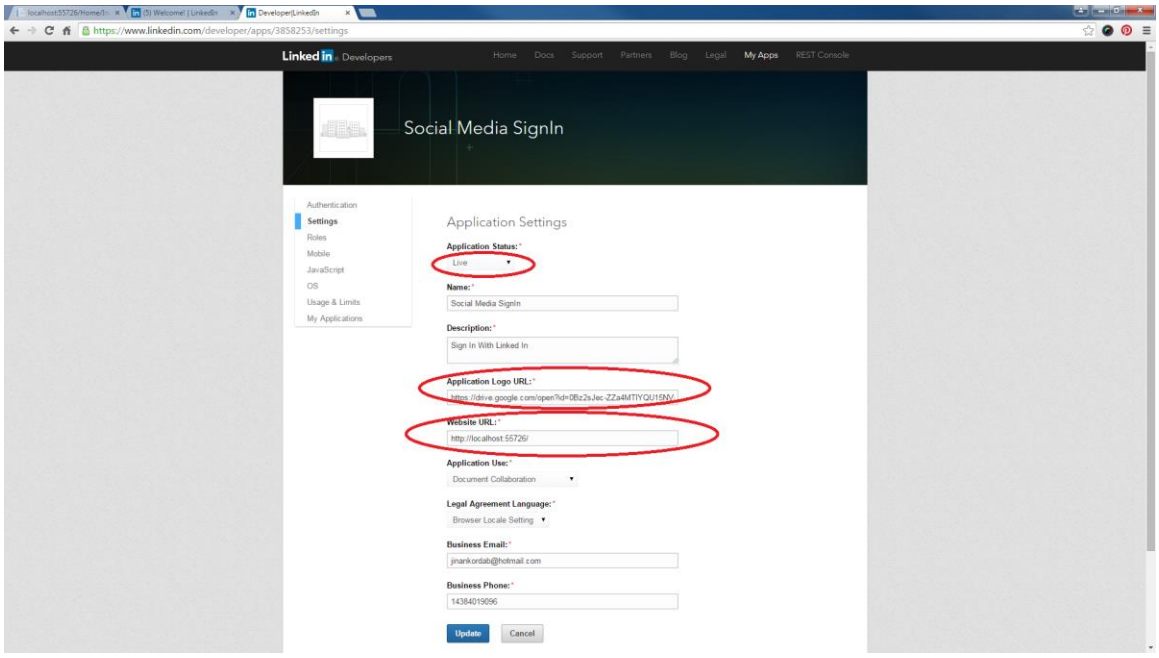
Authentication Section

Please see the highlighted red options that need to be selected:



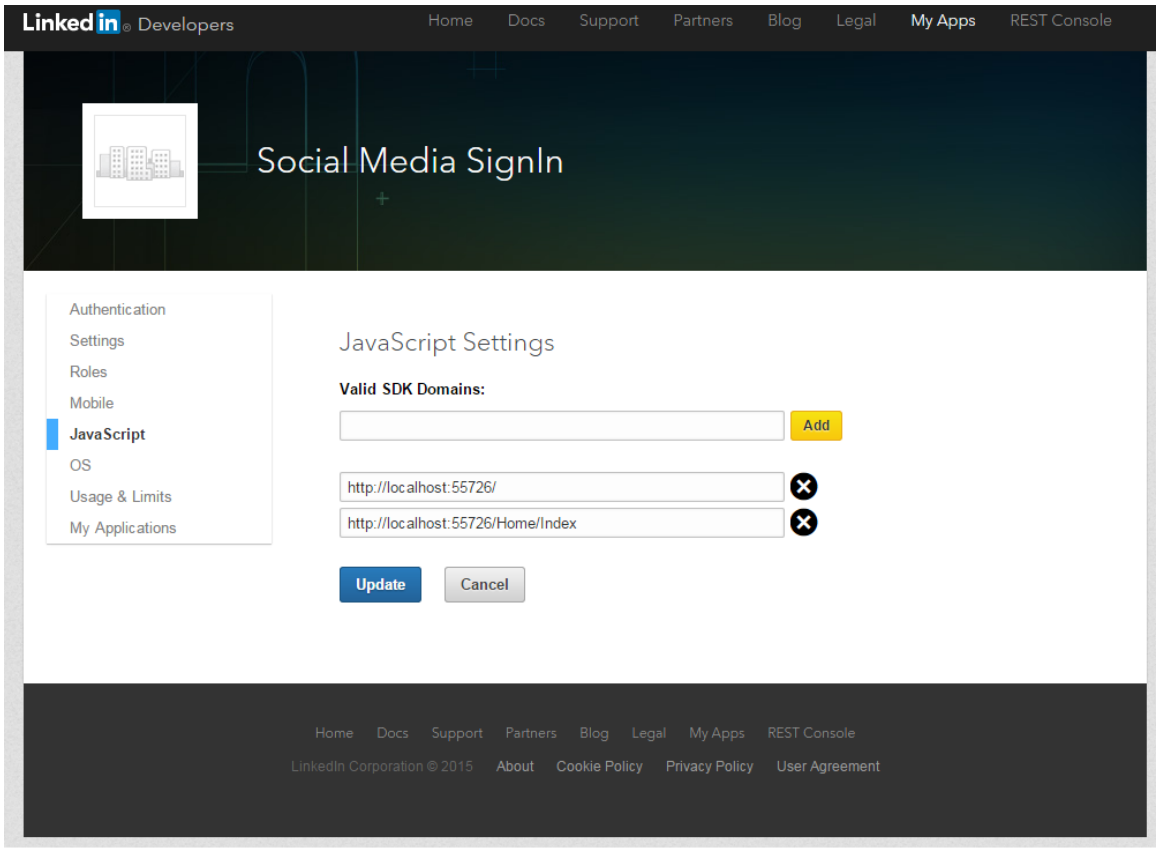
Settings Section

Application Status needs to be put Live, Application Logo URL needs to be HTTPS, and Website URL is where you will be running your MVC application, or any other application.



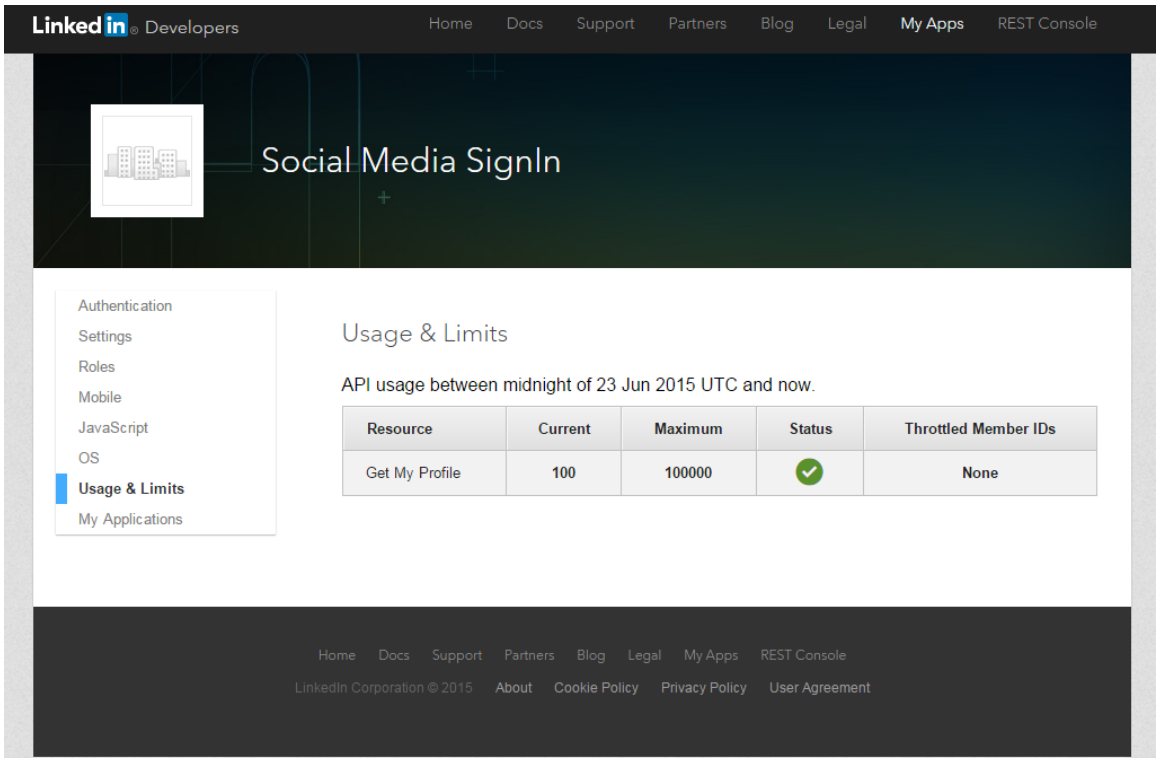
JavaScript Section

In this section, as seen below in the image, you need to put the URLs in your app where SDK will be put, like Javascript, and other frameworks, I have put two urls of my local app



Usage and Limits Section

Here you do not have to do anything, it will simply show you how many hits your app has, mine until now has 100 out of 100, 000.



Now The Code

Linked in Provides the following code to implement your sign-in:

In the head section you need to include your api key , as set by LinkedIn:

```
  <script type="text/javascript" src="//platform.linkedin.com/in.js">
    api_key:   xxxxxxxxxxxxxx  (your own api goes here)
</script>
```

Then we add linked in SIgn In Button :

```
<input type="button" onclick="onLinkedInLoad()" style="background-
image:url(../../Images/Sign-In-Small---Hover.png); width:293px; height:41px;
background-repeat:no-repeat"/>
```

With a call to `onLinkedInLoad()`

```
   function onLinkedInLoad() {
        IN.UI.Authorize().place();
        IN.Event.on(IN, "auth", function () { onLogin(); });
```

```
        IN.Event.on(IN, "logout", function () { onLogout(); });
    }
```

This function sets the functions to act after authentication and after log out. In this case we have onLogin and onLogout.

onLogin:

```
function onLogin() {
        IN.API.Profile("me").result(displayResult);
    }
```

onLogout :

```
    function onLogout() {
        IN.User.logout();
    }
```

And

displayResult function:

```
    function displayResult(profiles) {
        member = profiles.values[0];

        $.ajax({
            type: 'POST',
            url: '<%= Url.Action("AuthenticateUser", "Home") %>',
            data: {
                UID: member.id,
                FNAME: member.firstName,
                LNAME: member.lastName
            },
            dataType: 'html',
            success: function () {
                location.href = "Home/Index?f=" + member.firstName + "&l=" +
member.lastName;
            },
            error: function (req, status, err) {
                alert(err);
            }
              });


    }
```

That is all there is to implement log in. Official linkedInbutton is downloaded from their site as well: You can chceck these URLs for official linked in Buttons:

https://developer.linkedin.com/docs/signin-with-linkedin

https://developer.linkedin.com/downloads

And here is my view as in MVC:

```
<%@ Page Language="C#" MasterPageFile="~/Views/Shared/Site.Master"
Inherits="System.Web.Mvc.ViewPage" %>


<asp:Content ID="indexContent" ContentPlaceHolderID="MainContent" runat="server">

    <%if (Request["f"] != null && Request["l"] != null)
      { %>


    <div class="app-bar">
        <a class="app-bar-element">
            <span id="toggle-tiles-dropdown" class="mif-apps mif-2x">Welcome <%=
Request["f"].ToString() + " " +  Request["l"].ToString()%></span>
            <div class="app-bar-drop-container"
                data-role="dropdown"
                data-toggle-element="#toggle-tiles-dropdown"
                data-no-close="false" style="width: 324px;">
                <div class="tile-container bg-white">
                    <div class="tile-small bg-cyan">
                        <div class="tile-content iconic">
                            <span class="icon mif-onedrive"></span>
                        </div>
                    </div>
                    <div class="tile-small bg-yellow">
                        <div class="tile-content iconic">
                            <span class="icon mif-google"></span>
                        </div>
                    </div>
                    <div class="tile-small bg-red">
                        <div class="tile-content iconic">
                            <span class="icon mif-facebook"></span>
                        </div>
                    </div>
                    <div class="tile-small bg-green">
                        <div class="tile-content iconic">
                            <span class="icon mif-twitter"></span>
                        </div>
                    </div>

                    <div class="tile-small bg-green">
                        <div class="tile-content iconic">
                            <span class="icon mif-twitter"></span>
                        </div>
                    </div>

                </div>
            </div>
```

```
        </a>
    </div>









    <%}
      else
      { %>

    <% if (ViewData["newuserjustregistered"] != null &&
ViewData["newuserjustregistered"].ToString() == "TRUE")
       { %>

    <div class="login-form padding20 block-shadow" style="opacity: 1; -webkit-
transform: scale(1); transform: scale(1); -webkit-transition: 0.5s; transition:
0.5s;">


        <h1 class="text-light">Register</h1>
        <h1 class="text-enlarged">Step 2</h1>
        <hr class="thin">
        <br>

        <div class="input-control text full-size" data-role="input">
            Now,please <a href="#" onclick="onLinkedInLoad2(<%=
ViewData["LAST_REG_USR"].ToString() %>)">click here</a> to bind your account to
LinkedIn
        </div>
    </div>

    <%}
       else
       { %>

    <div class="login-form padding20 block-shadow" style="opacity: 1; -webkit-
transform: scale(1); transform: scale(1); -webkit-transition: 0.5s; transition:
0.5s;">

        <form action="/Home/RegisterUser" runat="server" method="post">
            <h1 class="text-light">Register</h1>
            <h1 class="text-enlarged">Step 1</h1>
            <hr class="thin">
            <br>
            <div class="input-control text full-size" data-role="input">
                <label for="user_login">First Name:</label>
                <input type="text" name="user_f_name" id="user_f_name"
style="padding-right: 39px;">
                <button class="button helper-button clear" tabindex="-1"
type="button"><span class="mif-cross"></span></button>
            </div>

            <br>
            <br>
```

```
            <div class="input-control text full-size" data-role="input">
                <label for="user_login">Last Name:</label>
                <input type="text" name="user_l_name" id="user_l_name"
style="padding-right: 39px;">
                <button class="button helper-button clear" tabindex="-1"
type="button"><span class="mif-cross"></span></button>
            </div>

            <br>
            <br>
            <div class="input-control text full-size" data-role="input">
                <label for="user_login">Email:</label>
                <input type="text" name="user_email" id="user_email"
style="padding-right: 39px;">
                <button class="button helper-button clear" tabindex="-1"
type="button"><span class="mif-cross"></span></button>
            </div>
            <br>
            <br>

            <div class="form-actions">
                <button type="submit" class="button primary">Register</button>
                <button type="button" class="button link"
onclick="clearRegForm()">Cancel</button>
            </div>
        </form>
    </div>


    <div class="login-form padding20 block-shadow" style="opacity: 1; -webkit-
transform: scale(1); transform: scale(1); -webkit-transition: 0.5s; transition:
0.5s;">

        <h1 class="text-light">Or,</h1>
        <hr class="thin">
        <br>
        <div>
            <input type="button" onclick="onLinkedInLoad()" style="background-
image: url(../../Images/Sign-In-Small---Hover.png); width: 293px; height: 41px;
background-repeat: no-repeat" /></div>


    </div>

    <%} %>
    <%} %>
</asp:Content>
```

My Master Page:

```
<%@ Master Language="C#" Inherits="System.Web.Mvc.ViewMasterPage" %>

<!DOCTYPE html>
<html lang="en">
<head runat="server">
```

```
<script type="text/javascript" src="//platform.linkedin.com/in.js">
api_key:   xxxxxxxxxxxx

</script>

<script type="text/javascript">


    var lastru;

    function onLinkedInLoad2(LRU) {
        IN.UI.Authorize().place();
        IN.Event.on(IN, "auth", function () { onLogin2(); });
        lastru = LRU;
    }
    function onLogin2() {
        IN.API.Profile("me").result(SaveResult);
    }

    function SaveResult(profiles) {
        member = profiles.values[0];

        $.ajax({
            type: 'POST',
            url: '<%= Url.Action("SaveRegisteringUserStep2", "Home") %>',
        data: {
            UID: member.id,
            FNAME: member.firstName,
            LNAME: member.lastName,
            IDENTITY: lastru
        },
        dataType: 'html',
        success: function () {
            location.href = "Index";
        },
        error: function (req, status, err) {
            alert(err);
        }
    });


}


function onLinkedInLoad() {
    IN.UI.Authorize().place();
    IN.Event.on(IN, "auth", function () { onLogin(); });
    IN.Event.on(IN, "logout", function () { onLogout(); });
}

function onLogin() {
    IN.API.Profile("me").result(displayResult);
}

function displayResult(profiles) {
    member = profiles.values[0];
```

```
        $.ajax({
            type: 'POST',
            url: '<%= Url.Action("AuthenticateUser", "Home") %>',
            data: {
                UID: member.id,
                FNAME: member.firstName,
                LNAME: member.lastName
            },
            dataType: 'html',
            success: function () {
                location.href = "Home/Index?f=" + member.firstName + "&l=" +
member.lastName;
            },
            error: function (req, status, err) {
                alert(err);
            }
        });


    }


    function clearRegForm() {
        $("#user_f_name").val("");
        $("#user_l_name").val("");
        $("#user_email").val("");

    }

    function RegisterUser(FIRST_NAME, LAST_NAME, EMAIL) {
        $.ajax({
            type: 'POST',
            url: '<%= Url.Action("RegisterUser", "Home") %>',
            data: {
                firstName: USERID,
                lastName: FIRST_NAME,
                email: EMAIL
            },
            dataType: 'html',
            success: function () {
                alert('success');
            },
            error: function (req, status, err) {
                alert(err);
            }
        });
    }


    </script>

    <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>

    <link href="../../docs/css/metro.css" rel="stylesheet">
    <script src="../../docs/js/metro.js"></script>
    <link href="../../docs/css/metro-icons.css" rel="stylesheet">
</head>
```

```
<body>
    <asp:ContentPlaceHolder ID="MainContent" runat="server">
    </asp:ContentPlaceHolder>
</body>
</html>
```

And my Model:

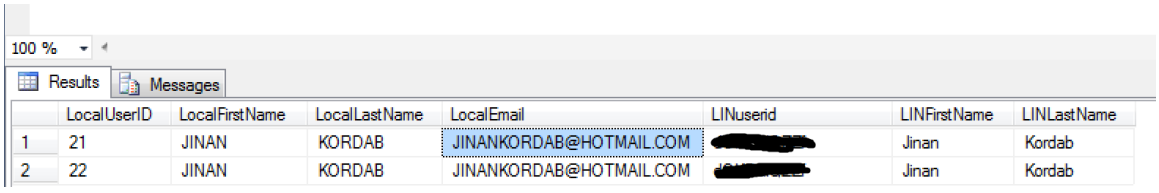
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace LogIn.Models
{
    public class UserAuthentication
    {
        public System.Data.DataTable AuthenticateUser(string UID, string FNAME,
string LNAME)
        {
            string connectionString = "data source=PORPRG_2177\\SQLEXPRESS;Initial
Catalog=LOGIN;Integrated Security=True";
            string query = "SELECT  * FROM [LOGIN].[dbo].[Users] WHERE [LINuserid]
=  '" + UID + "' AND [LINFirstName] = '" + FNAME + "'AND [LINLastName] = '" +
LNAME + "' ORDER BY LocalUserID DESC";

            System.Data.SqlClient.SqlConnection conn = new
System.Data.SqlClient.SqlConnection(connectionString);
            System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand(query, conn);
            System.Data.DataTable dt = new System.Data.DataTable();
            conn.Open();
            System.Data.SqlClient.SqlDataAdapter da = new
System.Data.SqlClient.SqlDataAdapter(cmd);
            da.Fill(dt);
            conn.Close();
            da.Dispose();

            return dt;
        }

        public int  RegisterUser(string firstName, string lastName, string email)
        {
            int ident;

            System.Data.SqlClient.SqlConnection connection = new
System.Data.SqlClient.SqlConnection("data source=PORPRG_2177\\SQLEXPRESS;initial
catalog=LOGIN;Integrated Security=True");
            string SQL_to_register = "INSERT INTO
Users([LocalFirstName],[LocalLastName],[LocalEmail])   VALUES ('" + firstName +
"','" + lastName + "','" + email + "')  select SCOPE_IDENTITY() ";
```

```
            System.Data.SqlClient.SqlCommand command = new
System.Data.SqlClient.SqlCommand(SQL_to_register, connection);
            connection.Open();
            Object objc = command.ExecuteScalar();
            ident = Convert.ToInt32(objc);
            connection.Close();
            return ident;
        }

        public void SaveRegisteringUserStep2(string UID, string FNAME, string
LNAME, string idtty)
        {
            System.Data.SqlClient.SqlConnection connection = new
System.Data.SqlClient.SqlConnection("data source=PORPRG_2177\\SQLEXPRESS;initial
catalog=LOGIN;Integrated Security=True");
            string SQL_to_register = "UPDATE Users SET [LINuserid] = '" + UID +
"', [LINFirstName] = '" + FNAME + "', [LINLastName] = '" + LNAME + "' WHERE
[LocalUserID] = " + idtty;
            System.Data.SqlClient.SqlCommand command = new
System.Data.SqlClient.SqlCommand(SQL_to_register, connection);
            connection.Open();
            command.ExecuteNonQuery();
            connection.Close();
        }



    }
}
```

The model class helps in additional security layer.

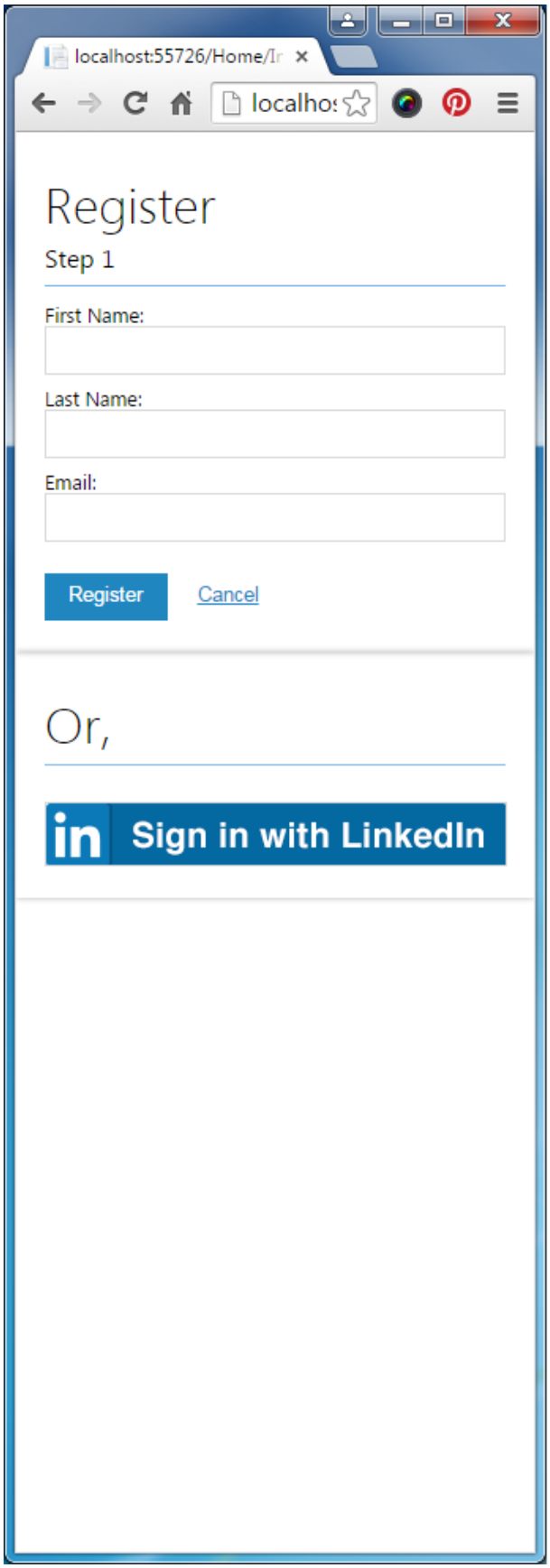
The image below demonstrates the options a user given to REGISTER. Registering stores his information on SQL server, and then matched against the LinkedIn credentials.

I actually created a table where I store both, users info, as well as linked in info, attributed to the same user. This was it is easier to **identify** users later on.

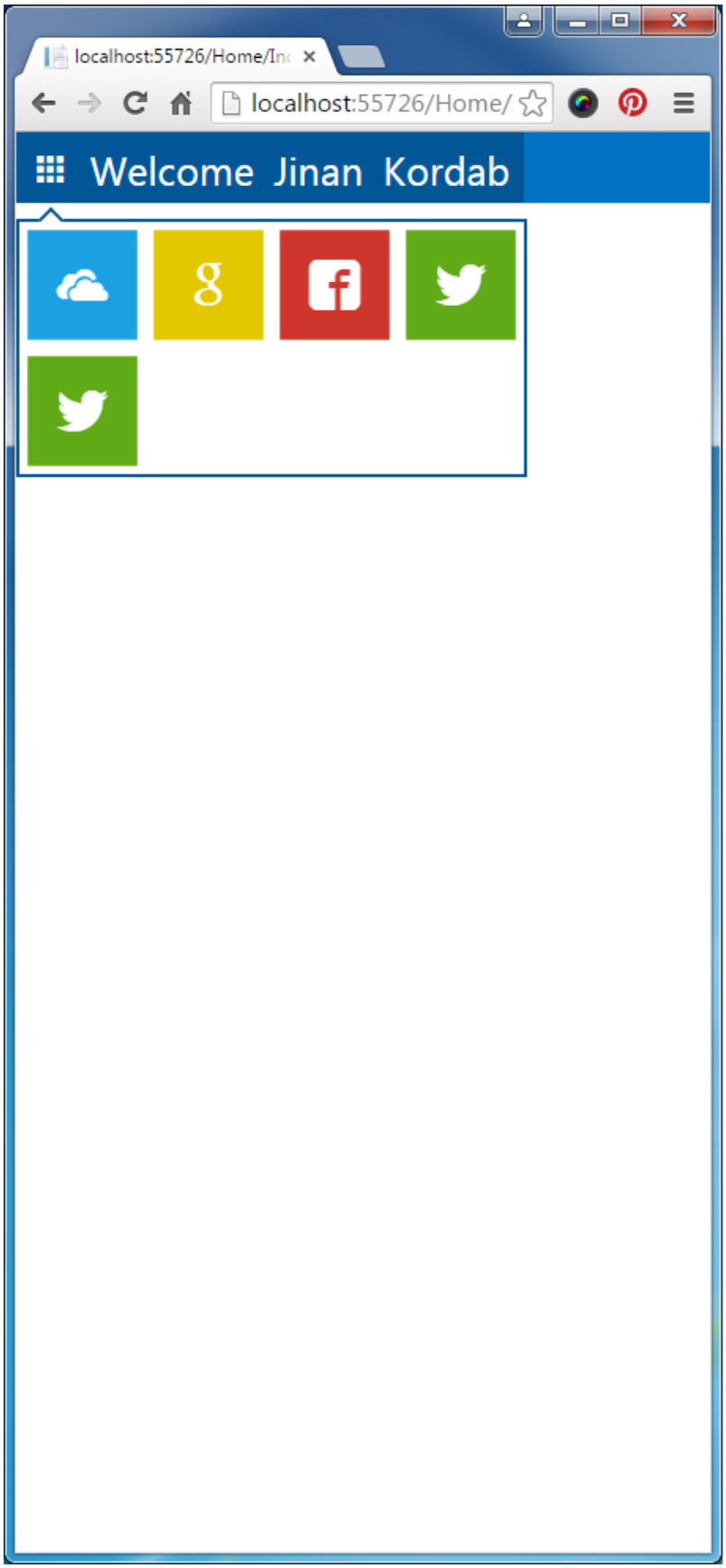


	LocalUserID	LocalFirstName	LocalLastName	LocalEmail	LINuserid	LINFirstName	LINLastName
1	21	JINAN	KORDAB	JINANKORDAB@HOTMAIL.COM	[redacted]	Jinan	Kordab
2	22	JINAN	KORDAB	JINANKORDAB@HOTMAIL.COM	[redacted]	Jinan	Kordab

And here is a screenshot of my app:


localhost:55726/Home/Ir
localhos
Register
Step 1
First Name:
Last Name:
Email:
Register
Cancel
Or,
Sign in with LinkedIn

Another screenshot after user logs and authenticates successfully:


localhost:55726/Home/In
localhost:55726/Home/
Welcome Jinan Kordab