**Viewer Module**

**Sign In & Sign Up**

Firstly, let’s start with login and register control. These controls, buttons, icons etc. made by helper bootstrap classes and for opening this small pop up windows using bootstrap modal control. Almost every website you can’t use some of controls without register yourself to their database. When you start to attempt sign up our website, you need to enter your ‘name, surname, email and password’. In these control I got some help from available asp.net validators.

All the fields can not be left blank before tap the register button. If you tap, you will see ‘required’ text at the under the textbox which you left blank. This control handled by ‘Asp:RequiredFieldValidator’. This validator doesn’t let you left blank these field while you are registering. Also we have a ‘confirm password’ field and for this control we I used ‘Asp:CompareValidator’. This validator taking 2 parameters as a control which are ‘ControlToValidate’ and ‘ControlToCompare’. These are controls to ‘password’ and ‘confirm password’ are same or not. E-mail field just taking an e-mail format string with simple ‘TextMode’ control. If you make a mistake while you writing your e-mail, a little popup will warn you about it.

You password must be at least 6 character and these control handled by ‘Asp:RegularExpressionValidator’. This is works as same like ‘requiredfieldvalidator’ which is show you a text under the password field after you entering your password without 6 character. But, this control doesn’t concern about the post back is send with clicking button or not without at least 6 character. So, I handle this situation at code behind with simple ‘textbox.text.length’ property.

Lastly, let’s check the text which is backgrounded with light blue at right above the sing up window. This field is made for warn user to making mistake. For example, if entered email is registered before or password field less than 6 character, a post back comes and show you a simple warning in this field.

If you write all register fields correctly after checking you will be registered the database. After registering, you are automatically signed in the website and your session start to keeping.

And, login control. Which is taking e-mail and password for login the website. It is simple checking the fields correctness via database at code behind and if correct signed in you, if not warn you with text at right above the control.

**Login With Facebook**

This button works for signed in you via your facebook account, if you have. Now, let’s look at how it work. For creating such a control which taking your facebook information and register these information to our database, I used ‘Facebook SDK’ for .Net framework. This kit has some useful methods for making such a control and obtain permission from facebook. Before I start using it, I create a facebook web API in developers page in facebook. Which gives me ‘AppID and AppSecret’, and for this API’s Url I have our website address. This means this ‘AppID and AppSecret’ will only works for this Url which I gave.

After clicking this button, my first function creating a login url for my api. It uses Facebook SDK’s FacebookClient class and inside it GetLoginUrl method. This method taking several parameters to creating url.

*FacebookTask.GetLoginUrl(*

*new*

*{*

*client\_id = AppID,*

*client\_secret = AppSecret,*

*redirect\_uri = CallBackUrl,*

*response\_type = "code",*

*scope = Scope,*

*});*

For taking this url I also give a call back url which is our website main page. That means after facebook login is done come back to this url.And at the call back url page loading method I check for is there any come any request which has a key “code”. As you can see above method, we describe response\_type which is equalt to “code” key. After getting this “code”, so we can redirect our user to facebook page and asking for her/him permission.

*dynamic result = FacebookTask.Post("oauth/access\_token",*

*new*

*{*

*client\_id = AppID,*

*client\_secret = AppSecret,*

*redirect\_uri = CallBackUrl,*

*code = code*

*});*

*return result.access\_token;*

I posted the kind of information about who we are, what we want and what will is come back, to facebook. If user give permission a simple method (getUserInfo) taking these information for using this “access\_token”. This token is approximately 50 character and use for only a login and logout for each user, can’t be repeat. I keep this “access\_token” in a variable, so while this user wants to log out, I used the code to log out in him/her facebook account also. After a user signed in to our website, I insert the user information to database for using it for each action in website.

**Session Management**

The basic stateless nature of web sites and web applications are rather very confusing to anyone who is starting with web development. ASP.NET provides various ways of handling the states.

Im using in-proc session mode. Using In-Proc session handling then all the session information will be stored in the server memory. This is the default session state in ASP.NET. This approach works fine as long as we are keeping small session information. Since the memory location was handled by the ASP.NET worker thread only it involves a considerable overhead for the worker thread to manage these. Also, since this is in the memory of server, chances are that large session information would lead to more memory usage and thus lowering the performance. I set up my session timeout as 60 minutes.

The problem with In-Proc mode session mode is that if the size of session data becomes large then it effects performance. Also, since the session data is in memory of server the application restart(ASP.NET) will effectively flush all the data.

**Search Part**

After writing searching user name or surname or e-mail, hitting enter button on keyboard, this written value on text box sending to search.aspx page via url like [http://www.site.com/search?s=’text](http://www.site.com/search?s='text)’. This text sending to search page as a ‘Request.QueryString’ method. This method check the “s” values for null or not when page is loading. If user send a text to search page with using upper menu textbox. “s” value will be equals to this method and this page get the string. After that sending the string to database and check for valid users. There is two function for search task. If written text is a word or more than one, this functions works for one or more words searching. Also there can be search with e-mail address. After that as you guess searching result come with a list to screen which includes user name, surname, email and description, then users can do click and go that user page (site.com/url).

**Main Page**

That is the page when you make connect to our website. A simple and useful page will welcome you. There is top 20 photos, which are get most like from other users. Also, under the photos you can see project name, at upper right photo description, upper left part has photo rate stars and top of the photo user image and name. Under the photos, there is another part for comments.

These control getting from database with ‘sql commands’ and putting in an ‘asp list view’ which covered by an ‘asp update panel’ in ‘Default.aspx’ page. In that pages most of the controls made by using asp.net controls. List view used for many items can be putting the same html & asp controls, and update panel used for liking or commenting actions refreshing dynamically. In that pages when you click the users name or images, you will redirect to that users page. This action handled by ‘asp hyperlink’, this control take a route for sending to regular page which is that user ‘url’. I am going to talk about this ‘url’ selection in profile page part.

Like and comment button working dynamically in main page. If user did not signed in the website, these button clicking action fires the login modal control and this modal page comes immediately to screen for warn to user like “you can’t use that controls without signed in”. Unfortunately, if user logged in after clicking a like button or sending comment, this action which is what you want to do before signed in did not sending to database and clicking action won’t fire. User should send like again or comment again. If user logged in before using these controls, like button clicking action working without need refresh, user will see like is sent, hearth painted to red and count increased. This controls handled with all at code behind and simple sql connections. The results is come back and ‘update panel’ shows you the result. If like is not sent or any problem occurs, the hearth will not be painted. Also sending comment works like same, after you sent a comment to a photo, this comment which you sent will be shown you at that moment and comment count will increasing via getting new result from ‘sql server’.

In the comment field users can also click the user names or pictures which are sent by user comment to any photo, they can directly go to that user viewing photograph page. This controls also handled with “hyperlink” via url routing. If there is any question like ‘how the hyperlink works?’, these control taking a property which is “NavigateUrl”. This property taking a “RouteValueDictionary” property. First one is the url routing name which I created from “global.asax” file. The second one is the key which are will be equal to that “global.asax” configuration parameter.

GetRouteUrl("User", new System.Web.Routing.RouteValueDictionary { { "userUrl", Eval("UserUrl").ToString()

**Profile Page**

Users can go to their profile path with clicking ‘profile’ button at the top menu. This page firstly checking for valid session. If user not logged in, this page will not be load and can’t be access. After checking session state, profile page has some ‘text box’ controls for making change for users profiles. These are ‘name, surname, description, avatar, url’ for editable and ‘id and e-mail’ text boxes can’t be editable after registiration. User profile picture have to be maximum 3 megabyte size. This checking doing from code behind with ‘PostedFile.ContentLength’ property from ‘.net framework HttpPostedFile’ class. The ‘url’ field allows the user to get the address he/she wants after website domain like ([www.site.com/URL](http://www.site.com/URL)). This property is unique and after trying to change, controls to database and if is not valid warn to user. Default ‘url’ is same as user ‘id’.

In this page another button which is redirect you to “changepassword.aspx” page. This page has 3 field “current password, new password and confirm new password”. Validation between ‘new password’ and ‘confirm new password’ fields handled with “asp comparevalidator” control. This is taking two property one of them “ControlToValidate” and “ControlToCompare”. If these field not same, a red text will warn you user right under the relevant text box. After attempt to change password, firstly checking current password is correct or not. If not the password will not be changed for sure.

**User Page**

The user page is showing user sharing photographs, projects and likes. At the top of the page there is purple windows. At the left side you can see this user name, description and e-mail information, at the right side user avatar. Bottom of the page there is three buttons. These are projects, photos and likes button and these elements counts. These button click event does not post back. These are working with ‘javascript’ and when you click it you will see the clicked item information which are already loaded to your browser and keeping hide until you click for see that details.

Projects part contains, you can the user sharing projects name, a sample photo from that project, and the number which is show the how many photos has in that project. If you click the any project, you directly go to that projects photos. After this, you can click any photo and a modal (bootstrap) page is opening. This modal page has some information for that photo. Which are user name who sent the photo, photo description, photos project name and at the bottom a like button with count. You also can sent a like that photo from that modal page. These control all handled from code behind at that page using C#. For clicking like there no need any post back for seeing like is sent or not. That is working with “updatepanel” any working at that moment with increasing like count.

At the like button page contains which photo is liked from which user. These control listing with “listview” control. And, also in that page user can click the photo for open that photo modal page for information or liked user can click for go to that user page. Still “hyperlink” control helped me with that all.

**Logout Action**

This action handled with a page. If a user click to log out button from that fixed menu at the top, this button will redirect to “logout.aspx” page. This page has not any ‘front end’ viewing. I just code for page loading event. This code contains session cleaning and facebook connection property. If a user connected with facebook, this button also log out facebook account for that user. For facebook log out, like I told above, I’m keeping “token string” when user connected with facebook to our website. And I use this token for log out that user again from facebook and website with session cleaning.