***Lab 9-Authentication and Authorization Demo***

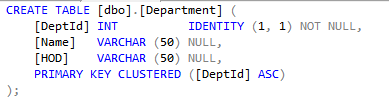
*Create a new MVC Application:*

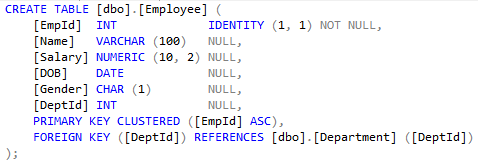
*File > New > Project > Project Name: LINQwithASP.NET > Click OK Button > Select Empty and click OK button.*

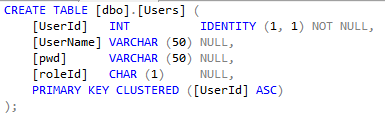
*Project will be create.*

*Right click on App\_Data folder > Add > New Item > SQl Server Database > Add*

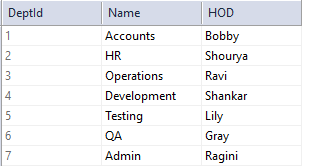
*Now add the following tables:*

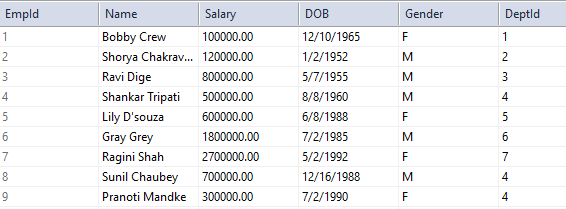
**

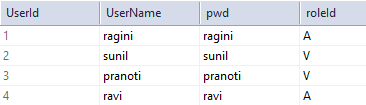
**

**

*Add few records in all the tables:*

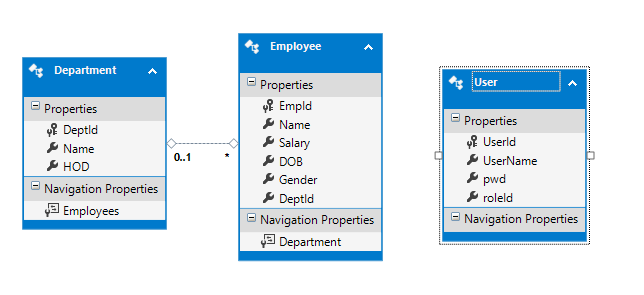
**

**

**

*Add Entity Data Model:*

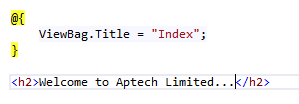
*Right click on Model Folder > Add > New Item > ADO.NET Entity Data Model > MyOrgnaisationModel > Add > EF Designer from Database > Next > MyOrgEntities > EF 6 > Next > Select all tables > MyOrgnaisationModel > Finish*

**

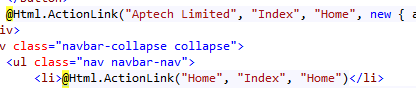
*Build the project*

*Right click Controller Folder > Add > Controller > Empty MVC Controller > Add > HomeController*

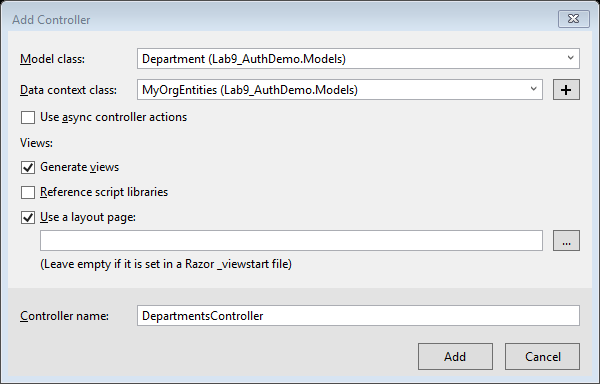
*Add empty view for Index action method. Add the following:*

**

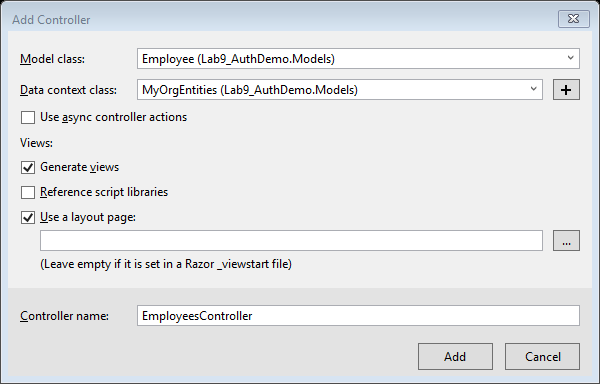
*Goto Layout.cshtml and Add the following code:*

**

*Right click Controller Folder > Add > Controller > Controller with Views using Entity Framework > Add*

**

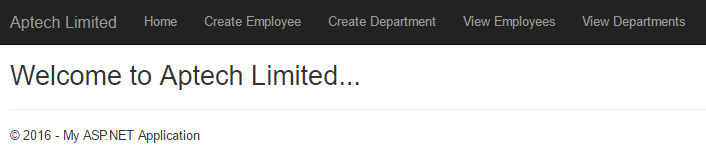
*Right click Controller Folder > Add > Controller > Controller with Views using Entity Framework > Add*

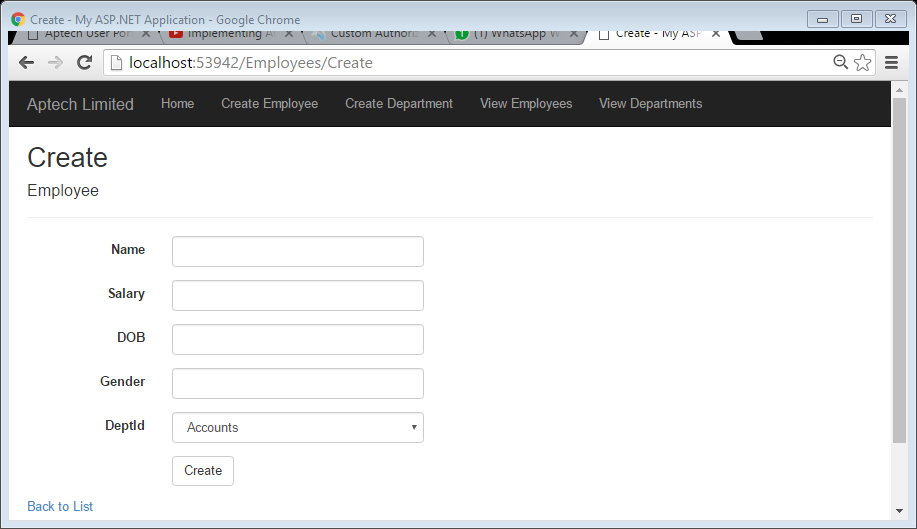
**

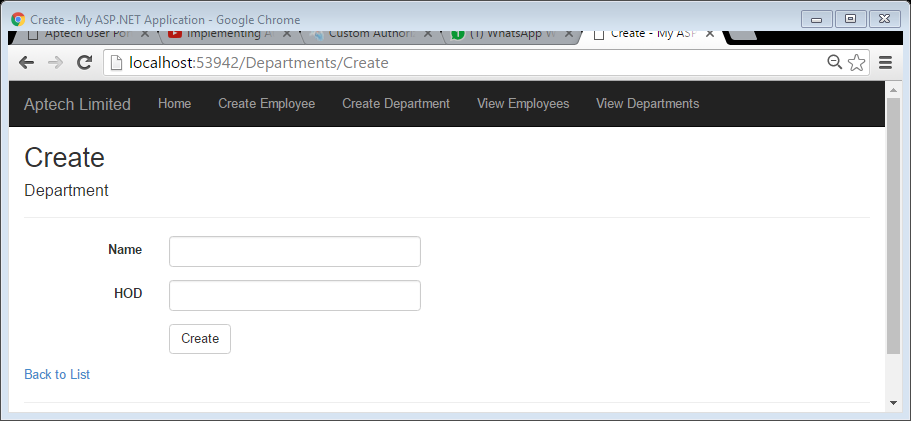
*Goto Layout.cshtml and Add the following code:*

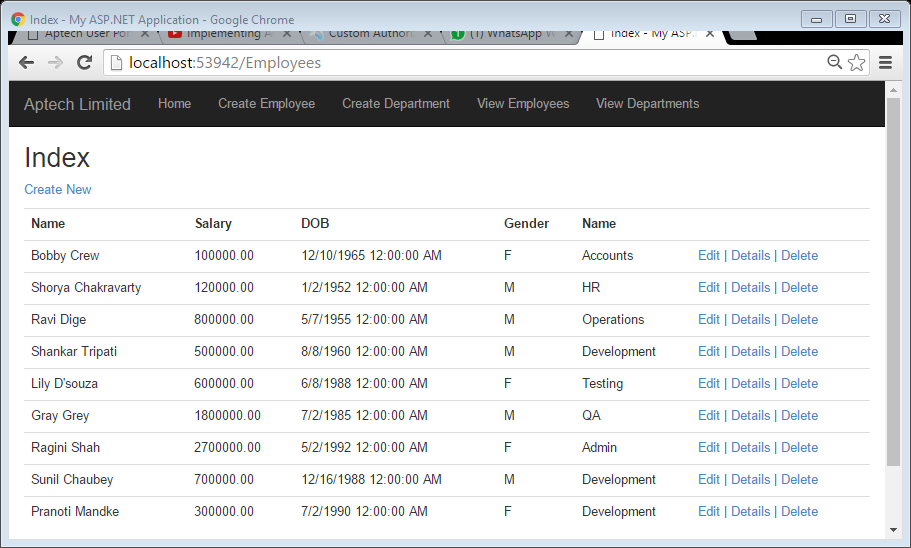
**

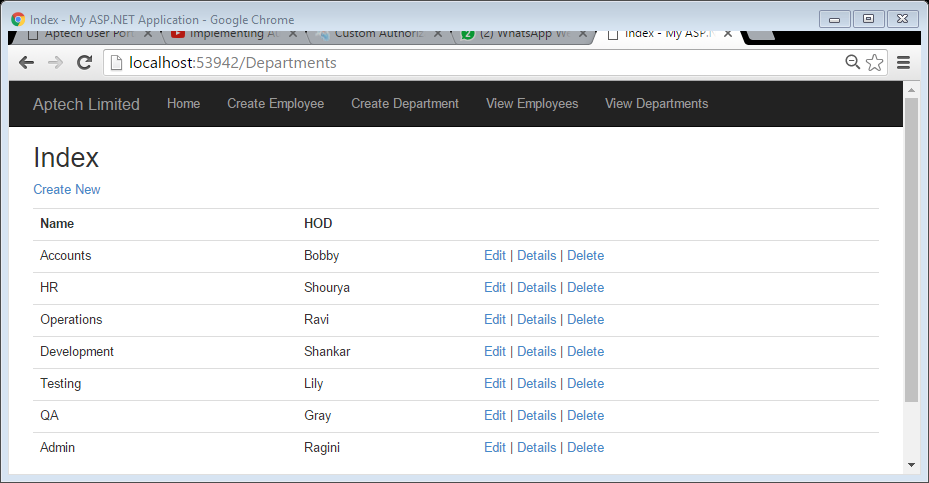
*Now we shall execute the code*

**

**

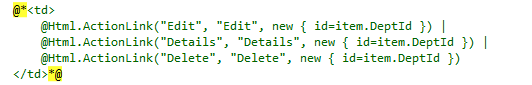
**

**

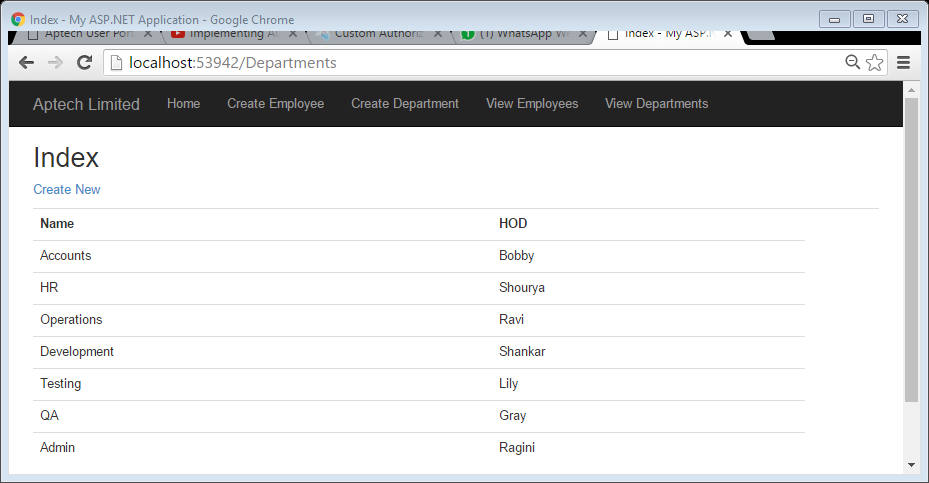
**

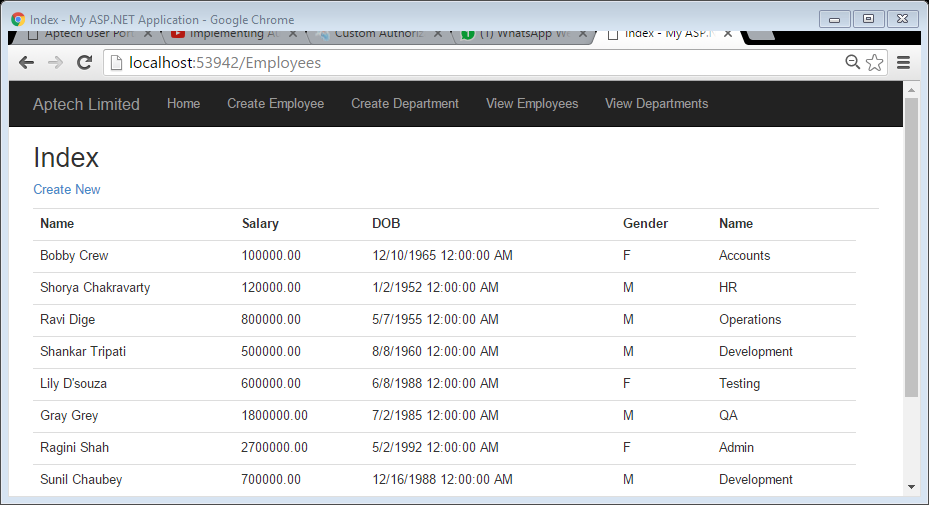
*We shall remove all the Edit, Details and Delete Links:*

*Comment the following code in Index.cshtml of Employee and Department*

**

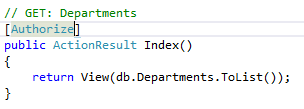
*Output will look like:*

**

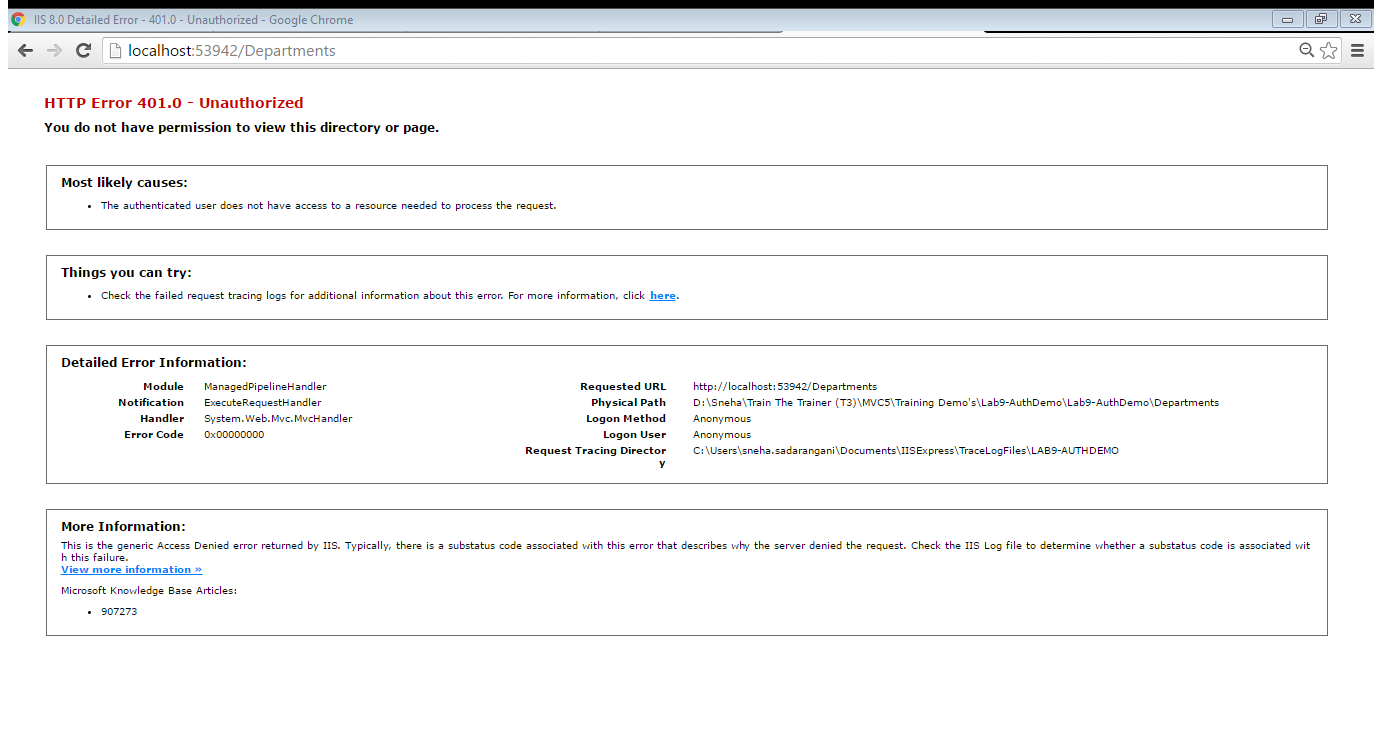
**

*Now all the links are available for Guest. We need to Authorize a user to access the links:*

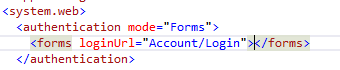
*In the Department Controller we will write the [Authorize] above the action method u want to authorize.*

**

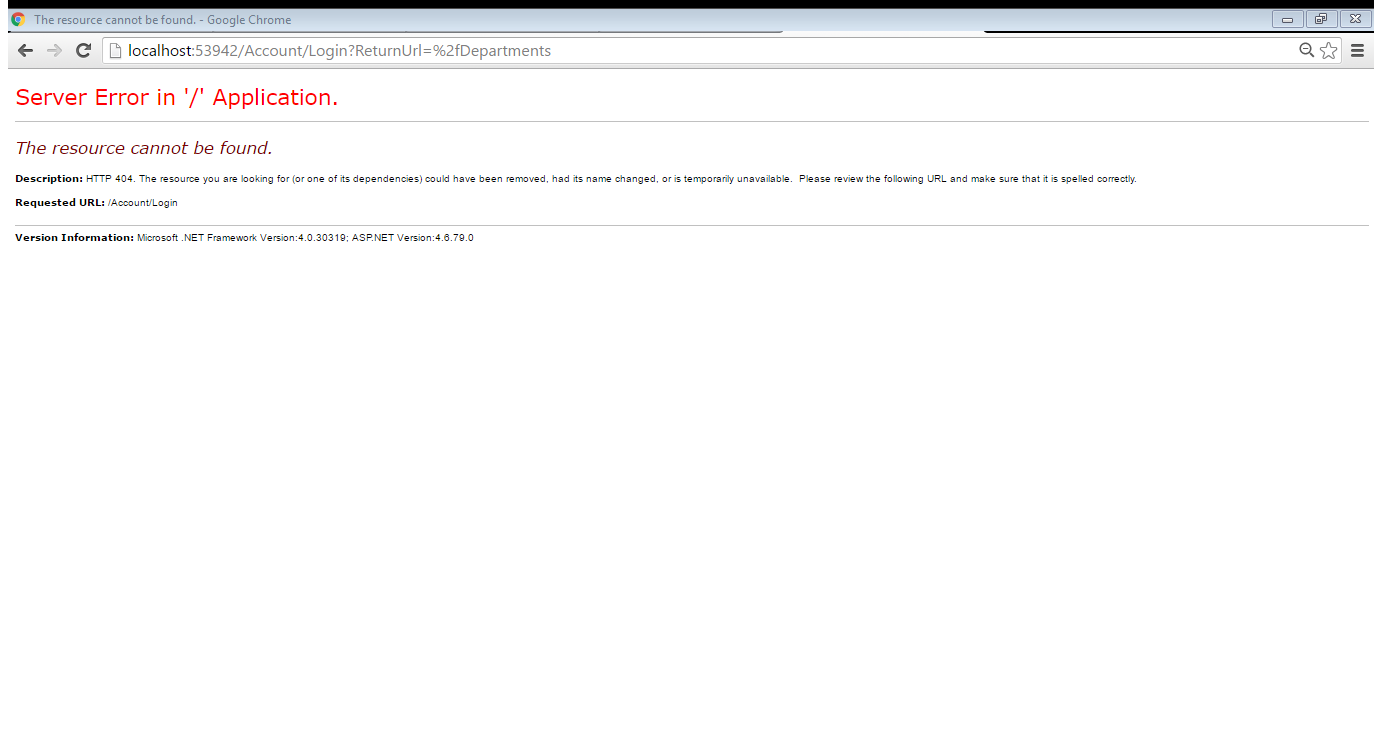
*You will get the following error:*

**

*We want to redirect the user to Login page. To do so goto web.config under <system.web> write the following code:*

**

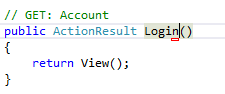
*Now it will give the following error:*

**

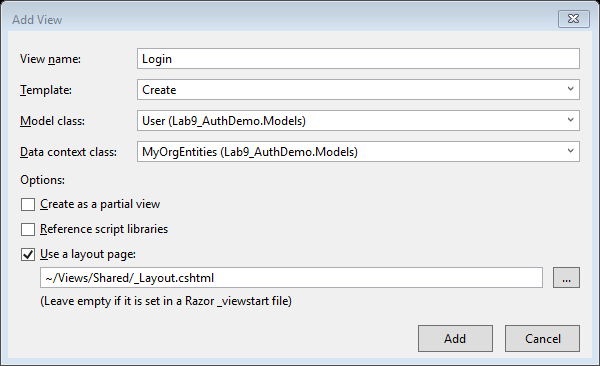
*We need to add Account Controller:*

*Right click Controller Folder > Add > Controller > Empty MVC Controller > Add > AccountController*

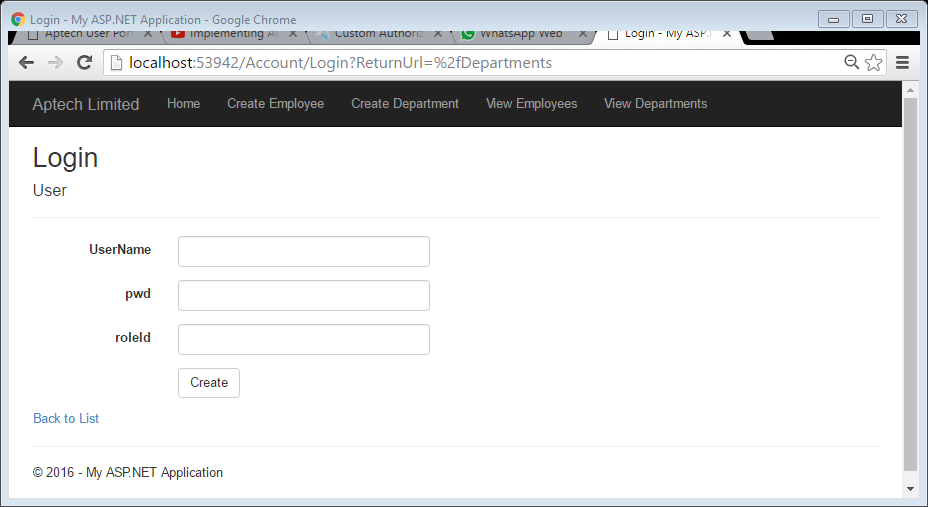
*Change the Index action name to Login*

**

*Add View for login*

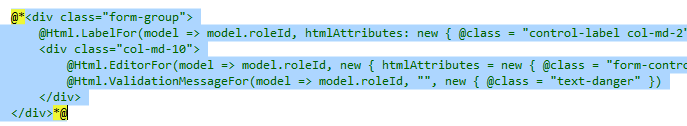
**

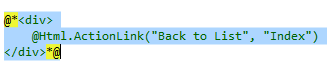
*Following will be the output*

**

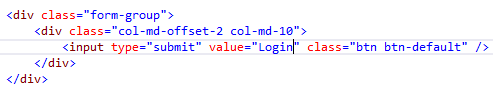
*We need to comment the following code:*

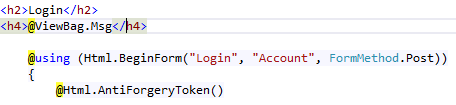
**

**

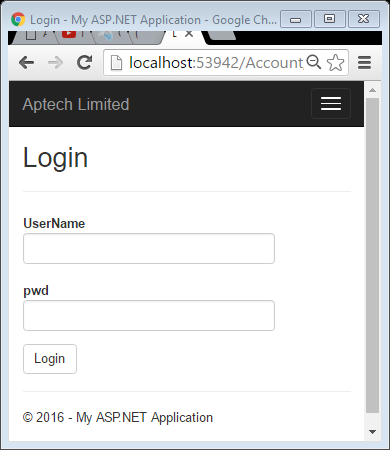
**

*Modify the following code:*

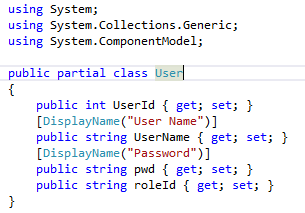
**

**

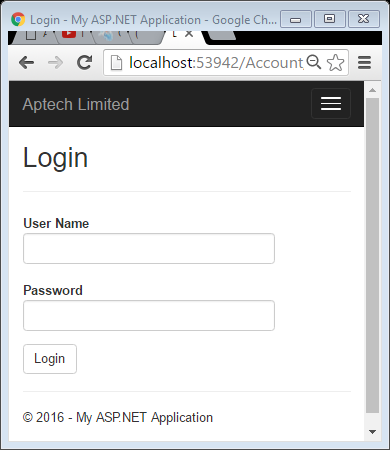
*The output will look like:*

**

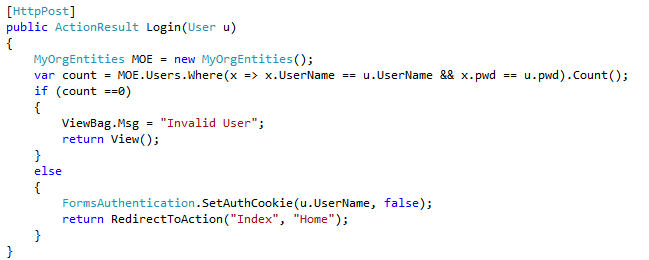
*Now we need to change Label values, Goto User.cs under Models folder and make the following changes:*

**

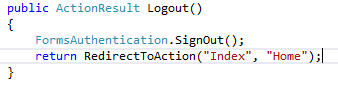
*Output will look like:*

**

*Task to be performed on Login click. Goto Account Controller and write the following code:*

**

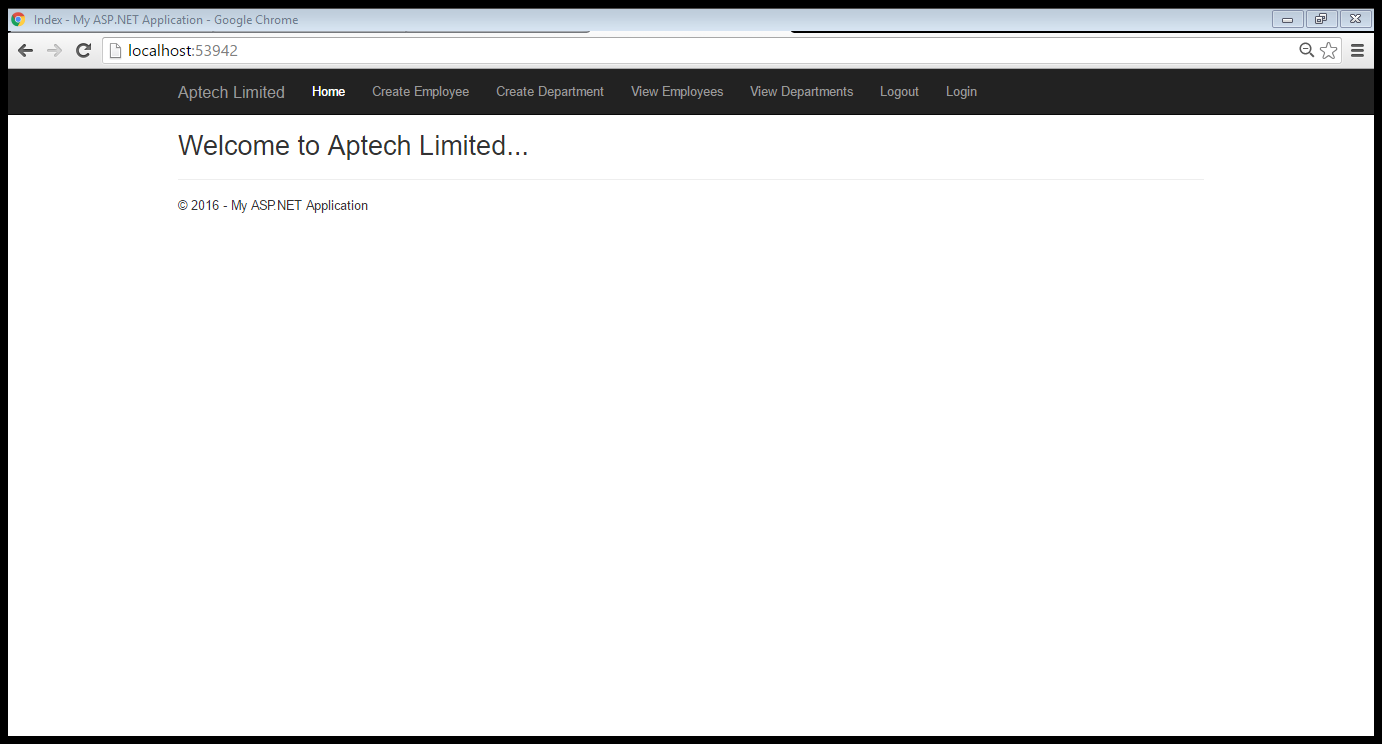
*For Logout write the following code in the Account Controller:*

**

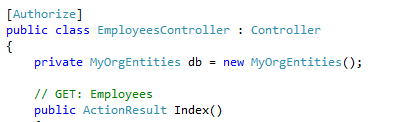
*Adding links of Login and Logout. Goto Layout.cshtml*

**

*The output will be:*

**

*Now we have given secured access to one link if we have to give secure access to all the action methods we write [Authorize] on Controller as shown in the code.*

**

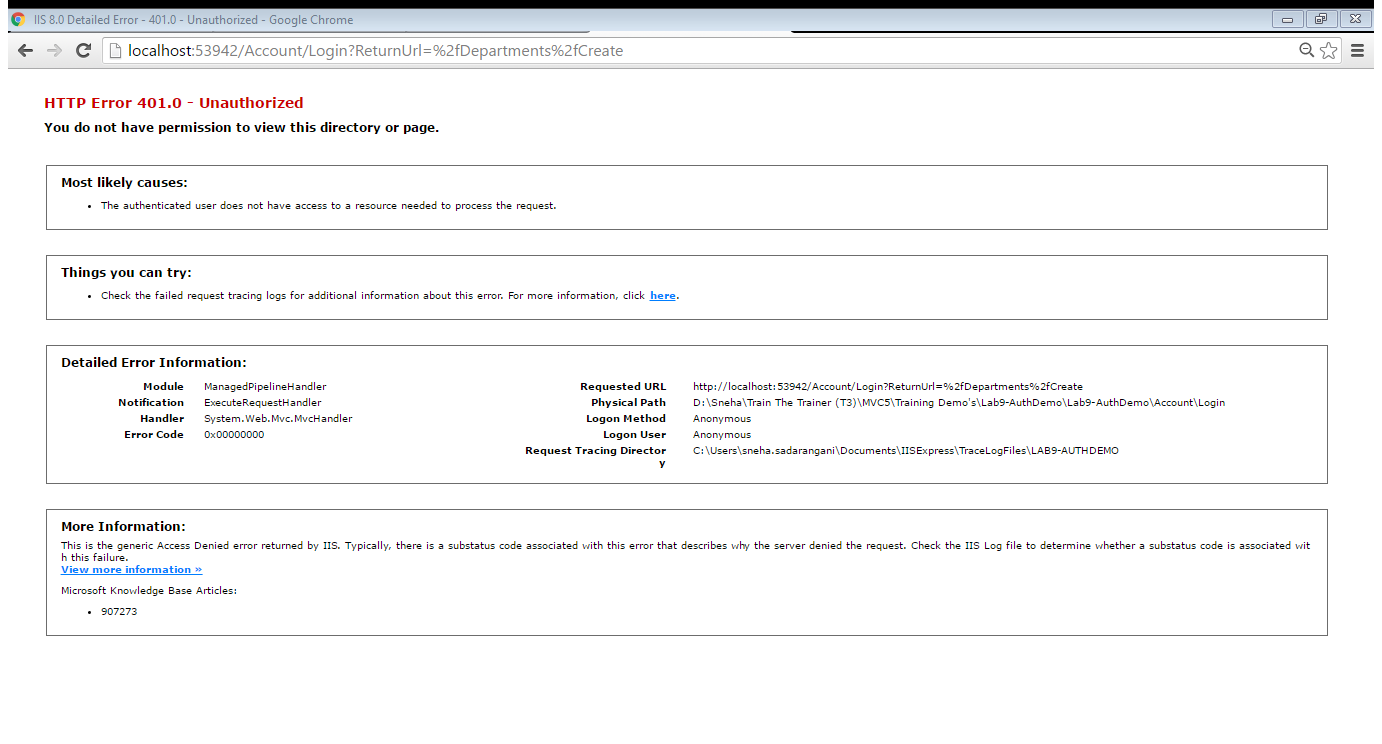
Authorization Filter Implementation

*Now if you try to access View Departments, View Employees or Create Employees it will as for Login.*

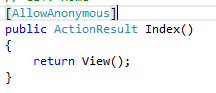
*If we want to authorize and secure the whole code then we make the modification in global.asax*

**

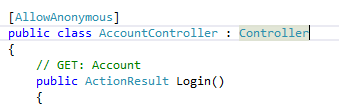
*After executing this code all the pages will have no access.*

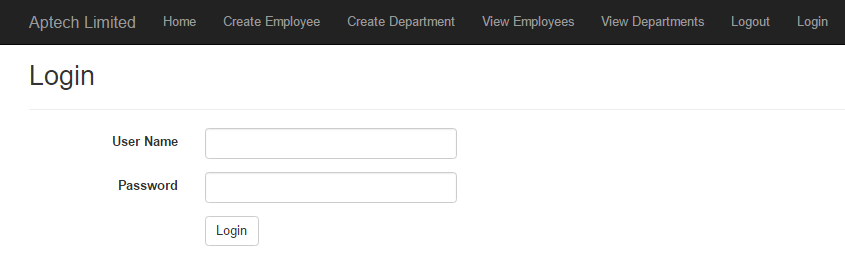
**

*So to allow access to the Home and Login page we shall write [AllowAnonymous] above the each action method as shown:*

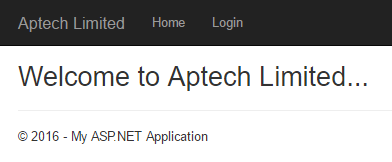
**

Action Filter Implementation

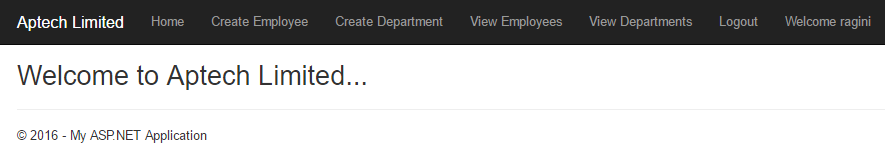
**

*Now the website will work properly. *

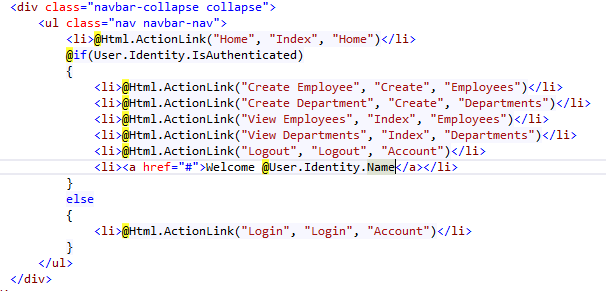
*Every time we run the website we are able to see Login and Logout. We will make it more user friend. Like when the Guest visits, the view will be as follows:*

**

*When the user Log’s In, the view will be as follows:*

**

*This can be achieved by writing the following code in the Layout.cshtml*

**

*Now the program works irrespective of the type of user. Now we shall implement the role based working of links:*

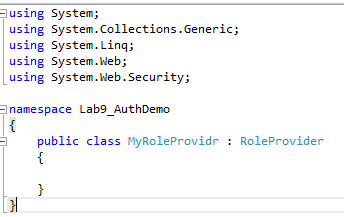
*If the RoleID of the user “V” then access will be View Department and View Employees. If RoleId is “A” all links can be accessed.*

*To perform the above we need the following:*

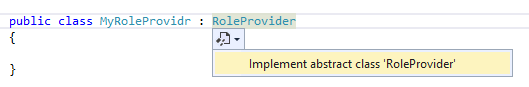
1. *Custom Role Provider*
2. *Changes in Configuration to implement Custom Role Provider*
3. *Corresponding implementation in necessary Controllers*

***Creating a Custom Role Provider:***

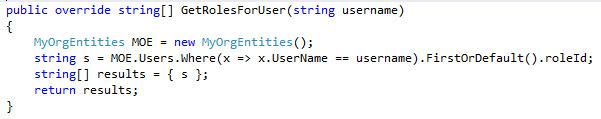
*Add a class directly to the Project named MyRoleProvider.cs, write following code:*

**

*RoleProvider is an abstract class so implement all the abstract methods:*

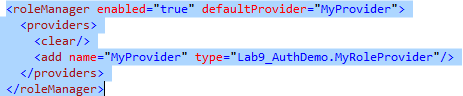
**

*After implementation go to GetRolesForUser Method and implement the following code:*

**

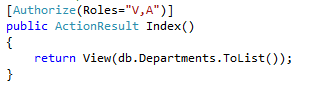
***Making modification in Configuration:***

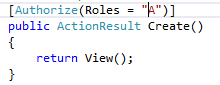
*Go to web.config, under <system.web> write the following code:*

**

***Implementing Roles in required Controllers/action methods:***

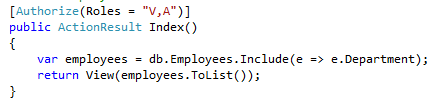
*We need to work on the links which needs to Authorize. Go to DepartmentsController and perform the following task:*

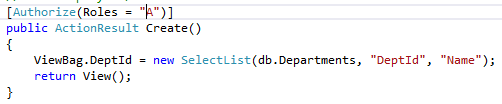
**

**

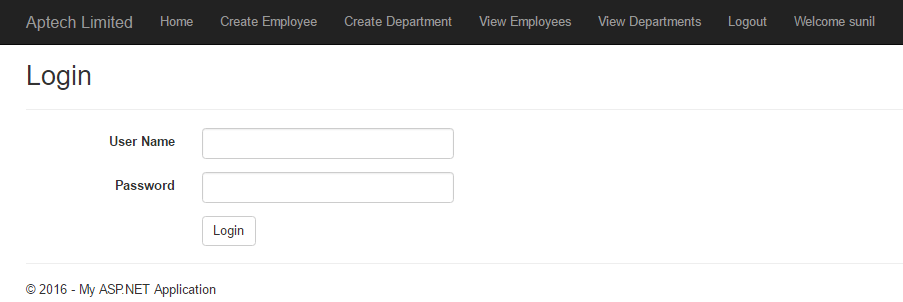
*[Authorize(Roles=”V,A”)] means Users having RoleId V and A can access the link. [Authorize(Roles=”A”)] means Users having RoleId A can access the link.*

*Go to EmployeesController and perform the following task:*

**

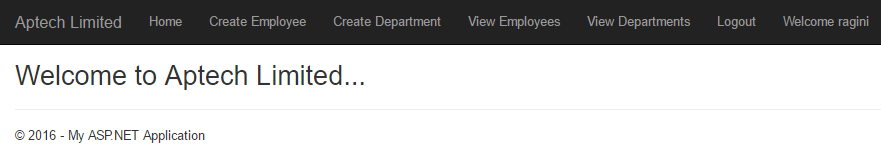
**

*Now when we access the project using the user having V roleid. It will ask to login:*

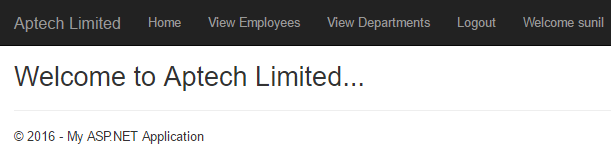
**

*Now we want to show the links as per the user login:*

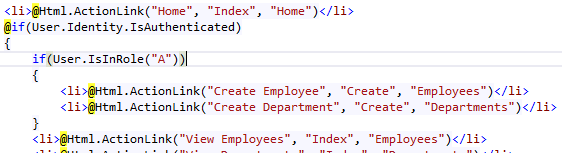
*If the admin logs in output should be like:*

**

*And if the viewer logs in output should be like:*

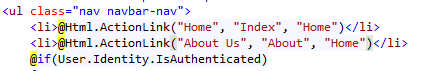
**

*To perform the above we need to make the following code changes in layout.cshtml*

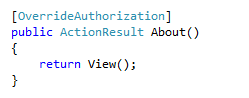
**

*Now we shall implement* ***Override authorization*** *filter*

*Add an action Link in \_Layout.cshtml*

**

*Add Action Method in HomeController*

**