**Step-By-Step**

**Creating MVC Application**

* File [Menu] 🡪 New 🡪 Project 🡪Select Asp.Net Web Application.
* Change the name to ‘WebApp’.
* Browse the Location to Desktop/Custom Token.
* Select ‘EMPTY’ Template and check ‘MVC’.

**Creating a Controller**

* Login, Registaration, Forgot Password Screen
* Right Click on the Controller🡪 Add Empty MVC Controller🡪 Change name to ‘LoginController'.
* Rename the Default ‘Index’ in ActionResult to ‘LoginPage’.
* RightClick on the *LoginPage* and create a View. (Without Selecting any Partial View and Layout)
* Create new ActionResult for RegistrationPage & ForgotPasswordPage also. And create view without Selecting any Partial View and Layout.
* Create a folder 'Content', and add files (css, js, etc) for Login Screen.
* Button Clicks : Ajax Calls
* Place ajax calls on the Login Button Click
* Place the Code in Login Controller.
* Add the HTTP Redirection along with the key value in web.config
* Add all the related Design code. (Test the Login and Check)
* Create a Partial View.
* In View, Create a Folder ‘PartialViews’.
* Right Click on the ‘PartialViews’ Folder, Add View.
* Enter a View name 'HeaderNavBar' and select ‘Create as a partial view’ checkbox.
* Create another View in 'PartialViews' with the name 'FooterNavBar'.
* Create a Folder ‘Shared’ in *Views*.
* Right Click on the ‘Shared’ Folder, Add View with name ' \_MetaTags' (Without selecting any Checkbox, just empty).
* Creating Home Controller.
* Right Click on the Controller🡪 Add Empty MVC Controller🡪 Change name to ‘HomeController'.
* Create a View for ‘HomeController’. (Without selecting any Checkbox, just empty)
* Rename the Default ‘Index’ in ActionResult to ‘Home’.
* Add a Layout dynamically in ‘HomeController’ View, and browse the file or place as null.
* In Model folder, Create a Sub Folder with name ’SharedEntities’.
* Create a class 'SE\_Users'.
* Place *partial* to the class *SE\_Users*.

**Creating Security Features**

* Creating CSRF Token
* Create a Class file with the name ‘CSRF\_AntiForgeryToken’ in App\_Start Folder.
* Write down the related Functions in it.
* Place '*@Html.AntiForgeryToken()'* in design code.
* Create a *JsonResult* Method with the name ‘CheckAunthentication‘ in *LoginController.*
* Above the *LoginController,* place HttpPost and the CSRF token Class name (ie., CSRF\_AntiForgeryToken)
* Creating ContentSecurityPolicy (CSP)
* Create a Class file with the name ‘ContentSecurityPolicy’ in App\_Start Folder.
* Write down the related Functions in it.
* Place Key Value in AppSettings (Web.Config).
* Place [ContentSecurityPolicy] Method at CheckAuthenication, LoginPage, RegistrationPage, ForgotPasswordPage.
* Creating XSS Protection
* Create a Class file with the name ‘XssProtection’ in App\_Start Folder.
* Write down the related Functions in it.
* Place Strict-Transport-Security in web.config under <system.webServer> - <httpProtocol> -<customHeaders>.
* Creating FilterConfig class
* Create a class 'FilterConfig.cs' in App\_Start folder.
* Add filters CSP, Xss.
* In Global.asax , Register the FilterConfig.
* asdf

**Creating API Calls**

* Add References
* Right click on References, Select Assemblies -> Framework and select
* System.Net
* System.Net.Http
* Add Reference and bowser the file form the folder System.Web.Http.dll.
* Install System.Net.Http.Formatting
* Install the Libraries
* Update: Newtonsoft.Json (To Latest Version)
* Create a Folder ‘Helper’
* Create a class ‘ApiHelper’. – Place related code.
* Create a ‘GlobalVariables’. – Please related code.
* Add related key values Web.config.
* Create a class ‘OAuthHelper’.
* Create a class ‘SE\_AccessToken’ under ‘SharedEntities’.
* Pass API URL's

**Creating RSA**

* Create a Folder 'Helper'.
* Create a class 'RSAPattern' in *Helper* Folder. - Write down the related code in it.
* Place Public Key and Private Key in GlobalVariables
* Placed Public Key in Web.Config.
* Note:
* Public Key should be kept on the Client Side(i.e., MVC).
* Private Keyshould be kept on the Server Side (i.e., Web API).

**Exception Handling - Storing in txt Log**

* Creating a class 'ExceptionLogging' in *Helper* Folder.
* Write the related code in it.
* Create a Folder 'ErrorLog'.
* Add Key Value in web.config.
* Place the code in any catch block to store in txt file.

**Error Page**

* Create a Controller 'ErrorController'.
* Create a ActionResult 'Error404' and also create a view.
* Design the View page.
* Add errorPage.css in Content Folder.