LAB WORKSHEET 9

Student Name: Sanjeev Kumar Pandey	**UID: 20MCA1096**
Branch: MCA	**Section/Group:20MCA1/A**
Semester: 4th	**Date of Performance: 26/04/2022**
Subject Name: AWAD	**Subject Code: 20CAA-755**

1. Task to be done:

1. Design User Registration Form Using MVC.

2. Steps for experiment/practical:

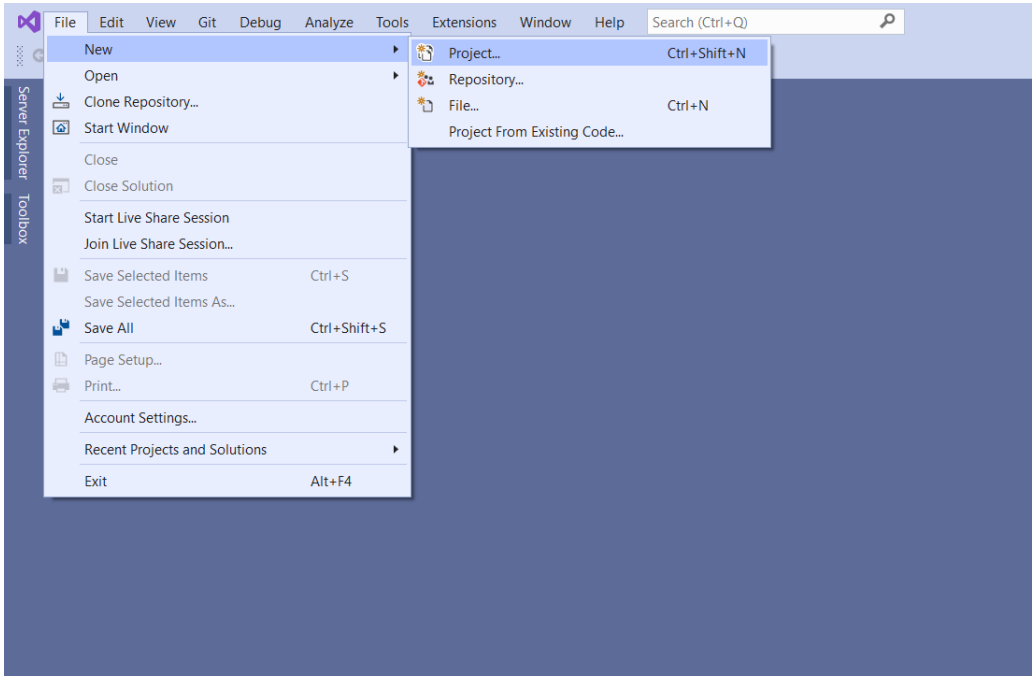
Task 1:

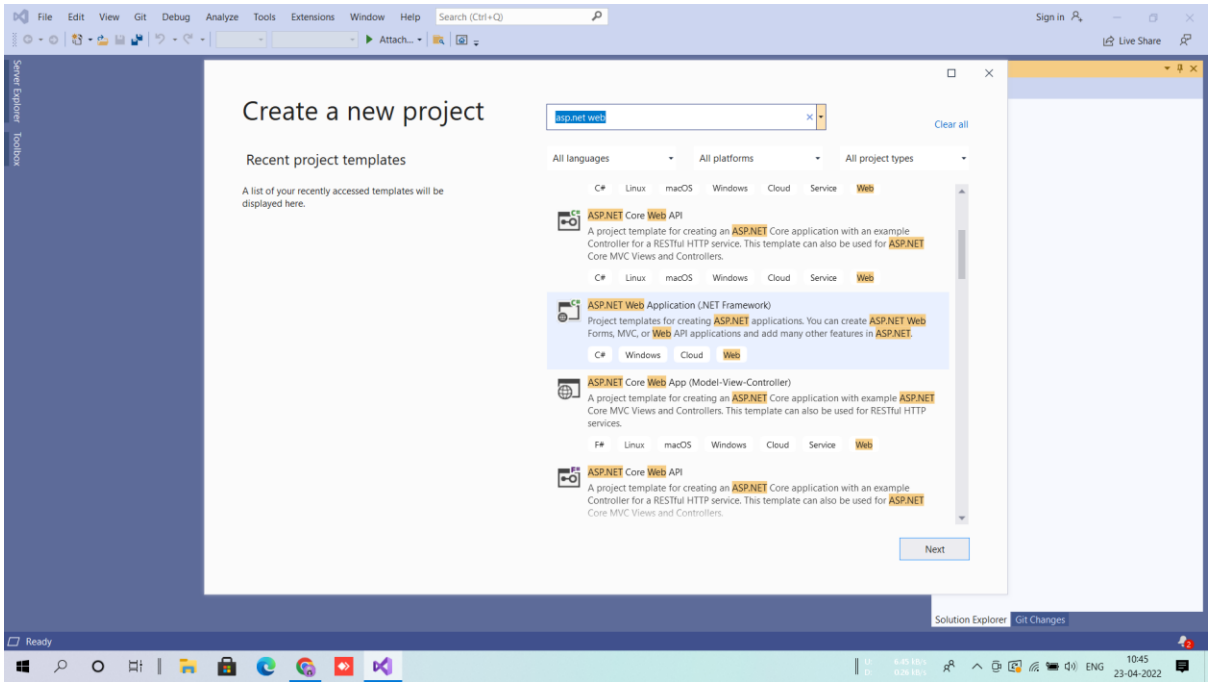
Step 1:

Download the latest Visual Studio version or Use Visual Studio 2013 or above. (I am Using Visual Studio 2019)

Step 2:

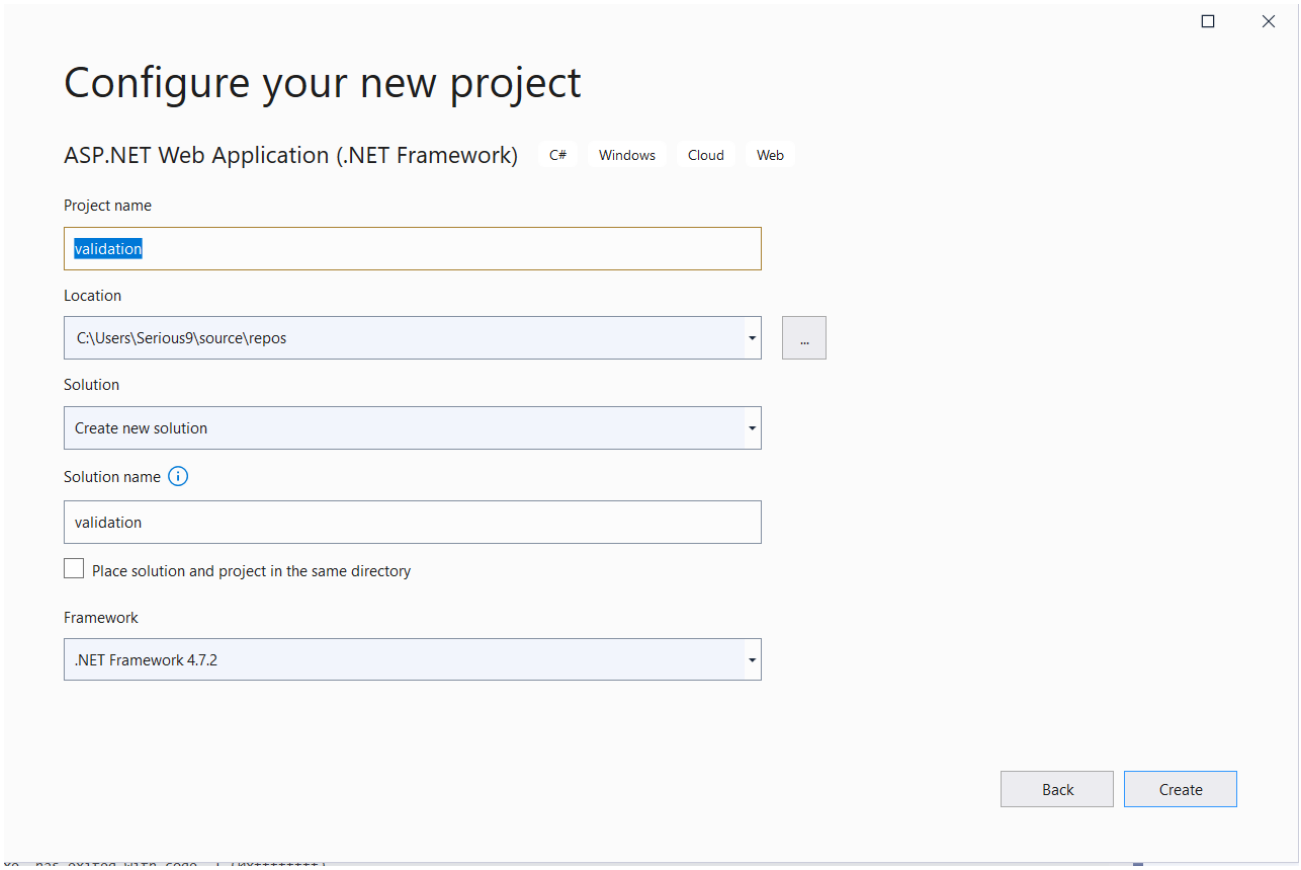
Open Visual Studio 2017 and click on File Menu ->New ->Project





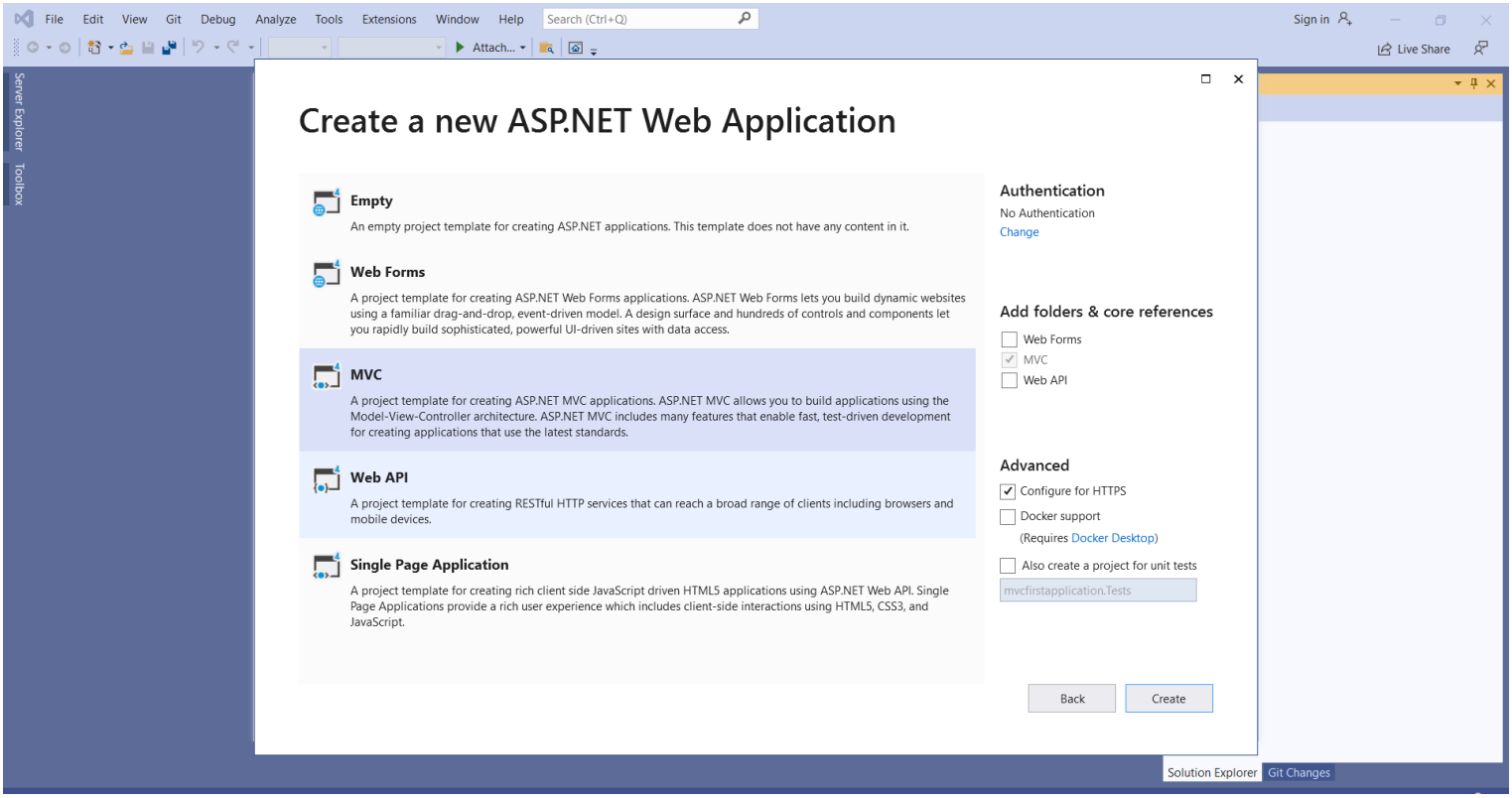
Step 3:

In the left side template expand Visual C# and select Web -> ASP.NET Web Application. Enter the name of your application. and if you want to change project location then click on browse and set location.



Step 4:

Select MVC and Check checkbox MVC for adding folder and core reference. (Note-:on Right side Authentication-No Authentication) Click on OK

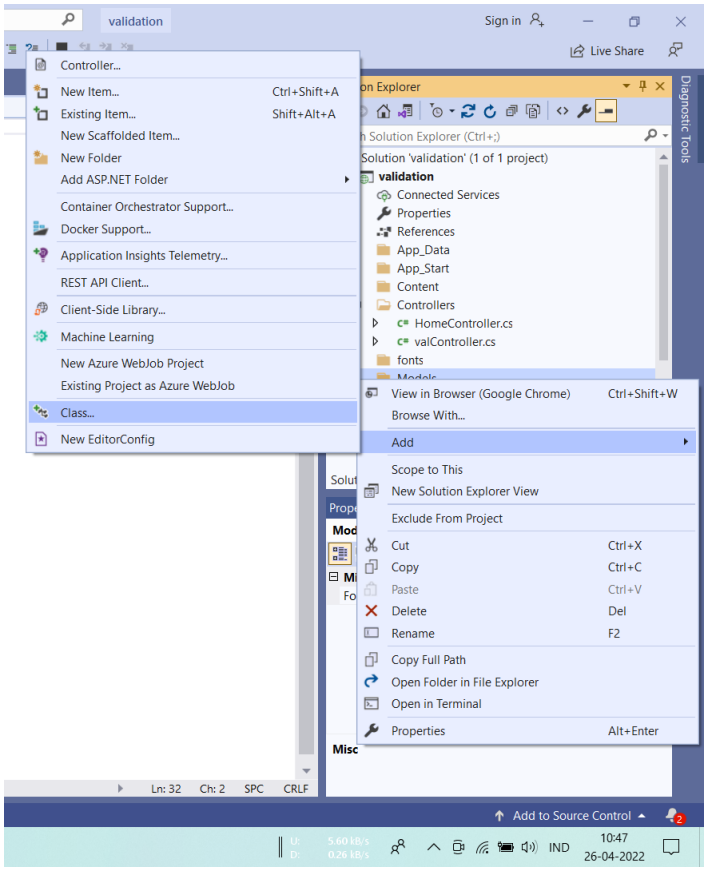


Step 5:

When you click ok, Visual Studio automatically creates the MVC Application. On the right side you will see the following folders:

App_Data : Used for Database File Stored
App_Start
Content: Where we store css files and any other client file.
Controller: Provides action
Fonts: Required for Text or any other action
Models: Domain classes are here
Views: In the View name after controller, it is used for displaying view to the user.

Firstly we working onModel: Right Click on Model Folder -> Add -> class -> Name it Validation



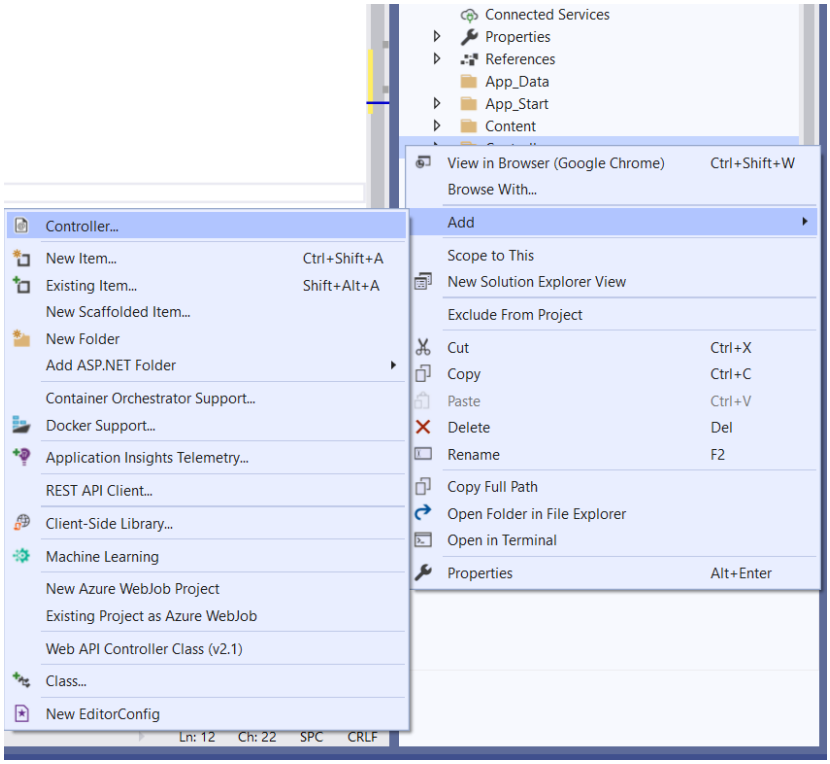
Step 6:

Add the below code in Model

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

namespace validation.Models
{
public class Validation
{
[Required]
[Display(Name = "Username")]
[StringLength(15, ErrorMessage = "Name should not exceed 15 characters")]
public string name { get; set; }

[Required]
```

```
[RegularExpression("^[a-z0-9_\\+-]+(\\.[a-z0-9_\\+-]+)*@[a-z0-9-]+(\\.[a-z0-9]+)*\\.([a-z]{2,4})$", ErrorMessage = "Email is not valid")]
public string Email { get; set; }

[Required]
[Range(18, 30, ErrorMessage = "Age between 18-30")]
public int Age { get; set; }

[Required(ErrorMessage ="Password is Required")]
public string Password { get; set; }

[Required]
[Display(Name ="Confirm Password")]
[Compare("Password",ErrorMessage = "Confirm Password do not match Password")]
public string Cp {get;set;}
}
}
```

Step 7:We want to work with Controller
Right click on controller folder -> add -> Controller -> Name it [valController]



Step 8:

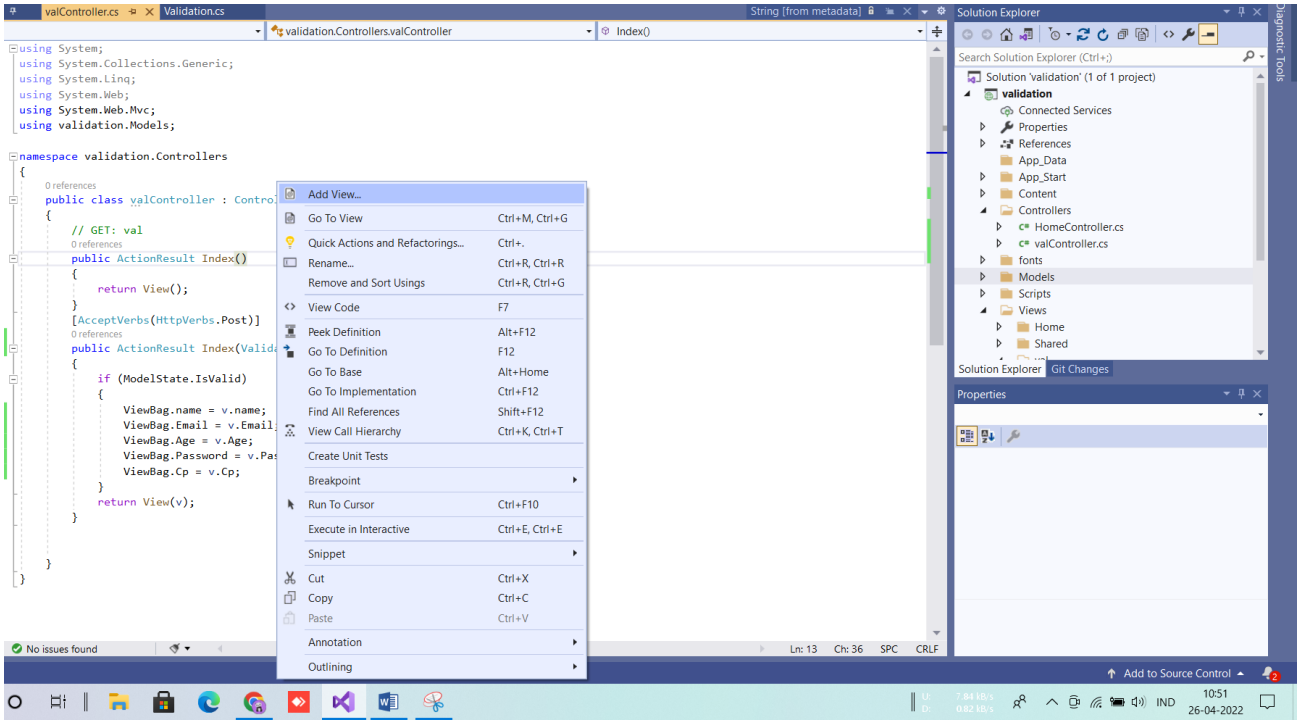
Modify your Controller code with the following code

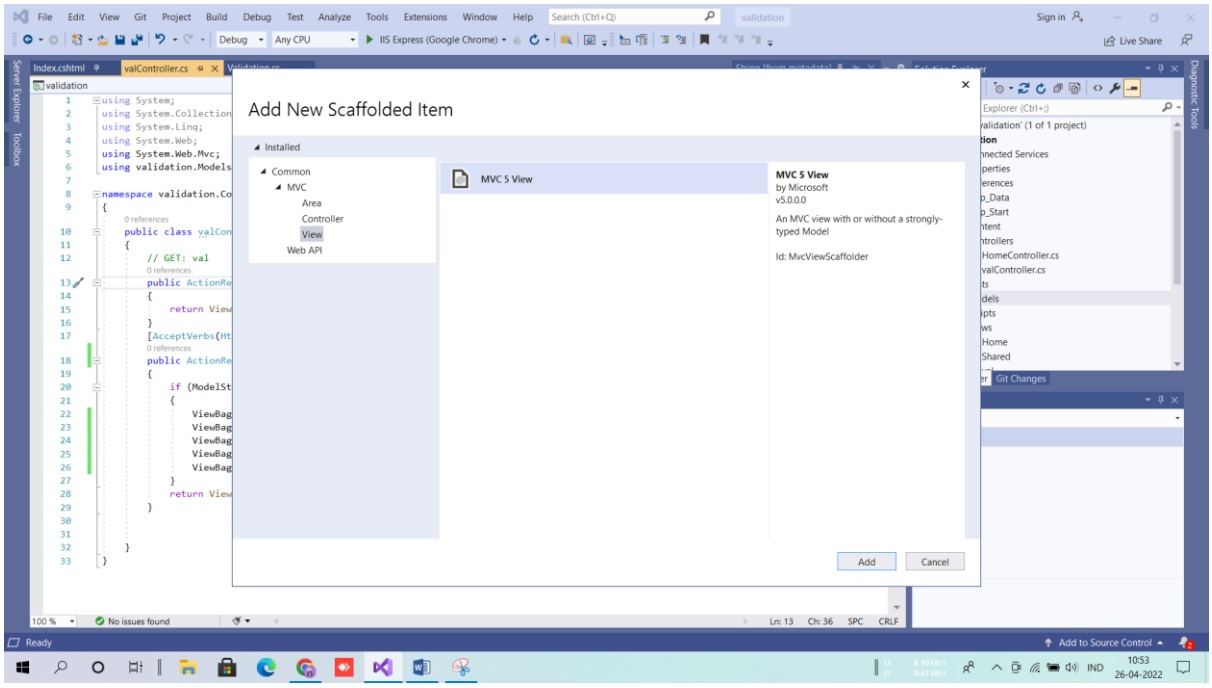
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using validation.Models;

namespace validation.Controllers
{
public class valController : Controller
{
// GET: val
public ActionResult Index()
{
return View();
}
[AcceptVerbs(HttpVerbs.Post)]
public ActionResult Index(Validation v)
{
if (ModelState.IsValid)
{
ViewBag.name = v.name;
ViewBag.Email = v.Email;
ViewBag.Age = v.Age;
ViewBag.Password = v.Password;
ViewBag.Cp = v.Cp;
}
return View(v);
}
}
}
```

Step 9:

Now we are working with View.

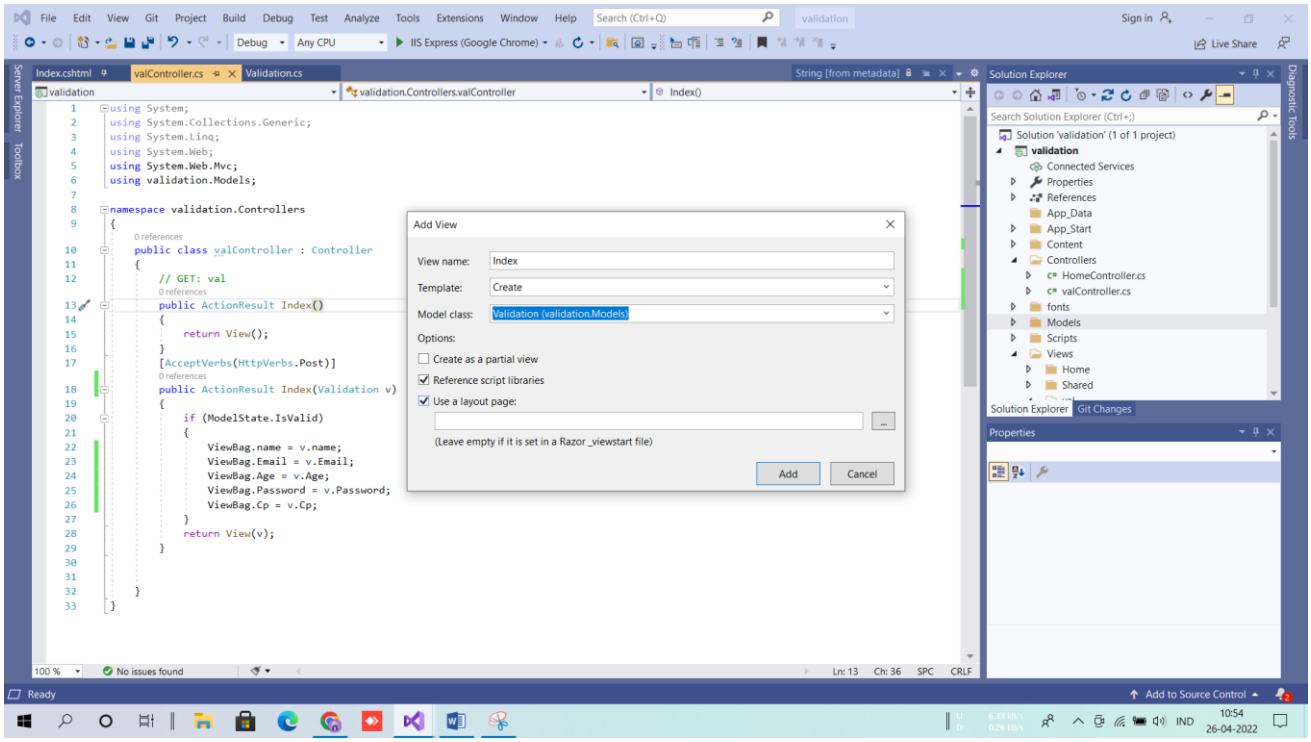
Go to controller then right click and then add view





Step 10:

Now add view name,choose template,choose model class and click on use a layout page and then click on add.



Step 11:

After clicking on add, view will be created with following code

```
@model validation.Models.Validation

@{
ViewBag.Title = "Index";
}

<h2>Index</h2>


@using (Html.BeginForm())
{
@Html.AntiForgeryToken()

<div class="form-horizontal" style="background-color:aqua;font-family:'Bell MT';font-size:large;">
<h1 style="text-align:center">Registration Page</h1>
<hr />
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
<div class="form-group">
```

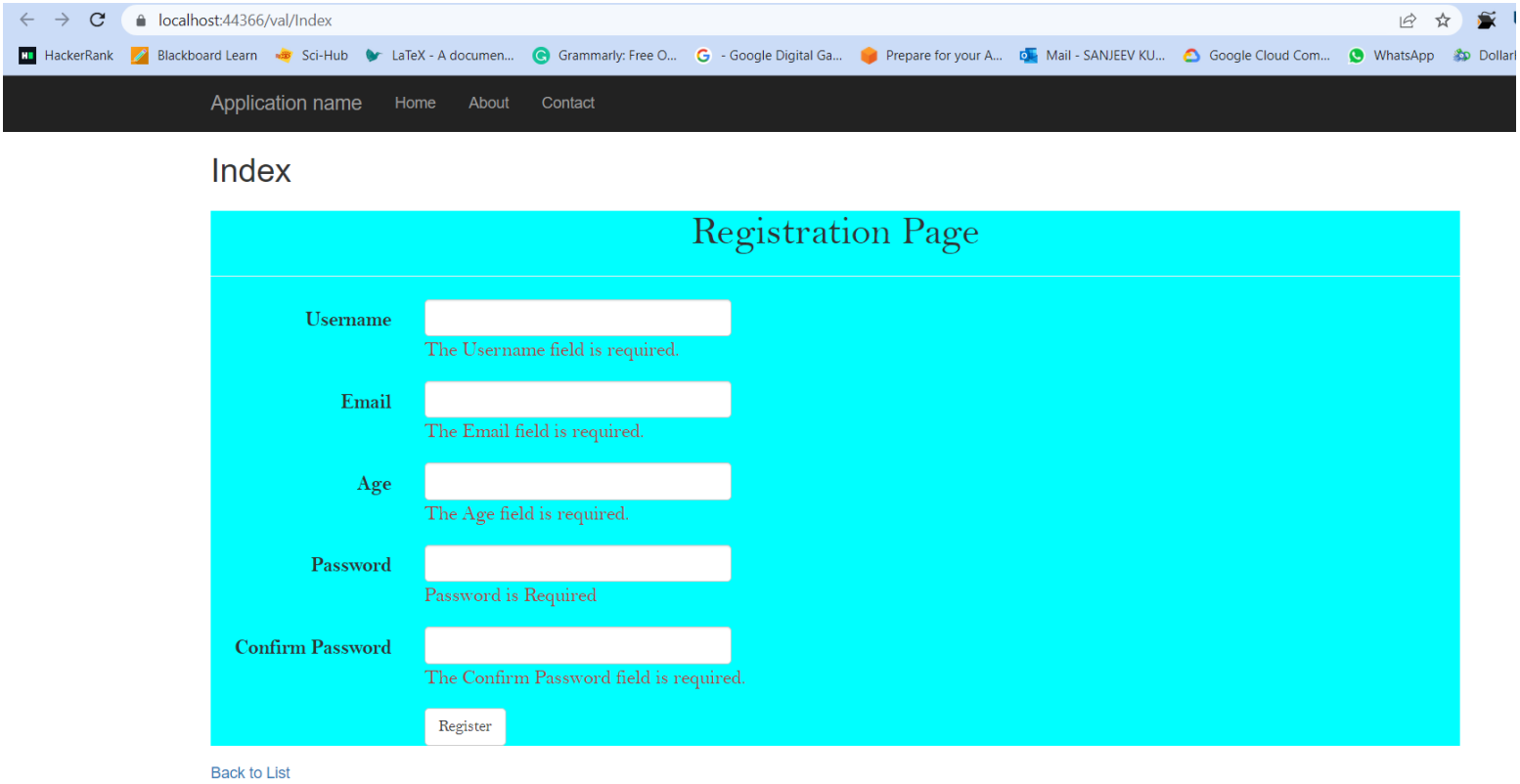
```
@Html.LabelFor(model => model.name, htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model => model.name, new { htmlAttributes = new { @class = "form-control" } })
@Html.ValidationMessageFor(model => model.name, "", new { @class = "text-danger" })
</div>
</div>

<div class="form-group">
@Html.LabelFor(model => model.Email, htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model => model.Email, new { htmlAttributes = new { @class = "form-control" } })
@Html.ValidationMessageFor(model => model.Email, "", new { @class = "text-danger" })
</div>
</div>

<div class="form-group">
@Html.LabelFor(model => model.Age, htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model => model.Age, new { htmlAttributes = new { @class = "form-control" } })
@Html.ValidationMessageFor(model => model.Age, "", new { @class = "text-danger" })
</div>
</div>

<div class="form-group">
@Html.LabelFor(model => model.Password, htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
```

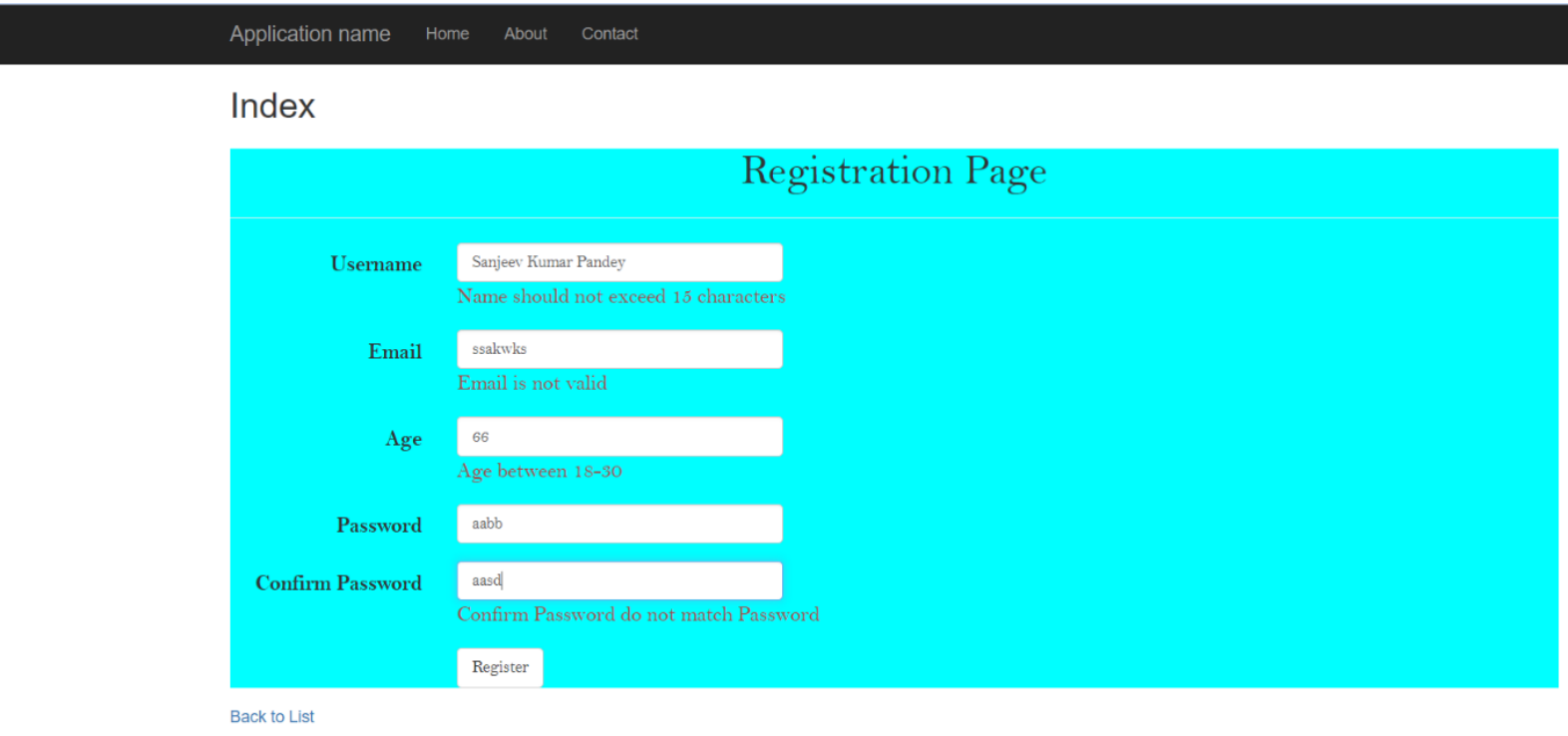
```
@Html.EditorFor(model => model.Password, new { htmlAttributes = new {
@class = "form-control" } })
@Html.ValidationMessageFor(model => model.Password, "", new { @class
= "text-danger" })
</div>
</div>

<div class="form-group">
@Html.LabelFor(model => model.Cp, htmlAttributes: new { @class =
"control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model => model.Cp, new { htmlAttributes = new {
@class = "form-control" } })
@Html.ValidationMessageFor(model => model.Cp, "", new { @class =
"text-danger" })
</div>
</div>

<div class="form-group">
<div class="col-md-offset-2 col-md-10">
<input type="submit" value="Register" class="btn btn-default" />
</div>
</div>
</div>
}

<div>
@Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
@Scripts.Render("~/bundles/jqueryval")
}
```

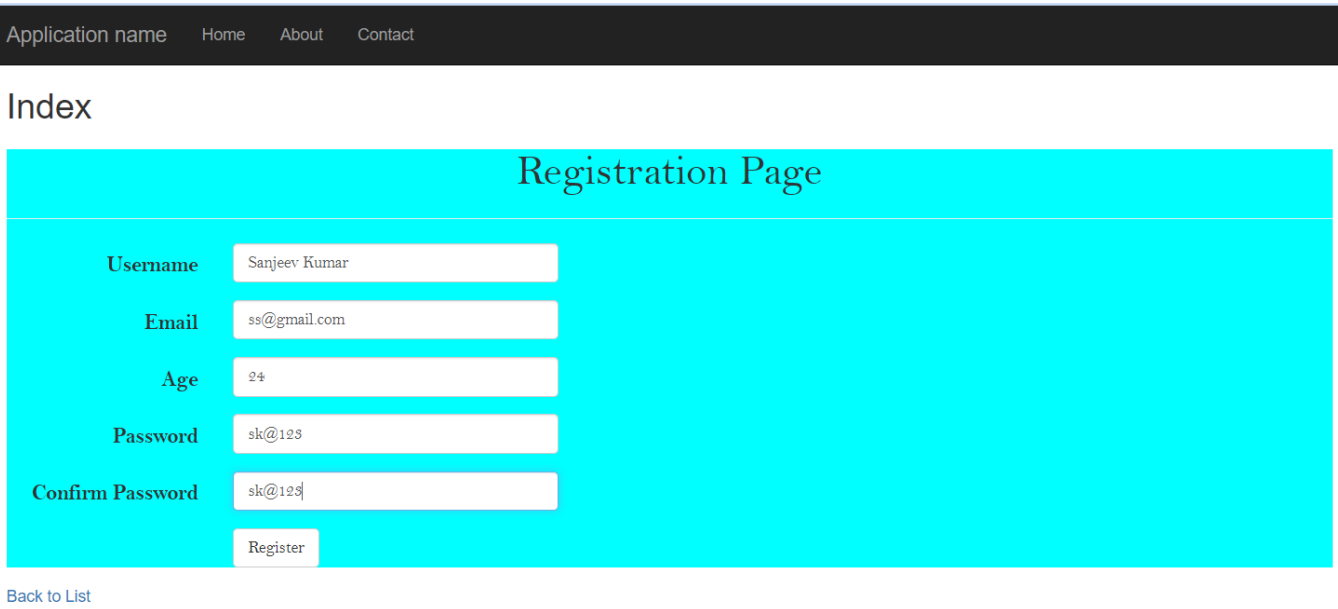
Step 12:

Run your Project by pressing the F5 key or clicking on Green Run Button. This view appears on screen.

OUTPUT:


localhost:44366/val/Index
HackerRank
Blackboard Learn
Sci-Hub
LaTeX - A documen...
Grammarly: Free O...
- Google Digital Ga...
Prepare for your A...
Mail - SANJEEV KU...
Google Cloud Com...
WhatsApp
Application name
Home
About
Contact
Index
Registration Page
Username
The Username field is required.
Email
The Email field is required.
Age
The Age field is required.
Password
Password is Required
Confirm Password
The Confirm Password field is required.
Register
Back to List


Application name
Home
About
Contact
Index
Registration Page
Username
Sanjeev Kumar Pandey
Name should not exceed 15 characters
Email
ssakwks
Email is not valid
Age
66
Age between 18-30
Password
aabb
Confirm Password
aasd
Confirm Password do not match Password
Register
Back to List



Learning outcomes (What I have learnt):

1. Learnt about mvc model
2. Learnt how to create registration page using an mvc model