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:

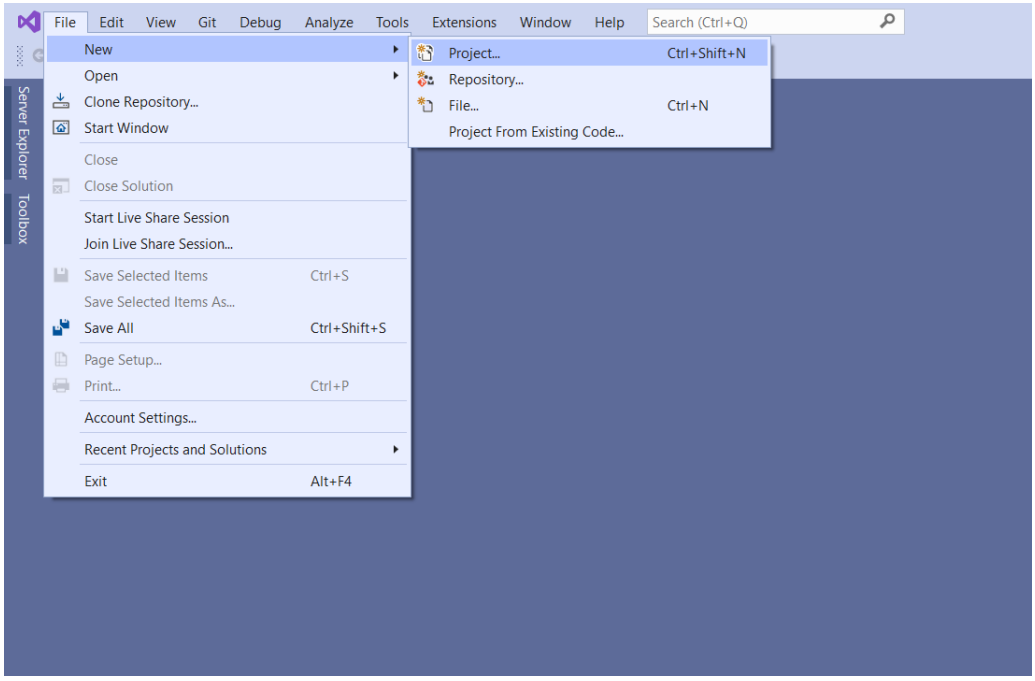
Task 1:

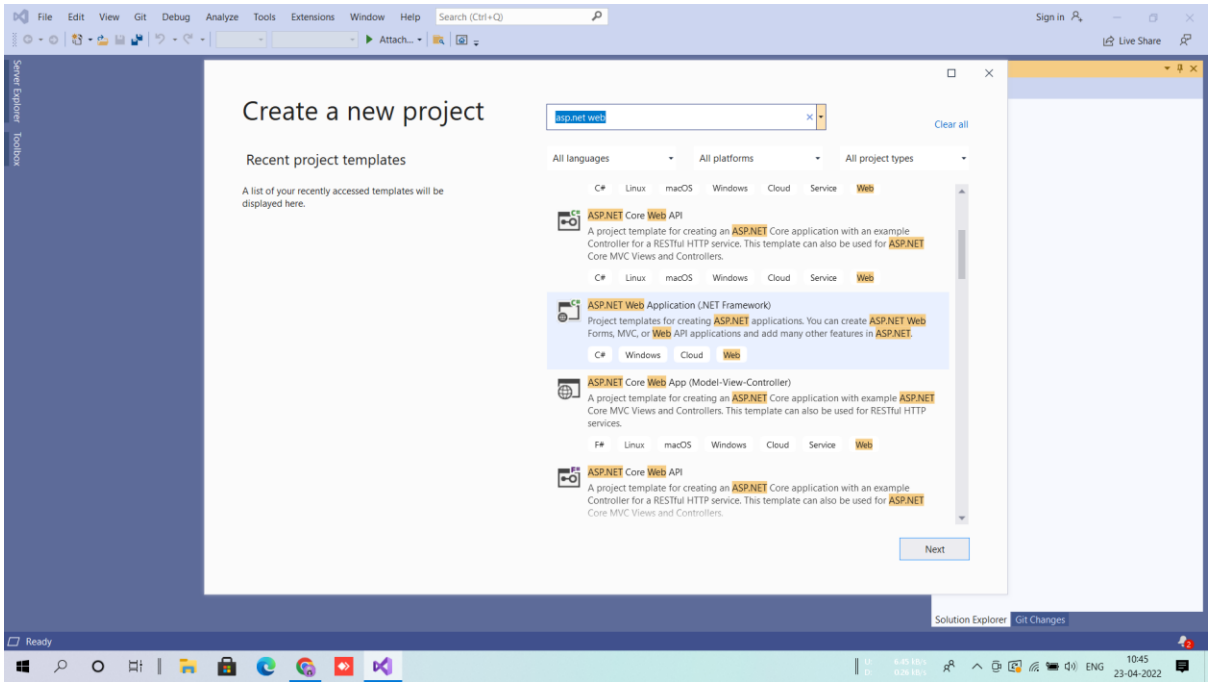
Step 1:

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

Step 2:

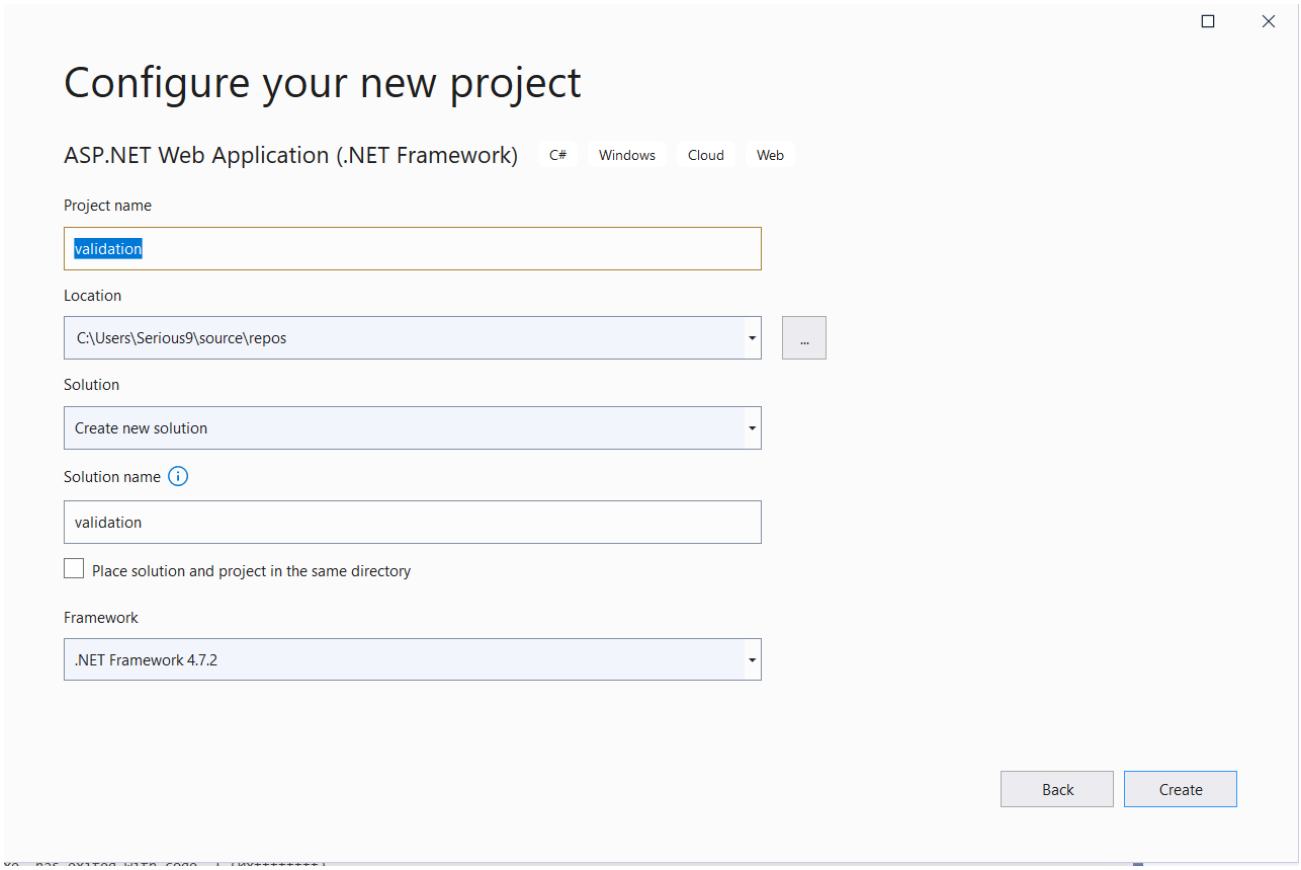
Open Visual Studio 2017 and click on File Menu ->New ->Project





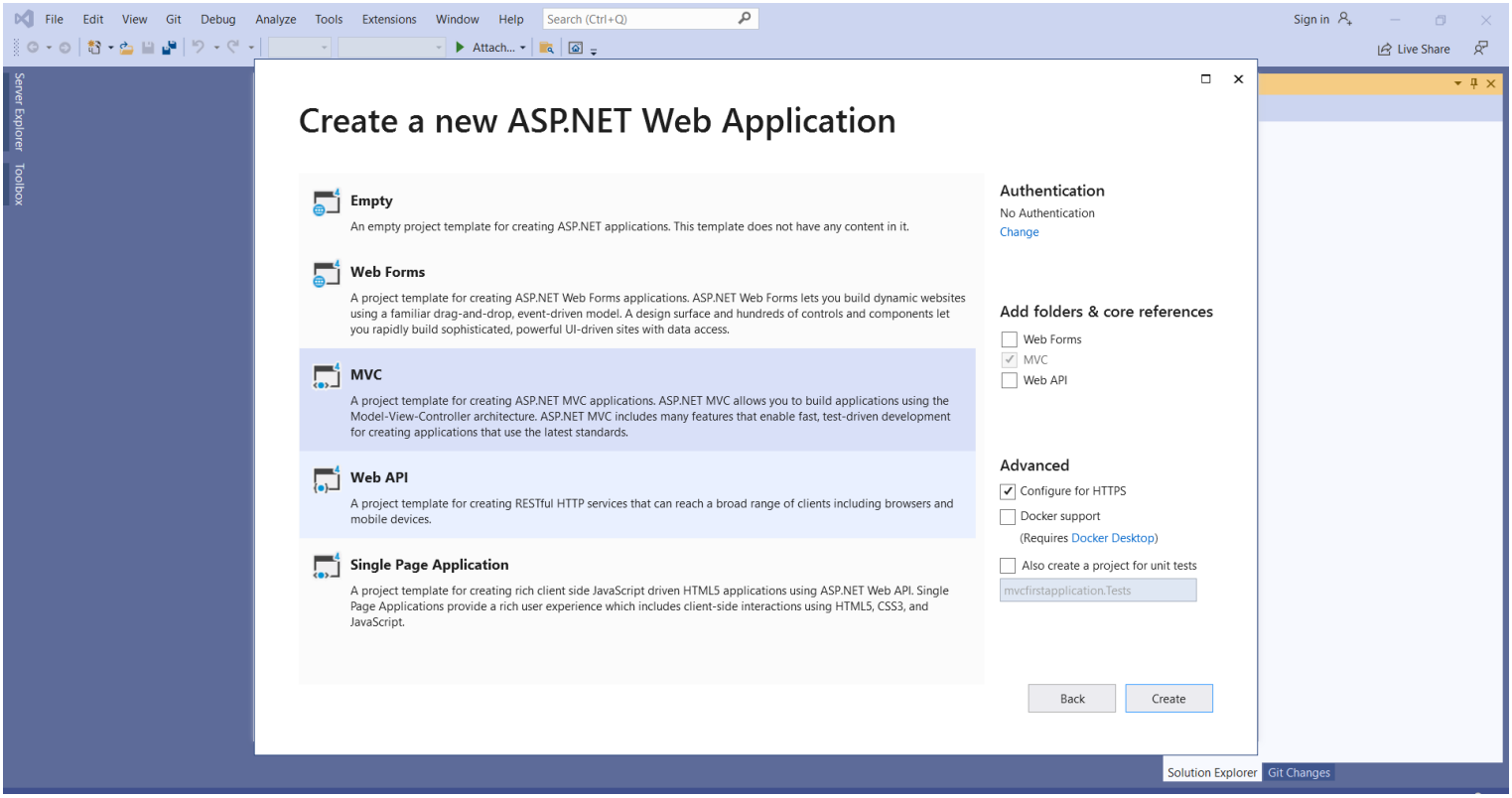
Step 3:

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

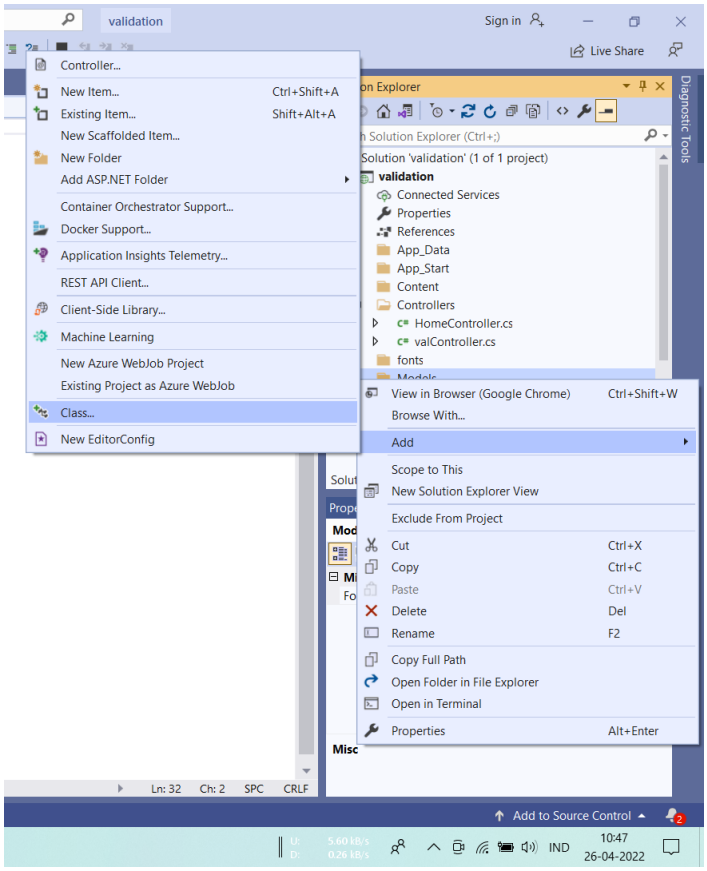


Step 5:

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



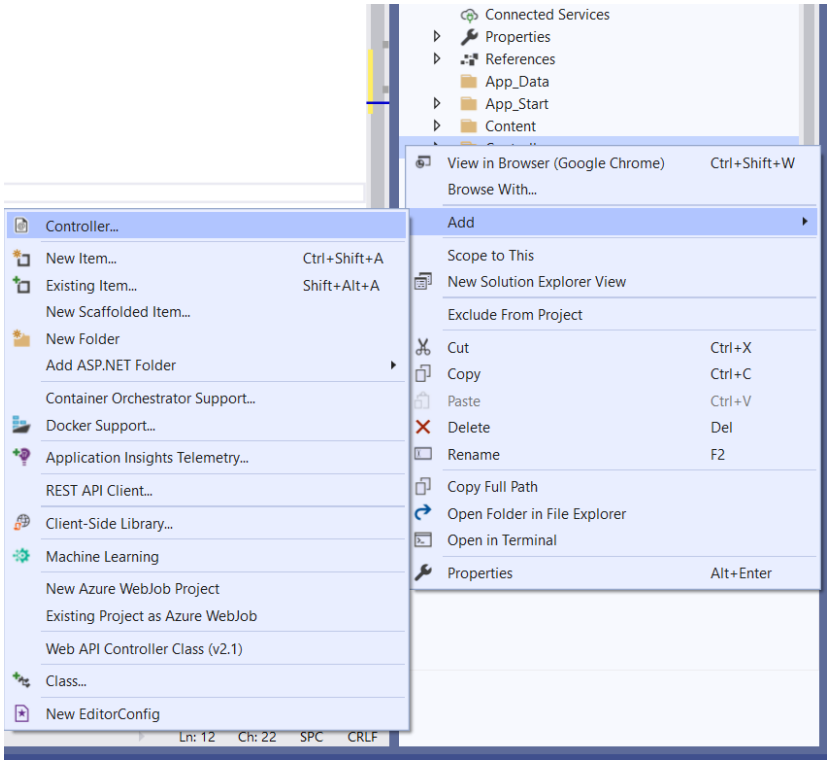
Step 6:

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:

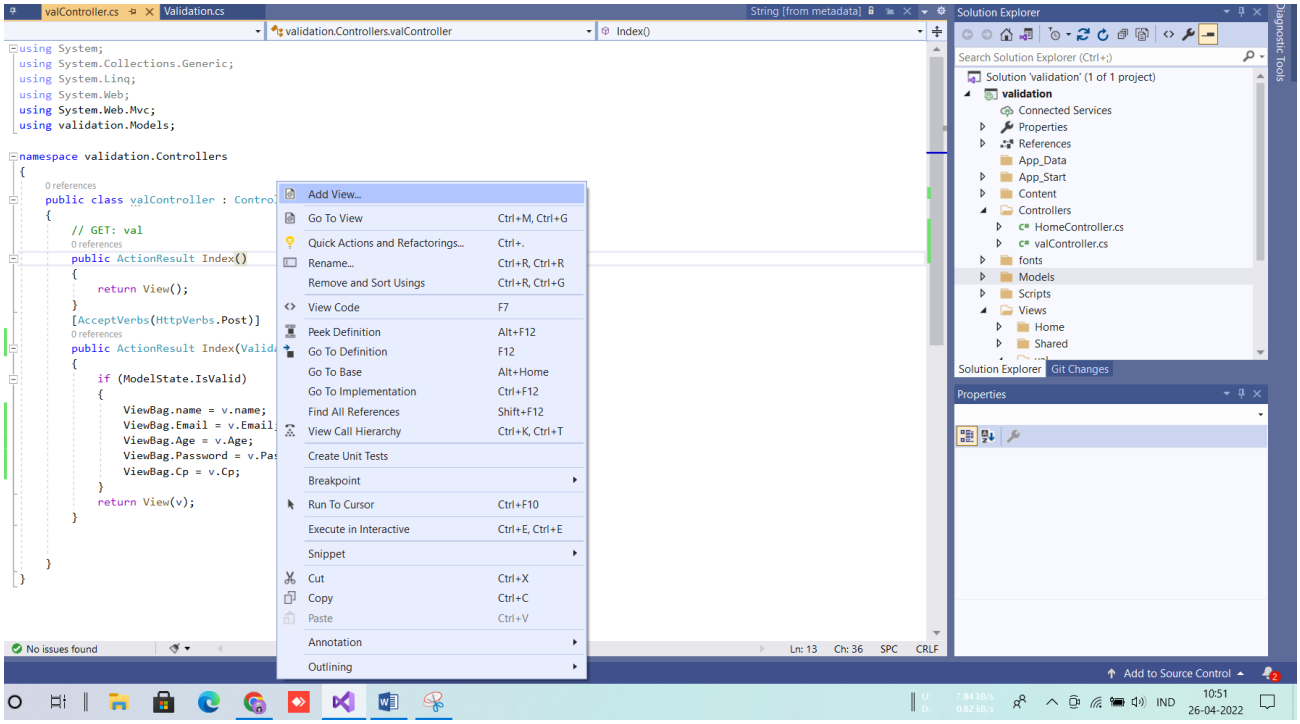
Modify your Controller code with the following code

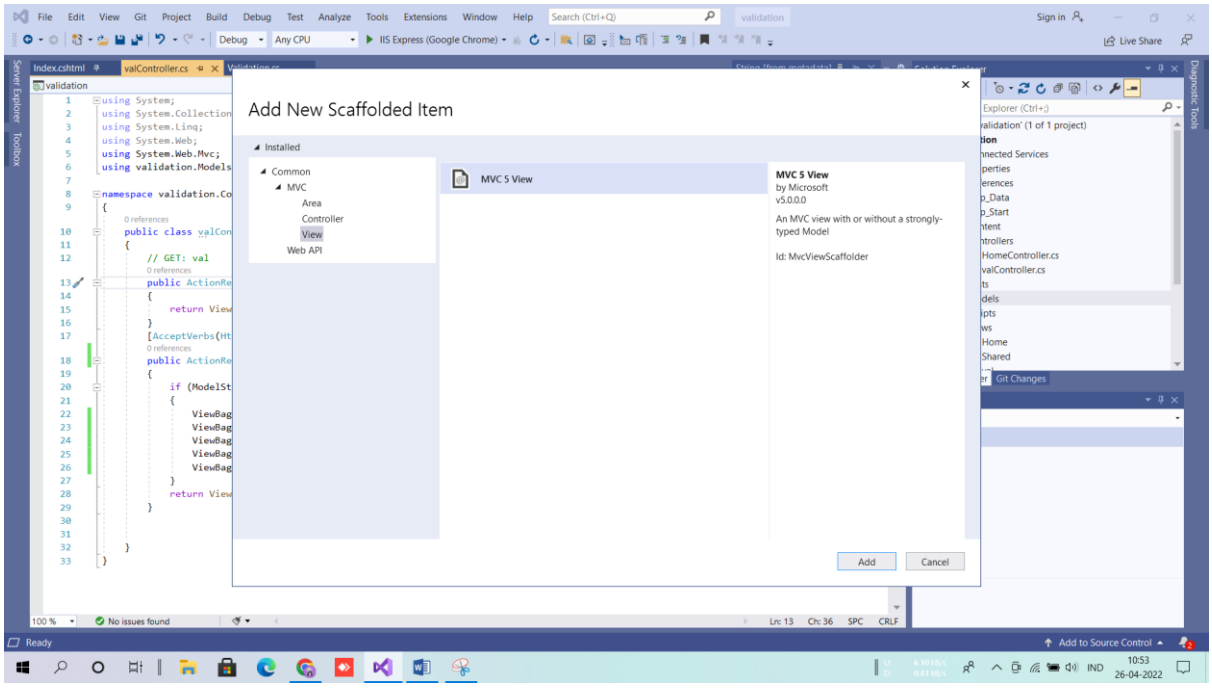
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using validation.Models;

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.

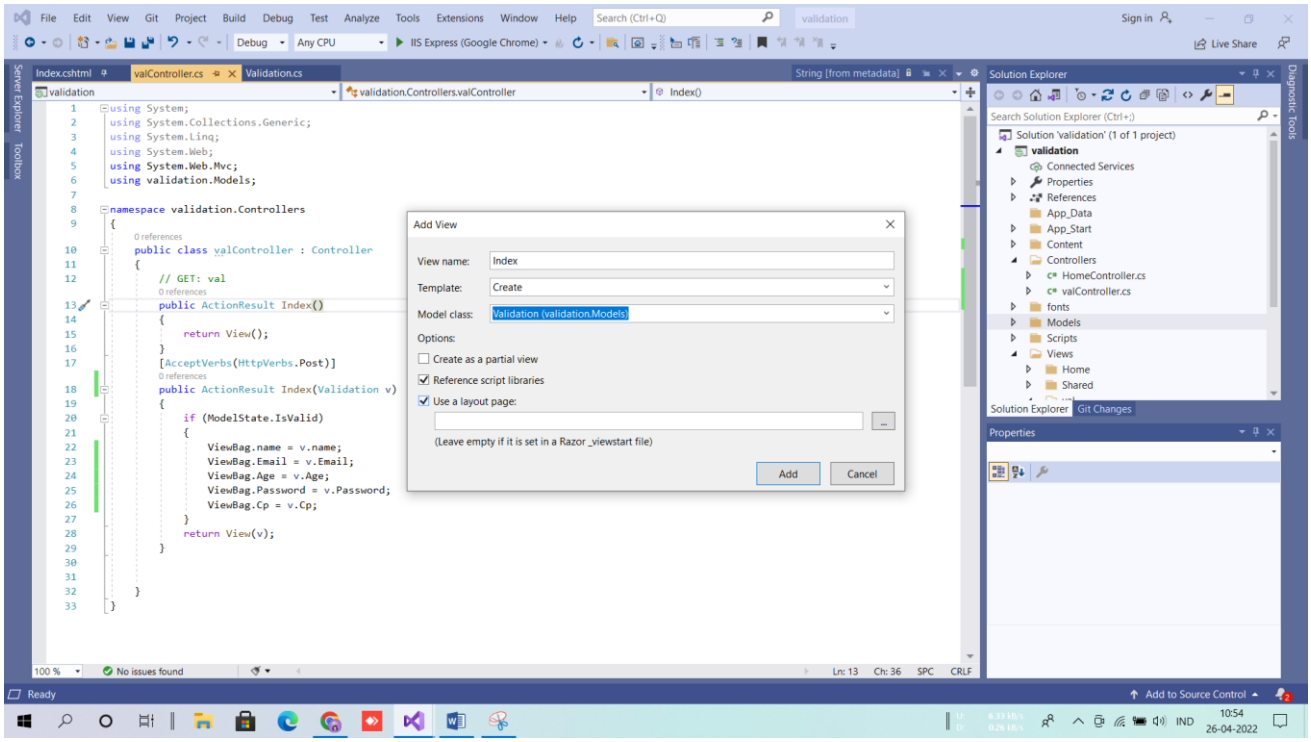
Go to controller then right click and then add view





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">
```

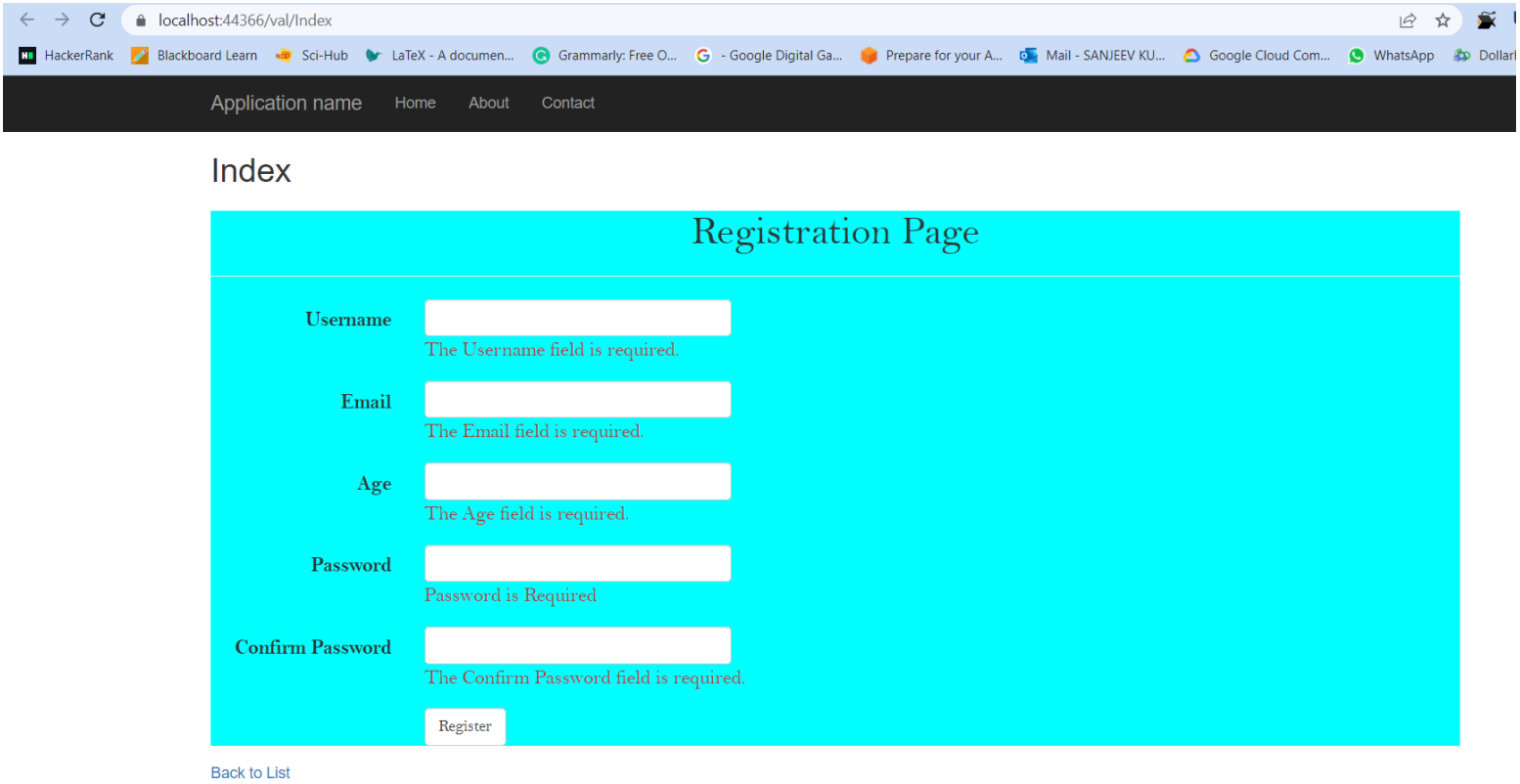
```
@Html.LabelFor(model => model.name, htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model => model.name, new { htmlAttributes = new { @class = "form-control" } })
@Html.ValidationMessageFor(model => model.name, "", new { @class = "text-danger" })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

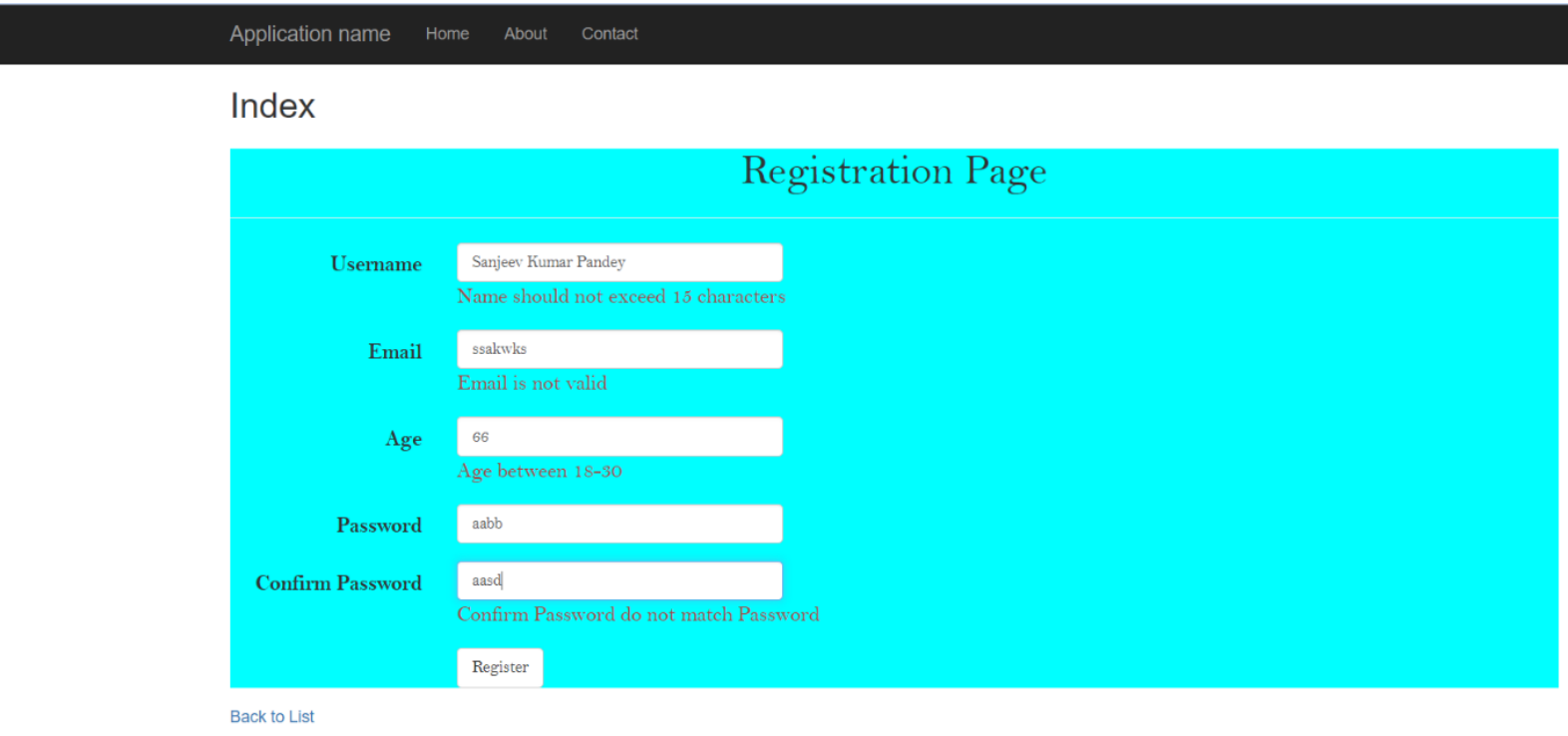
```
@Html.EditorFor(model => model.Password, new { htmlAttributes = new {
@class = "form-control" } })
@Html.ValidationMessageFor(model => model.Password, "", new { @class
= "text-danger" })
</div>
</div>

<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")
}
```

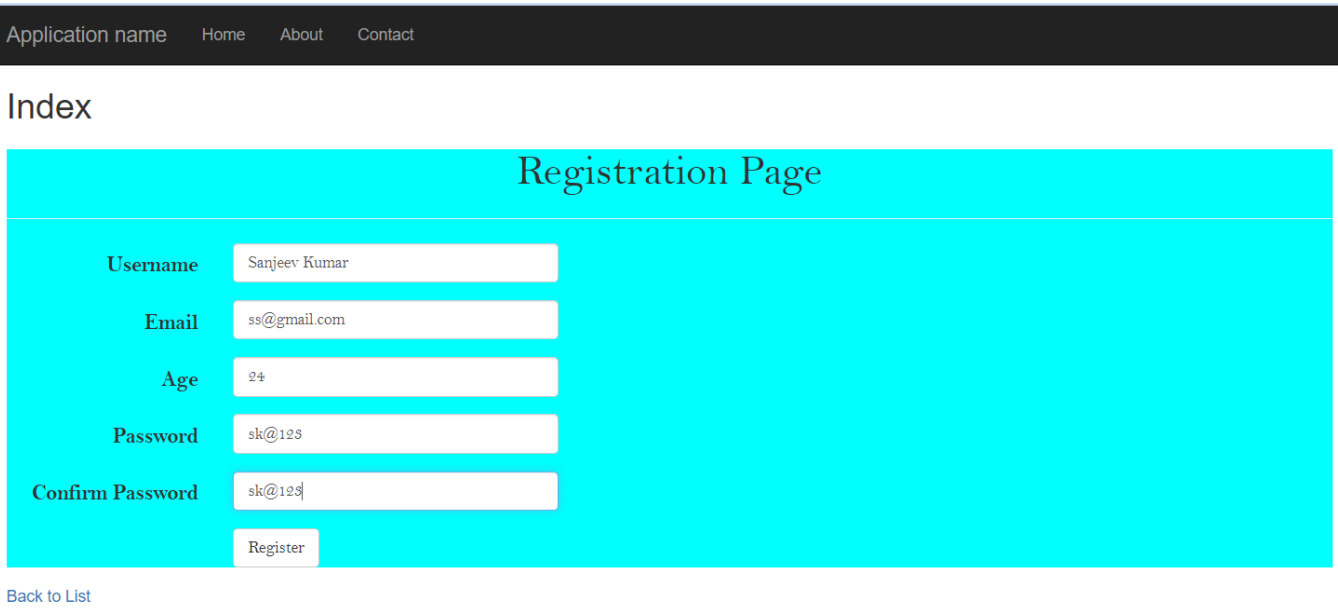
Step 12:

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

LAB WORKSHEET 10

Student Name: Sanjeev Kumar Pandey	**UID: 20MCA1096**
Branch: MCA	**Section/Group:20MCA1/A**
Semester: 4th	**Date of Performance: 30/04/2022**
Subject Name: AWAD	**Subject Code: 20CAA-755**

1. **Task to be done:**

1.Write the steps to host website in asp.net.

2. **Steps for experiment/practical:**

For hosting a website on IIS, follow these steps:

Step 1: IIS Installation.

IIS is a protocol server which is used to host a website on server. IIS stands for Internet Information services.

To install IIS start with windows start icon -> Control panel, Programs and Features , then click Turn Windows features on or off.

Figure 1

After that click windows features on or off, the Windows Features pop up window opens.

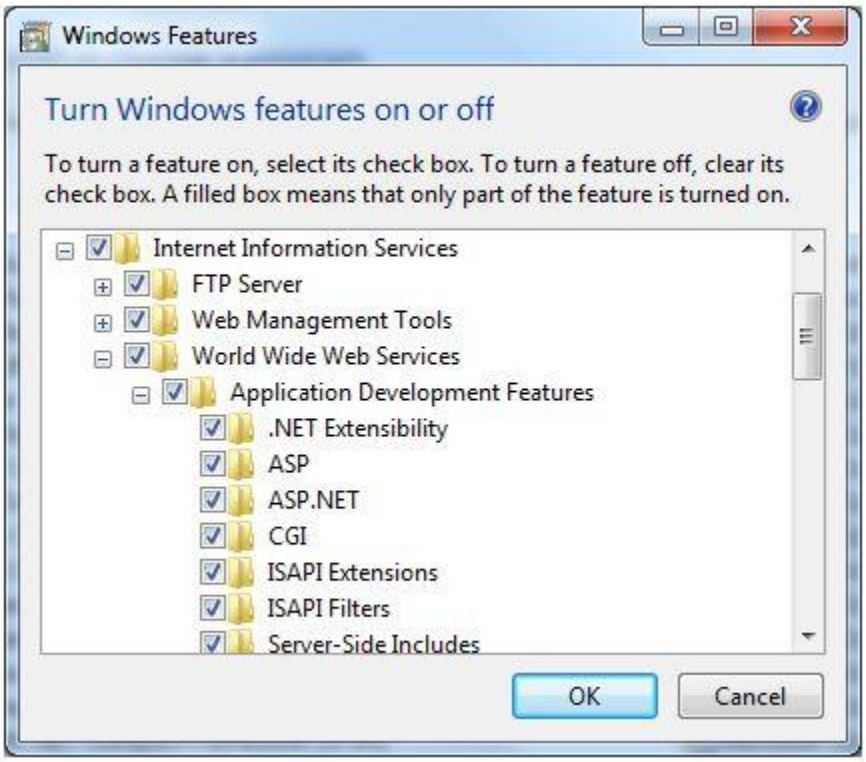


Figure 2: Windows features

Step 2: Enable ASP.NET Features on IIS

Expand the Application Development Features in that enable the ASP.NET application features such as:

1. .NET Extensibility
2. ASP
3. ASP.NET
4. CGI
5. ISAPI Extensions
6. ISAPI Filters
7. Server-Side Includes.

After enabling the ASP.NET features on IIS hit Ok. The IIS server Manager is successfully enabled in start control list.

Step 3: Install .NET Framework.

Now we have to install .NET framework version in our system because the aspx web page needs to run the platform of .net framework.

Figure 3: Setup.exe

Then the file is saved, click the saved exe file and the extract field process window will appear. The file extraction process completed and the following authorization window will appear, check I agree box then hit Install. Then the download and installation progress will appear in the same window.

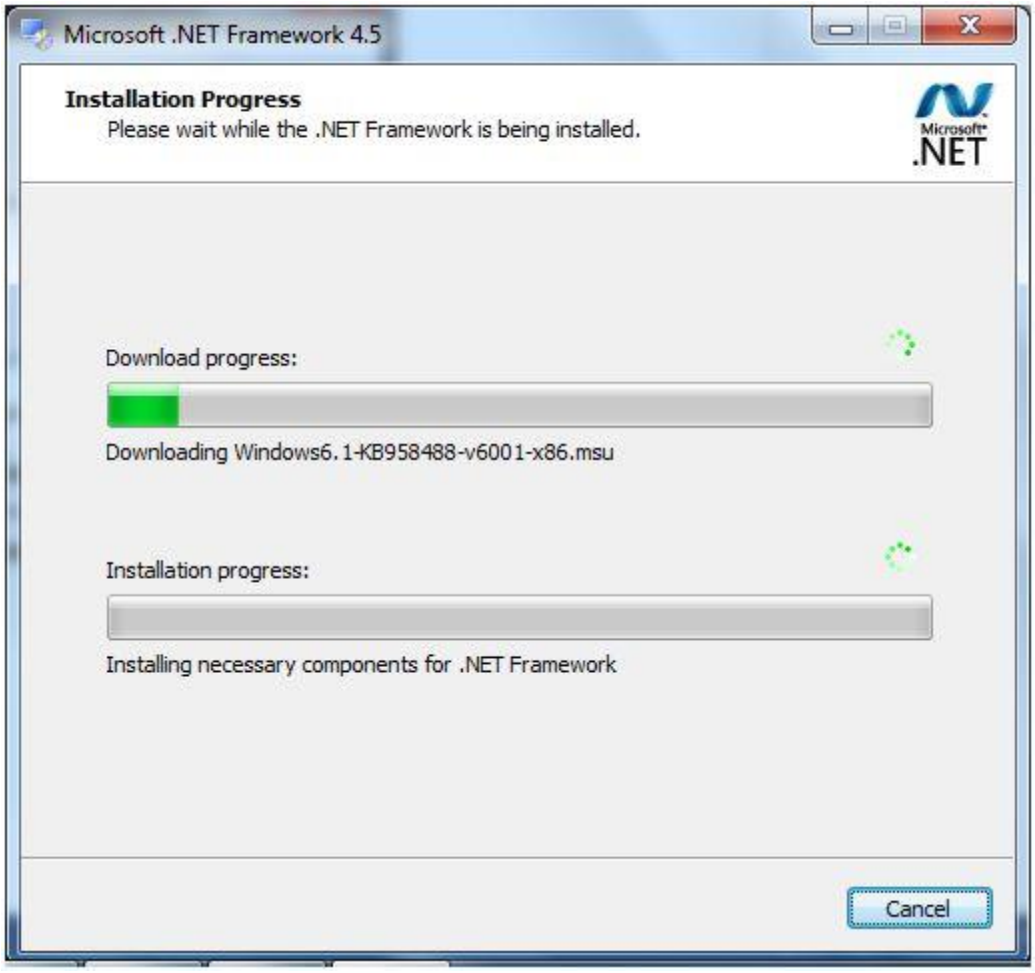


Figure 4: Installation Process

After the downloading and Installation progress are completed hit the finish button to complete the Installation. Now your IIS is ready to host ASP.NET web sites.

Step 4: Hosting a site.
Select the website folder which one you want to host it into your IIS.

Start IIS server, in that click site, right click on site and then choose add web site

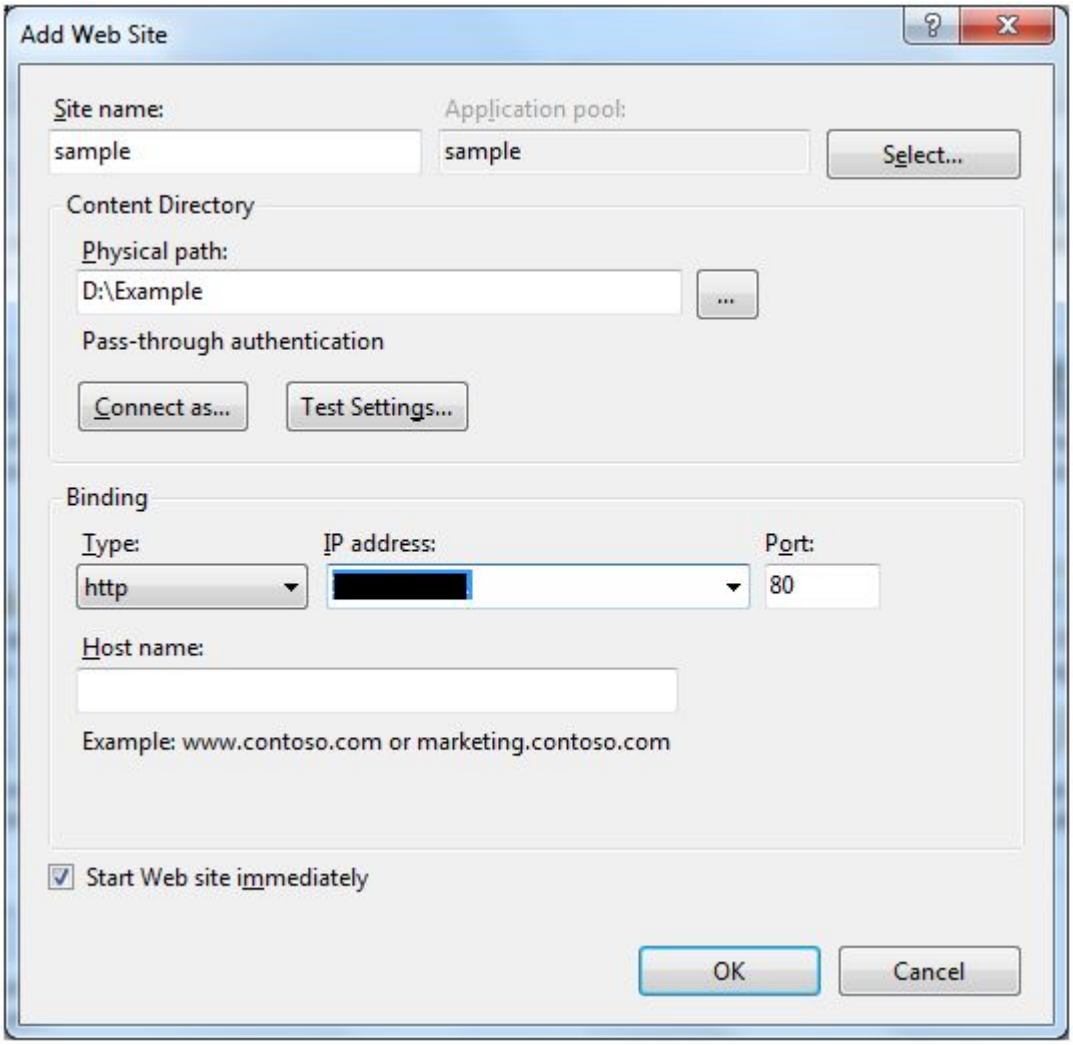


Figure 5: Add web site window

Add site name, path of that site folder, then assign the IP address or enter the host name of that particular site like www.example.com and then hit ok.

After the process is completed your site is placed under default web site under Sites.

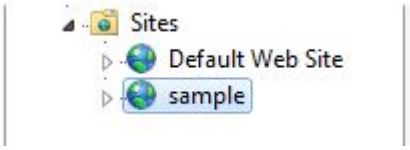


Figure 6

At the same time there is one application pool is automatically created while adding the web site.

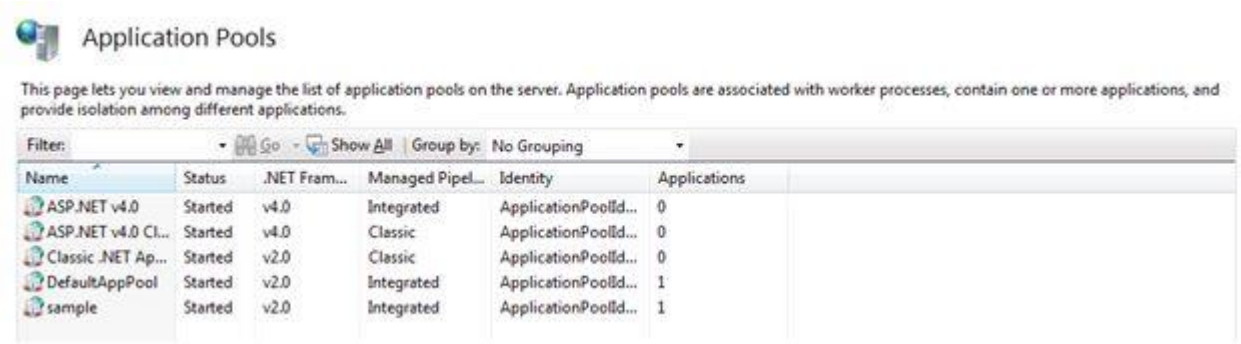


Figure 7: Application Pool

Here I have a simple web page like default.aspx in your website there is a database connectivity available which means then you have to follow the steps to add the database into application pool.

Step 5: Database Connectivity

In the application pool panel right click the application name like sample, then choose Advanced Settings option.

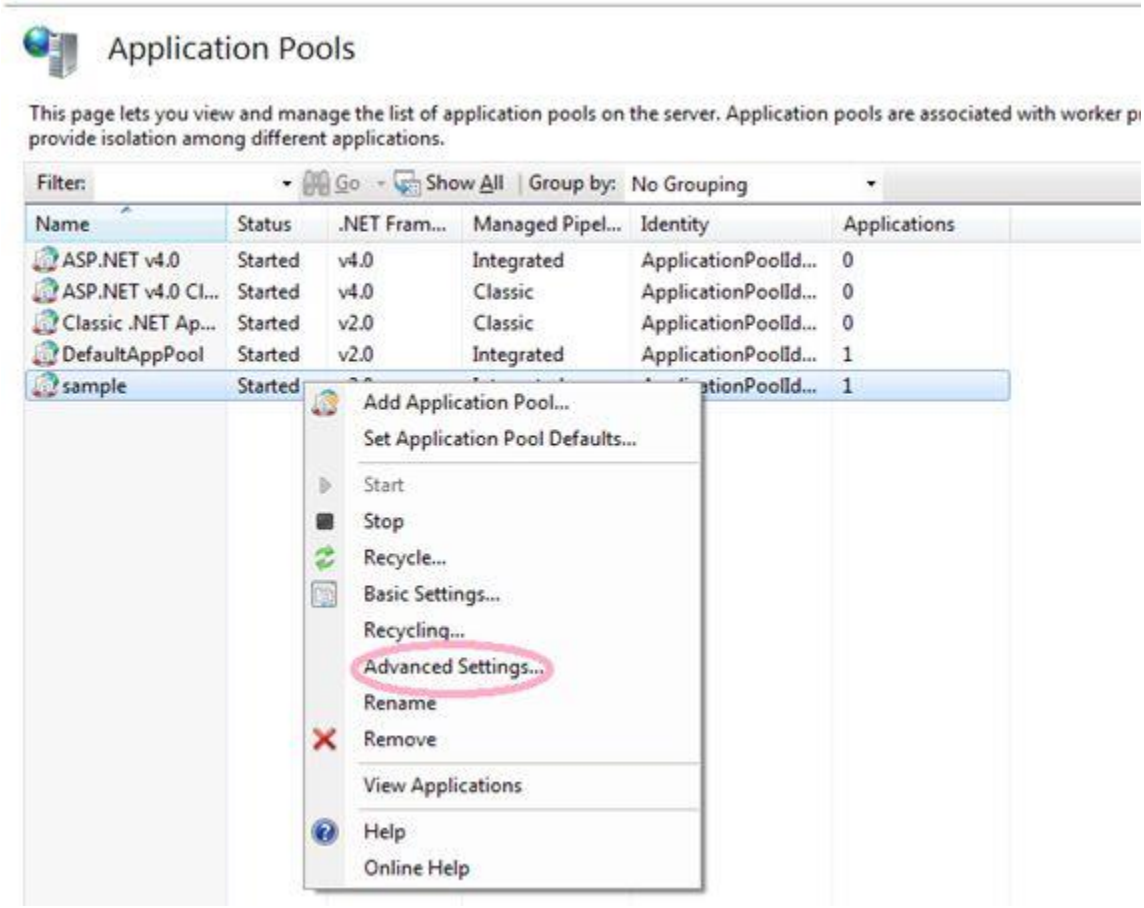


Figure 8: Advanced settings Option

The settings window will open in that window select .net framework version like v2.0. Based on our pages it automatically chooses the framework version. Then select managed pipeline mode is Integrated or Classic.

Then in the process model tab select Identity then choose the following:

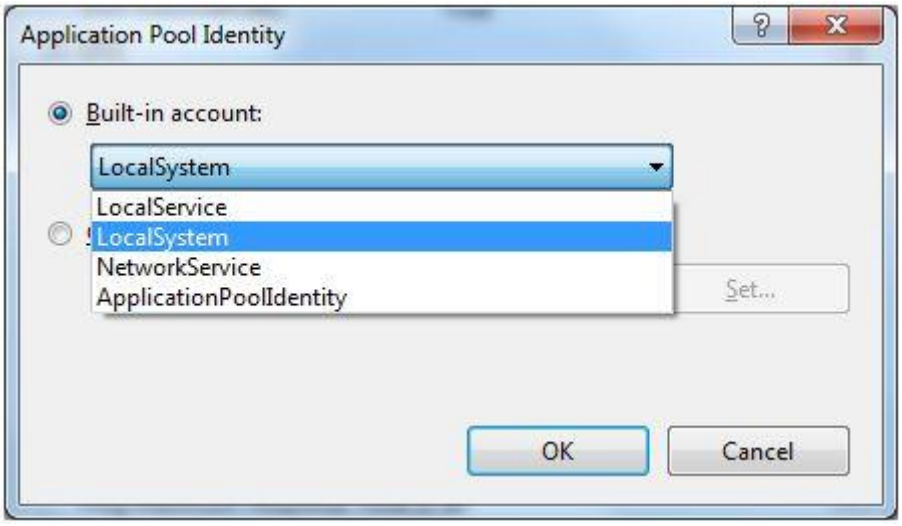


Figure 9: Pool Identity

Select LocalSystem because our database connectivity is placed in our system SQL Server so we need to choose LocalSystem then only our database is connected. Then hit OK to close the window.

Step 6: Conclusion.

In the Actions panel choose Browse link

Figure 10: Browse

Now your site will appear in the browser like the following figure:

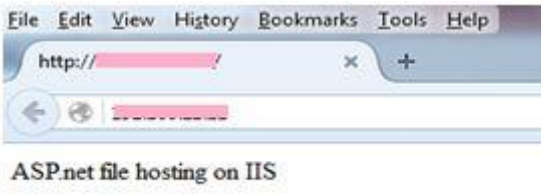


Figure 11: Browser

Now your application is ready to be browsed on any other system through domain name (Globally) or locally connected LAN network (IP Address) computers.

Learning outcomes (What I have learnt):

1. **Learnt steps to host a website**
2. **Learnt how to host a website on iis**

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:

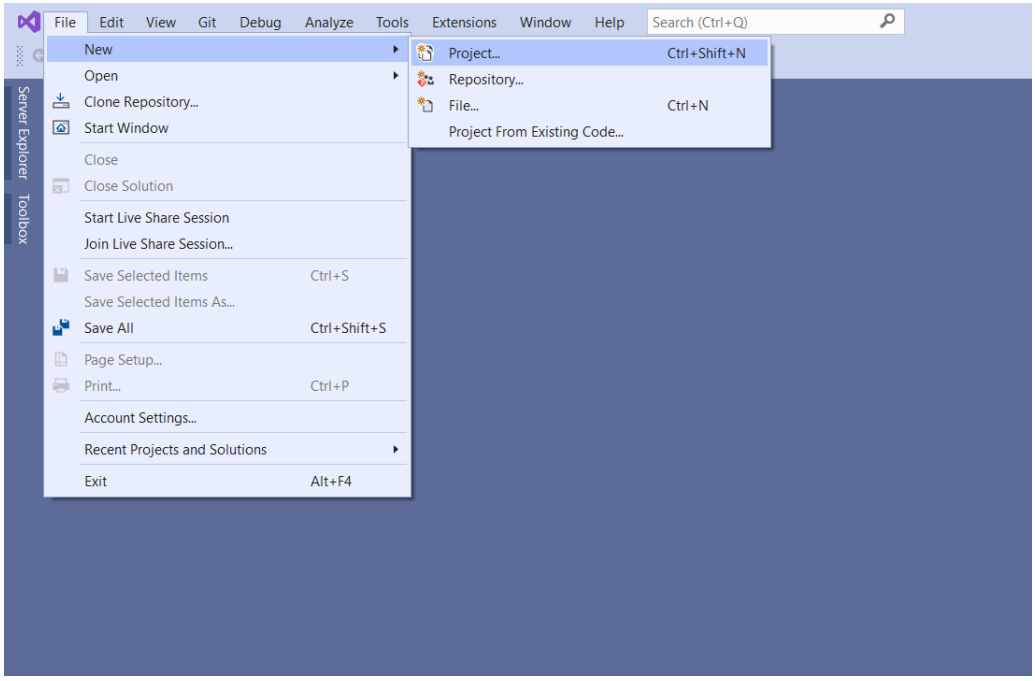
Task 1:

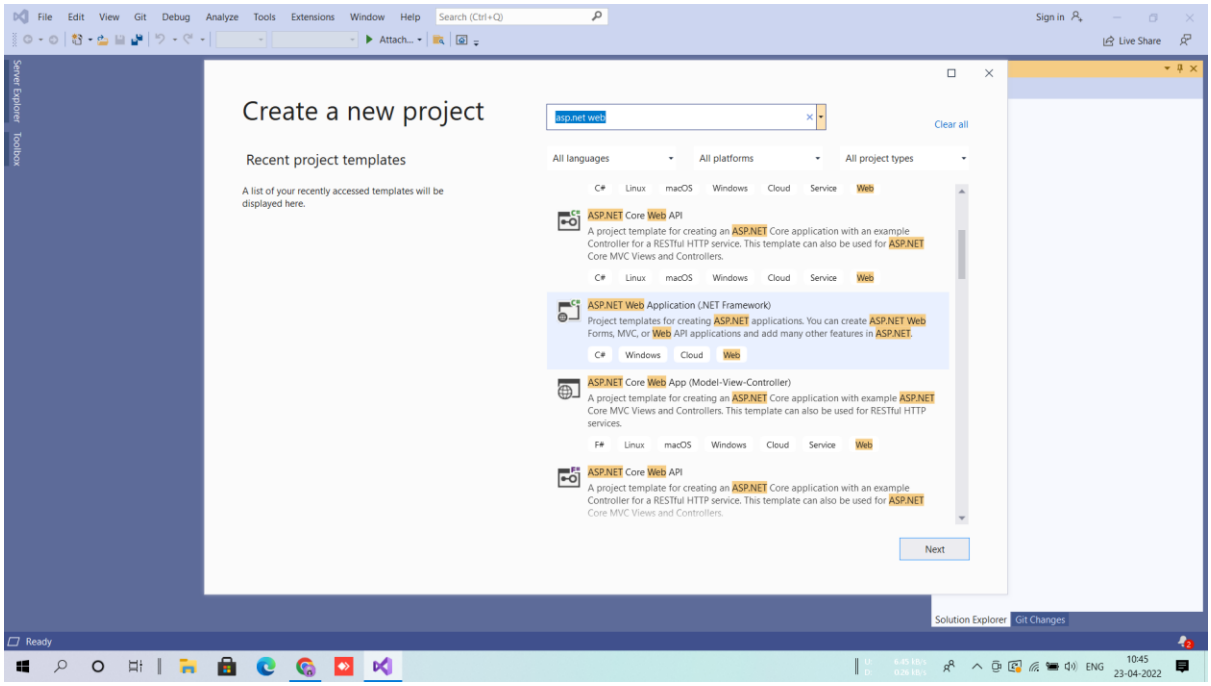
Step 1:

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

Step 2:

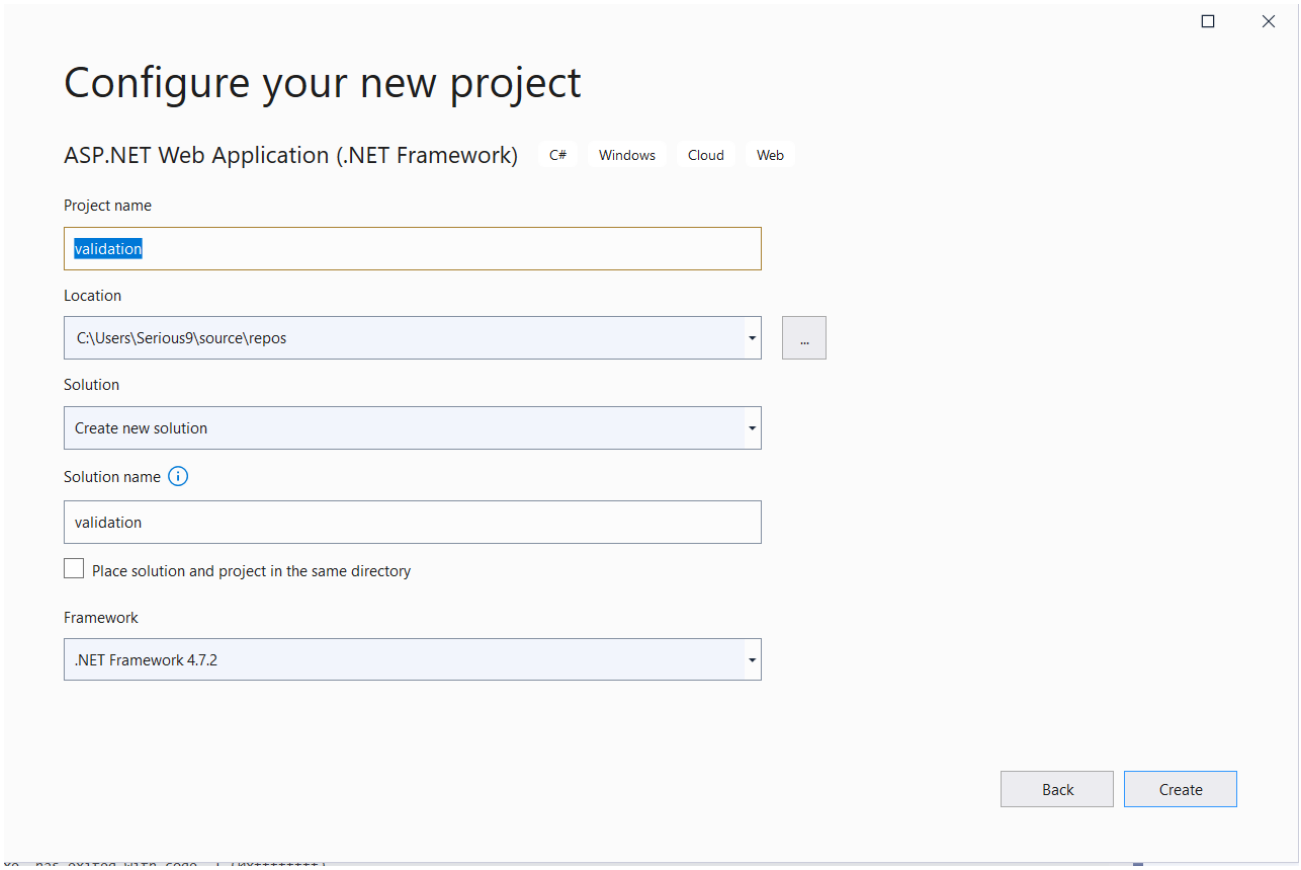
Open Visual Studio 2017 and click on File Menu ->New ->Project





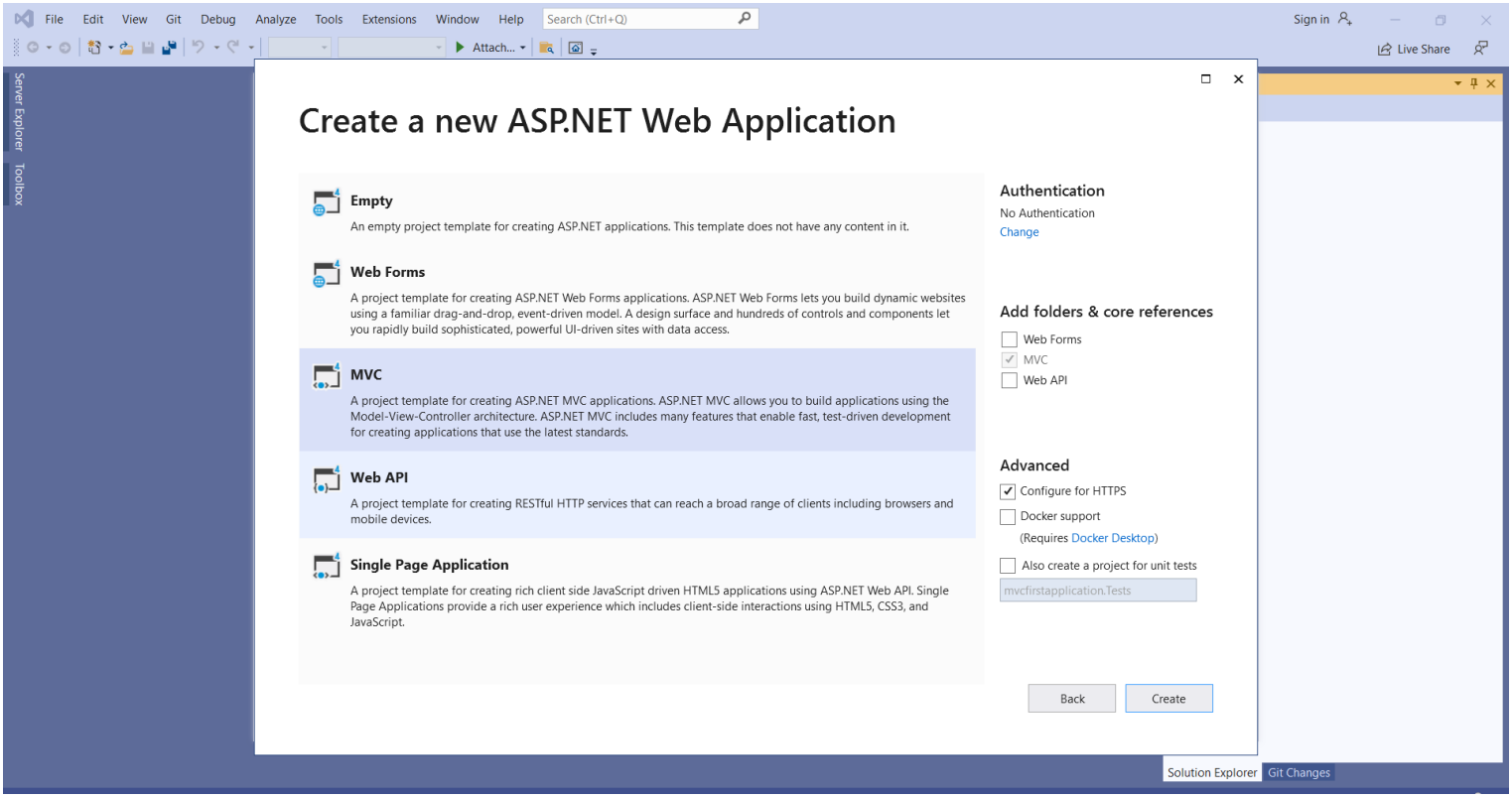
Step 3:

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

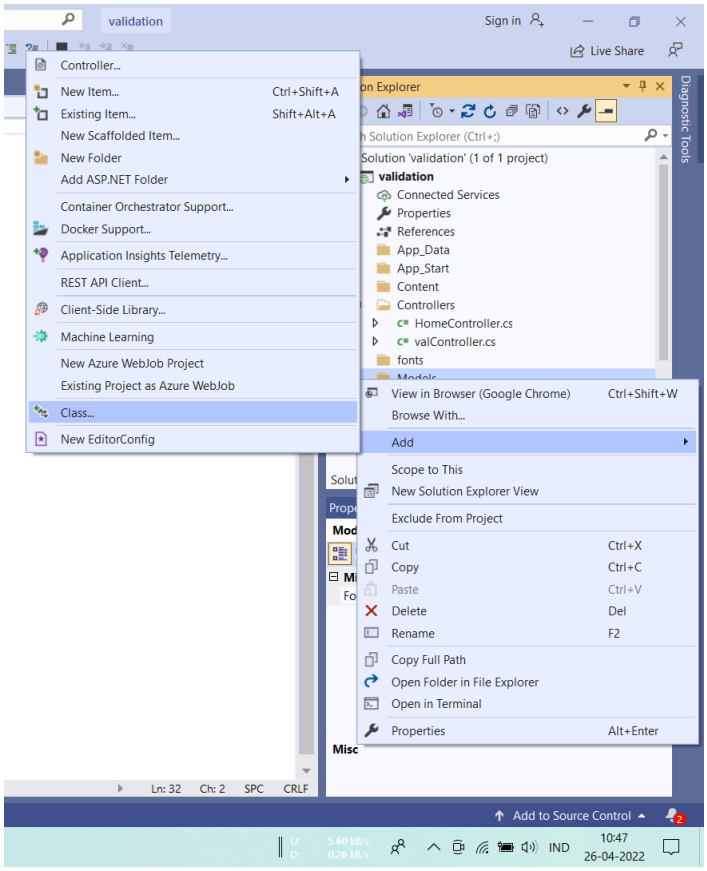


Step 5:

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



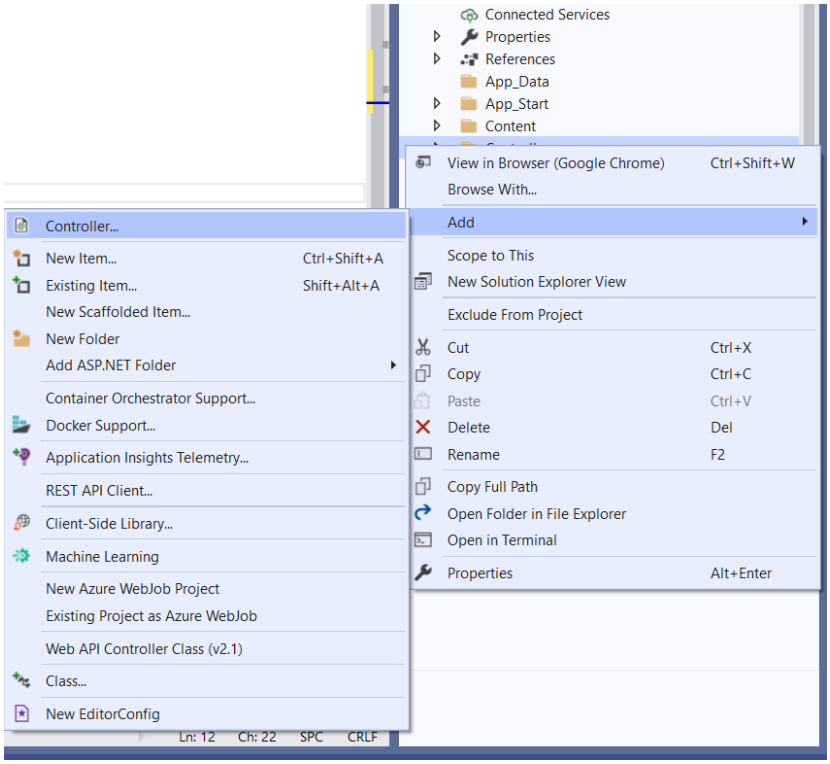
Step 6:

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:

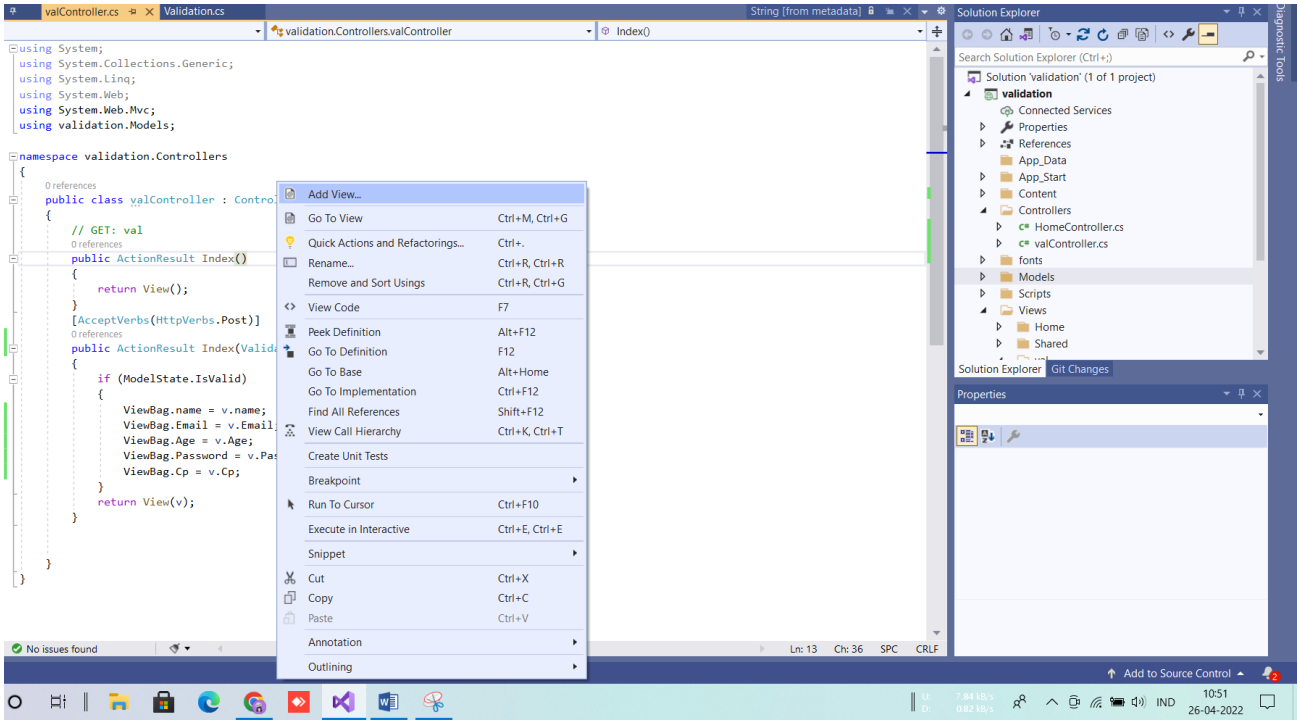
Modify your Controller code with the following code

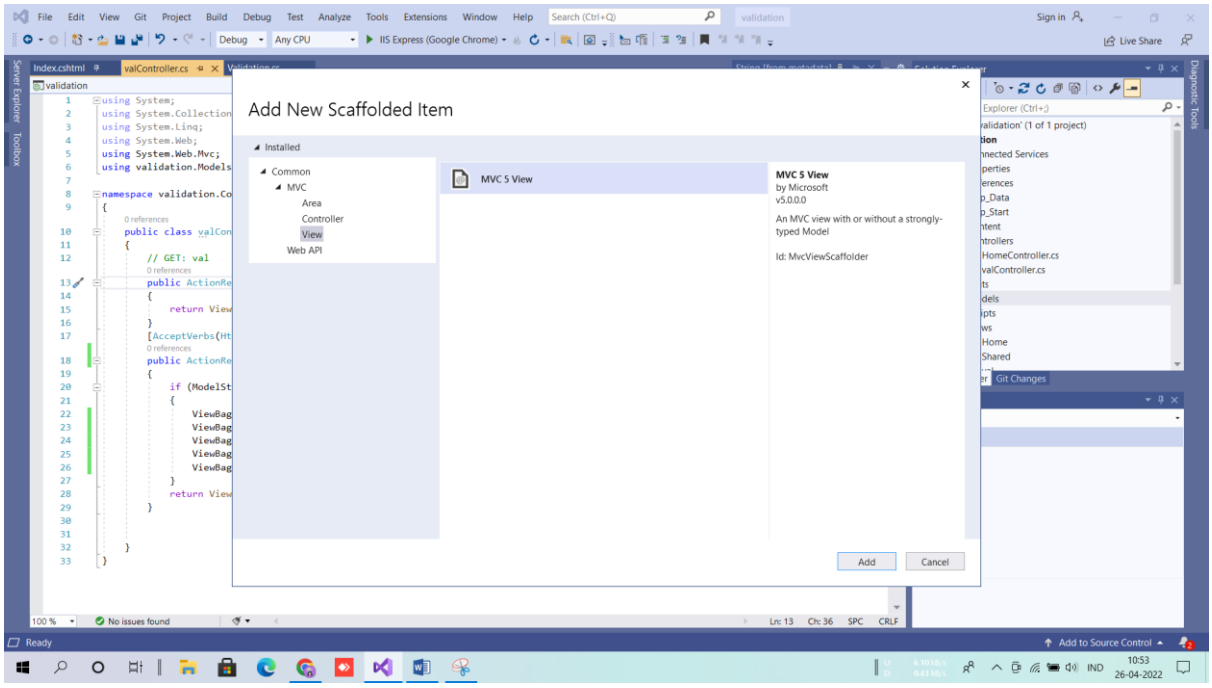
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using validation.Models;

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.

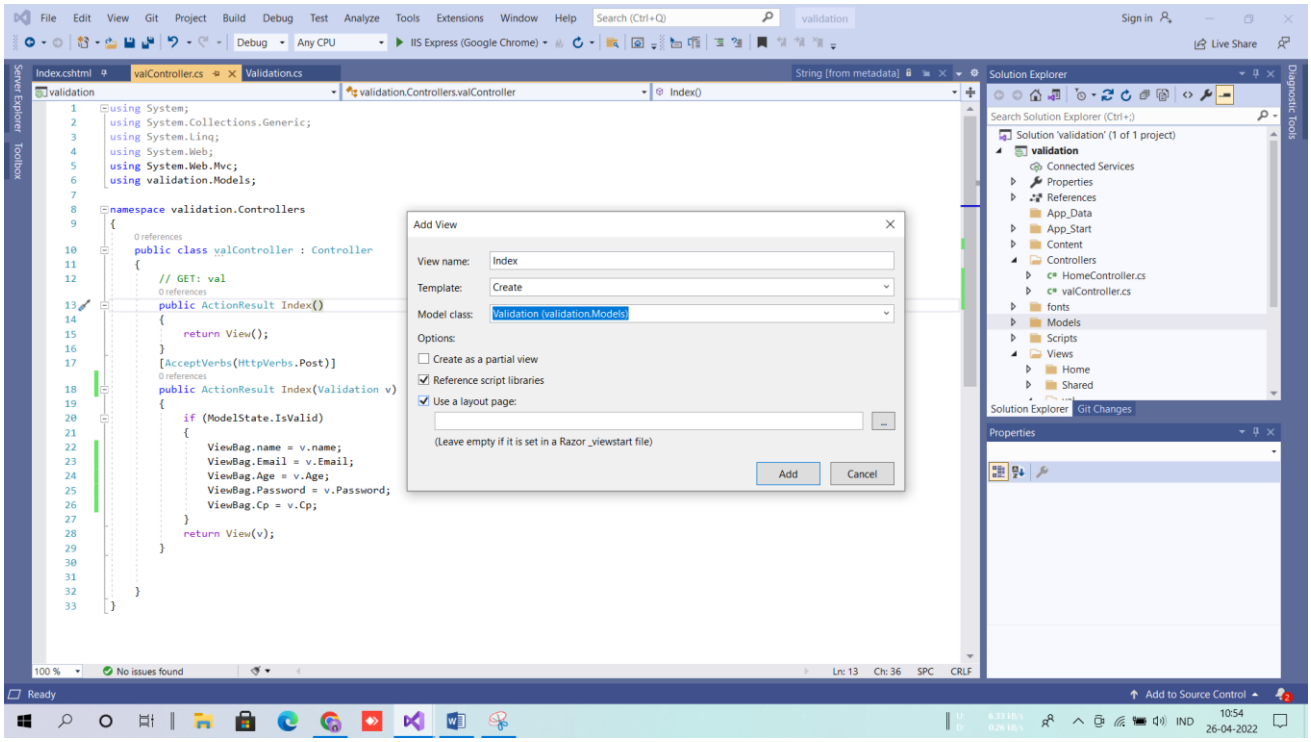
Go to controller then right click and then add view





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">
```

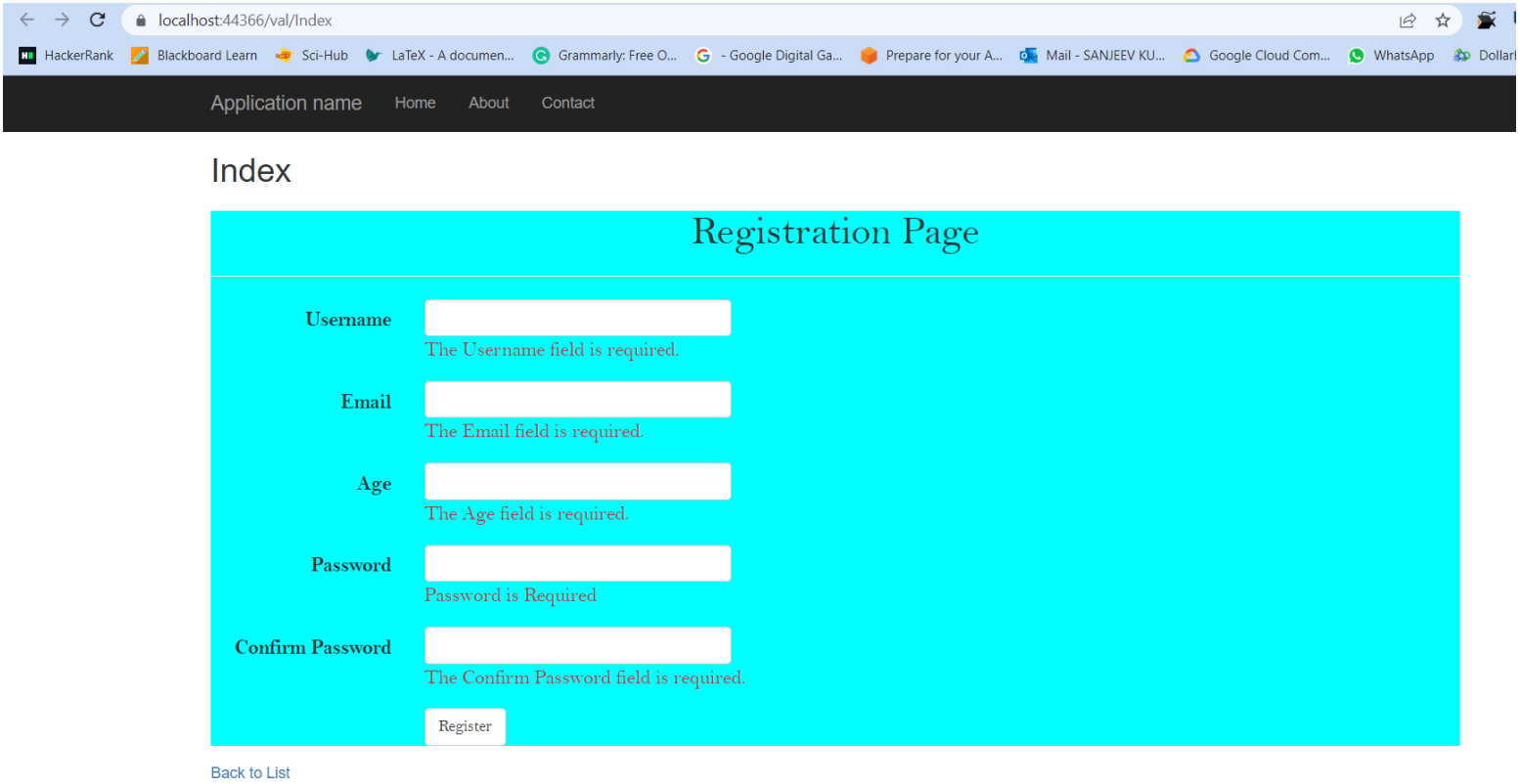
```
@Html.LabelFor(model => model.name, htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model => model.name, new { htmlAttributes = new { @class = "form-control" } })
@Html.ValidationMessageFor(model => model.name, "", new { @class = "text-danger" })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

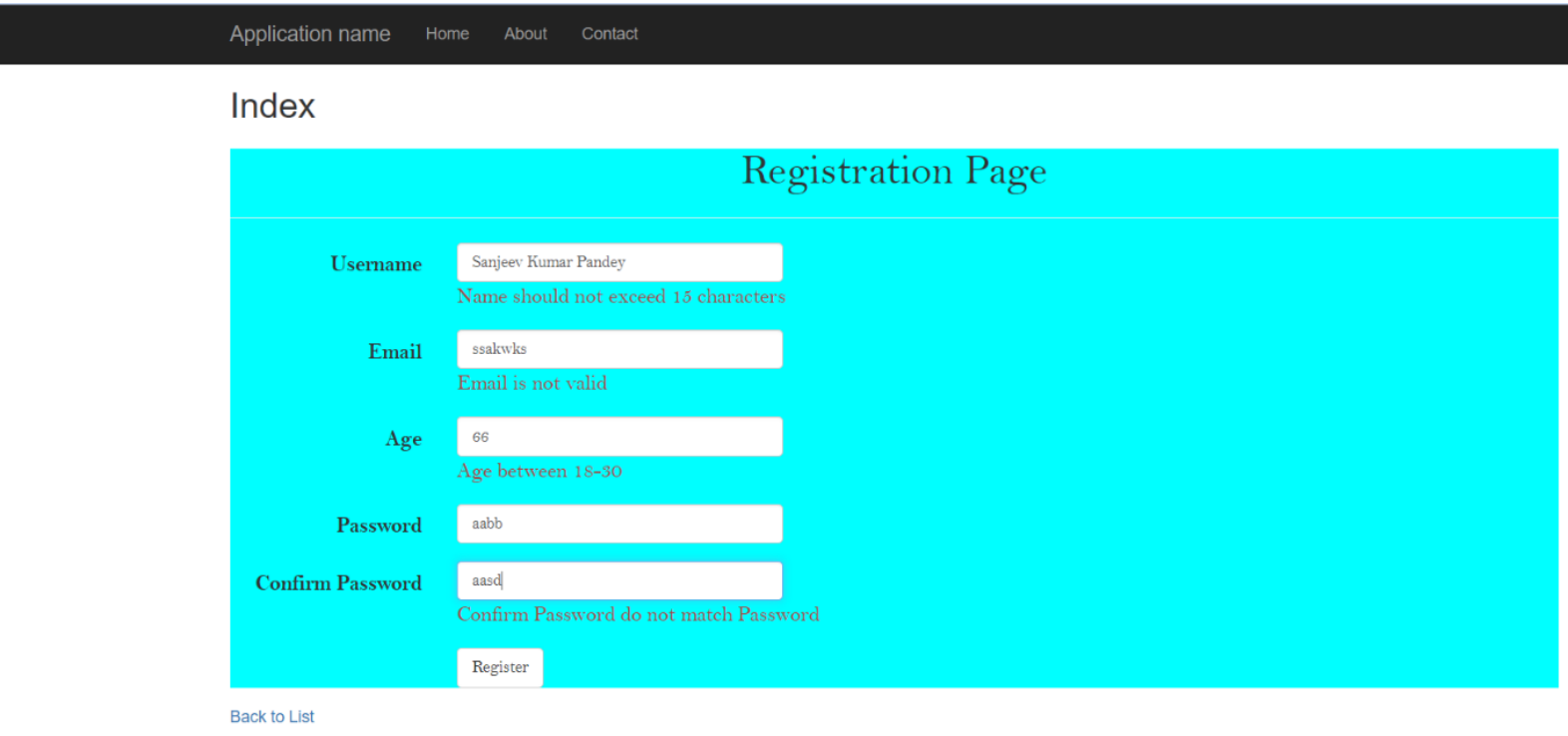
```
@Html.EditorFor(model => model.Password, new { htmlAttributes = new {
@class = "form-control" } })
@Html.ValidationMessageFor(model => model.Password, "", new { @class
= "text-danger" })
</div>
</div>

<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")
}
```

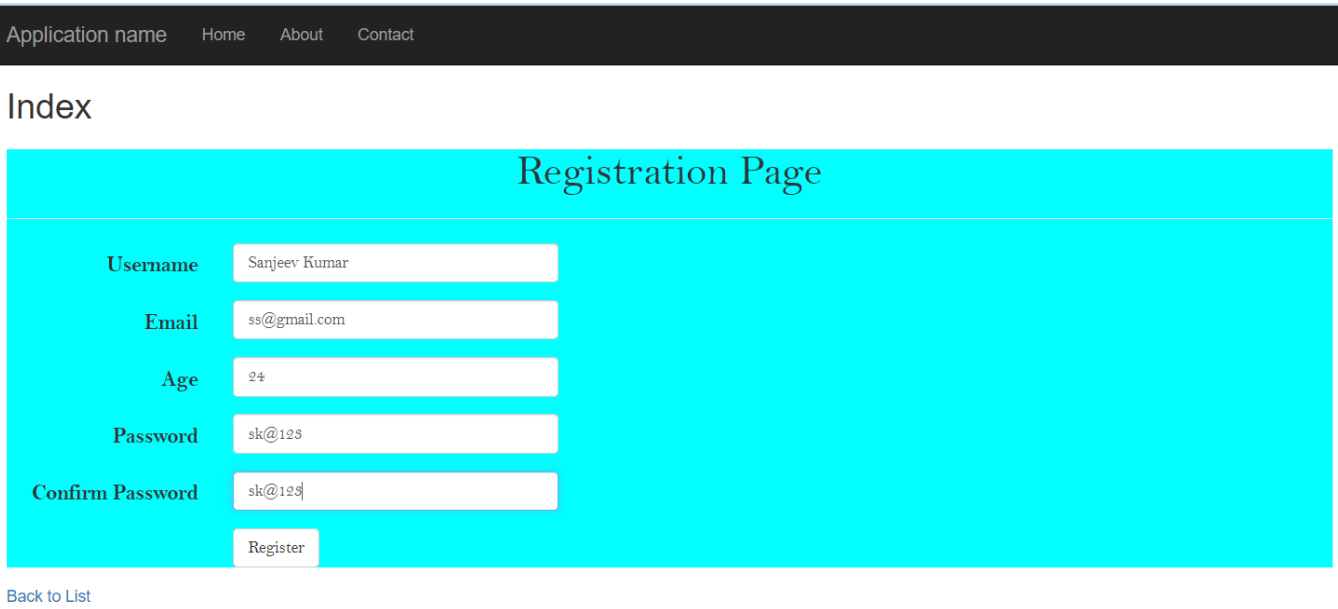
Step 12:

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

LAB WORKSHEET 10

Student Name: Sanjeev Kumar Pandey	**UID: 20MCA1096**
Branch: MCA	**Section/Group:20MCA1/A**
Semester: 4th	**Date of Performance: 30/04/2022**
Subject Name: AWAD	**Subject Code: 20CAA-755**

1. **Task to be done:**

1.Write the steps to host website in asp.net.

2. **Steps for experiment/practical:**

For hosting a website on IIS, follow these steps:

Step 1: IIS Installation.

IIS is a protocol server which is used to host a website on server. IIS stands for Internet Information services.

To install IIS start with windows start icon -> Control panel, Programs and Features , then click Turn Windows features on or off.

Figure 1

After that click windows features on or off, the Windows Features pop up window opens.

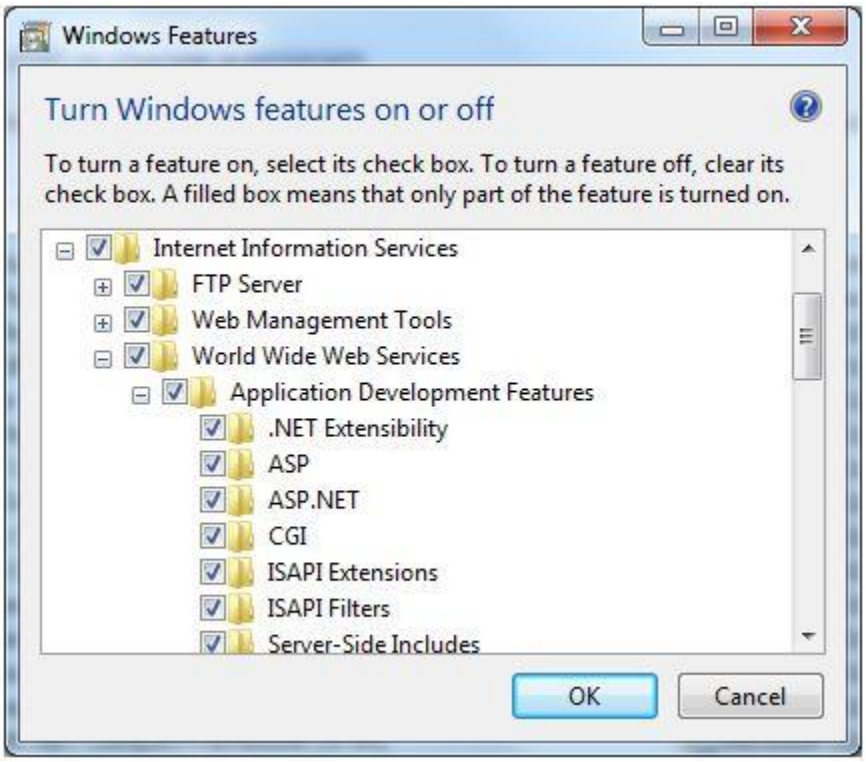


Figure 2: Windows features

Step 2: Enable ASP.NET Features on IIS

Expand the Application Development Features in that enable the ASP.NET application features such as:

1. .NET Extensibility
2. ASP
3. ASP.NET
4. CGI
5. ISAPI Extensions
6. ISAPI Filters
7. Server-Side Includes.

After enabling the ASP.NET features on IIS hit Ok. The IIS server Manager is successfully enabled in start control list.

Step 3: Install .NET Framework.

Now we have to install .NET framework version in our system because the aspx web page needs to run the platform of .net framework.

Figure 3: Setup.exe

Then the file is saved, click the saved exe file and the extract field process window will appear. The file extraction process completed and the following authorization window will appear, check I agree box then hit Install. Then the download and installation progress will appear in the same window.

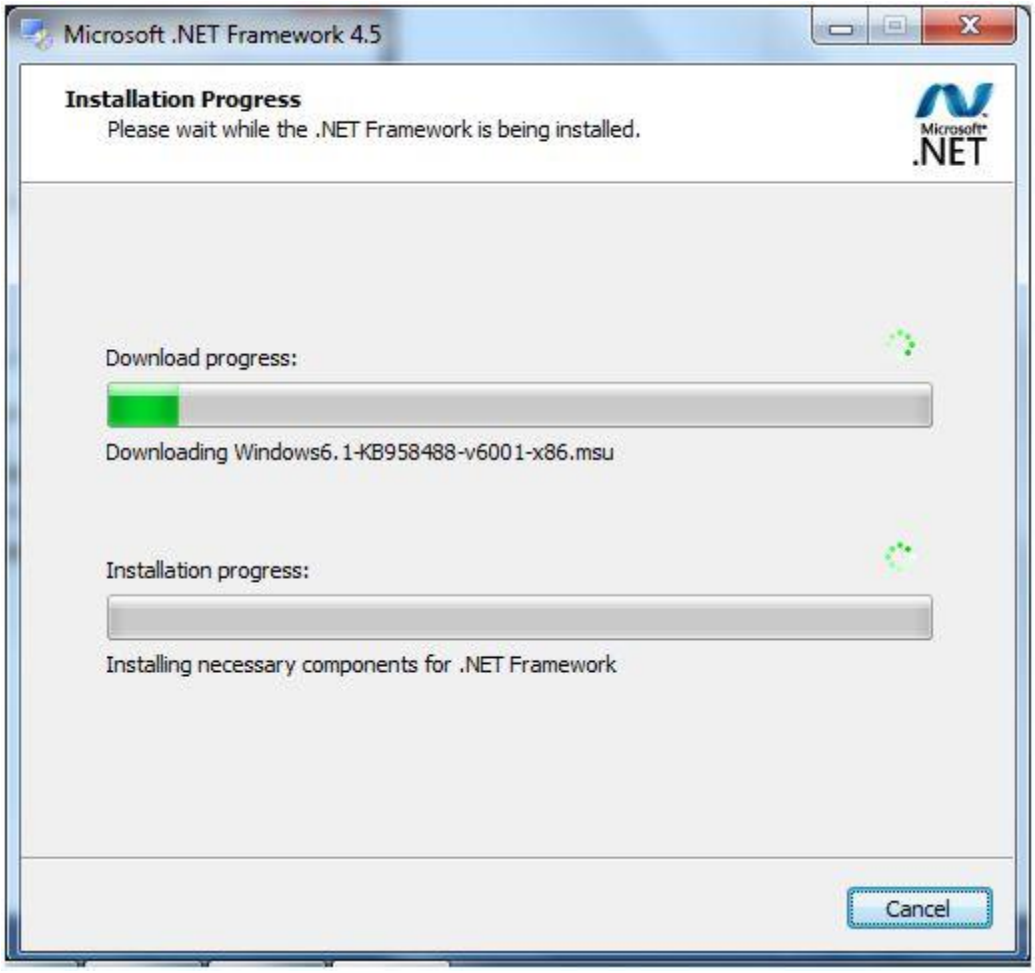


Figure 4: Installation Process

After the downloading and Installation progress are completed hit the finish button to complete the Installation. Now your IIS is ready to host ASP.NET web sites.

Step 4: Hosting a site.
Select the website folder which one you want to host it into your IIS.

Start IIS server, in that click site, right click on site and then choose add web site

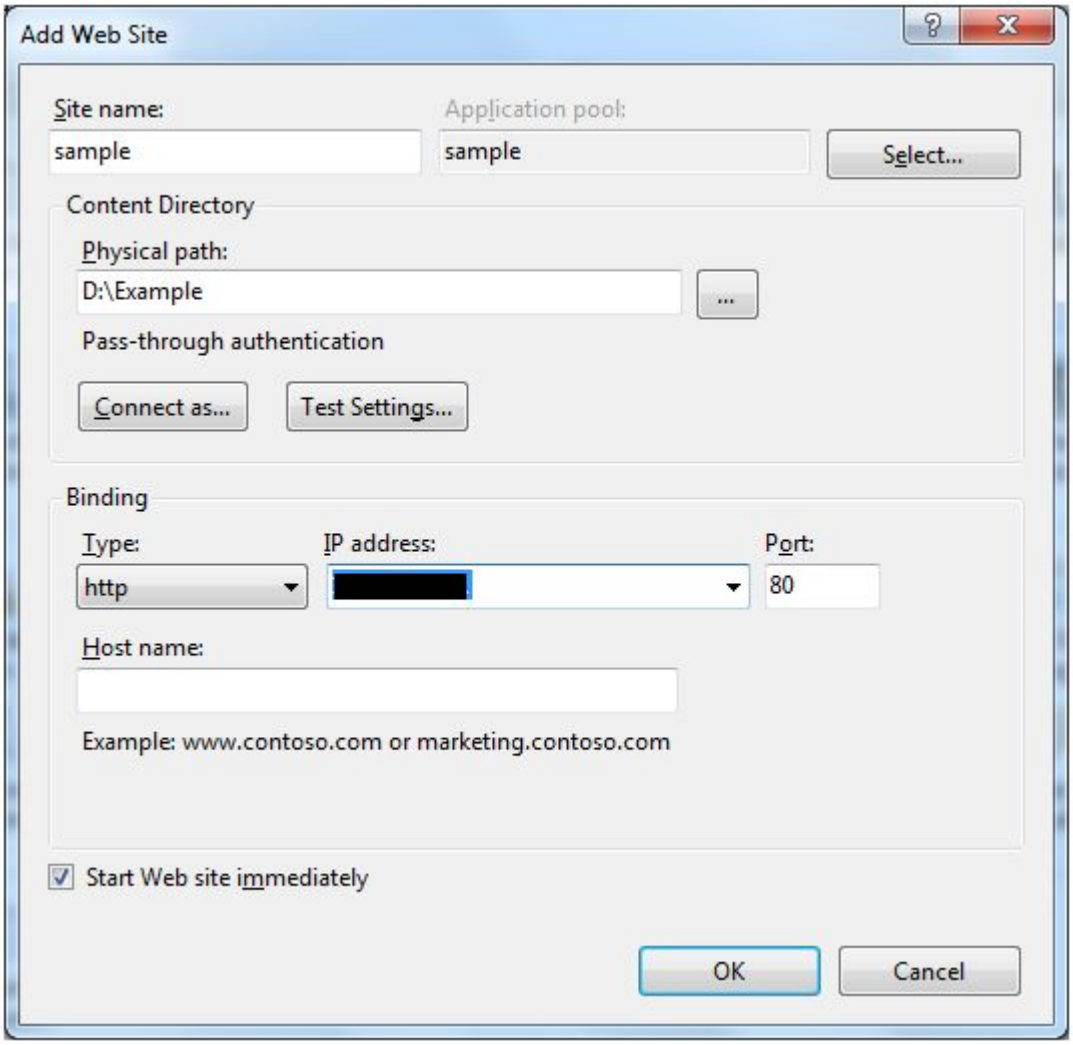


Figure 5: Add web site window

Add site name, path of that site folder, then assign the IP address or enter the host name of that particular site like www.example.com and then hit ok.

After the process is completed your site is placed under default web site under Sites.

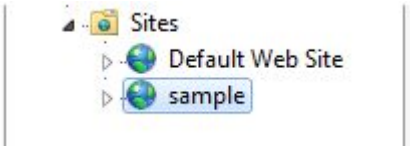


Figure 6

At the same time there is one application pool is automatically created while adding the web site.

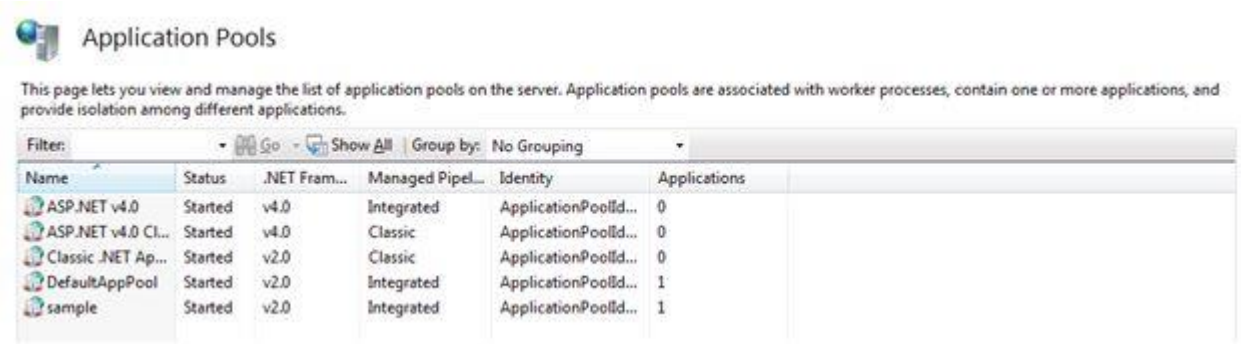


Figure 7: Application Pool

Here I have a simple web page like default.aspx in your website there is a database connectivity available which means then you have to follow the steps to add the database into application pool.

Step 5: Database Connectivity

In the application pool panel right click the application name like sample, then choose Advanced Settings option.

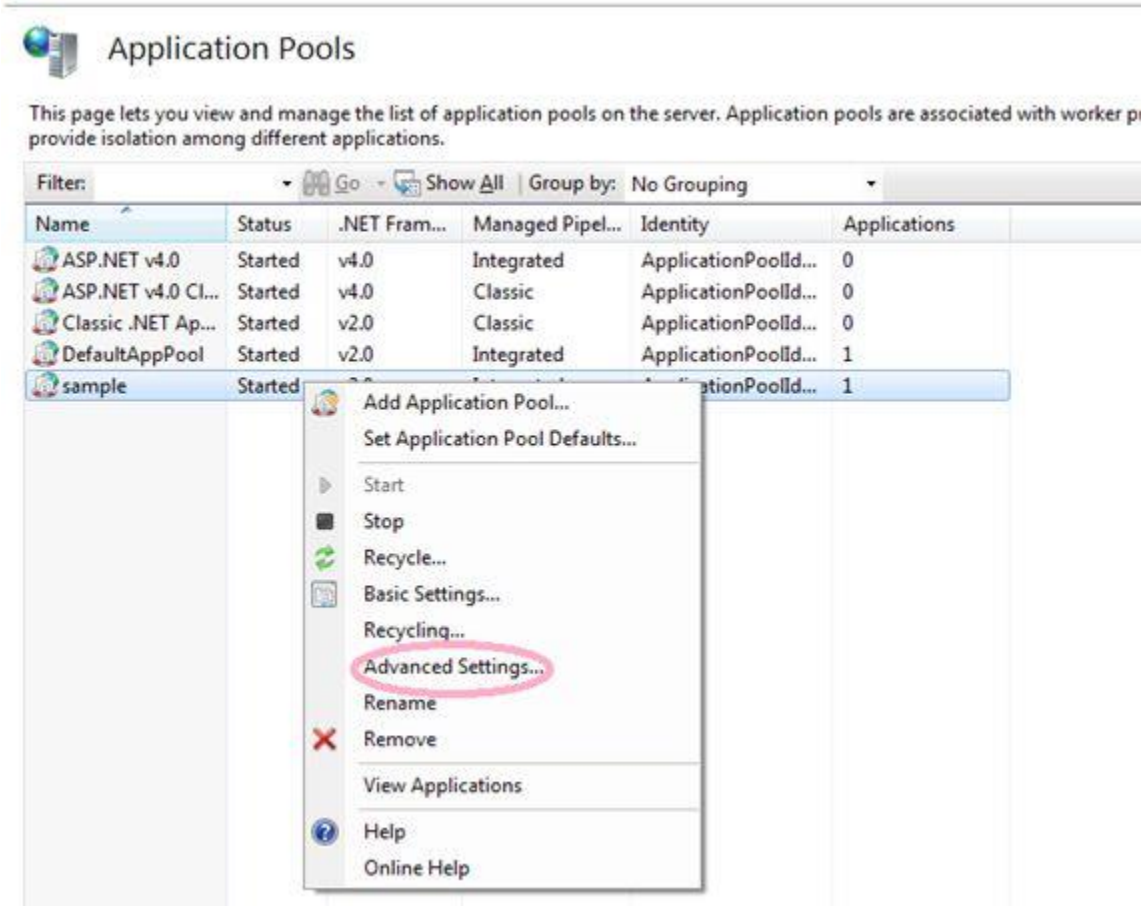


Figure 8: Advanced settings Option

The settings window will open in that window select .net framework version like v2.0. Based on our pages it automatically chooses the framework version. Then select managed pipeline mode is Integrated or Classic.

Then in the process model tab select Identity then choose the following:

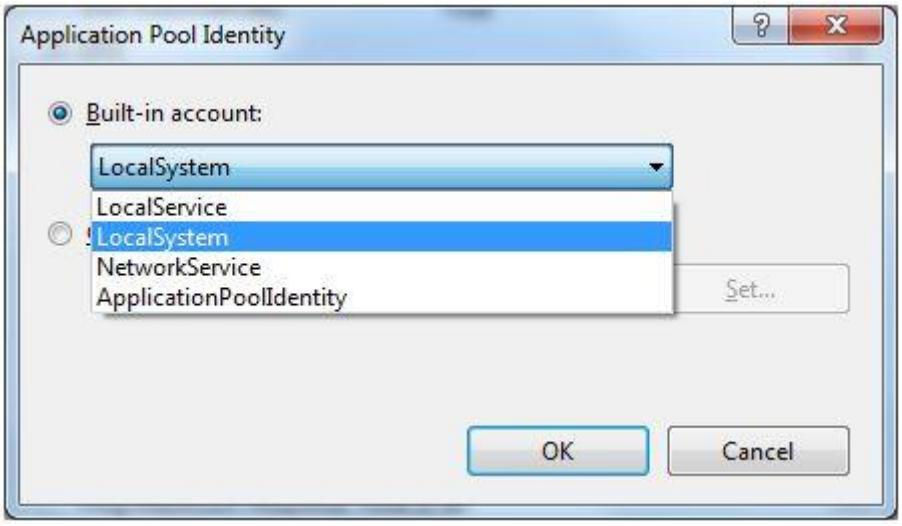


Figure 9: Pool Identity

Select LocalSystem because our database connectivity is placed in our system SQL Server so we need to choose LocalSystem then only our database is connected. Then hit OK to close the window.

Step 6: Conclusion.

In the Actions panel choose Browse link

Figure 10: Browse

Now your site will appear in the browser like the following figure:

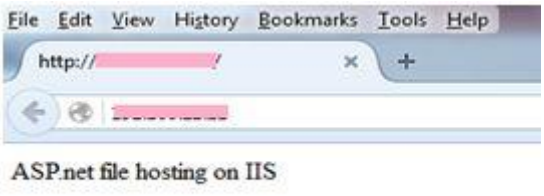


Figure 11: Browser

Now your application is ready to be browsed on any other system through domain name (Globally) or locally connected LAN network (IP Address) computers.

Learning outcomes (What I have learnt):

1. **Learnt steps to host a website**
2. **Learnt how to host a website on iis**

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:

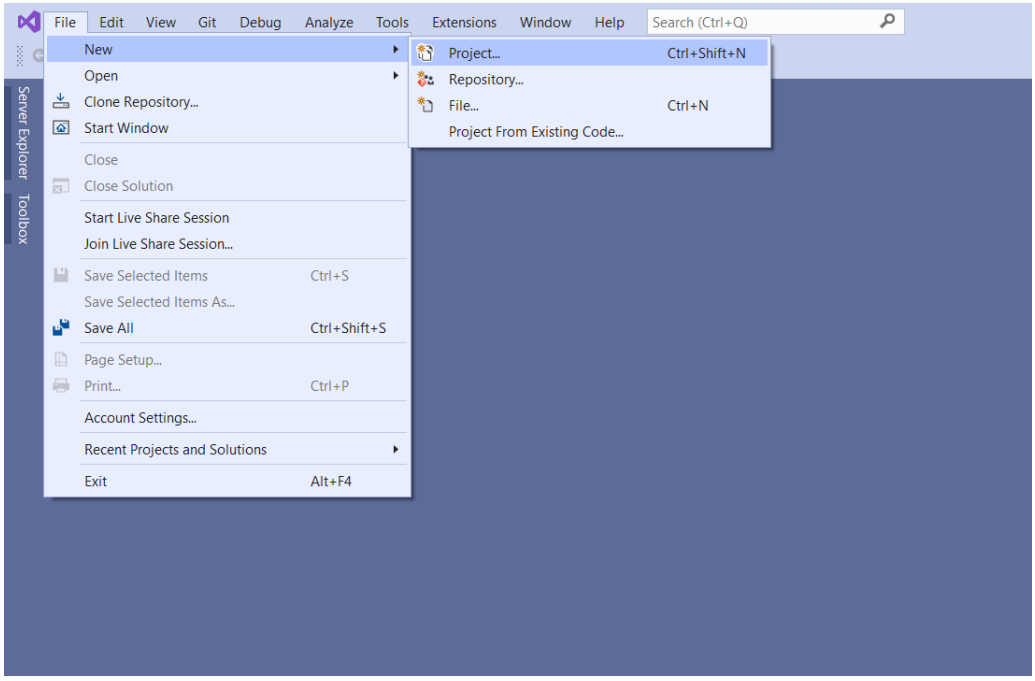
Task 1:

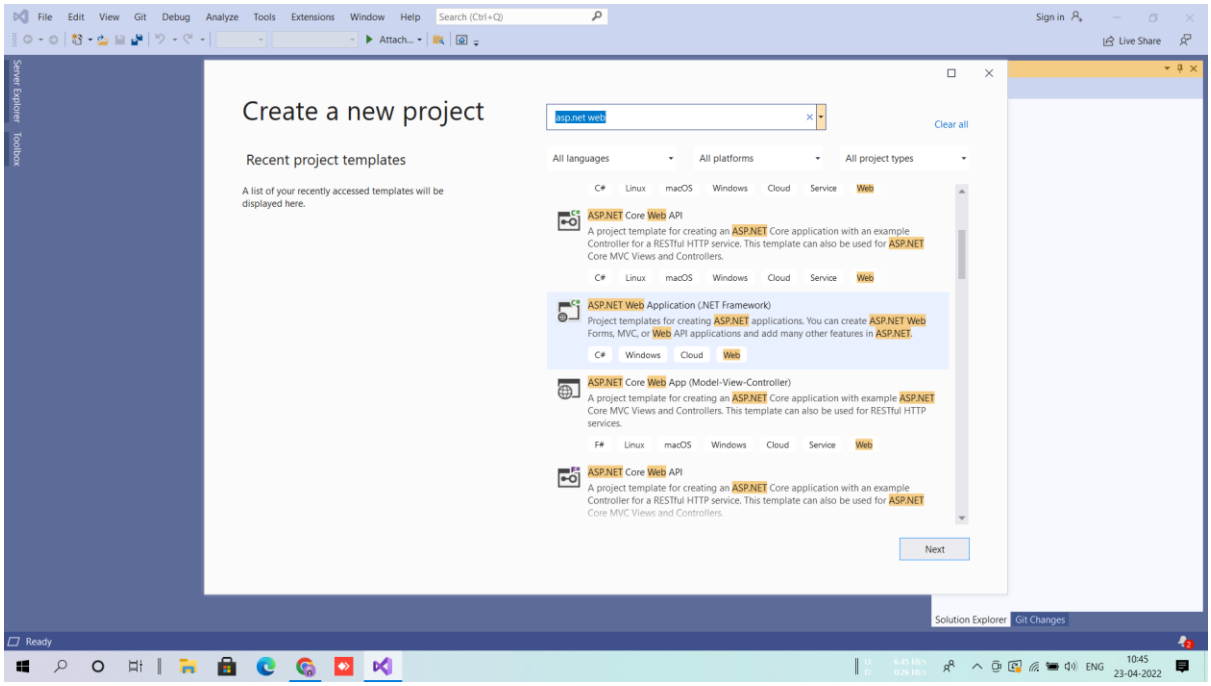
Step 1:

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

Step 2:

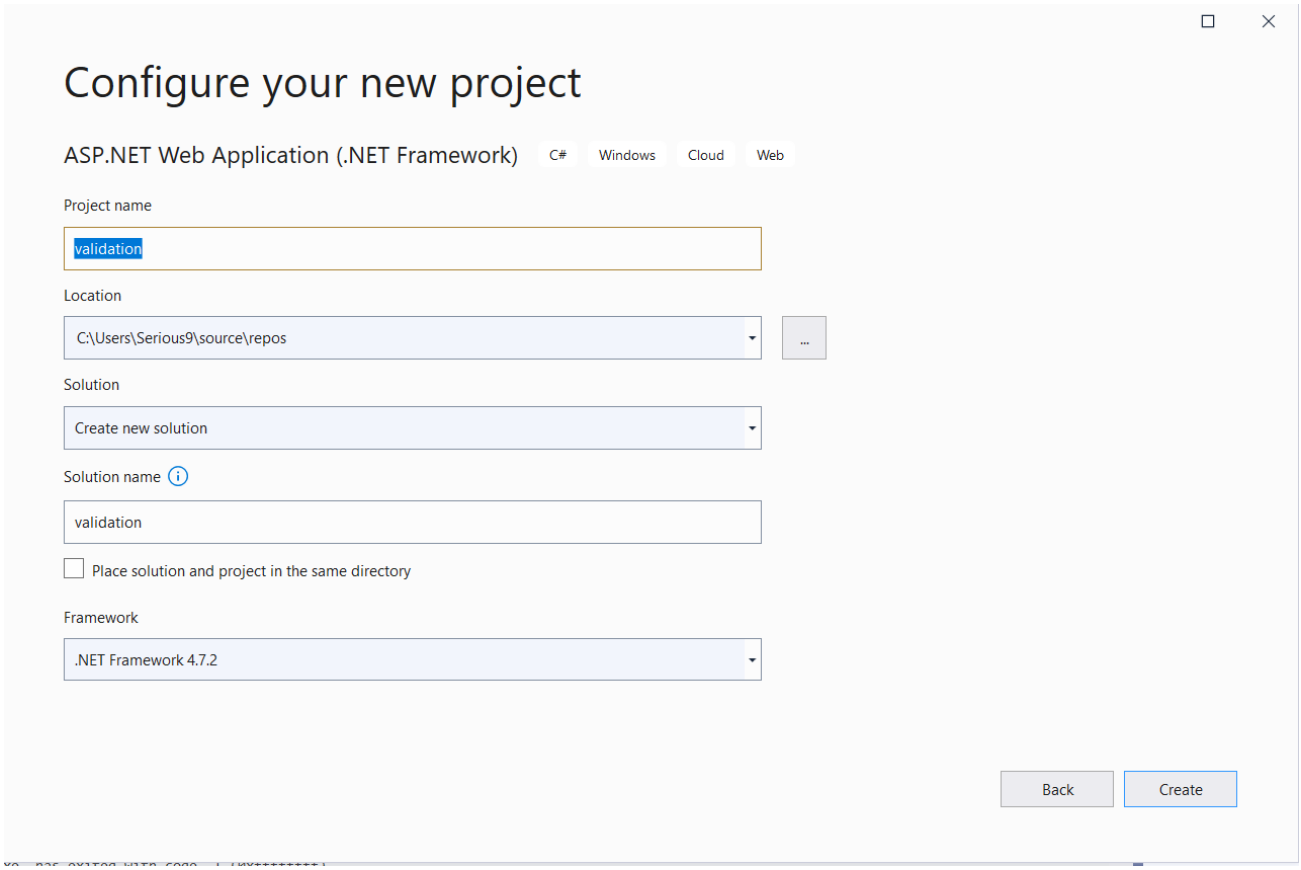
Open Visual Studio 2017 and click on File Menu ->New ->Project





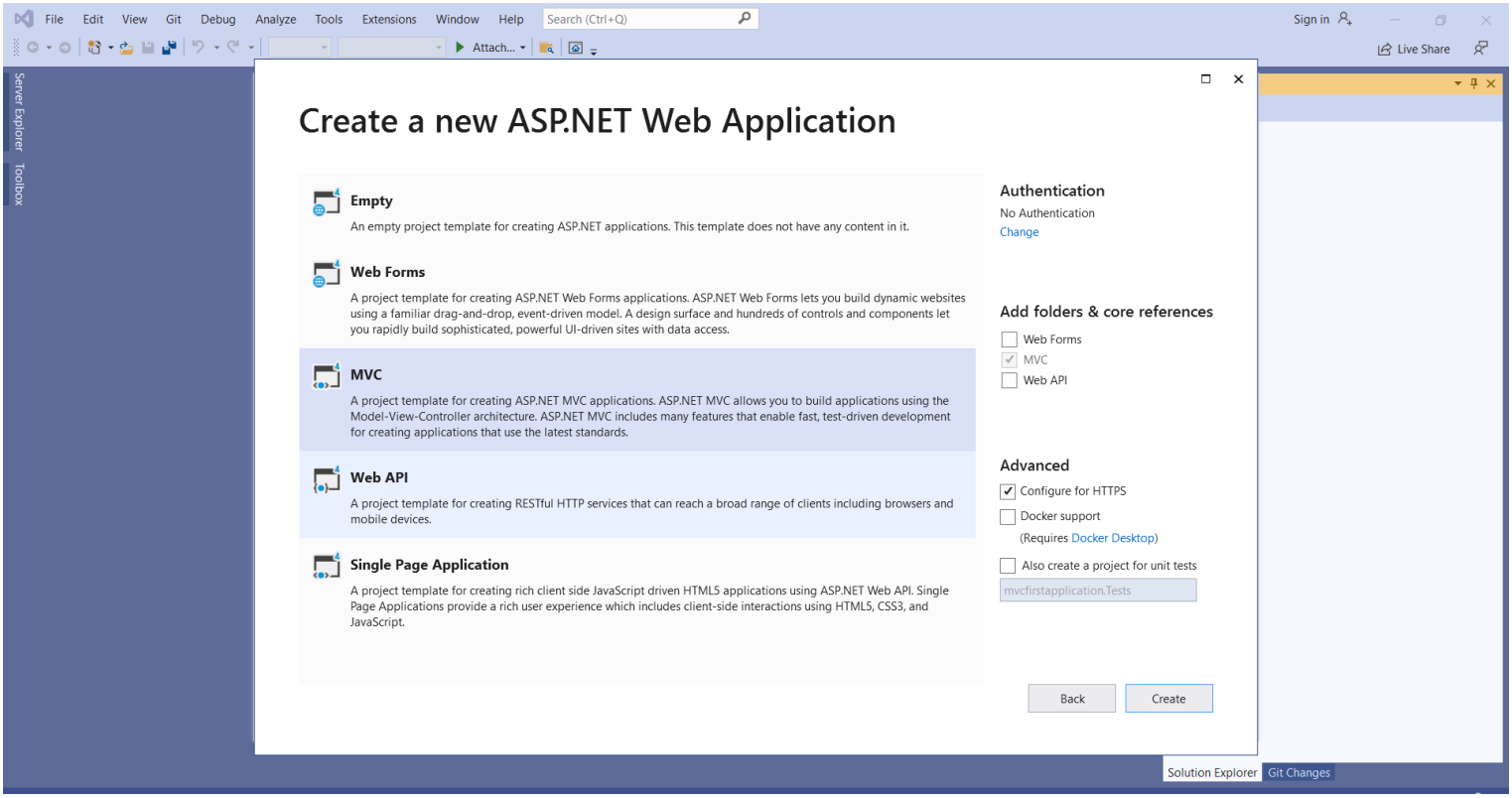
Step 3:

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

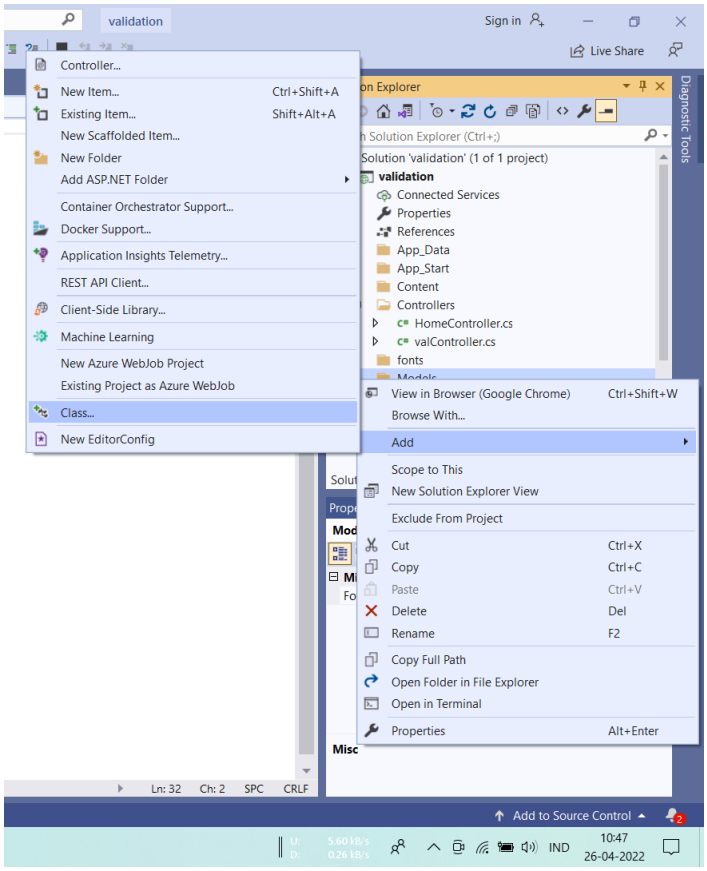


Step 5:

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



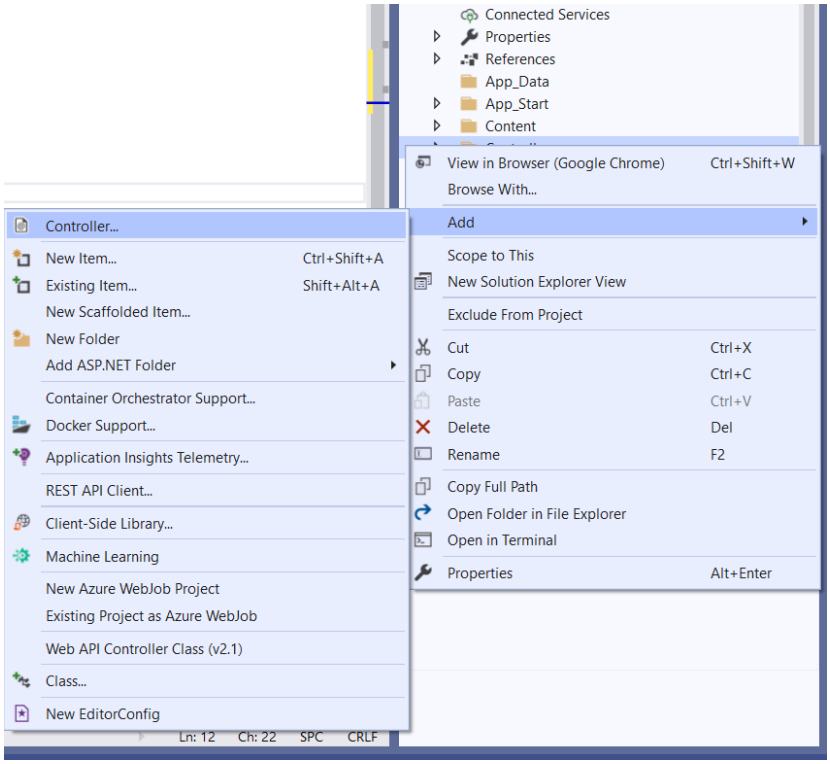
Step 6:

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:

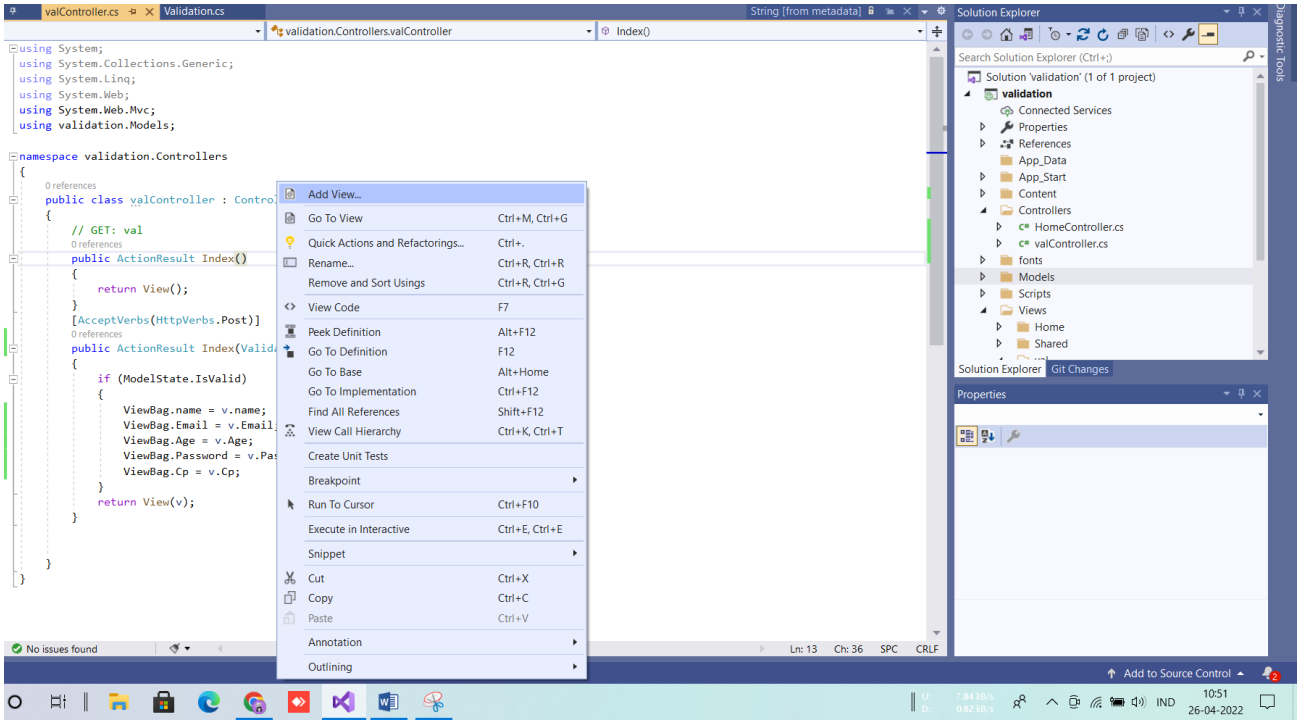
Modify your Controller code with the following code

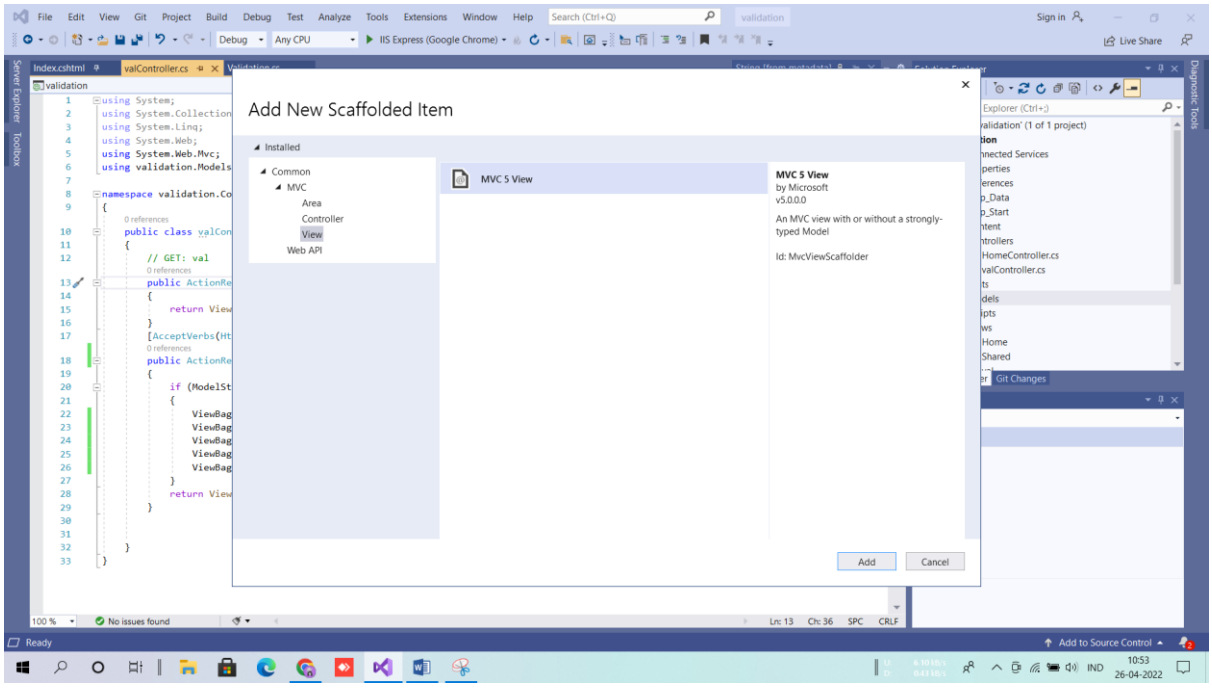
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using validation.Models;

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.

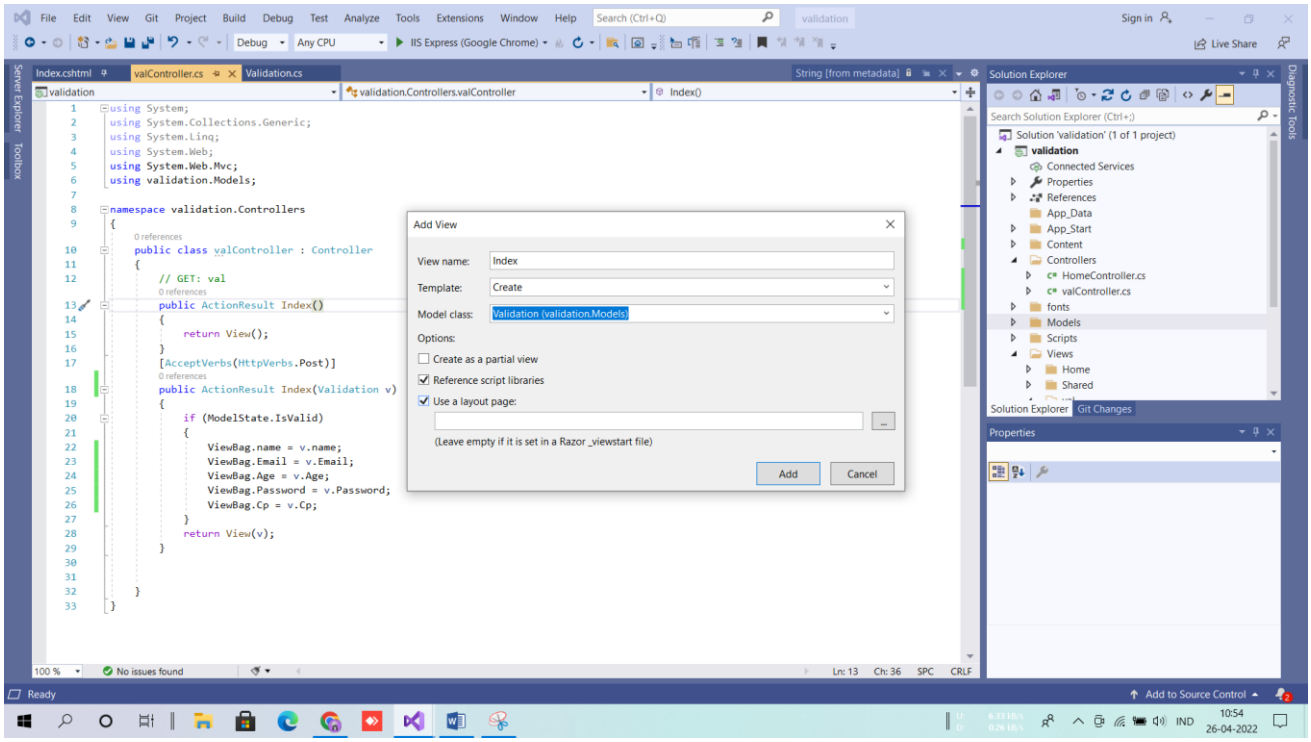
Go to controller then right click and then add view





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">
```

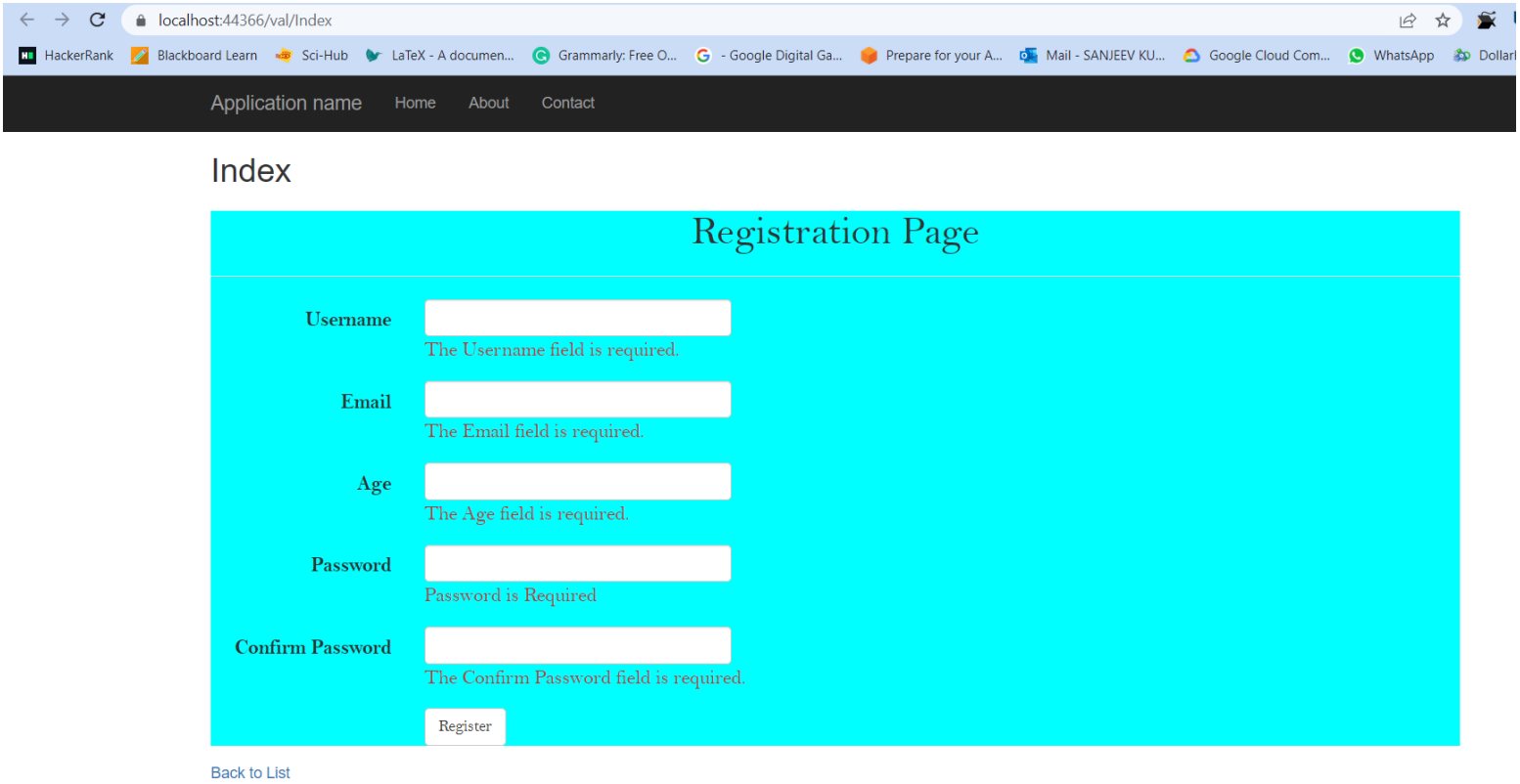
```
@Html.LabelFor(model => model.name, htmlAttributes: new { @class = "control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model => model.name, new { htmlAttributes = new { @class = "form-control" } })
@Html.ValidationMessageFor(model => model.name, "", new { @class = "text-danger" })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

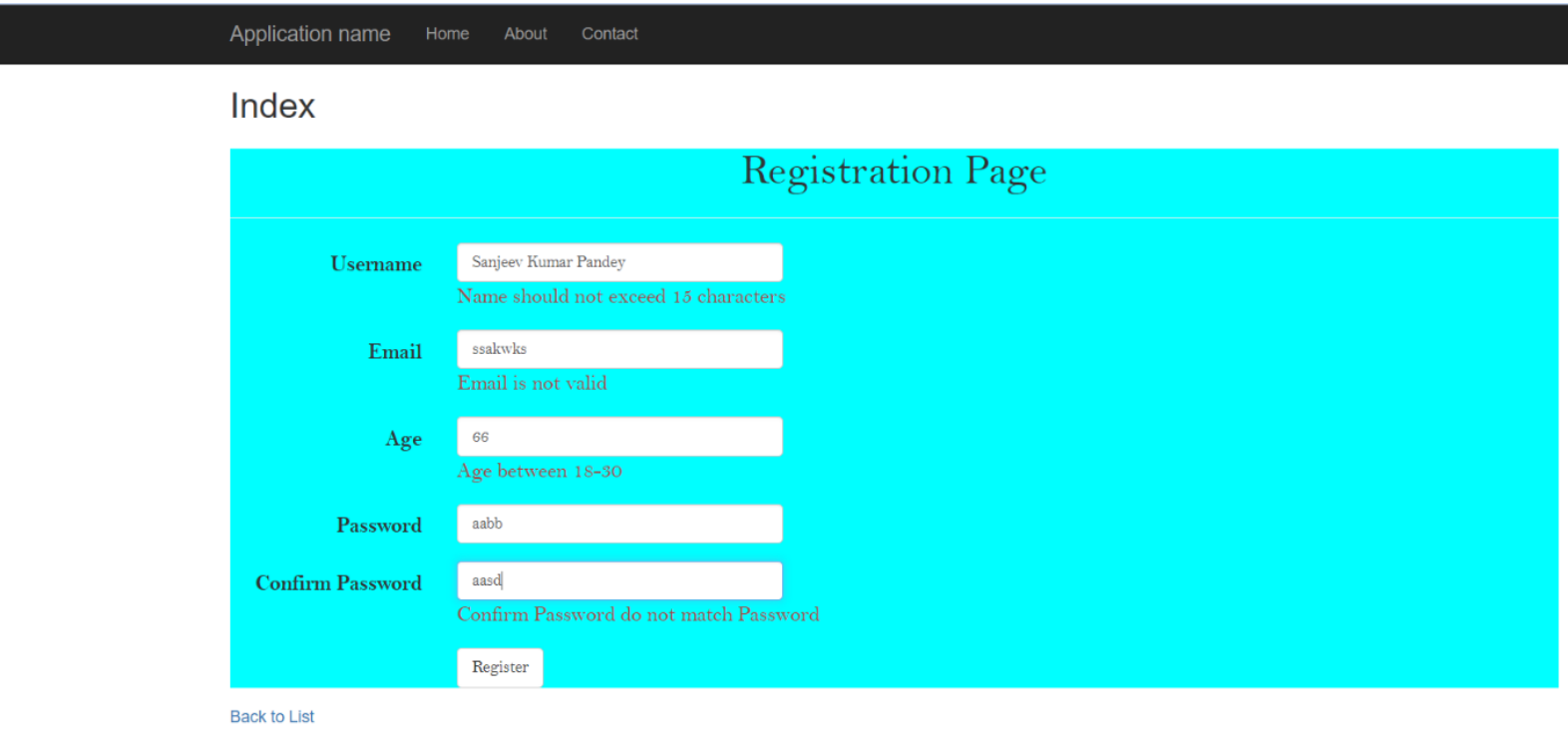
```
@Html.EditorFor(model => model.Password, new { htmlAttributes = new {
@class = "form-control" } })
@Html.ValidationMessageFor(model => model.Password, "", new { @class
= "text-danger" })
</div>
</div>

<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")
}
```

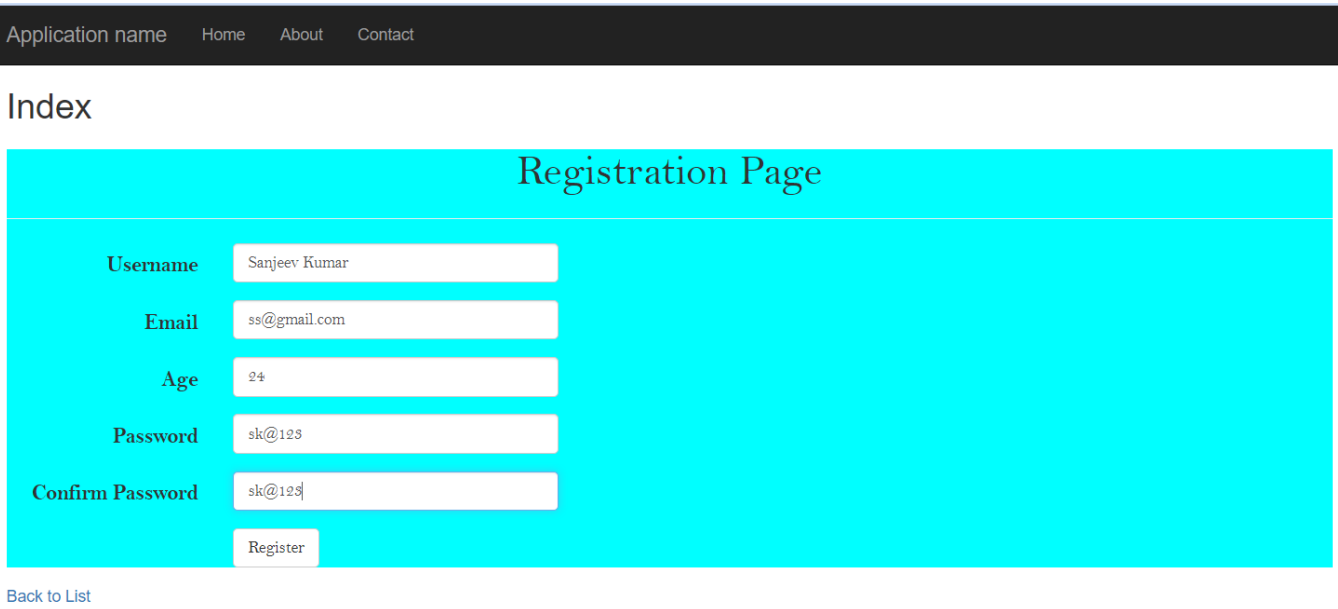
Step 12:

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

LAB WORKSHEET 10

Student Name: Sanjeev Kumar Pandey	**UID: 20MCA1096**
Branch: MCA	**Section/Group:20MCA1/A**
Semester: 4th	**Date of Performance: 30/04/2022**
Subject Name: AWAD	**Subject Code: 20CAA-755**

1. **Task to be done:**

1.Write the steps to host website in asp.net.

2. **Steps for experiment/practical:**

For hosting a website on IIS, follow these steps:

Step 1: IIS Installation.

IIS is a protocol server which is used to host a website on server. IIS stands for Internet Information services.

To install IIS start with windows start icon -> Control panel, Programs and Features , then click Turn Windows features on or off.

Figure 1

After that click windows features on or off, the Windows Features pop up window opens.

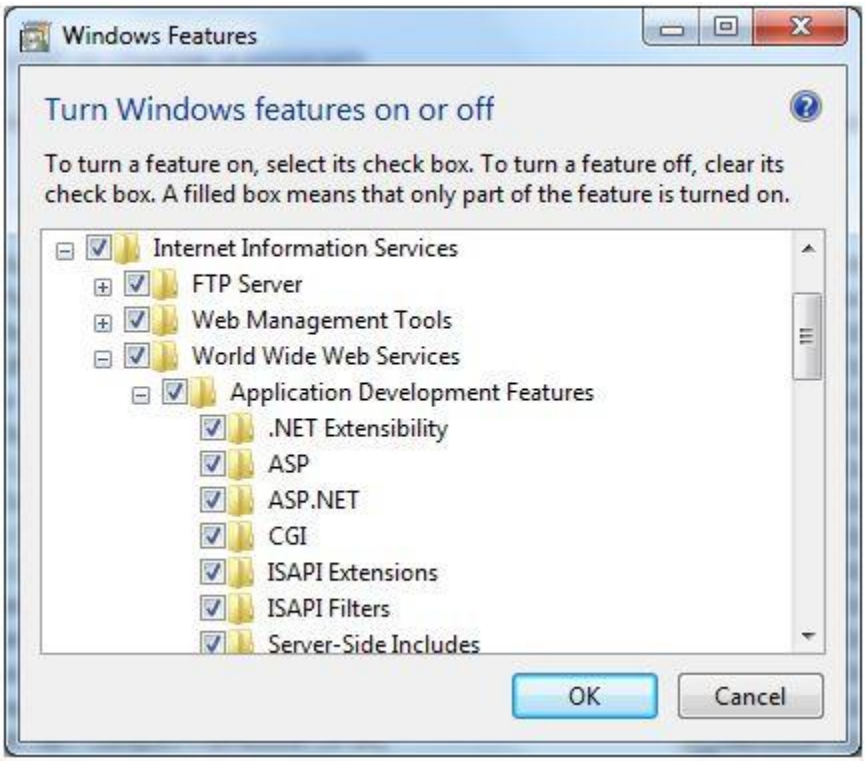


Figure 2: Windows features

Step 2: Enable ASP.NET Features on IIS

Expand the Application Development Features in that enable the ASP.NET application features such as:

1. .NET Extensibility
2. ASP
3. ASP.NET
4. CGI
5. ISAPI Extensions
6. ISAPI Filters
7. Server-Side Includes.

After enabling the ASP.NET features on IIS hit Ok. The IIS server Manager is successfully enabled in start control list.

Step 3: Install .NET Framework.

Now we have to install .NET framework version in our system because the aspx web page needs to run the platform of .net framework.

Figure 3: Setup.exe

Then the file is saved, click the saved exe file and the extract field process window will appear. The file extraction process completed and the following authorization window will appear, check I agree box then hit Install. Then the download and installation progress will appear in the same window.

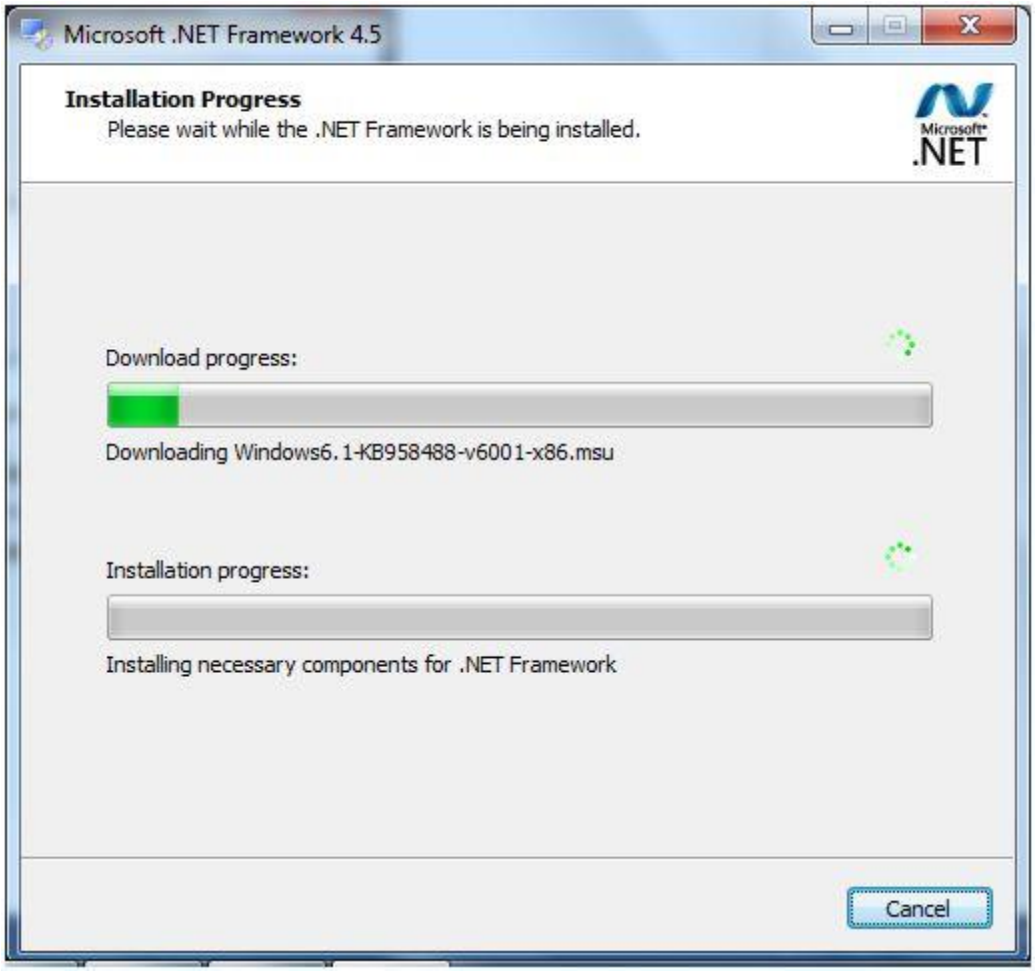


Figure 4: Installation Process

After the downloading and Installation progress are completed hit the finish button to complete the Installation. Now your IIS is ready to host ASP.NET web sites.

Step 4: Hosting a site.
Select the website folder which one you want to host it into your IIS.

Start IIS server, in that click site, right click on site and then choose add web site

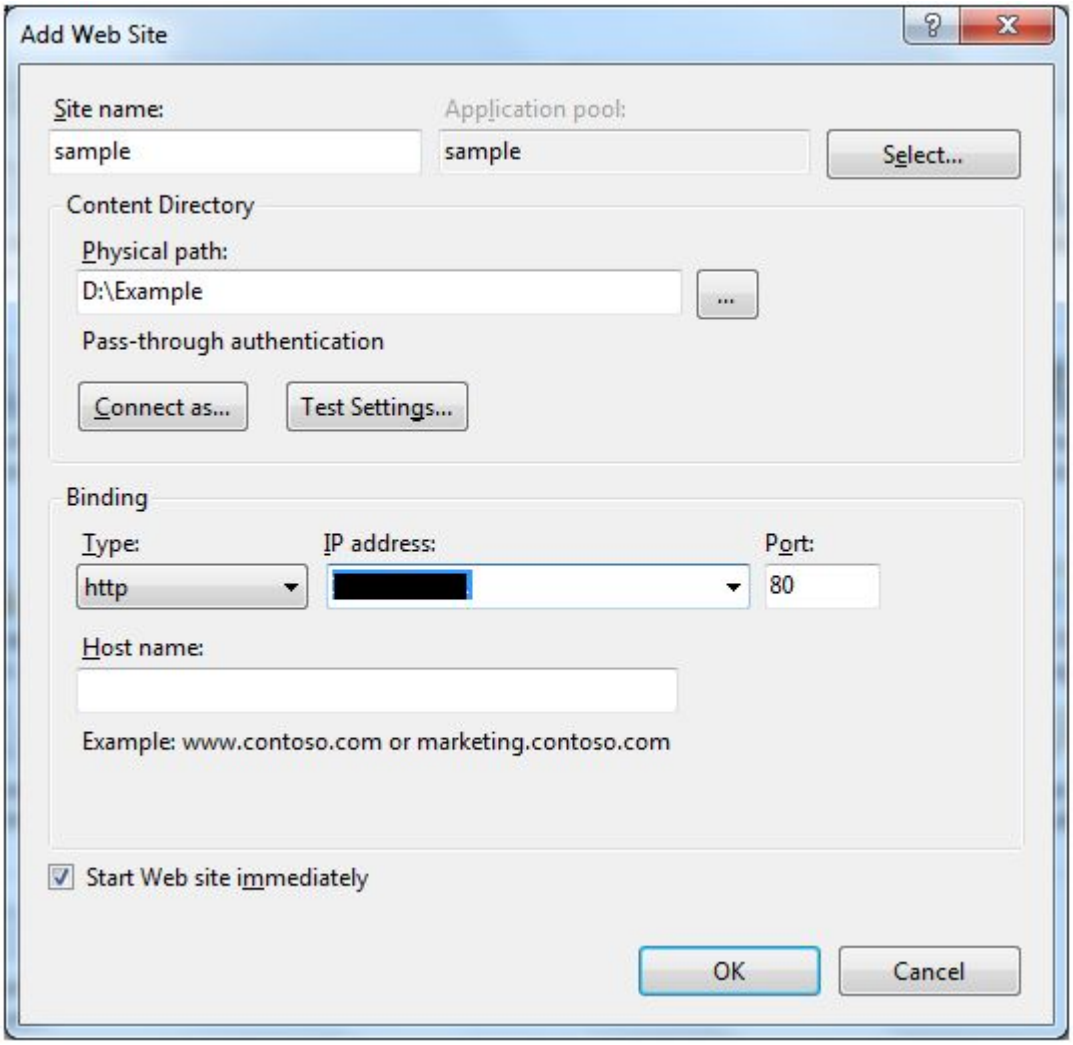


Figure 5: Add web site window

Add site name, path of that site folder, then assign the IP address or enter the host name of that particular site like www.example.com and then hit ok.

After the process is completed your site is placed under default web site under Sites.

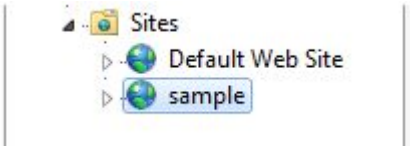


Figure 6

At the same time there is one application pool is automatically created while adding the web site.

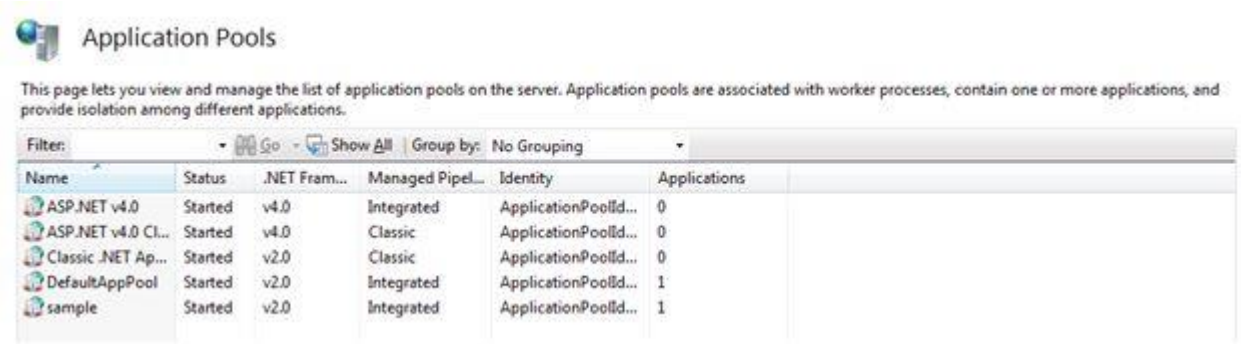


Figure 7: Application Pool

Here I have a simple web page like default.aspx in your website there is a database connectivity available which means then you have to follow the steps to add the database into application pool.

Step 5: Database Connectivity

In the application pool panel right click the application name like sample, then choose Advanced Settings option.

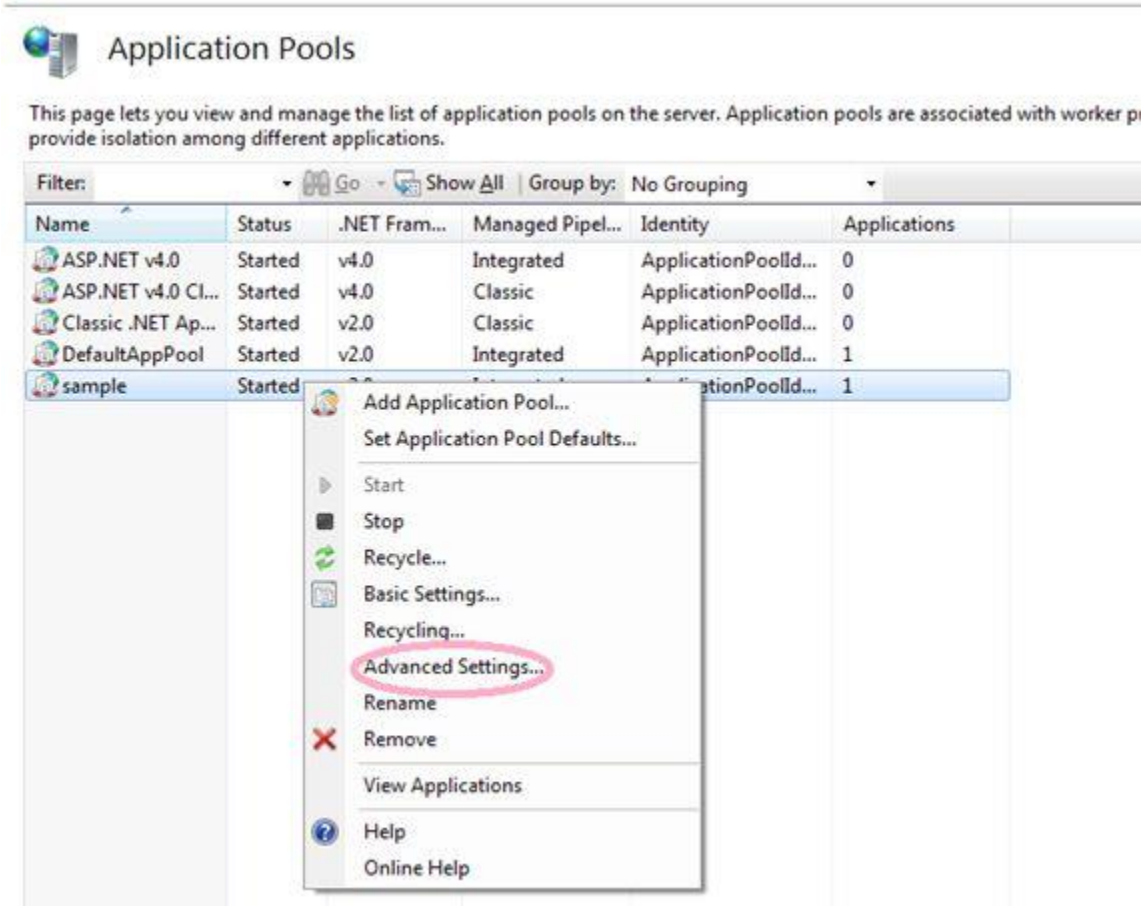


Figure 8: Advanced settings Option

The settings window will open in that window select .net framework version like v2.0. Based on our pages it automatically chooses the framework version. Then select managed pipeline mode is Integrated or Classic.

Then in the process model tab select Identity then choose the following:

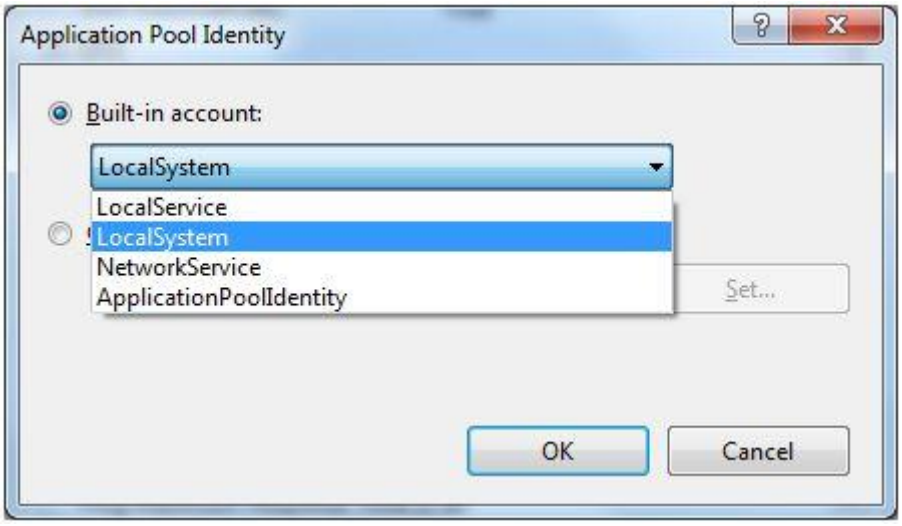


Figure 9: Pool Identity

Select LocalSystem because our database connectivity is placed in our system SQL Server so we need to choose LocalSystem then only our database is connected. Then hit OK to close the window.

Step 6: Conclusion.

In the Actions panel choose Browse link

Figure 10: Browse

Now your site will appear in the browser like the following figure:

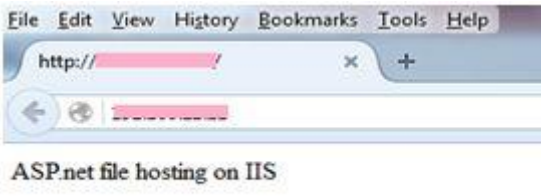


Figure 11: Browser

Now your application is ready to be browsed on any other system through domain name (Globally) or locally connected LAN network (IP Address) computers.

Learning outcomes (What I have learnt):

1. **Learnt steps to host a website**
2. **Learnt how to host a website on iis**