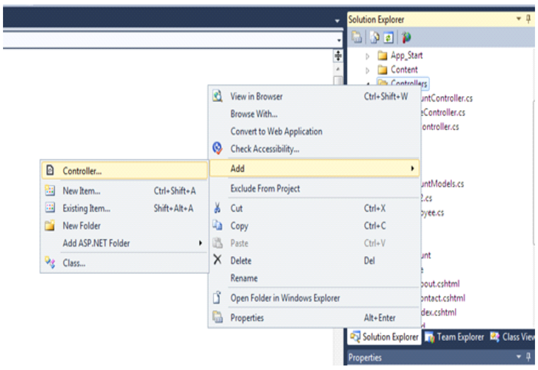
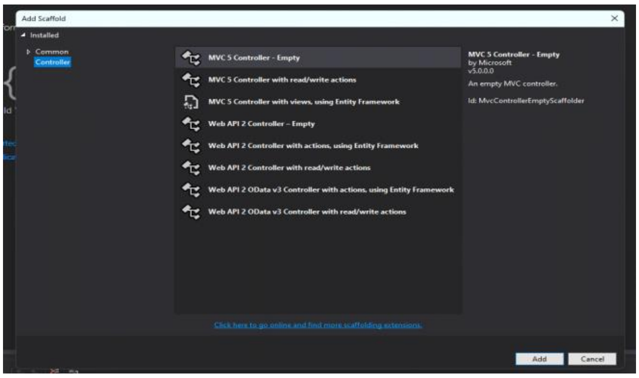
## 1. Design & Develop ASP.NET MVC Controller.

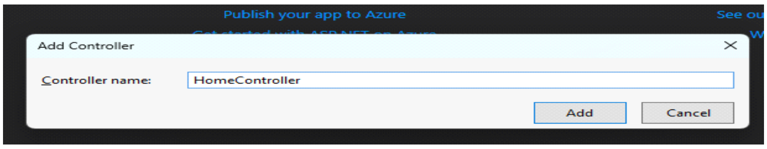
How to create a controller:-

Go to solution explorer Right-click on “Controller” Folder >> Click on Add >> click on “Controller” as follow.




Add Scaffold
Installed
Common
Controller
MVC 5 Controller - Empty
MVC 5 Controller with read/write actions
MVC 5 Controller with views, using Entity Framework
Web API 2 Controller – Empty
Web API 2 Controller with actions, using Entity Framework
Web API 2 Controller with read/write actions
Web API 2 OData v3 Controller with actions, using Entity Framework
Web API 2 OData v3 Controller with read/write actions
Click here to go online and find more scaffolding extensions.
Add
Cancel

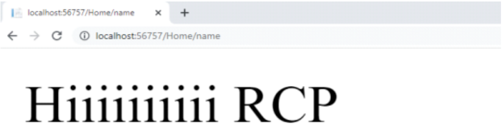
MVC 5 Controller - Empty
by Microsoft
v5.0.0.0
An empty MVC controller.
Id: MvcControllerEmptyScaffolder



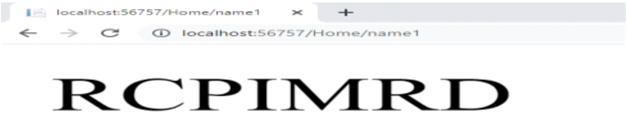
Controller name: HomeController

HomeController.cs

```
using System.Linq;
using System.Web;
using System.Web.Mvc;
namespace controllerpro.Controllers
{
   public class HomeController : Controller
   {
      // GET: Home

      public string name()
      {
         return "Hiiiiiiiiii RCP";
      }
      public string name1()
      {
         return "RCPIMRD";
      }
   }
}
```



```
Hiiiiiiiiii RCP
```



```
RCPIMRD
```

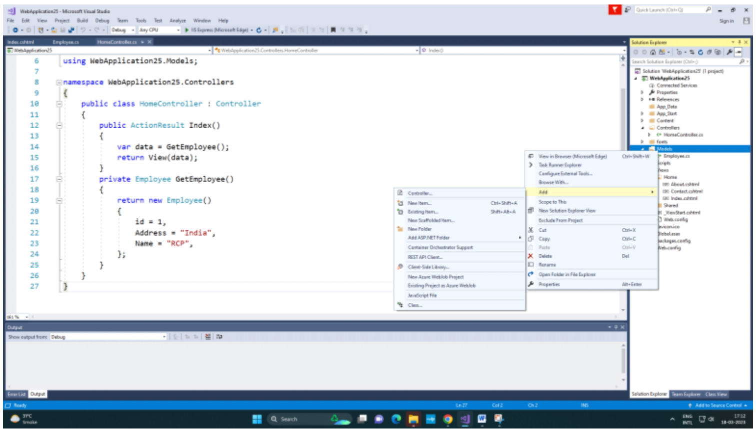
**2. Design & Develop Model Templates using Metadata for data values.**

- Steps to create ASP.NET MVC 4 Application using Visual Studio.
- Open => Visual Studio and go to File >> New >> Click on Project as follow.
- Select “ASP.NET Web Application” and provide a meaningful name like “WebApplication25” and Click on “Ok” button.
- Select the “Empty” template and check the “MVC” checkbox from “New Web Application” window as follow.

**Below file is HomeController.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using WebApplication25.Models;   //this file as to be included

namespace WebApplication25.Controllers
{
   public class HomeController : Controller
   {
      public ActionResult Index()
      {
         var data = GetEmployee();
         return View(data);
      }
      private Employee GetEmployee()
      {
         return new Employee()
         {
            id = 1,
            Address = "India",
            Name = "RCP",
         };
      }
   }
 }
```

- After coding above code, now to add Model class.
- Look for the Model is Solution Explorer.
- Right click on Model Click on Add Select Class give name to the class Employee.cs

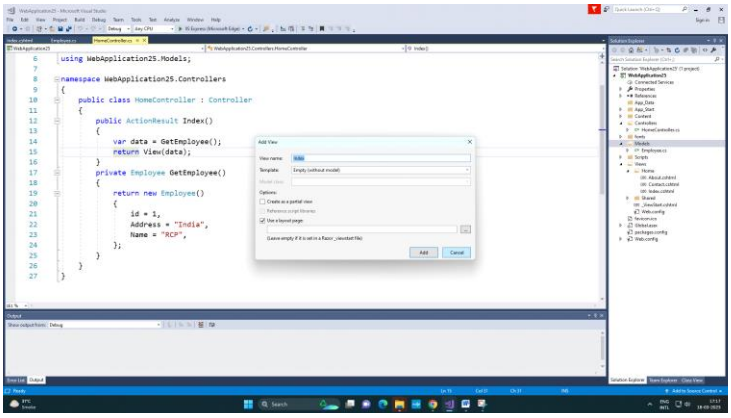


```
using WebApplication25.Models;

namespace WebApplication25.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            var data = GetEmployee();
            return View(data);
        }
        private Employee GetEmployee()
        {
            return new Employee()
            {
                id = 1,
                Address = "India",
                Name = "RCP",
            };
        }
    }
}
```

### Do the following code in Employee.cs file

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

 namespace WebApplication25.Models
 {
     public class Employee
     {
         public int id { get; set; }
         public string Name { get; set; }
         public string Address { get; set; }
     }
 }
```

Right click in the Index() and add Index.cshtml



Add View

View name: Index

Template: Empty (without model)

Options:

□ Create as a partial view

□ Reference script libraries

☑ Use a layout page

(Leave empty if it is set in a Razor _viewstart file)

### Index.cshtml

- Make sure we add created model in the Index.cshtml
- First line @model contains the Applicationname. Models.ClassName
- In this example WebApplication25 is Application name and Employee is ClassName

```
@model  WebApplication25.Models.Employee
@{
    ViewBag.Title = "Index";
}

<body>
    <center>
        <h1>Model Templates using Metadata for data values. </h1>
        <table style="border:5px solid black; ">
            <tr>
                <th style="border:5px solid red;">ID</th>
                <th style="border:5px solid red;">@Model.id</th>
            </tr>
            <tr>
                <th style="border:5px solid red;">Name</th>
                <th style="border:5px solid red;">@Model.Name</th>
            </tr>
            <tr>
                <th style="border:5px solid red;">Address</th>
                <th style="border:5px solid red;">@Model.Address</th>
            </tr>
        </table>
    </center>
</body>
```

**3. Demonstrate ASP.NET MVC for Model Validation.**

Open Microsoft Visual studio Click on file NewProject Dialog Box will appear, Select Asp.net Web application (.net Framework)  click ok next dialog box will appear select MVC and click ok. New Web application will be open

You can rename the controller name by :- Right click on HomeController and select rename option and you can rename it. In this example StudentController.cs name is given

Following is StudentController.cs file

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

namespace WebApplication7.Controllers
{
    public class StudentsController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult About()
        {
            ViewBag.Message = "Your application description page.";

            return View();
        }

        public ActionResult Contact()
        {
            ViewBag.Message = "Your contact page.";

            return View();
        }
    }
}
```

Open  Index.cshtml file by right clicking on view() in to ActionResult Index() and do the following code

Index.cshtml file

```
@model WebApplication7.Models.Student
@{
    ViewBag.Title = "Index";
}

<h2>Validation Form</h2>
```

```
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">

        <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.Contact, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Contact, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Contact, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Create" class="btn btn-default" />
            </div>
        </div>
    </div>
}
<div>
    @Html.ActionLink("Back to List", "Index")
</div>
@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

Right click on Models in Solution Explorer click on Add click on class Rename the model Class As Student.cs and do the following code

Student.cs file

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

namespace WebApplication7.Models
{
    public class Student
    {
        public int ID { get; set; }
        // -- Validating Student Name
        [Required(ErrorMessage = "Name is required")]
        [MaxLength(12)]
        public string Name { get; set; }
        // -- Validating Email Address
        [Required(ErrorMessage = "Email is required")]
        [EmailAddress(ErrorMessage = "Invalid Email Address")]
        public string Email { get; set; }
        // -- Validating Contact Number
        [Required(ErrorMessage = "Contact is required")]
        [DataType(DataType.PhoneNumber)]
        [RegularExpression(@"^\(?([0-9]{3})\)?[-. ]?([0-9]{3})[-. ]?([0- 9]{4})$", ErrorMessage = "Not a valid Phone number")]
        public string Contact { get; set; }
    }
}
```

**4. Design & develop to demonstrate working with razor engine.**

Open Microsoft Visual studio Click on file NewProject Dialog Box will appear, Select Asp.net Web application (.net Framework) click ok next dialog box will appear select MVC and click ok. New Web application will be open

HomeController.cs file

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using WebApplication28.Models;//This Line as to be included

namespace WebApplication28.Controllers
{
   public class HomeController : Controller
   {
      public ActionResult Index()
      {
         return View();
      }
   }
}
```

Now add model class right click on model in solution explorer click on add button-then select class and name it as Student_Model.cs

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

namespace WebApplication28.Models
{
   public class Student_Model
   {
      public int student_id { get; set; }
      public string studnet_name { get; set; }
      public string student_address { get; set; }
      public string student_context { get; set; }
      public string student_city { get; set; }
      public string male { get; set; }
      public string Female { get; set; }
   }
}
```

Next go to HomeController and right click on View() in Index().Then do the coding in Index.cshtml file

```
@model WebApplication28.Models.Student_Model //Include this line after adding
                                                 //Student_Model.cs
@{
   ViewBag.Title = "Index";
}


<h1>Student Information</h1>
<div class="row">

    <label>Student Name</label>
    @Html.TextBoxFor(m => m.studnet_name, new { @class = "form-control" })
    <label>Student Address</label>
    @Html.TextAreaFor(m => m.student_address, new { @class = "form-control" })
    <label>Student Contact</label>
    @Html.TextBoxFor(m => m.student_context, new { @class = "form-control" })

    <label>Student City</label>
    @Html.DropDownListFor(m=>m.student_city , new List<SelectListItem> {
    new SelectListItem { Text="--Select City--",Value="0"},
    new SelectListItem { Text="Pune",Value="1"},
    new SelectListItem { Text="Mumbai",Value="2"},
    new SelectListItem { Text="Indore",Value="3"},
    new SelectListItem { Text="Dhule",Value="4"}
    } ,new {@class="form-control" })

    <label>Male</label>
    @Html.RadioButtonFor(m => m.male, "Male", new { @class = "form-control" })
    <label>Female</label>
    @Html.RadioButtonFor(m => m.male, "Female", new { @class = "form-control"
})

</div>
<hr />
<div class="row">
    <button type="submit" class="btn btn-danger">Save Record</button>

</div>
```

**5. Design & develop to demonstrate working in html helper.**

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

namespace htmlhelper.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult About()
        {
            ViewBag.Message = "Your application description page.";

            return View();
        }

        public ActionResult Contact()
        {
            ViewBag.Message = "Your contact page.";

            return View();
        }
    }
}
```

Index.cshtml

```
@{
    ViewBag.Title = "Index";
}
<html>
<h2>Index</h2>
<body>
    <div>
        @helper MyListHelper(string[] str)
        {
            <ul>
                @foreach (var item in str)
                {
                    <li>@item</li>
                }
            </ul>
    }
    <div>
        <lable>Ex 1-NAME LIST</lable>
        <div>
            @MyListHelper(new string[] { "TOM", "HARRY", "JOHN" })
        </div>
    </div>

    <div>
        <lable>
            Ex 2-programming Lang</label>
            <div>
                @MyListHelper(new string[] { ".net", "MVC", "JAVA" })
            </div>
    </div>
</body>
</html>
```

**About.cshtml**

```
@{
    ViewBag.Title = "About";
}

<html>
<h2>About</h2>

<body>
    <div>
        @helper MyListHelper(string[] str)
        {
            <ul>
                @foreach (var item in str)
                {
                    <li>@item</li>
                }
            </ul>
        }

</html>
@{
   string[] strBooks = new string[] { "C#.NET", "ASP.NET MVC", "ASP.NET CORE",
"VB.NET", "WEB API" };
}
<div id="div1" style="background-color:yellow;">
    Book Name List: @MyListHelper(strBooks)
</div>
```

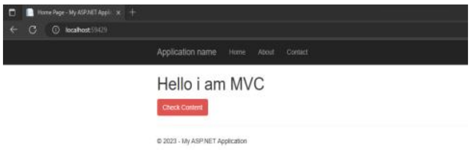
## 6. Design & Develop to demonstrate adding dynamic content to a razor view.

HomeController.cs File

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
namespace DynamicRazorView.Controllers
{
   public class HomeController : Controller
   {
         public ActionResult Index()
         {
            return View();
         }
         public ActionResult Content()
         {
            @TempData["Check"] = "Hello i am MVC";
            return RedirectToAction("index");
         }
         public ActionResult About()
         {
            ViewBag.Message = "Your application description page.";
            return View();
         }
         public ActionResult Contact()
         {
            ViewBag.Message = "Your contact page.";
            return View();
         }
   }
}
```

Index.cshtml file

```
@{
         ViewBag.Title = "Home Page";
}
@using (Html.BeginForm("Content", "Home", FormMethod.Post))
{
         <h1>@TempData["Check"]</h1>
           <button type="submit" class="btn btn-danger">Check Content</button>
}
```



Hello i am MVC

© 2023 - My ASP.NET Application

**7. Design & develop to demonstrate partial views.**

HomeController.cs  file

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

namespace PartialView1.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult About()
        {
            ViewBag.Message = "Your application description page.";

            return View();
        }

        public ActionResult Contact()
        {
            ViewBag.Message = "Your contact page.";

            return View();
        }
    }
}
```

Index.cshtml file

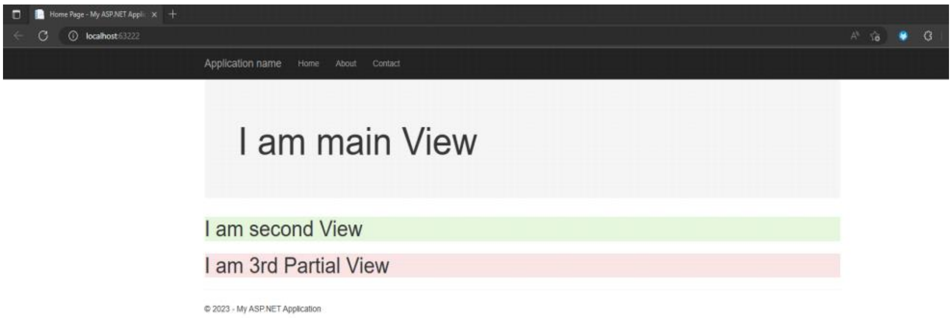
```
@{
    ViewBag.Title = "Home Page";
}

<div class="jumbotron">
    <h1>I am main View</h1>

</div>
<div class="card-body bg-success">
    @Html.Partial("~/Views/Home/About.cshtml")
</div>
<div class="card-body bg-danger">
    @Html.Partial("~/Views/Home/Contact.cshtml")
</div>
```

About.cshtml file

```
@{
    ViewBag.Title = "About";
}
<h1>I am second View</h1>
```

Contact.cshtml file

```
@{
    ViewBag.Title = "Contact";
}
<h1>I am 3rd Partial View</h1>
```



I am main View

I am second View

I am 3rd Partial View

© 2023 - My ASP.NET Application

**8. Demonstrate routing mechanism in ASP.NET MVC application.**

**HomeController.cs**

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

namespace _8_Routing_Pattern.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult About()
        {
            ViewBag.Message = "Your application description page.";

            return View();
        }

        public ActionResult Contact()
        {
            ViewBag.Message = "Your contact page.";

            return View();
        }
    }
}
```

**App_Start\RouteConfig.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Routing;

namespace _8_Routing_Pattern
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}",
                defaults: new { controller = "Home", action = "Index" }
            );
            routes.MapRoute(
              name: "Action",
               url: "{action}/{controller}/{id}",
           defaults: new { action = "Index", controller = "Home", id = UrlParameter.Optional },
            constraints: new { id = @"\d+" }
          );
        }
    }
}
```

**Global.asax.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Optimization;
using System.Web.Routing;

namespace _8_Routing_Pattern
{
    public class MvcApplication : System.Web.HttpApplication
    {
        protected void Application_Start()
        {
            AreaRegistration.RegisterAllAreas();
            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);
        }
    }
}
```

**9. Demonstrate routing with respect to using parameters, using constraints.**

**HomeController.cs**

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

namespace _8_Routing_Pattern.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult About(int ID)
        {
            ViewBag.Message = "Your application description page.";

            return View();
        }
        public ActionResult Contact()
        {
            ViewBag.Message = "Your contact page.";

            return View();
        }
    }
}
```

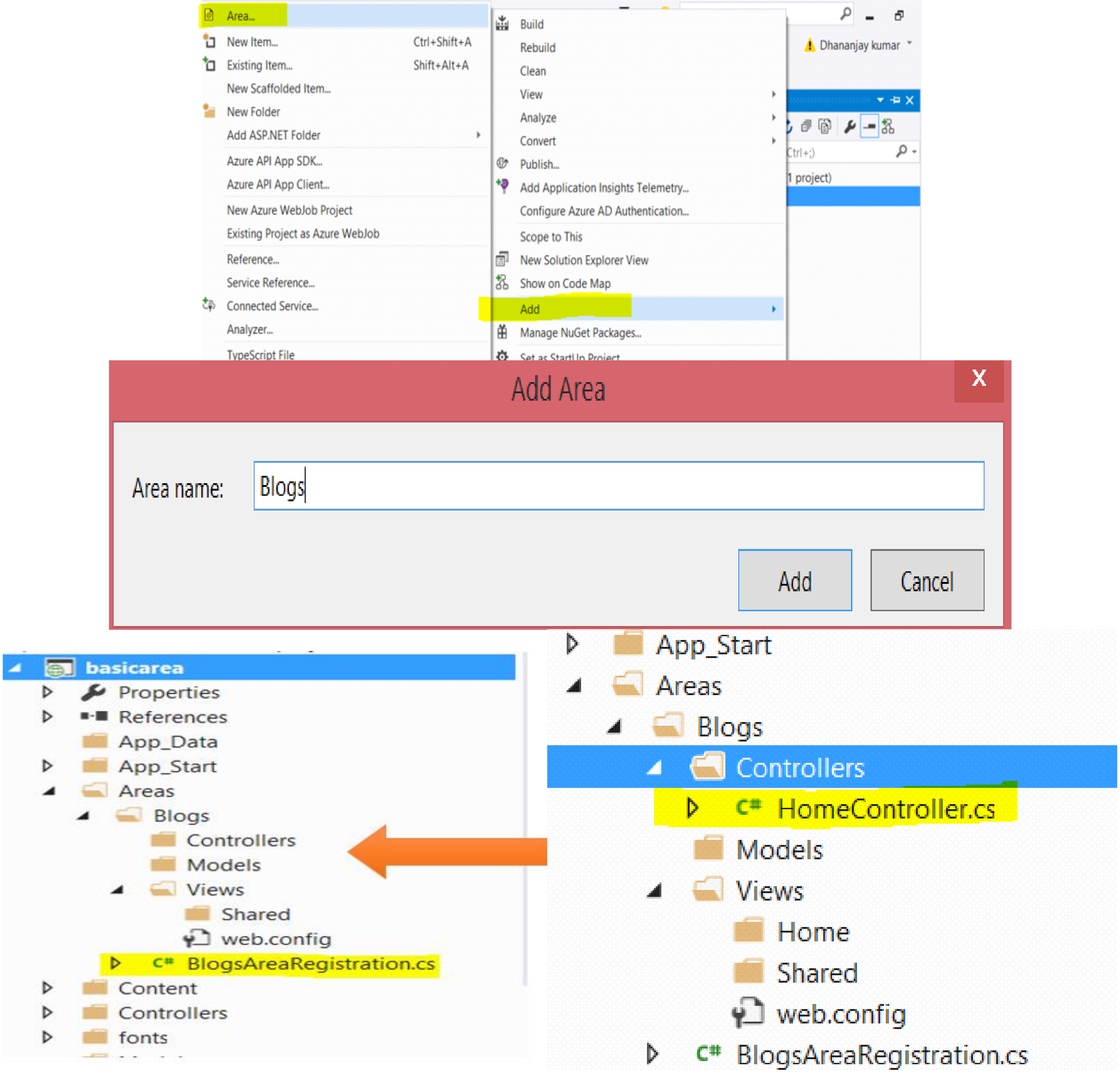
**App_Start\RouteConfig.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Routing;

namespace _8_Routing_Pattern
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}",
                defaults: new { controller = "Home", action = "Index" }
            );
            routes.MapRoute(
              name: "Action",
               url: "{action}/{controller}/{id}",
           defaults: new { action = "Index", controller = "Home", id = UrlParameter.Optional },
            constraints: new { id = @"\d+" }
          );
        }
    }
}
```

**Global.asax.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Optimization;
using System.Web.Routing;

namespace _8_Routing_Pattern
{
    public class MvcApplication : System.Web.HttpApplication
    {
        protected void Application_Start()
        {
            AreaRegistration.RegisterAllAreas();
            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);
        }
    }
}
```

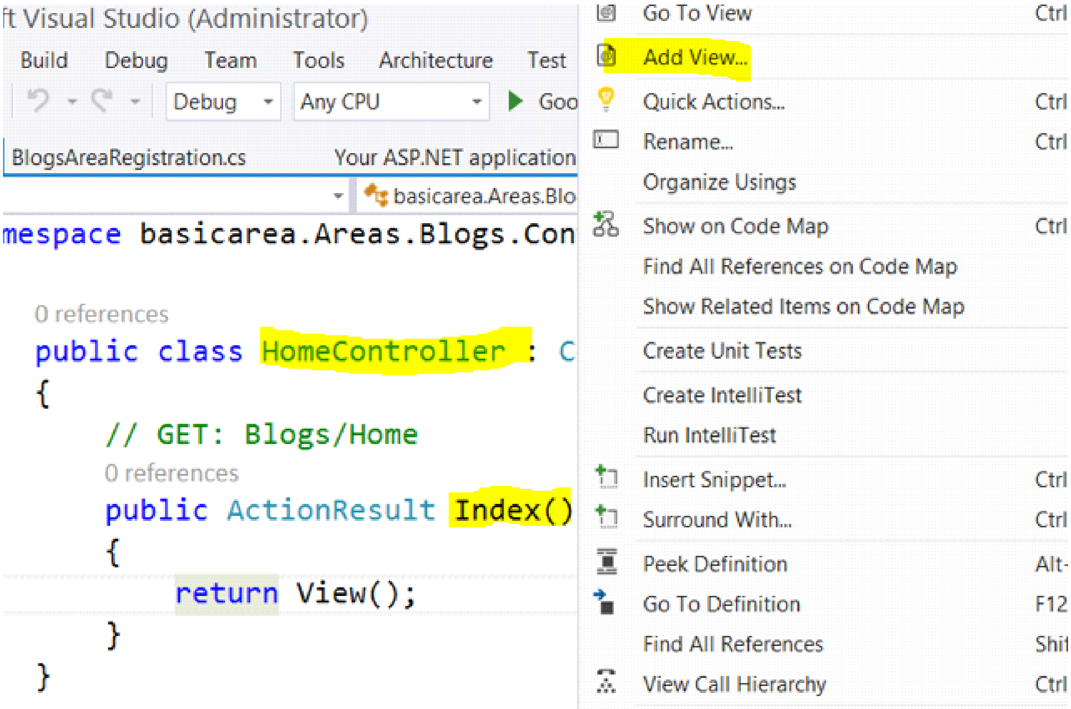
## 10. Demonstrate actions in areas for ASP.NET MVC application.



Area name: Blogs

**HomeController:**

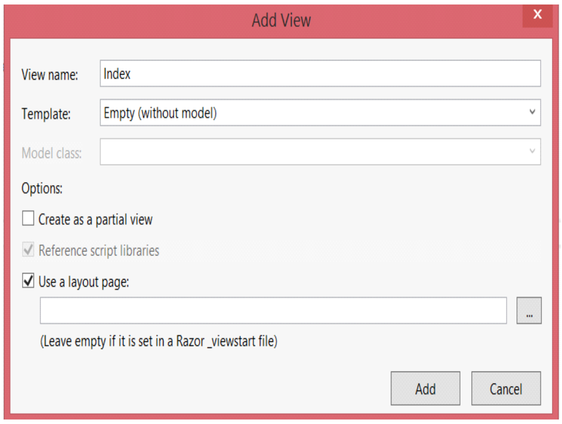
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace Assi10.Areas.Blogs.Controllers
{
    public class HomeController : Controller
    {
        // GET: Blogs/Home
        public ActionResult Index()
        {
            return View();
        }
    }
}
```



```
mespace basicarea.Areas.Blogs.Con

    0 references
    public class HomeController : C
    {
        // GET: Blogs/Home
        0 references
        public ActionResult Index()
        {
            return View();
        }
    }
```

**Index.cs:**

```
@{
    ViewBag.Title = "Index";
}

<h2>Blogs Area Home Index</h2>
```



Add View

View name: Index

Template: Empty (without model)

Model class:

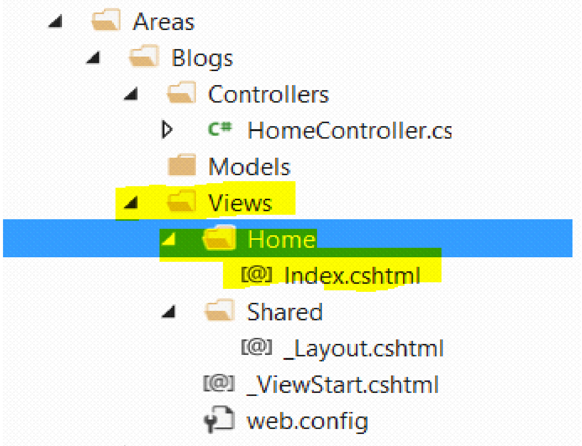
Options:

☐ Create as a partial view

☑ Reference script libraries

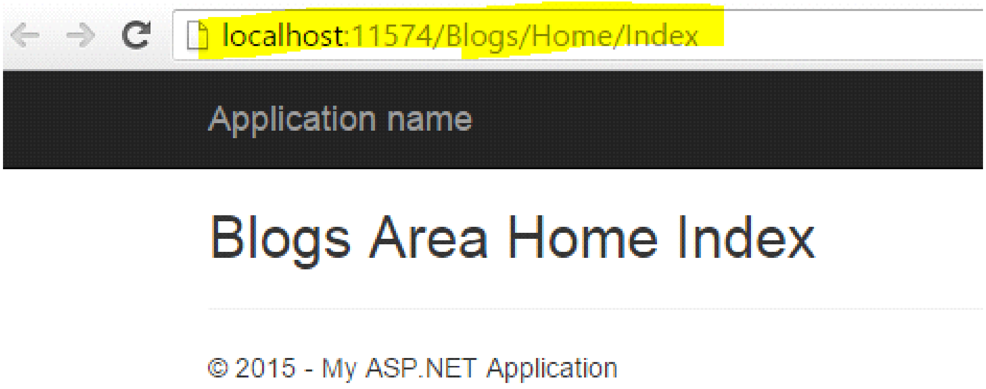
☑ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)



**Globle.asax:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Optimization;
using System.Web.Routing;

namespace Assi10
{
    public class MvcApplication : System.Web.HttpApplication
    {
        protected void Application_Start()
        {
            AreaRegistration.RegisterAllAreas();
            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);
        }
    }
}
```

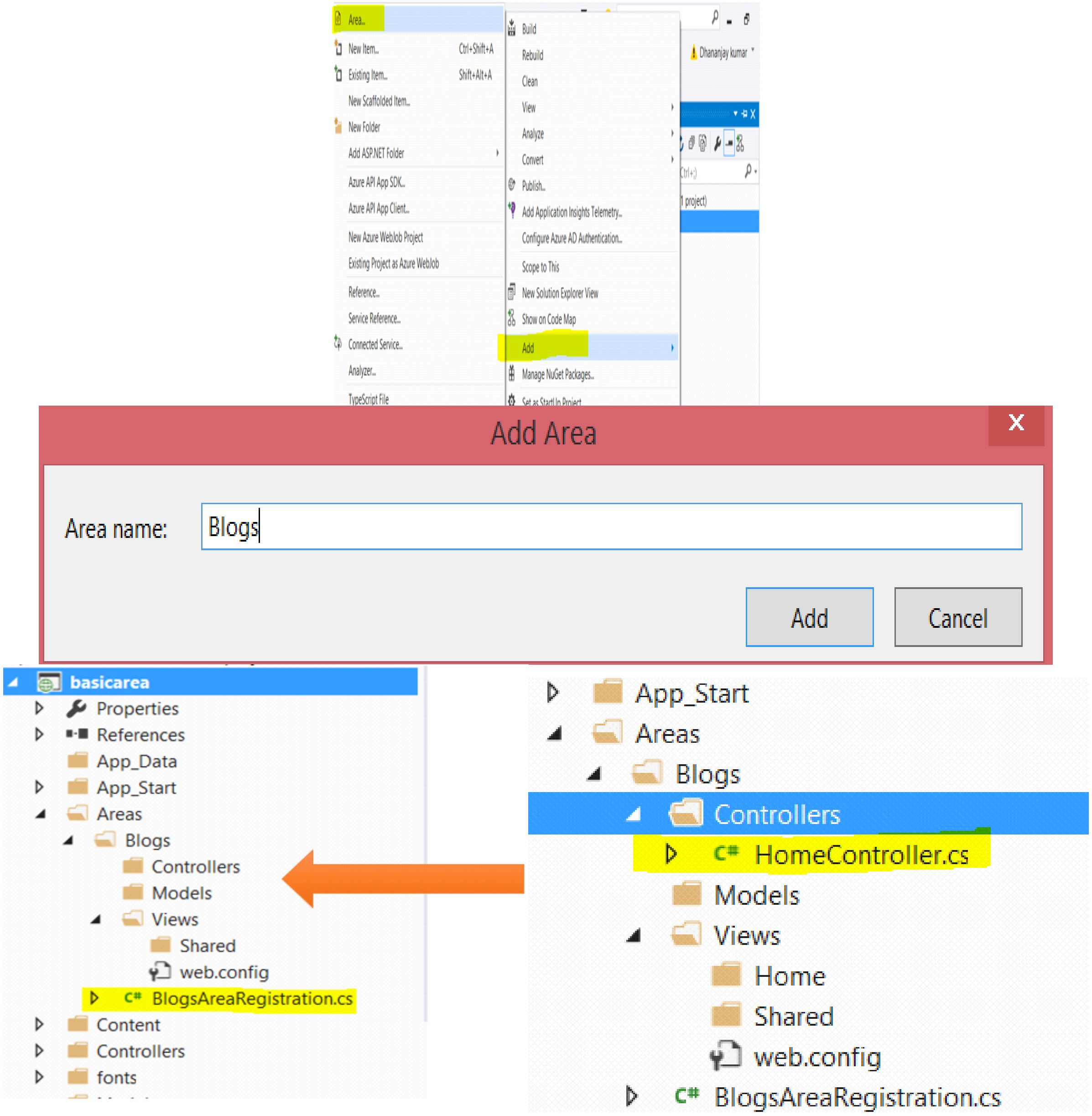


localhost:11574/Blogs/Home/Index

Blogs Area Home Index

© 2015 - My ASP.NET Application

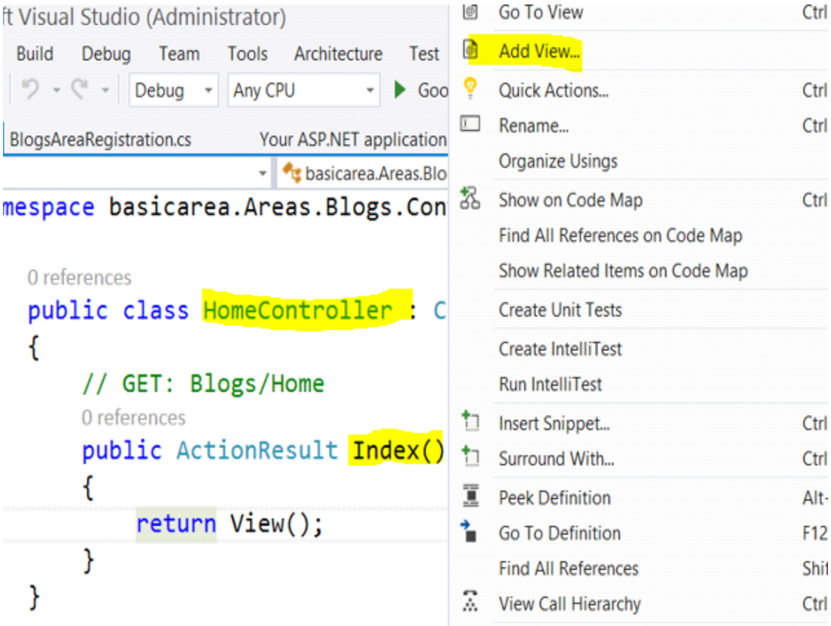
## 11. Demonstrate routing & URL generation with areas in ASP.NET MVC.



Area name: Blogs

**HomeController:**

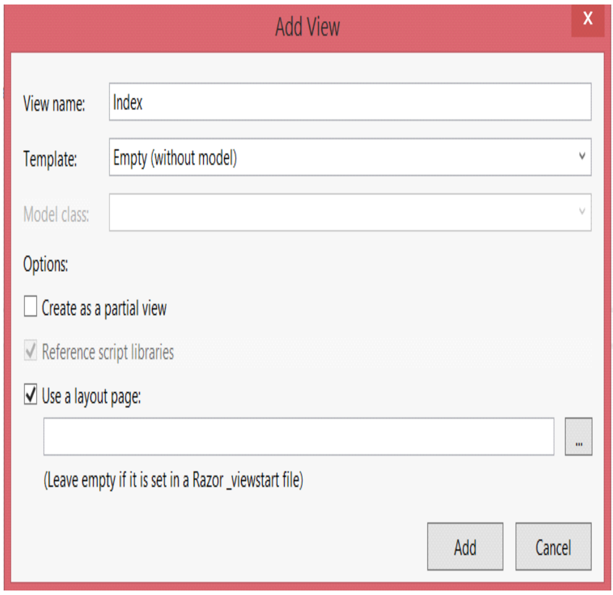
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace Assi10.Areas.Blogs.Controllers
{
    public class HomeController : Controller
    {
        // GET: Blogs/Home
        public ActionResult Index()
        {
            return View();
        }
    }
}
```



```
mespace basicarea.Areas.Blogs.Con

  0 references
  public class HomeController : C
  {
      // GET: Blogs/Home
      0 references
      public ActionResult Index()
      {
          return View();
      }
  }
```

**Index.cs:**

```
@{
    ViewBag.Title = "Index";
}

<h2>Blogs Area Home Index</h2>
```



Add View

View name: Index

Template: Empty (without model)

Model class:

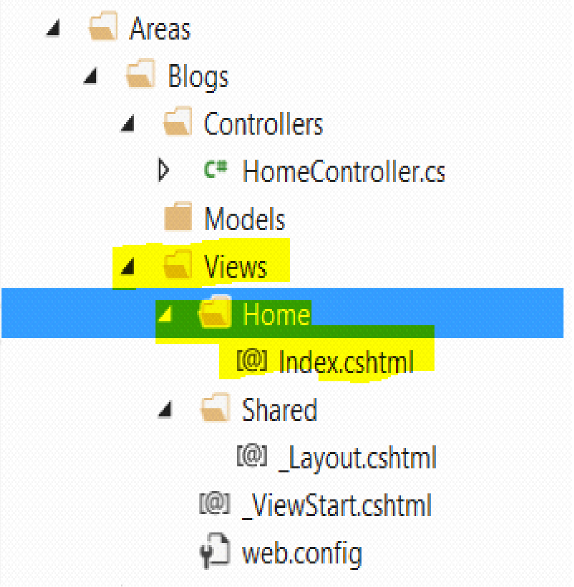
Options:

□ Create as a partial view

☑ Reference script libraries

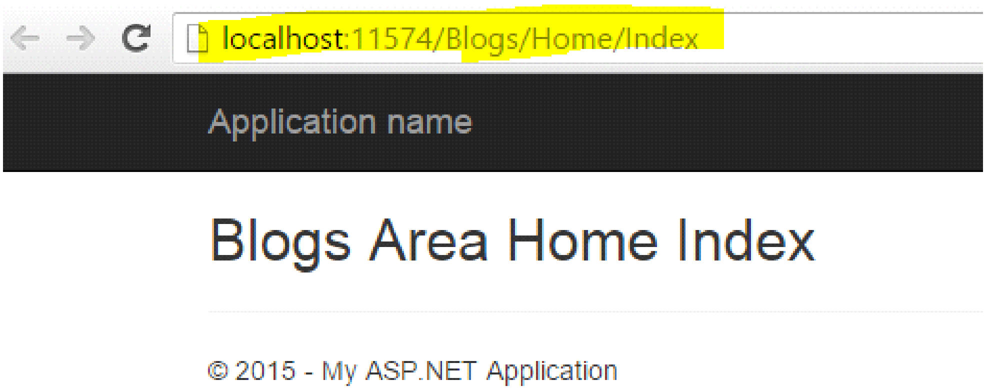
☑ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)



**Globle.asax:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Optimization;
using System.Web.Routing;

namespace Assi10
{
    public class MvcApplication : System.Web.HttpApplication
    {
        protected void Application_Start()
        {
            AreaRegistration.RegisterAllAreas();
            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);
        }
    }
}
```



localhost:11574/Blogs/Home/Index

Application name

Blogs Area Home Index

© 2015 - My ASP.NET Application

**12 Design & Develop sample ASP.NET MVC application using JQuery.**

HomeController:

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

namespace _1jquery.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult About()
        {
            ViewBag.Message = "Your application description page.";

            return View();
        }

        public ActionResult Contact()
        {
            ViewBag.Message = "Your contact page.";

            return View();
        }
    }
}
```

Index.cs:

```
@{
    ViewBag.Title = "Index";
}
<script src="~/Scripts/jquery-3.3.1.min.js"></script>
<script src="~/Scripts/JavaScript.js"></script>

<div id=" element">
    Hello Geeks Welcome To GeeksforGeeks
</div>
@using (Html.BeginForm())
{
    <H1>Hquery Using MVC</H1>
    @Html.Label("email", "Email:");
    @Html.TextBox("email");
    <input type="submit" id="submitButton" value="send" />
}
<body>
    <button type="button" class="hide-btn">Hide Paragraphs</button>
    <button type="button" class="show-btn">Show Paragraphs</button>
</body>
<script>
    $(document).ready(function () {
        $("#submitButton").on("click", function (e) {
            var email = $("#email").val();
            if (email == "") {
                e.preventDefault();
                alert("Please enter your email address first.");
            }
        });
    });
</script>
```

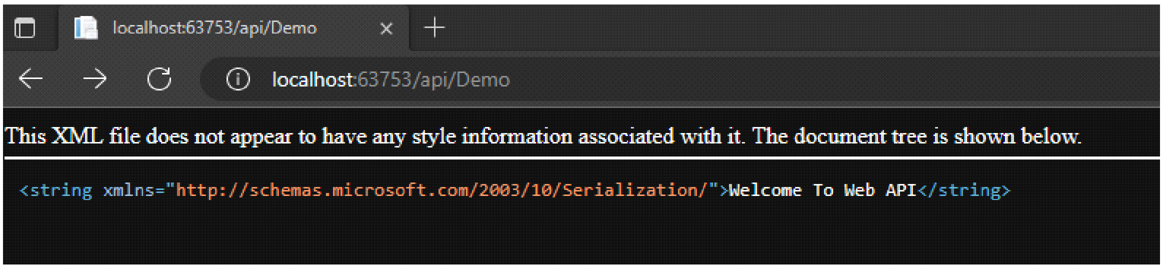
JavaScript.js:

```
$(document).ready(function () {
    //Display alert message after hiding paragraphs
    $(".hide-btn").click(function () {
        $("p").hide("slow", function () {
            //Code to be Executed
            alert("The hide effect is completed .");

        });
    });
    //Display alert message after showing paragraphs
    $(".show-btn").click(function () {
        $("p").show("slow", function () {
            //code to be executed
            alert("The show effect is completed.");
        });
    });

});
```

## 13. Design & Develop web API controllers for ASP.NET MVC application.

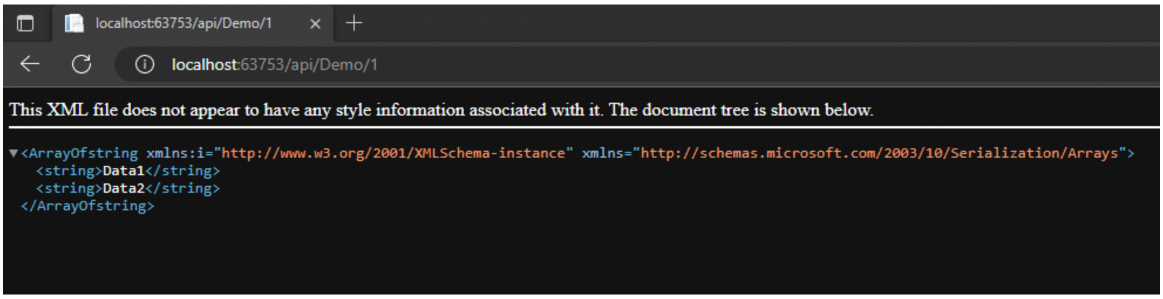
DemoController:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;

namespace webapi1.Controllers
{
   public class DemoController : ApiController
   {
      public string Get()
      {
         return "Welcome To Web API";
      }
      public List<string> Get(int Id)
      {
         return new List<string> {
            "Data1",
            "Data2"
         };
      }
   }
}
```



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Welcome To Web API</string>
```



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<ArrayOfstring xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
   <string>Data1</string>
   <string>Data2</string>
 </ArrayOfstring>
```

**14. Demonstrate database connectivity using ASP.NET MVC application.**

**StudentController.cs**

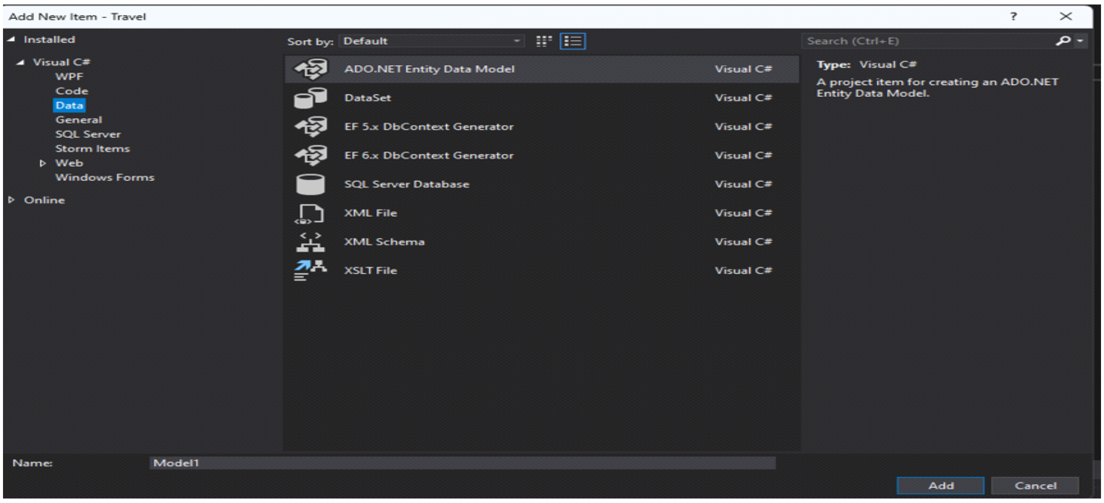
```
using database_connection.DataContext;
using database_connection.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace database_connection.Controllers
{
    public class StudentController : Controller
    {
        // GET: Student
        public ActionResult Index()
        {
            return View();
        }
        public ActionResult save(Student_Model model)
        {
            using (Db_AssiEntities _db = new Db_AssiEntities())
            {
                Student_Table table = new Student_Table
                {
                    ID = model.ID,
                    Name = model.Name,
                    Contact=model.Contact,
                    Email=model.Email
                };
                _db.Entry(table).State = System.Data.Entity.EntityState.Added;
                _db.SaveChanges();
            }
            return RedirectToAction("Index");
        }
    }
}
```

**Student_Model.cs**

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

namespace database_connection.Models
{
    public class Student_Model
    {
        public int ID { get; set; }
        // -- Validating Student Name
        [Required(ErrorMessage = "Name is required")]
        [MaxLength(12)]
        public string Name { get; set; }
        // -- Validating Email Address
        [Required(ErrorMessage = "Email is required")]
        [EmailAddress(ErrorMessage = "Invalid Email Address")]
        public string Email { get; set; }
        // -- Validating Contact Number
        [Required(ErrorMessage = "Contact is required")]
        [DataType(DataType.PhoneNumber)]
        [RegularExpression(@"^\(?([0-9]{3})\)?[-. ]?([0-9]{3})[-. ]?([0-9]{4})$", ErrorMessage = "Not a valid Phone number")]
        public string Contact { get; set; }

    }
}
```

**Index.cs**

```
@model database_connection.Models.Student_Model
@{
    ViewBag.Title = "Index";
}

@using (Html.BeginForm("save", "Student", FormMethod.Post))
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Student</h4>
        <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.Contact, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Contact, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Contact, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input id="Submit" type="submit" value="submit" />
            </div>
        </div>
    </div>
}
<div>
    @Html.ActionLink("Back to List", "Index")
</div>
@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

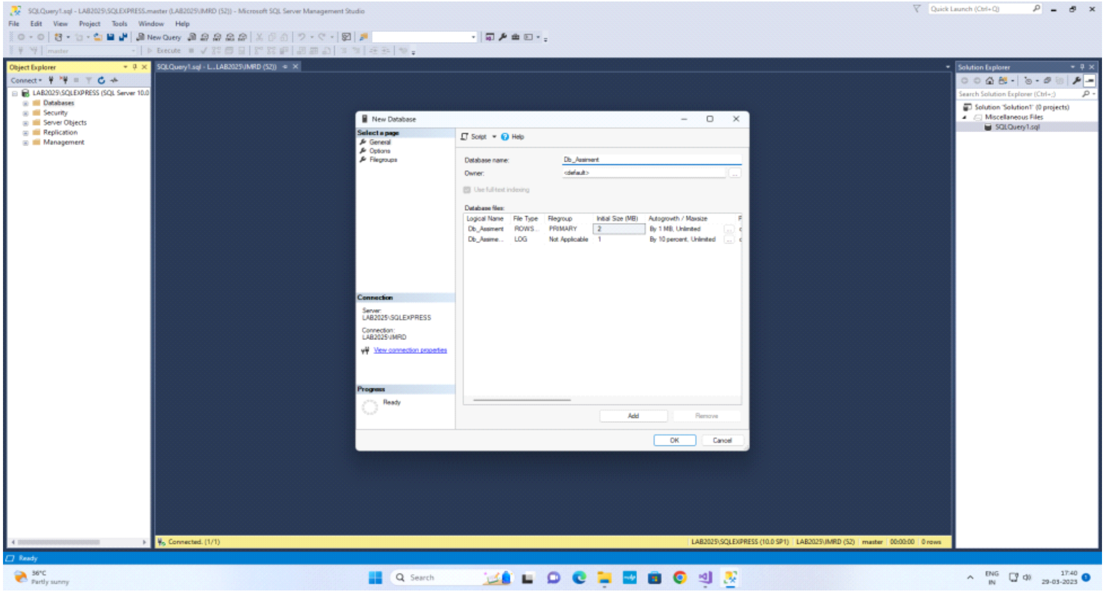


ADO.NET Entity Data Model — Visual C#
DataSet — Visual C#
EF 5.x DbContext Generator — Visual C#
EF 6.x DbContext Generator — Visual C#
SQL Server Database — Visual C#
XML File — Visual C#
XML Schema — Visual C#
XSLT File — Visual C#

**Type:** Visual C#

A project item for creating an ADO.NET Entity Data Model.

Name: Model1



Database name: Db_Assiment

Owner: <default>

□ Use full-text indexing

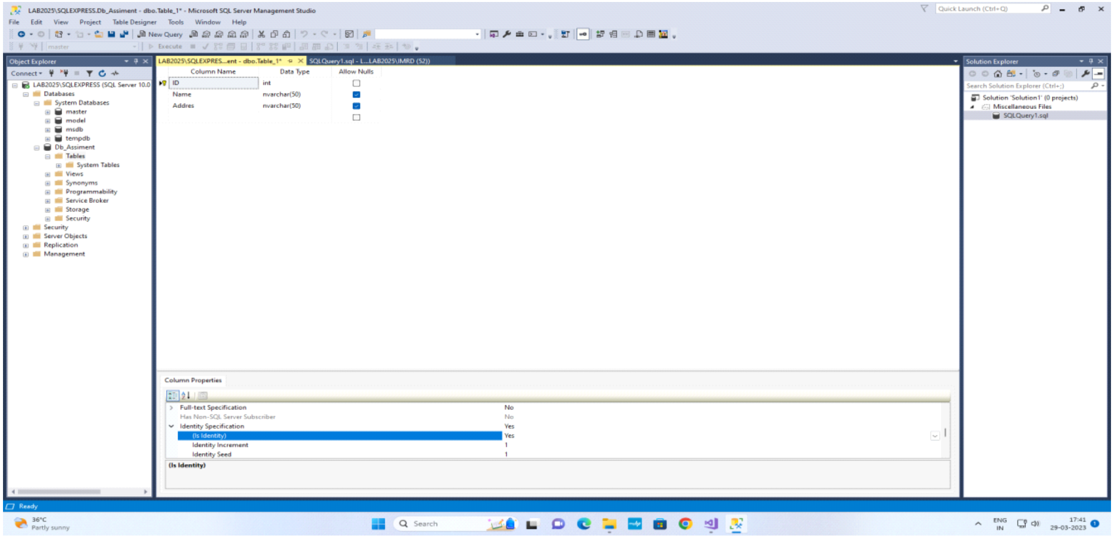
Database files:

| Logical Name | File Type | Filegroup | Initial Size (MB) | Autogrowth / Maxsize |
|---|---|---|---|---|
| Db_Assiment | ROWS... | PRIMARY | 2 | By 1 MB, Unlimited |
| Db_Assime... | LOG | Not Applicable | 1 | By 10 percent, Unlimited |

Server:
LAB2025\SQLEXPRESS

Connection:
LAB2025\IMRD

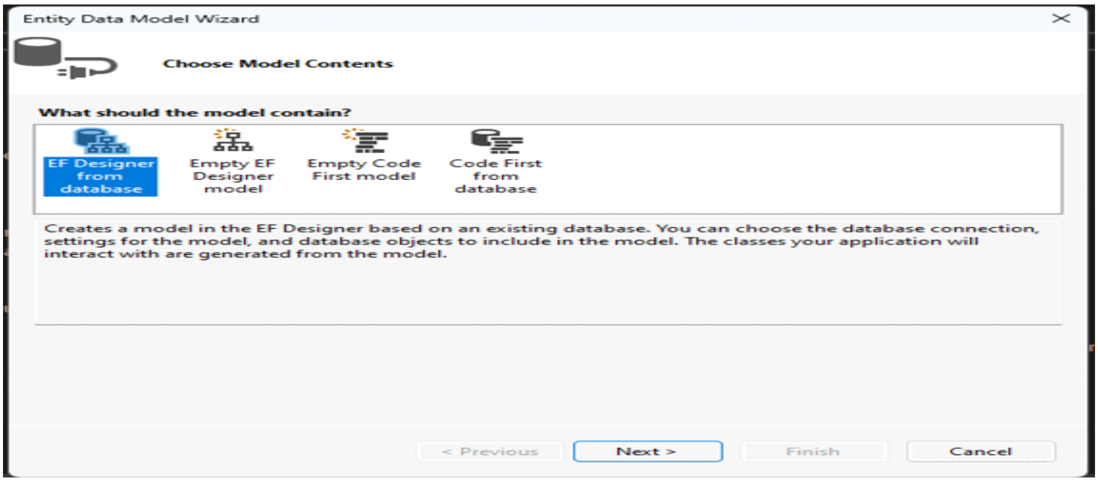
Connected. (1/1)


LAB2025\SQLEXPRESS.Db_Assiment - dbo.Table_1* - Microsoft SQL Server Management Studio
File Edit View Project Table Designer Tools Window Help
New Query
Execute
Object Explorer
Connect
LAB2025\SQLEXPRESS (SQL Server 10.0
Databases
System Databases
master
model
msdb
tempdb
Db_Assiment
Tables
System Tables
Views
Synonyms
Programmability
Service Broker
Storage
Security
Security
Server Objects
Replication
Management
LAB2025\SQLEXPRES....ent - dbo.Table_1*
SQLQuery1.sql - L...LAB2025\IMRD (52))
Solution Explorer
Solution 'Solution1' (0 projects)
Miscellaneous Files
SQLQuery1.sql
Column Properties
Ready

| Column Name | Data Type | Allow Nulls |
|---|---|---|
| ID | int | □ |
| Name | nvarchar(50) | ☑ |
| Addres | nvarchar(50) | ☑ |
|  |  | □ |

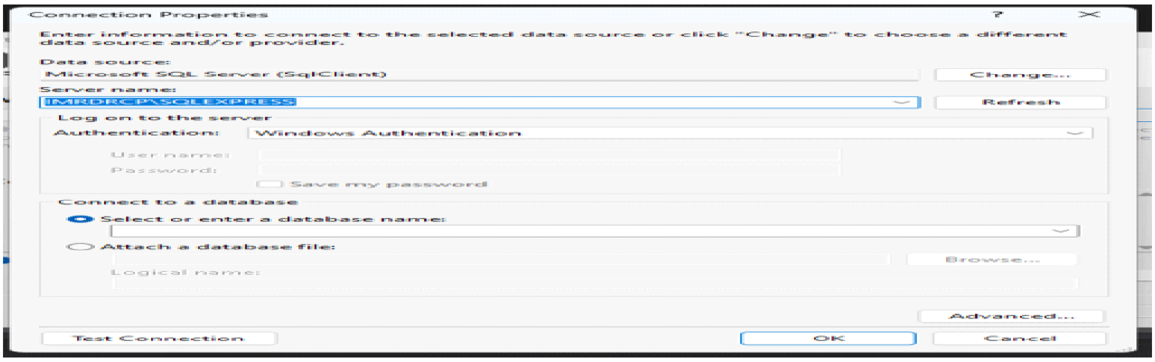
| | |
|---|---|
| Full-text Specification | No |
| Has Non-SQL Server Subscriber | No |
| Identity Specification | Yes |
| (Is Identity) | Yes |
| Identity Increment | 1 |
| Identity Seed | 1 |

(Is Identity)



What should the model contain?

Creates a model in the EF Designer based on an existing database. You can choose the database connection, settings for the model, and database objects to include in the model. The classes your application will interact with are generated from the model.



Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

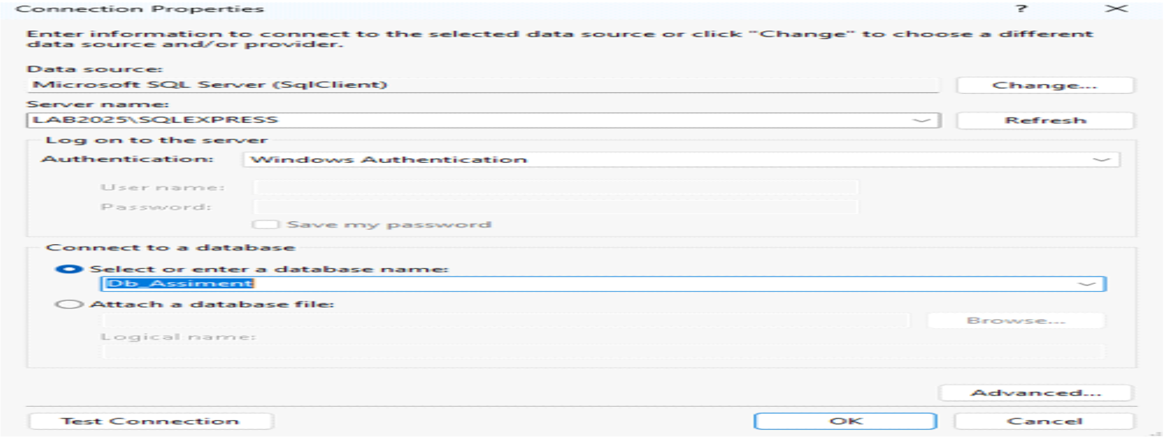
Data source:
Microsoft SQL Server (SqlClient)

Server name:
IMRDRCP\SQLEXPRESS

Log on to the server
Authentication: Windows Authentication

User name:
Password:
□ Save my password

Connect to a database
● Select or enter a database name:
○ Attach a database file:
Logical name:



Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

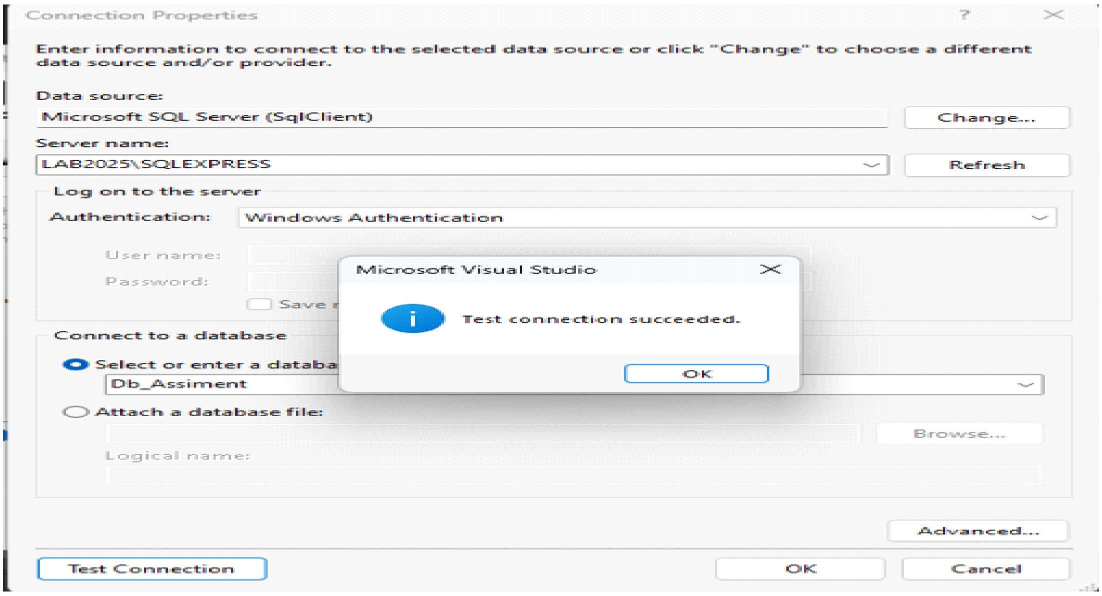
Data source:
Microsoft SQL Server (SqlClient)

Server name:
LAB2025\SQLEXPRESS

Log on to the server
Authentication: Windows Authentication

User name:
Password:
□ Save my password

Connect to a database
● Select or enter a database name:
Db_Assiment
○ Attach a database file:
Logical name:



Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient)

Server name:
LAB2025\SQLEXPRESS

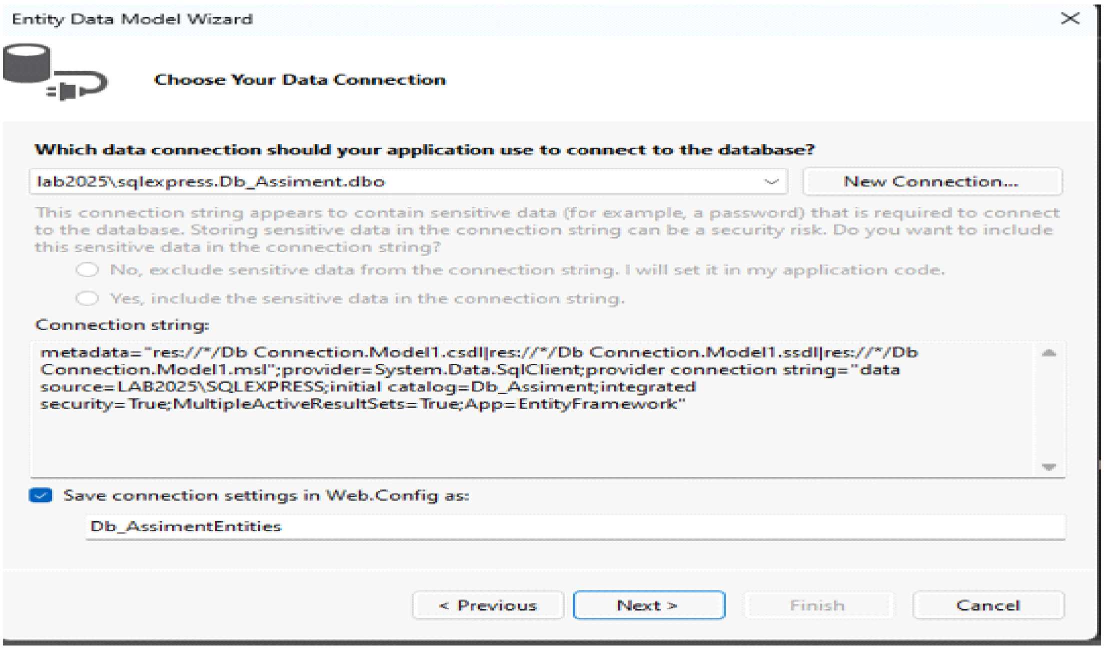
Log on to the server
Authentication: Windows Authentication

User name:
Password:
□ Save r

Connect to a database
● Select or enter a databa
Db_Assiment
○ Attach a database file:
Logical name:

Microsoft Visual Studio

Test connection succeeded.



Entity Data Model Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

lab2025\sqlexpress.Db_Asiment.dbo

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

○ No, exclude sensitive data from the connection string. I will set it in my application code.

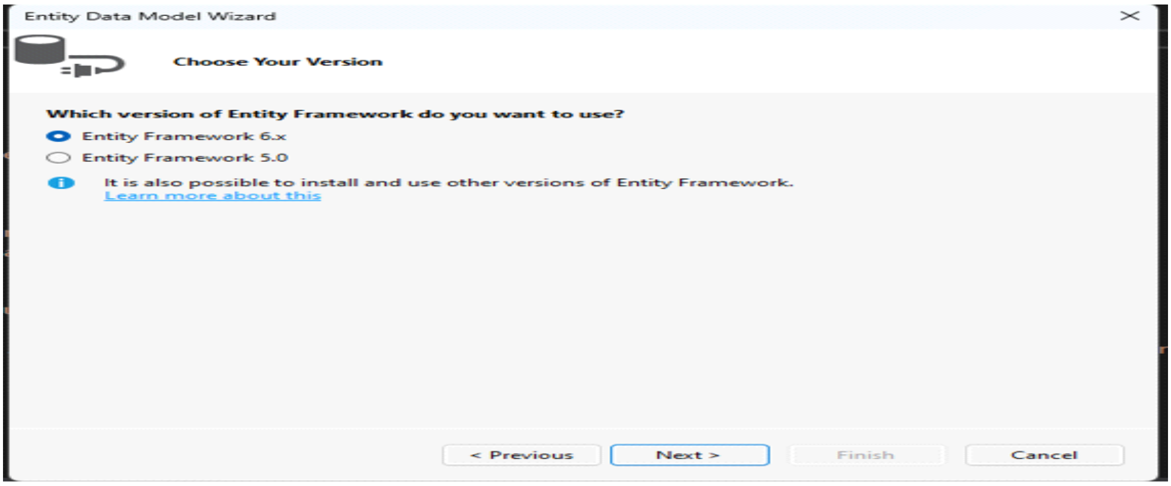
○ Yes, include the sensitive data in the connection string.

Connection string:

```
metadata="res://*/Db Connection.Model1.csdl|res://*/Db Connection.Model1.ssdl|res://*/Db
Connection.Model1.msl";provider=System.Data.SqlClient;provider connection string="data
source=LAB2025\SQLEXPRESS;initial catalog=Db_Asiment;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework"
```

☑ Save connection settings in Web.Config as:

Db_AsimentEntities



Entity Data Model Wizard

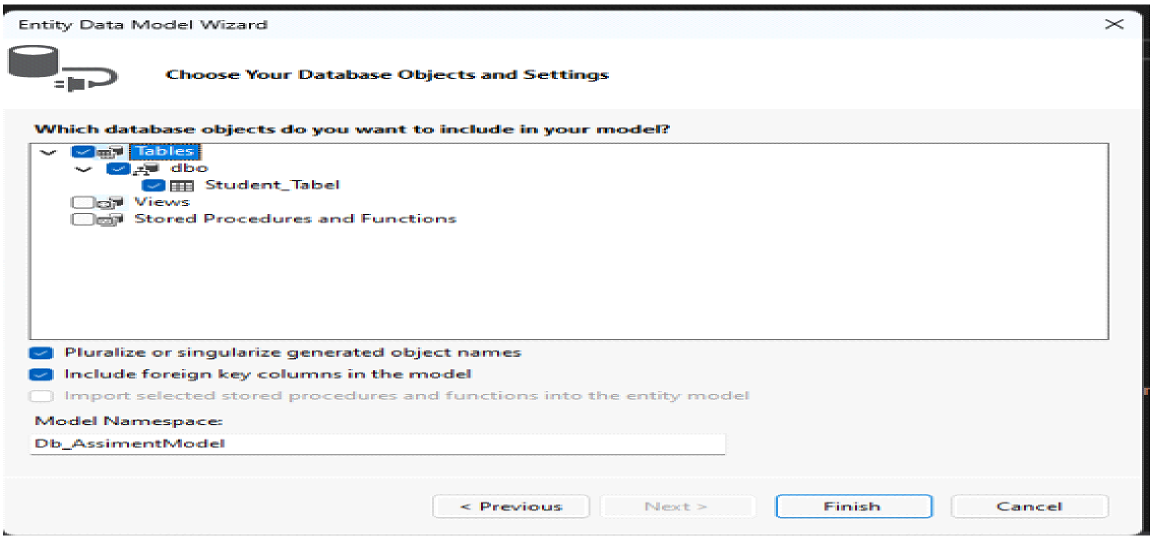
Choose Your Version

Which version of Entity Framework do you want to use?

● Entity Framework 6.x

○ Entity Framework 5.0

It is also possible to install and use other versions of Entity Framework.
Learn more about this



Entity Data Model Wizard

Choose Your Database Objects and Settings

Which database objects do you want to include in your model?

- ☑ Tables
  - ☑ dbo
    - ☑ Student_Tabel
- □ Views
- □ Stored Procedures and Functions

☑ Pluralize or singularize generated object names

☑ Include foreign key columns in the model

□ Import selected stored procedures and functions into the entity model

Model Namespace:

Db_AsimentModel

**15. Develop a Program in ASP.net MVC using Build In Helper**

StudentController:

```
//BuiltIn-HtmlHelper
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;


namespace WebApplication5.Controllers
{
    public class StudentController: Controller
    {
        public ActionResult SIndex()
        {
            return View();
        }

    }
}
```

Student - Index:

```
//BuiltIn-HtmlHelper
@{
    ViewBag.Title = "SIndex";
}

<h2>SIndex</h2>
<h1> I am from student controller and index action method</h1>
<div>
    @Html.ActionLink("Click me", "Index", "Home")
</div>
```

HomeController.cs

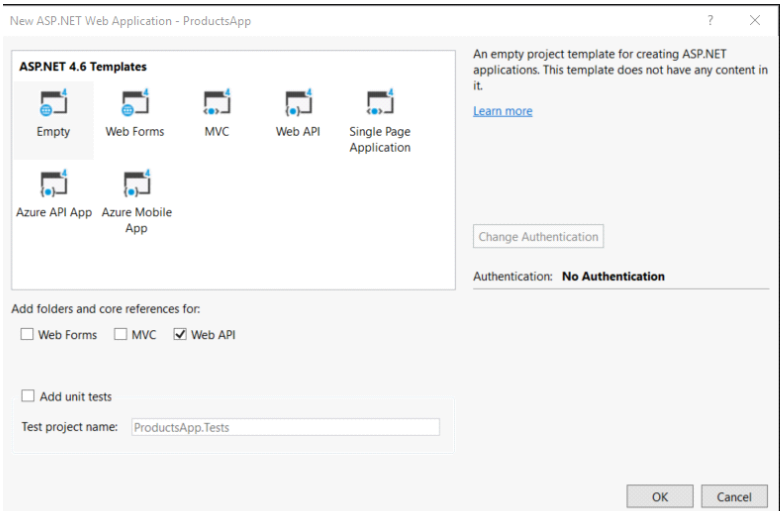
```
using System.Web;
using System.Web.Mvc;
namespace BuildInHtmlHelper.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            ViewBag.Message = "Modify this template to jump-start your ASP.NET MVC application.";
            return View();
        }
        public ActionResult About()
        {
            ViewBag.Message = "Your app description page.";
            return View();
        }

        public ActionResult Contact()
        {
            ViewBag.Message = "Your contact page.";

            return View();
        }
    }
}
```

Home –Index:

```
//BuiltIn-HtmlHelper
@{
    ViewBag.Title = "Index";
}
<body>
<h2>I am From home controller and index method</h2>
    <div>
        @Html.ActionLink("Click me","SIndex","Student")
        </div>
</body>
```

## 16. Design and Develop ASP.NET web-api and using in Web Application.

Open Microsoft Visual studio Click on file NewProject Dialog Box will appear, Select Asp.net Web application (.net Framework)  click ok next dialog box will appear select Empty and click on web API



New ASP.NET Web Application - ProductsApp

An empty project template for creating ASP.NET applications. This template does not have any content in it.

Authentication:  **No Authentication**

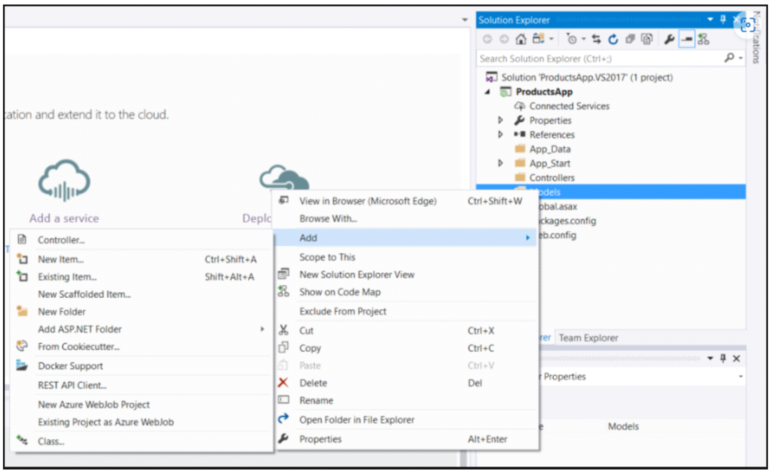
Add folders and core references for:

☐ Web Forms ☐ MVC ☑ Web API

☐ Add unit tests

Test project name: ProductsApp.Tests

## Adding a Model

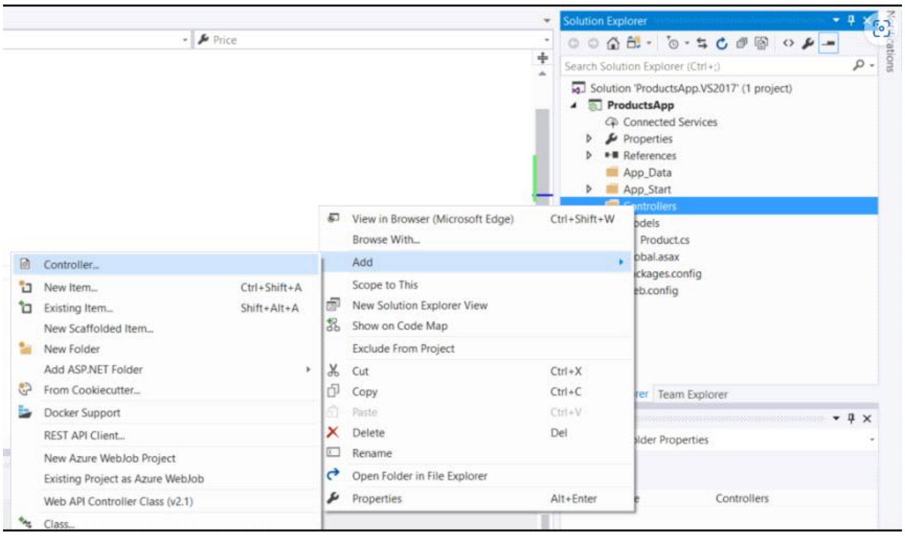


- Add the following code in model class: Product.cs

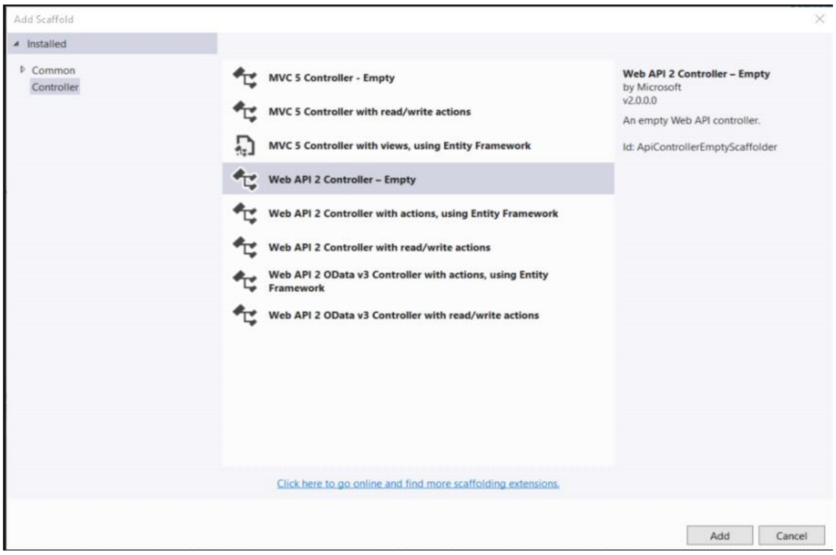
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WebApplication21.Models
{
    public class Product
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Category { get; set; }
        public decimal Price { get; set; }
    }
}
```

**Adding a Controller**

**In Solution Explorer, right-click the Controllers folder. Select Add and then select Controller.**



**In the Add Scaffold dialog, select Web API Controller - Empty. Click Add.**

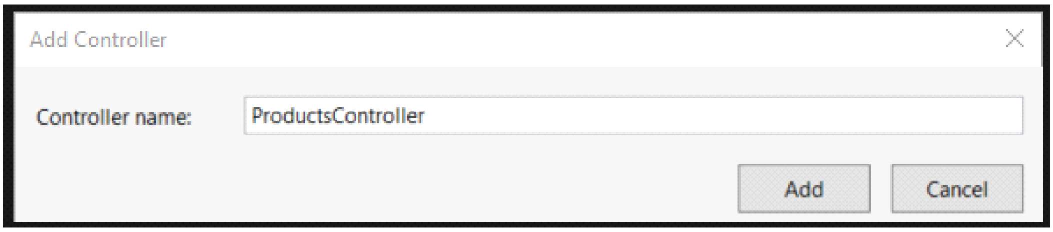

Add Scaffold
Installed
Common
Controller
MVC 5 Controller - Empty
MVC 5 Controller with read/write actions
MVC 5 Controller with views, using Entity Framework
Web API 2 Controller – Empty
Web API 2 Controller with actions, using Entity Framework
Web API 2 Controller with read/write actions
Web API 2 OData v3 Controller with actions, using Entity Framework
Web API 2 OData v3 Controller with read/write actions
Click here to go online and find more scaffolding extensions.
Add
Cancel

Web API 2 Controller – Empty
by Microsoft
v2.0.0.0

An empty Web API controller.

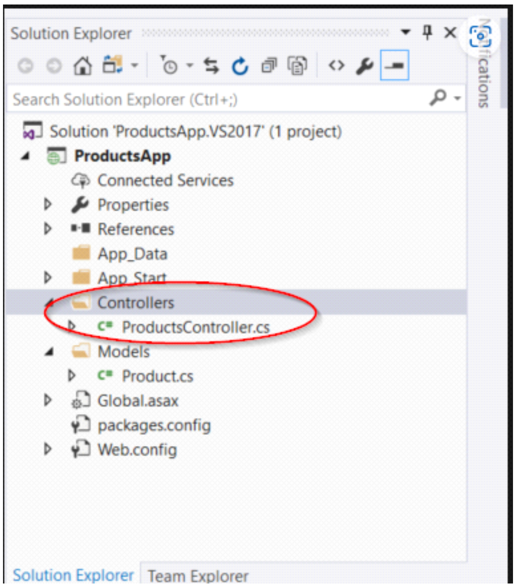
Id: ApiControllerEmptyScaffolder

**In the Add Controller dialog, name the controller "ProductsController". Click Add.**



Controller name: ProductsController

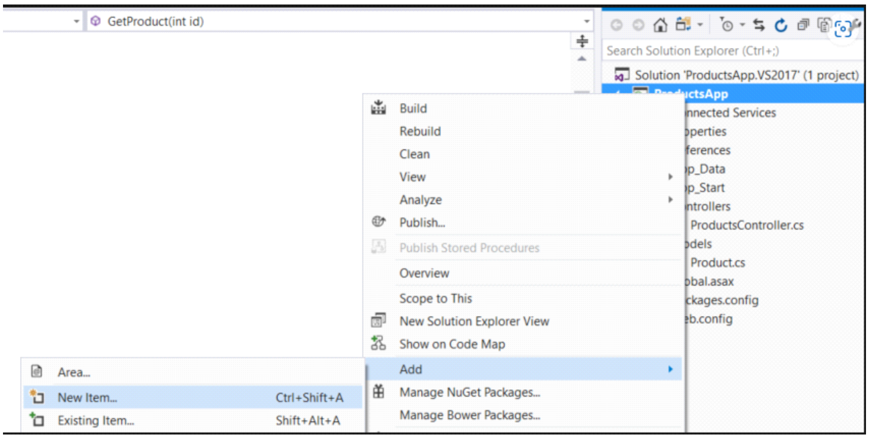
**The scaffolding creates a file named ProductsController.cs in the Controllers folder.**



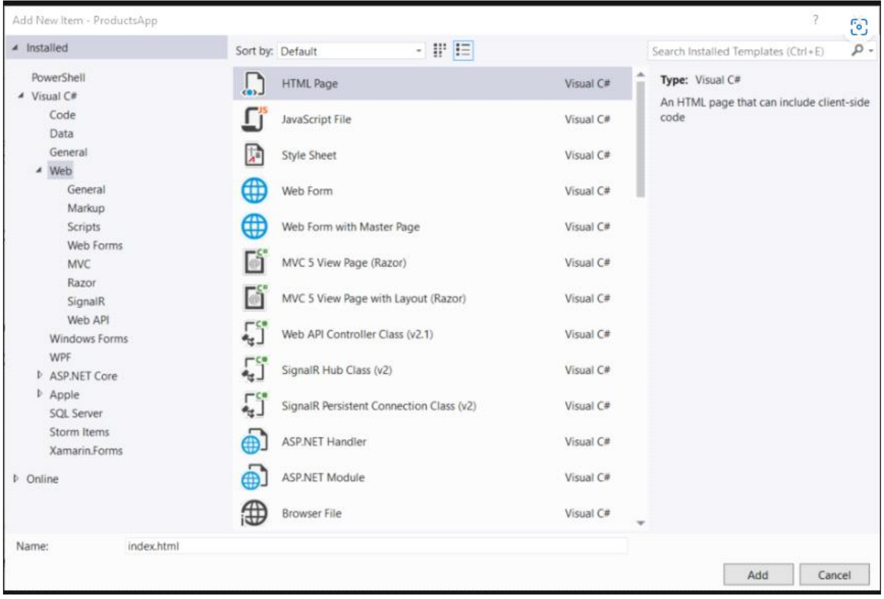
**If this file is not open already, double-click the file to open it. Replace the code in this file with the following.**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using WebApplication21.Models;

namespace WebApplication21.Controllers
{
    public class ProductsController : ApiController
    {
        Product[] products = new Product[]
        {
            new Product { Id = 1, Name = "Tomato Soup", Category = "Groceries", Price
= 1 },
            new Product { Id = 2, Name = "Yo-yo", Category = "Toys", Price = 3.75M },
            new Product { Id = 3, Name = "Hammer", Category = "Hardware", Price =
16.99M }
        };

        public IEnumerable<Product> GetAllProducts()
        {
            return products;
        }

        public IHttpActionResult GetProduct(int id)
        {
            var product = products.FirstOrDefault((p) => p.Id == id);
            if (product == null)
            {
                return NotFound();
            }
            return Ok(product);
        }
    }
}
```

## Calling the Web API with Javascript and jQuery



In the Add New Item dialog, select the Web node under Visual C#, and then select the HTML Page item. Name the page "HomePage.html".


Add New Item - ProductsApp
Installed
PowerShell
Visual C#
Code
Data
General
Web
General
Markup
Scripts
Web Forms
MVC
Razor
SignalR
Web API
Windows Forms
WPF
ASP.NET Core
Apple
SQL Server
Storm Items
Xamarin.Forms
Online
Sort by: Default
Search Installed Templates (Ctrl+E)
HTML Page Visual C#
JavaScript File Visual C#
Style Sheet Visual C#
Web Form Visual C#
Web Form with Master Page Visual C#
MVC 5 View Page (Razor) Visual C#
MVC 5 View Page with Layout (Razor) Visual C#
Web API Controller Class (v2.1) Visual C#
SignalR Hub Class (v2) Visual C#
SignalR Persistent Connection Class (v2) Visual C#
ASP.NET Handler Visual C#
ASP.NET Module Visual C#
Browser File Visual C#
Add
Cancel

Type: Visual C#

An HTML page that can include client-side code

Name: index.html

HomePage.html file

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Product App</title>
</head>
<body>
```

```
  <div>
    <h2>All Products</h2>
    <ul id="products" />
  </div>
  <div>
    <h2>Search by ID</h2>
    <input type="text" id="prodId" size="5" />
    <input type="button" value="Search" onclick="find();" />
    <p id="product" />
  </div>

  <script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-
2.0.3.min.js"></script>
  <script>
  var uri = 'api/products';

  $(document).ready(function () {
   // Send an AJAX request
   $.getJSON(uri)
     .done(function (data) {
      // On success, 'data' contains a list of products.
      $.each(data, function (key, item) {
       // Add a list item for the product.
       $('<li>', { text: formatItem(item) }).appendTo($('#products'));
      });
     });
  });

  function formatItem(item) {
   return item.Name + ': $' + item.Price;
  }

  function find() {
   var id = $('#prodId').val();
   $.getJSON(uri + '/' + id)
     .done(function (data) {
      $('#product').text(formatItem(data));
     })
     .fail(function (jqXHR, textStatus, err) {
      $('#product').text('Error: ' + err);
     });
  }
  </script>
</body>
</html>
```

**17. Demonstrate using of View bag in ASP.NET MVC.**

**Homecontroller-> view(Index.cshtml)->add Models.**

HomeController.cs

```
using System.Web;
using System.Web.Mvc;
using viewback.Models;
namespace viewback.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {

            ViewBag.MyList=new List<string>(){"john","kim","rock"};

           List<Employee>emplist =new List<Employee>()
              {
           new Employee(){Address="Shirpur",id=1,Name="John" },
           new Employee(){Address="Pune",id=2,Name="kim" },
           new Employee(){Address="Mumbai",id=3,Name="rock" },
              };
          ViewBag.MyEmpList=emplist;
           return View();
        }

      }

 }
```

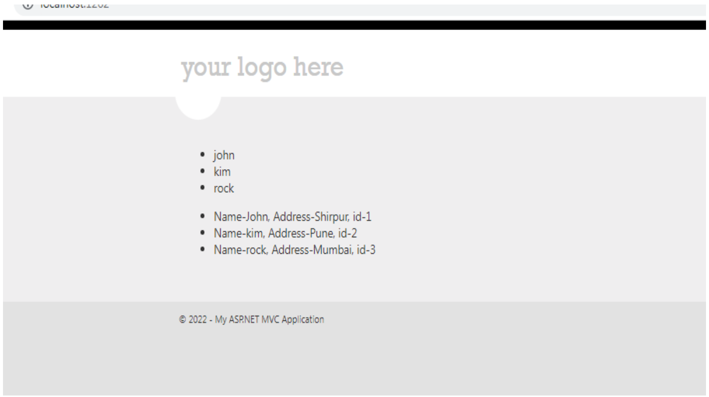
Index.cshtml

```
@{
    ViewBag.Title = "Home Page";
}
<h1>@ViewBag.MyCustomProperty</h1>

  <body>
  <div>
  <ul>
  @foreach (var item in ViewBag.MyList)
  {
      <li>@item</li>
  }
  </ul>
```

```
 </div>
 <div>
 <ul>
 @foreach (var item in ViewBag.MyEmplist)
 {
    <li>
    Name-@item.Name,
    Address-@item.Address,
    id-@item.id
    </li>
 }
 </ul>

 </div>
 </body>
```

**\Models\Employee.cs**

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

namespace viewback.Models
{
    public class Employee
    {
        public int id { get; set; }
        public string Name { get; set; }
        public string Address { get; set; }
    }
}
```



```
• john
• kim
• rock

• Name-John, Address-Shirpur, id-1
• Name-kim, Address-Pune, id-2
• Name-rock, Address-Mumbai, id-3

© 2022 - My ASP.NET MVC Application
```