TechLift

KICKSTART YOUR CAREER

WEB ENGINEERING

.Net

P@SHA

PAKISTAN PSEB SOFTWARE EXPORT BOARD

Week 3 - Day 15

Recap Project (Time: 1 hr)

The students will work in pairs and build a simple

Student Management System

They may take help from this video to complete the project.

Marked Project (Total marks: 20) (Time: 2 hrs)

The students will work individually and complete the following project:

Restaurant Web App

Marks will be calculated based on the ratio of correct coding and steps.

ASP.NET Web Application

Learning Objectives

By the end of this session, the students will have developed an understanding of:

- Separating Code between Controller and Views
- Controller Actions and Views
- Models in MVC

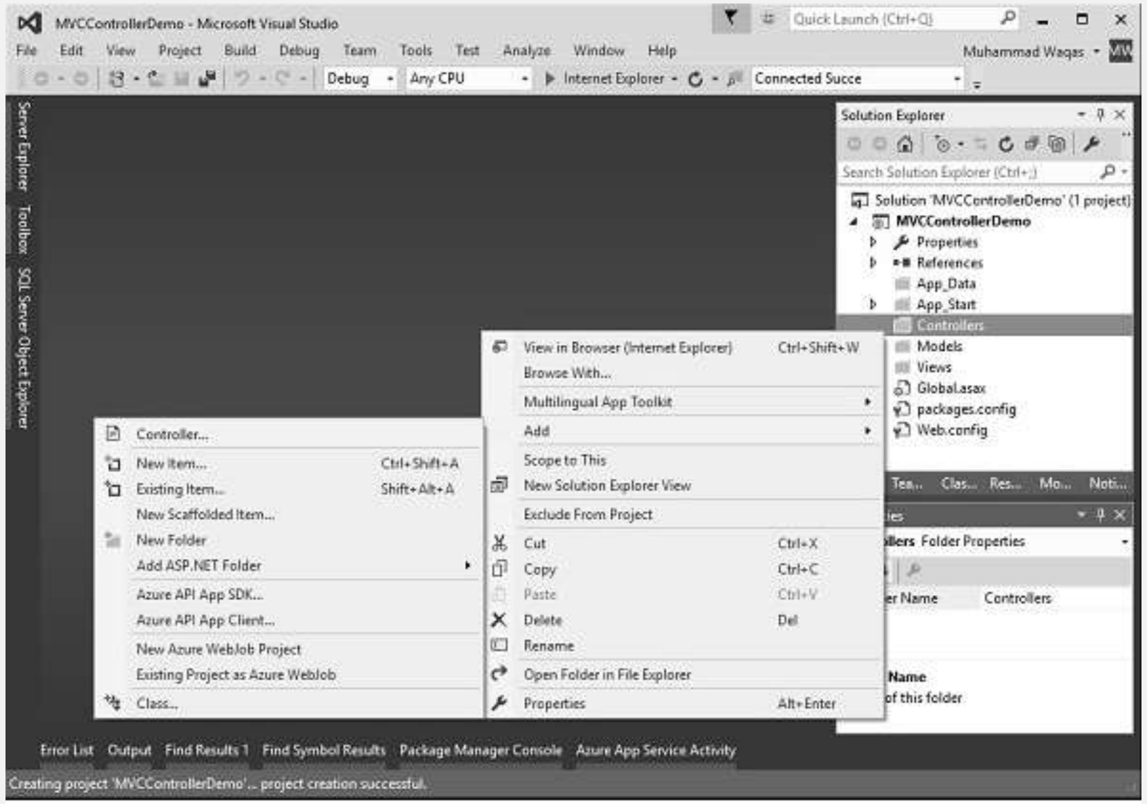
ASP.NET Web Applications

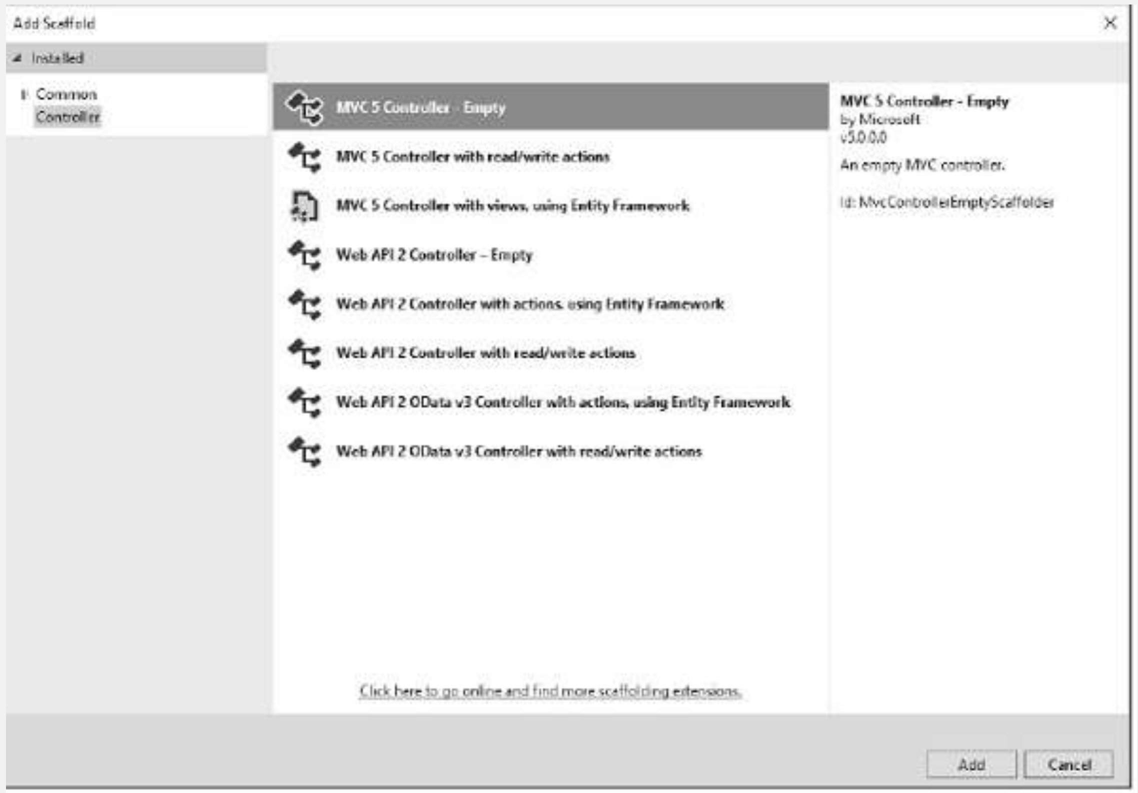
ASP.NET MVC: CONTROLLERS

Controllers are essentially the central unit of your ASP.NET MVC application. It is the 1st recipient, which interacts with incoming HTTP Request. So, the controller decides which model will be selected, and then it takes the data from the model and passes the same to the respective view, after that view is rendered. Actually, controllers are controlling the overall flow of the application taking the input and rendering the proper output.

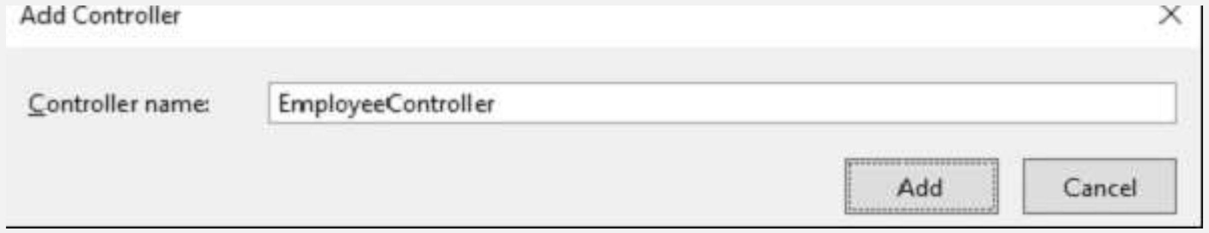
Controllers are C# classes inheriting from System.Web.Mvc.Controller, which is the builtin controller base class. Each public method in a controller is known as an action method, meaning you can invoke it from the Web via some URL to perform an action.

CONTROLLERS (..continued)

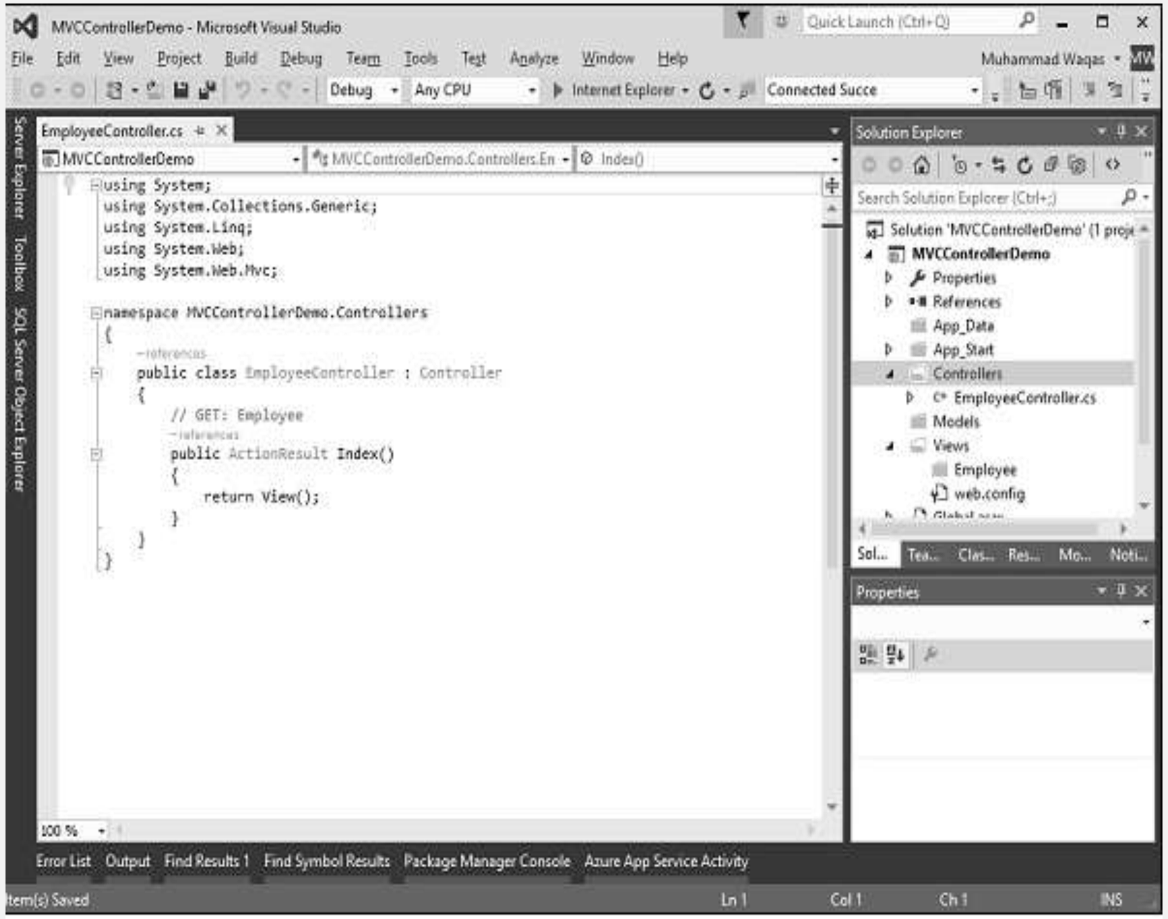

MVCControllerDemo - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
Any CPU
Internet Explorer
Solution Explorer
Solution 'MVCControllerDemo' (1 project)
MVCControllerDemo
Properties
References
App_Data
App_Start
Controllers
Models
Views
Global.asax
packages.config
Web.config
View in Browser (Internet Explorer) Ctrl+Shift+W
Browse With...
Multilingual App Toolkit
Add
Scope to This
New Solution Explorer View
Exclude From Project
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Rename
Open Folder in File Explorer
Properties Alt+Enter
Controller...
New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Scaffolded Item...
New Folder
Add ASP.NET Folder
Azure API App SDK...
Azure API App Client...
New Azure WebJob Project
Existing Project as Azure WebJob
Class...
Error List Output Find Results 1 Find Symbol Results Package Manager Console Azure App Service Activity
Creating project 'MVCControllerDemo'... project creation successful.


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
MVC 5 Controller - Empty
by Microsoft
v5.0.0.0
An empty MVC controller.
Id: MvcControllerEmptyScaffolder
Click here to go online and find more scaffolding extensions.
Add
Cancel

CONTROLLERS (..continued)



Controller name: EmployeeController


MVCControllerDemo - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
Any CPU
Internet Explorer
Solution Explorer
Solution 'MVCControllerDemo' (1 project)
MVCControllerDemo
Properties
References
App_Data
App_Start
Controllers
EmployeeController.cs
Models
Views
Employee
web.config
Properties
Error List Output Find Results 1 Find Symbol Results Package Manager Console Azure App Service Activity
Item(s) Saved
Ln 1 Col 1 Ch 1 INS

EmployeeController.cs

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

namespace MVCControllerDemo.Controllers
{
    public class EmployeeController : Controller
    {
        // GET: Employee
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

CONTROLLERS (..continued)

```
routes.MapRoute(
   "Employee", "Employee/{name}", new{
      controller = "Employee", action = "Search", name =
      UrlParameter.Optional });
```

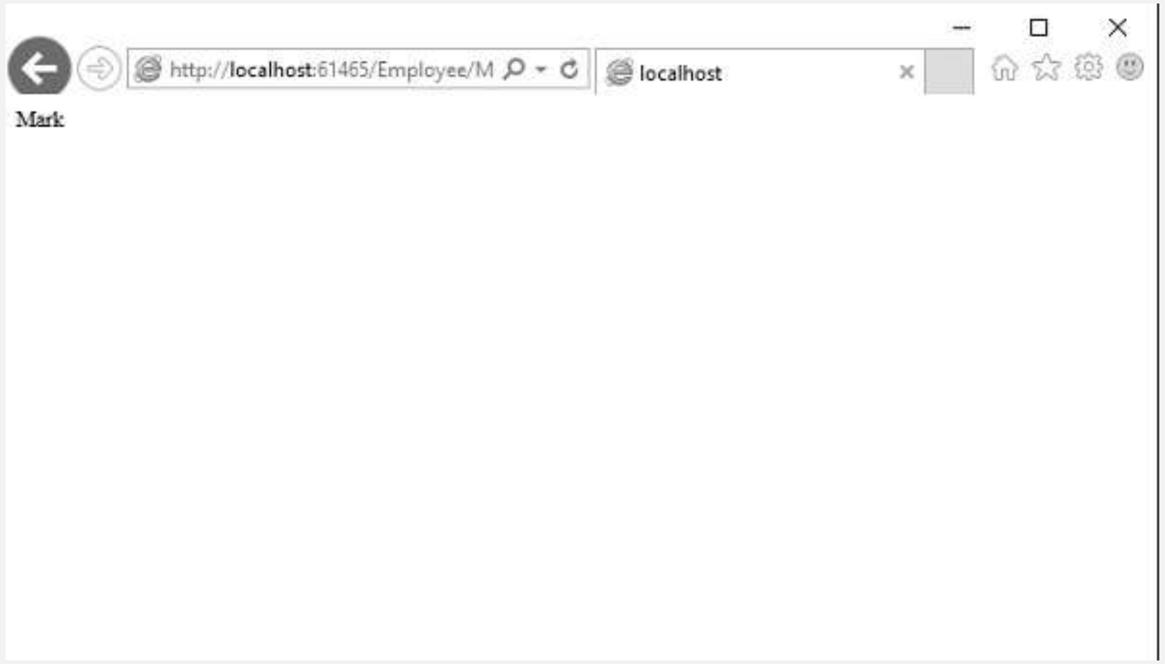
```
using System;
using System.Collections.Generic;
using System.Linq;

using System.Web;
using System.Web.Mvc;
using System.Web.Routing;

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

         routes.MapRoute(
            "Employee", "Employee/{name}", new{
               controller = "Employee", action = "Search", name = UrlParameter.Optional })

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

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

using System.Web;
using System.Web.Mvc;

namespace MVCControllerDemo.Controllers  {
   public class EmployeeController : Controller {
      // GET: Employee
      public ActionResult Search(string name){
         var input = Server.HtmlEncode(name);
         return Content(input);
      }
   }
}
```

CONTROLLERS (..continued)

The trainer will ask the students to add a custom route for Employee controller with the default Route and ask them to show the following output:

Once completed, click here to check the output.

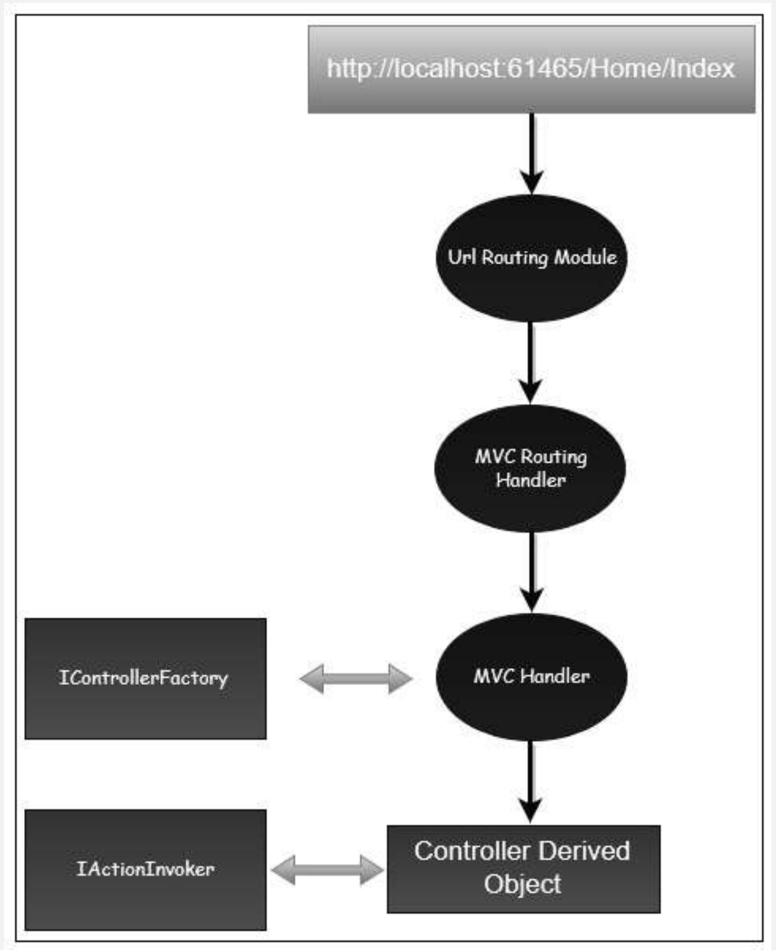


Mark

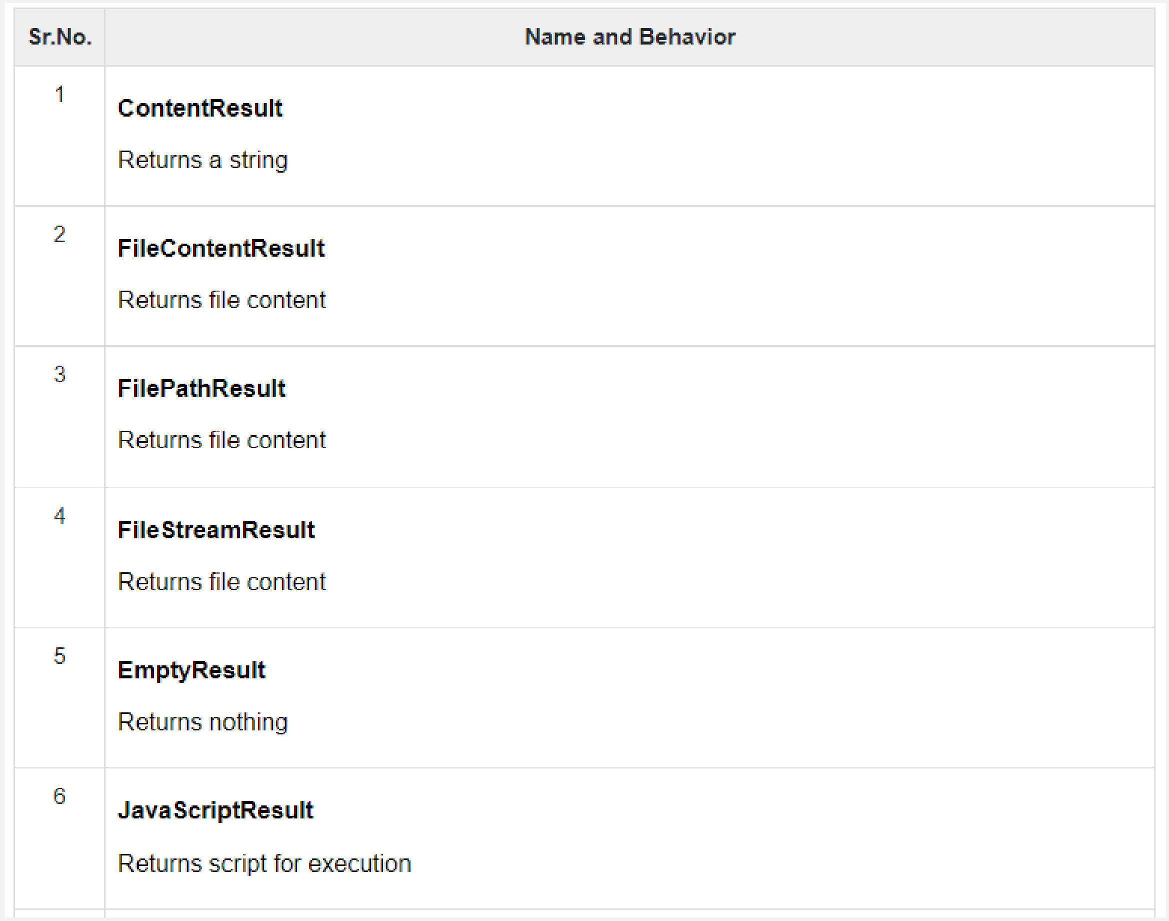
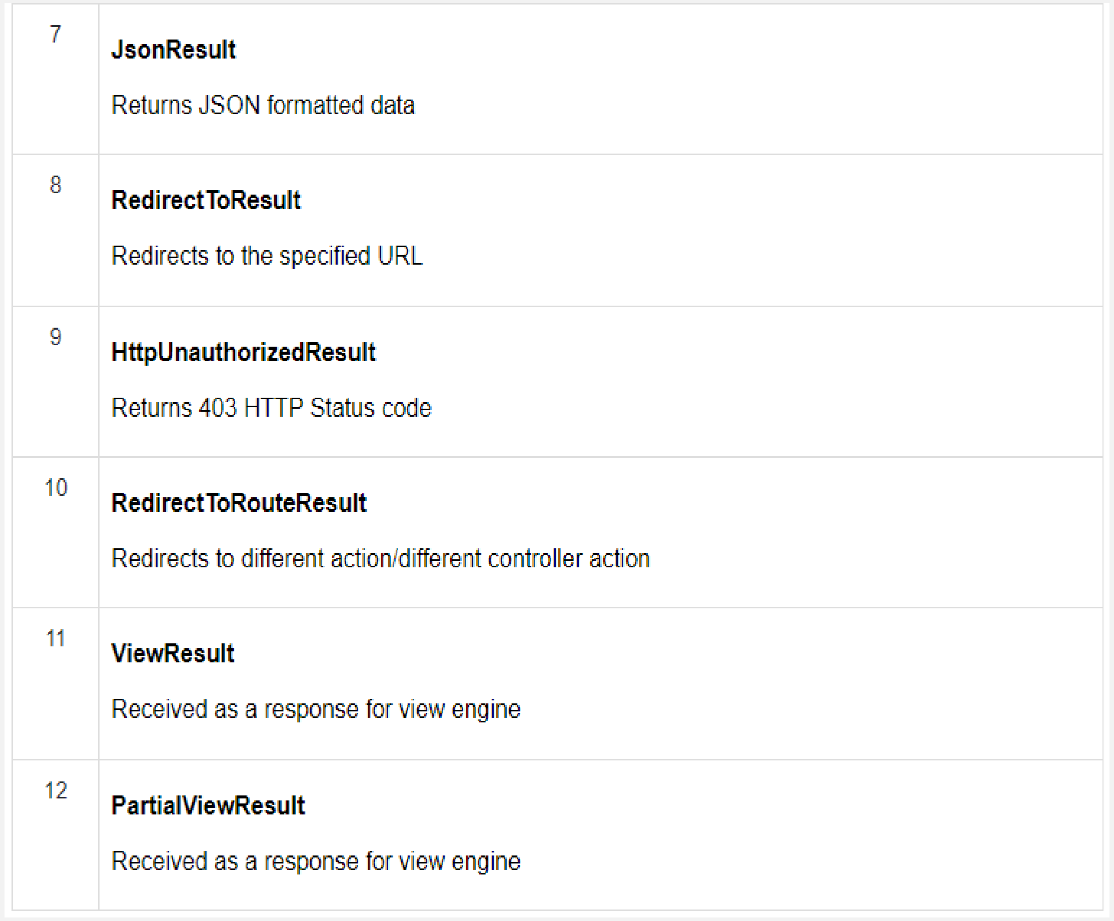
ASP.NET MVC: ACTIONS

ASP.NET MVC Action Methods are responsible to execute requests and generate responses to it. By default, it generates a response in the form of ActionResult. Actions typically have a one-to-one mapping with user interactions.

For example, enter a URL into the browser, click on any particular link, and submit a form, etc. Each of these user interactions causes a request to be sent to the server. In each case, the URL of the request includes information that the MVC framework uses to invoke an action method. The one restriction on action method is that they have to be instance method, so they cannot be static methods. Also there is no return value restrictions. So you can return the string, integer, etc.

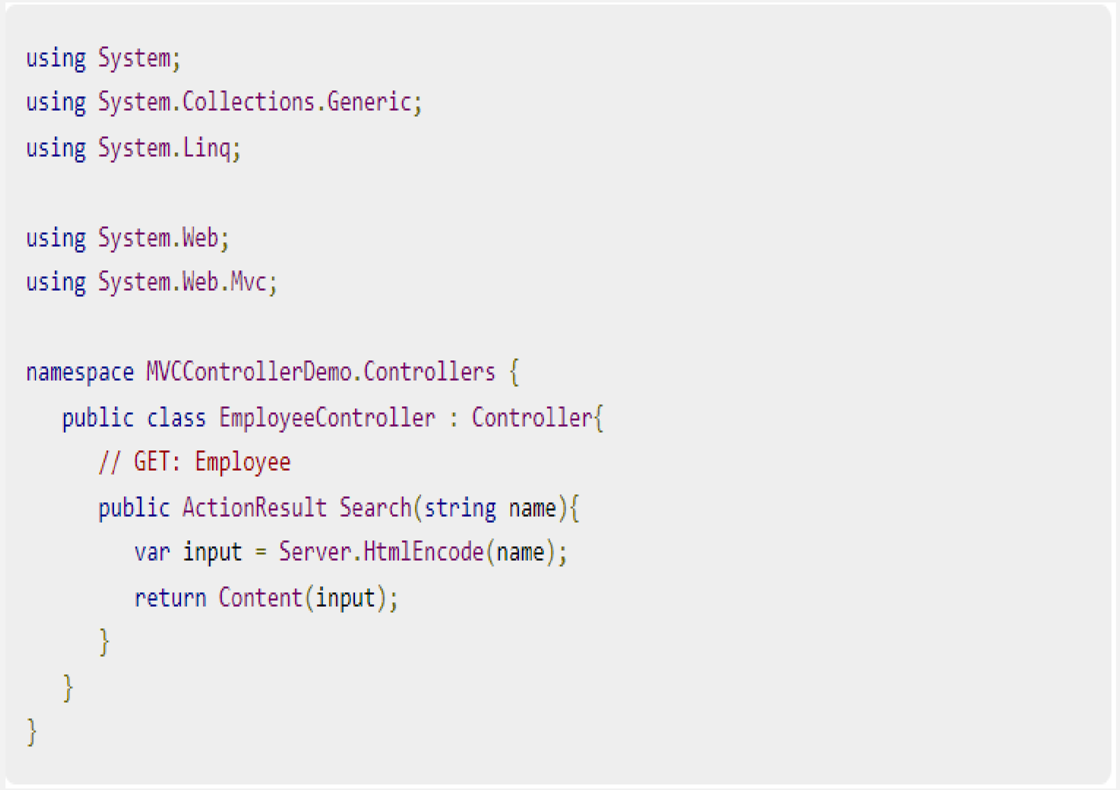


TYPES OF ACTIONS

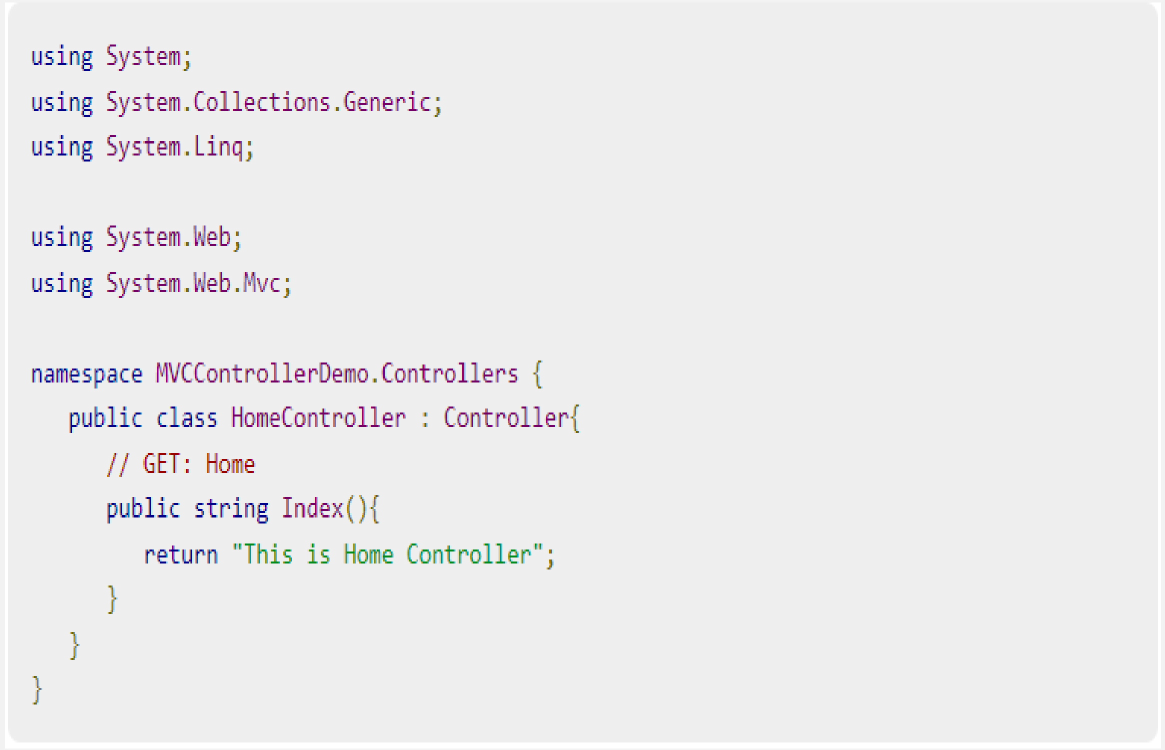
Sr.No.	Name and Behavior
1	**ContentResult** Returns a string
2	**FileContentResult** Returns file content
3	**FilePathResult** Returns file content
4	**FileStreamResult** Returns file content
5	**EmptyResult** Returns nothing
6	**JavaScriptResult** Returns script for execution
7	**JsonResult** Returns JSON formatted data
8	**RedirectToResult** Redirects to the specified URL
9	**HttpUnauthorizedResult** Returns 403 HTTP Status code
10	**RedirectToRouteResult** Redirects to different action/different controller action
11	**ViewResult** Received as a response for view engine
12	**PartialViewResult** Received as a response for view engine

EXAMPLE: ACTIONS



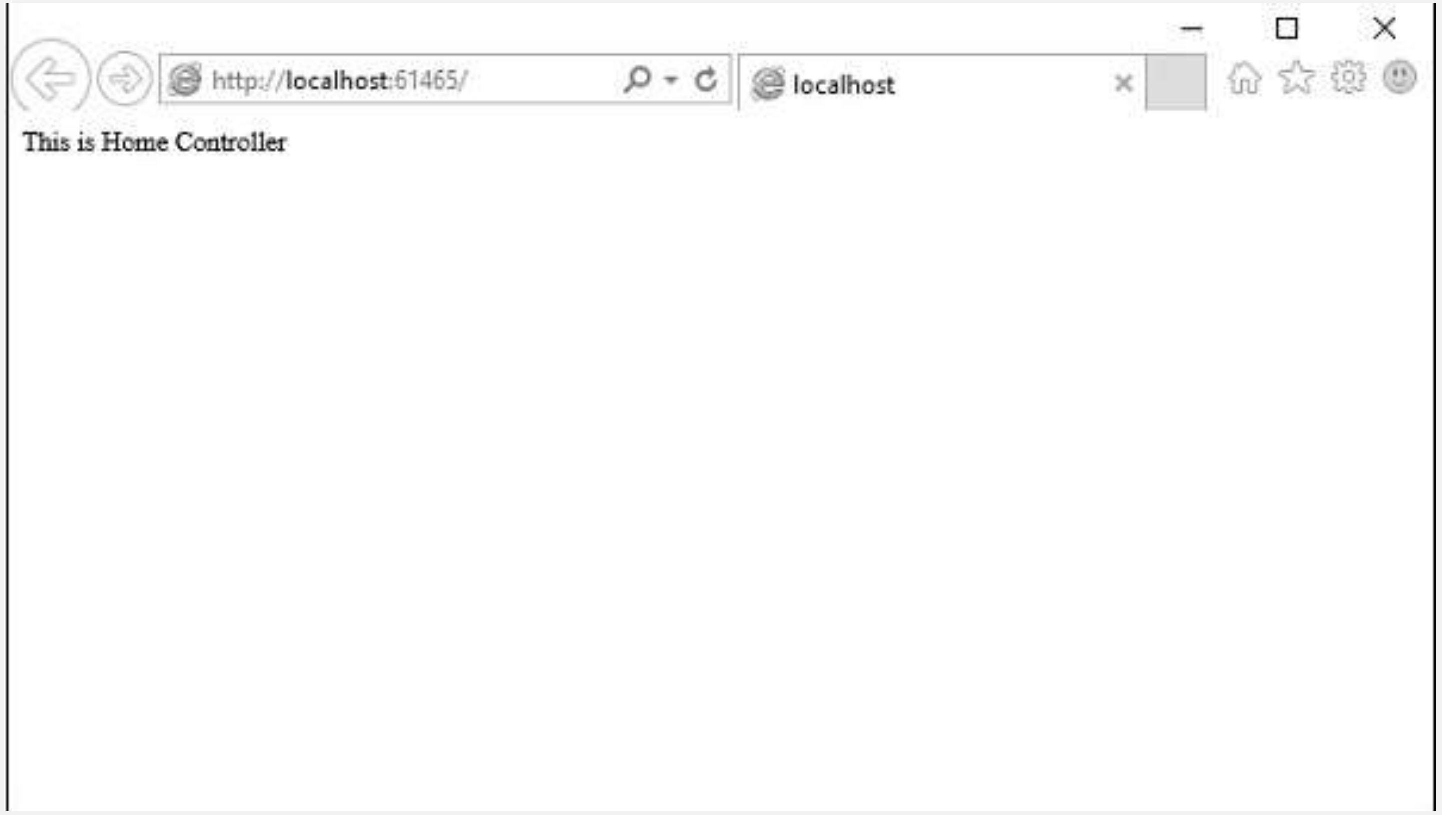
```
using System;
using System.Collections.Generic;
using System.Linq;

using System.Web;
using System.Web.Mvc;

namespace MVCControllerDemo.Controllers {
   public class EmployeeController : Controller{
      // GET: Employee
      public ActionResult Search(string name){
         var input = Server.HtmlEncode(name);
         return Content(input);
      }
   }
}
```

Adding Another Controller



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

using System.Web;
using System.Web.Mvc;

namespace MVCControllerDemo.Controllers {
   public class HomeController : Controller{
      // GET: Home
      public string Index(){
         return "This is Home Controller";
      }
   }
}
```

ACTIONS (OUTPUT)

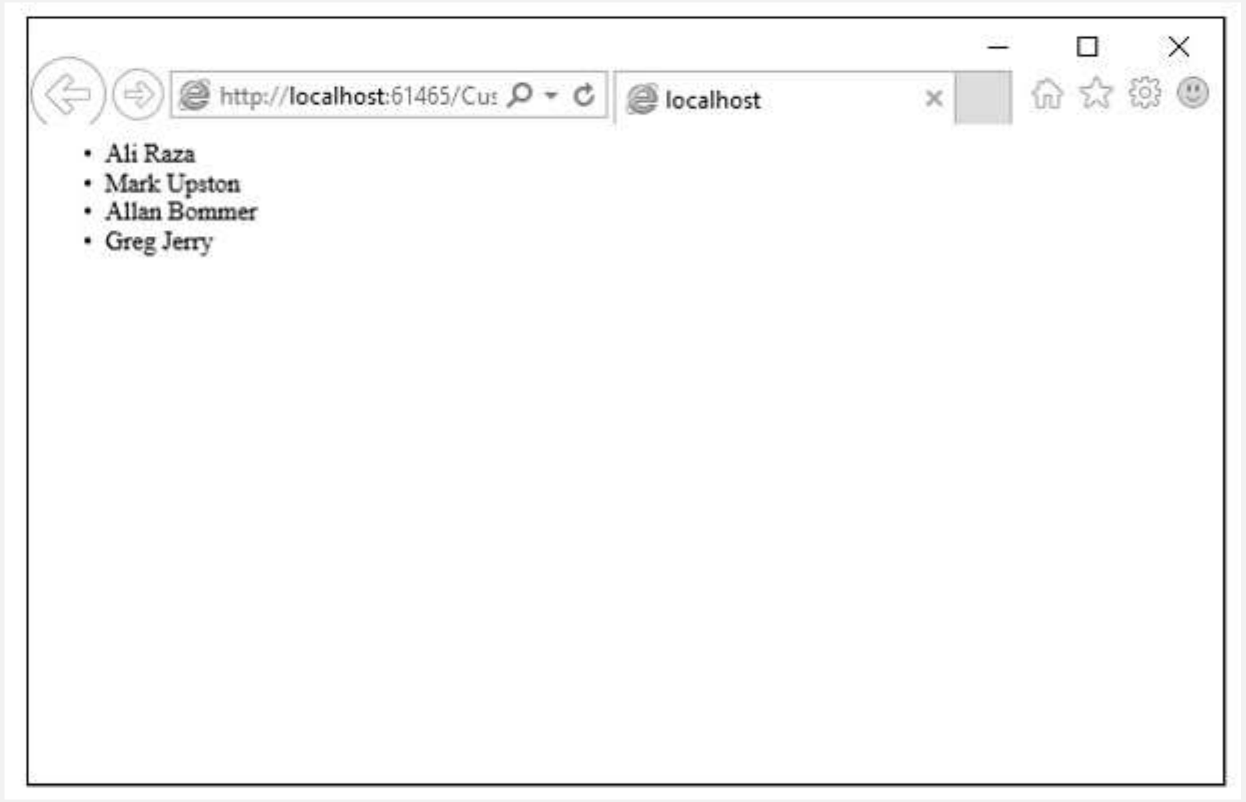


```
This is Home Controller
```

IMPLEMENTATION

The trainer will show the students the following output and ask the students to implement the code on their systems

Once completed, click here to check the output.

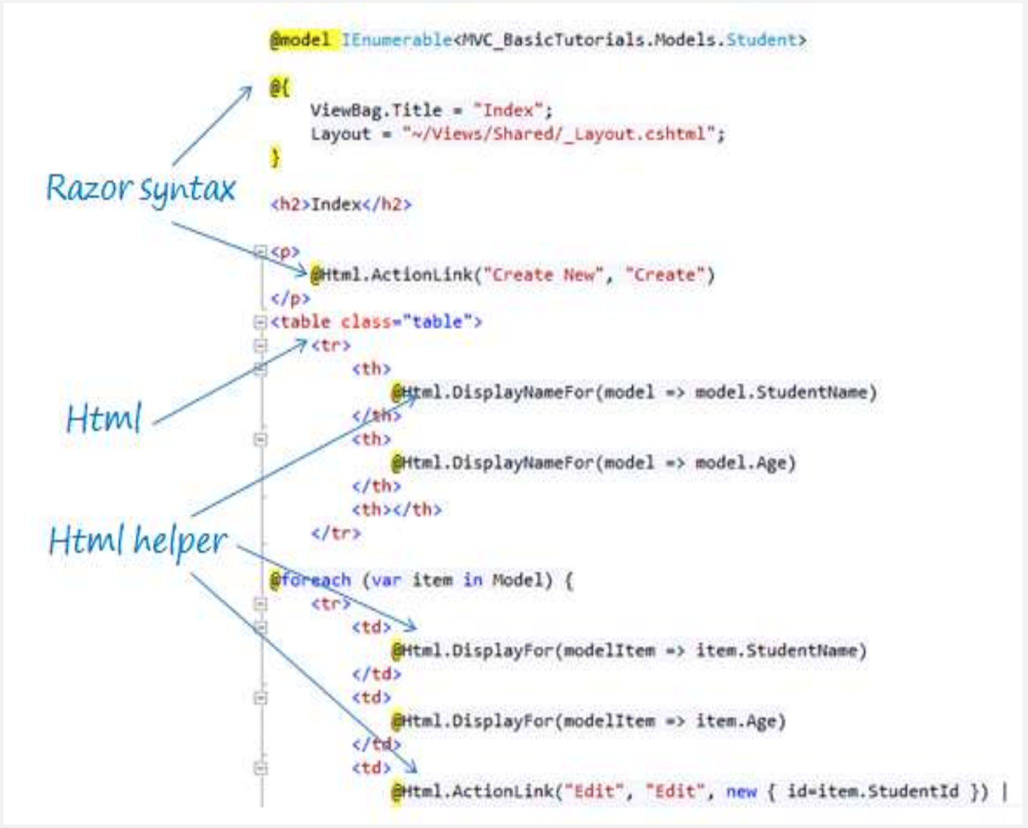


```
• Ali Raza
• Mark Upston
• Allan Bommer
• Greg Jerry
```

ASP.NET MVC: VIEWS

In an ASP.NET MVC application, there is nothing like a page and it also doesn't include anything that directly corresponds to a page when you specify a path in URL. The closest thing to a page in an ASP.NET MVC application is known as a View.

In ASP.NET MVC application, all incoming browser requests are handled by the controller and these requests are mapped to controller actions. A controller action might return a view or it might also perform some other type of action such as redirecting to another controller action.

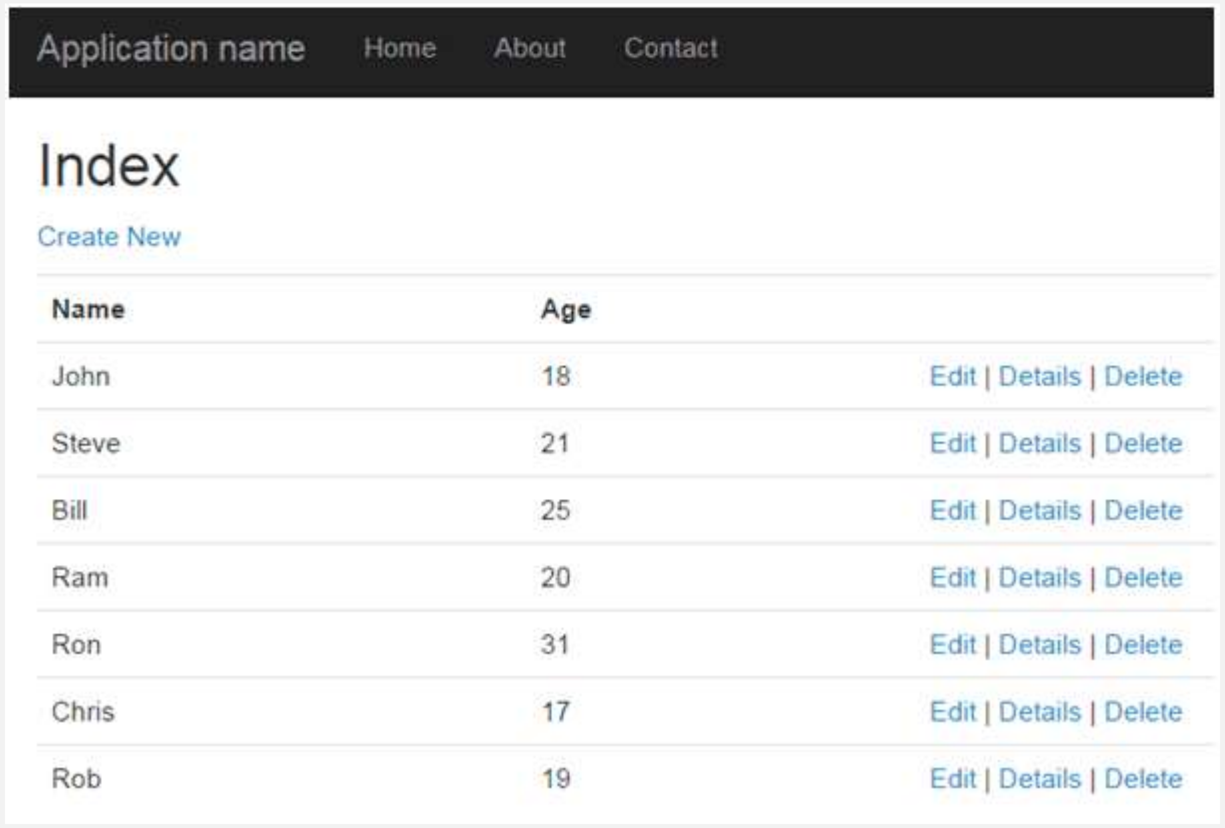


```
@model IEnumerable<MVC_BasicTutorials.Models.Student>

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.StudentName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Age)
        </th>
        <th></th>
    </tr>

@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.StudentName)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Age)
        </td>
        <td>
            @Html.ActionLink("Edit", "Edit", new { id=item.StudentId }) |
```

VIEWS IMPLEMENTATION

The trainer will show the students the following output and ask the students to implement the code on their systems

Once completed, click here to check the output.



Index

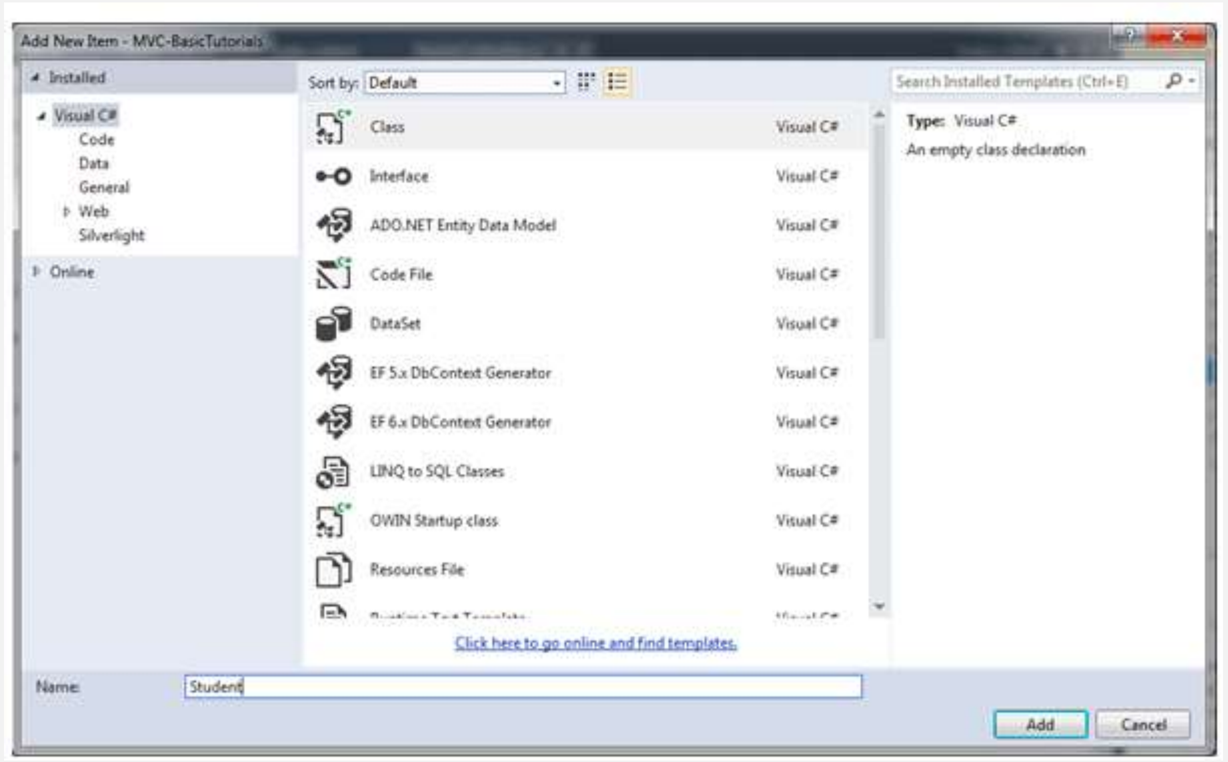
Create New

Name	Age	
John	18	Edit | Details | Delete
Steve	21	Edit | Details | Delete
Bill	25	Edit | Details | Delete
Ram	20	Edit | Details | Delete
Ron	31	Edit | Details | Delete
Chris	17	Edit | Details | Delete
Rob	19	Edit | Details | Delete

ASP.NET MVC: MODELS

The model classes represents domain-specific data and business logic in the MVC application. It represents the shape of the data as public properties and business logic as methods.

In the ASP.NET MVC Application, all the Model classes must be created in the Model folder.


Add New Item - MVC-BasicTutorials

Sort by: Default

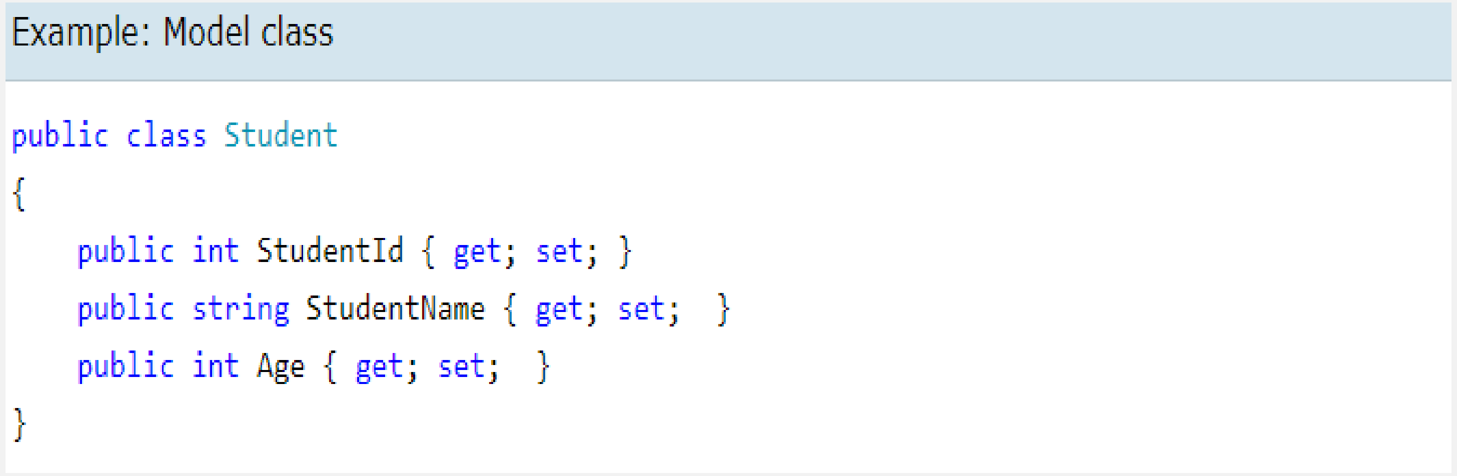
Type: Visual C#
An empty class declaration

Name: Student

Installed
Visual C#
Code
Data
General
Web
Silverlight
Online
Search Installed Templates (Ctrl+E)
Class Visual C#
Interface Visual C#
ADO.NET Entity Data Model Visual C#
Code File Visual C#
DataSet Visual C#
EF 5.x DbContext Generator Visual C#
EF 6.x DbContext Generator Visual C#
LINQ to SQL Classes Visual C#
OWIN Startup class Visual C#
Resources File Visual C#
Click here to go online and find templates.
Add
Cancel

ASP.NET MVC: MODELS (..continued)

Add a new Student class in model folder. We want this model class to store id, name, and age of the students. So, we will have to add public properties for Id, Name, and Age, as shown below.



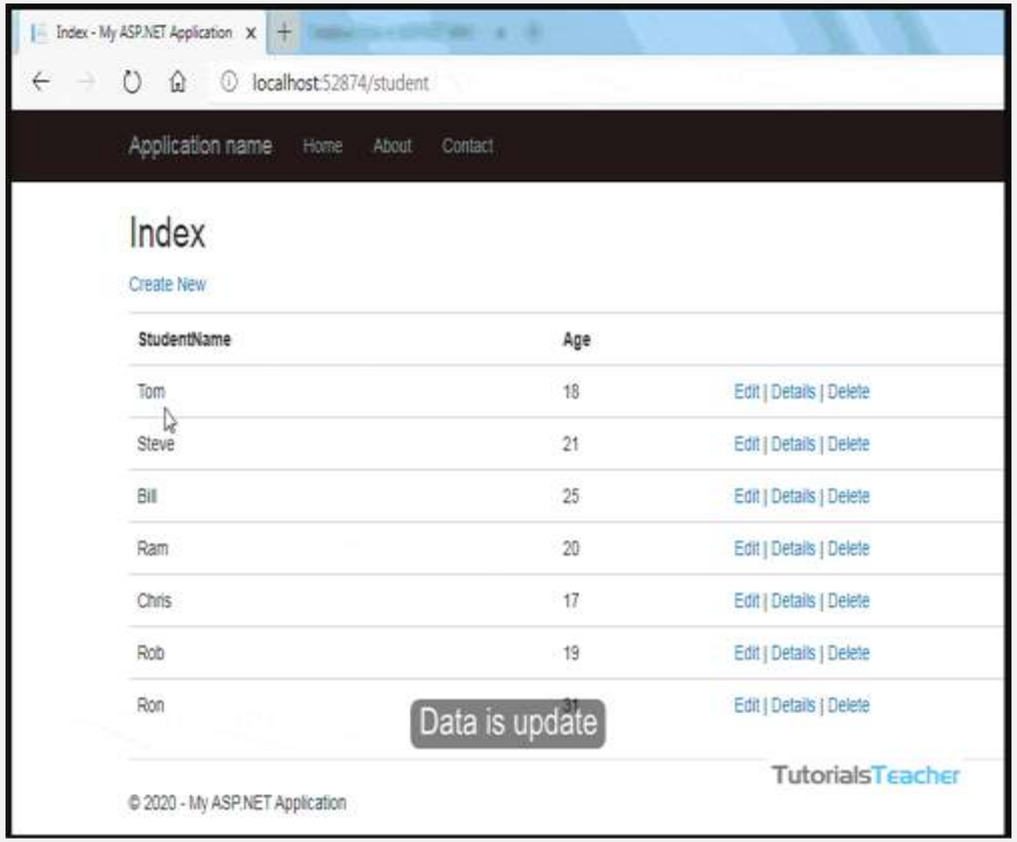
Example: Model class

```
public class Student
{
    public int StudentId { get; set; }
    public string StudentName { get; set;  }
    public int Age { get; set;  }
}
```

Homework

Create the list view in the Integrate Model, View, Controller. You will create the edit view where the users can edit the data.

Expected Output :



Index - My ASP.NET Application

localhost:52874/student

Application name Home About Contact

Index

Create New

StudentName	Age	
Tom	18	Edit | Details | Delete
Steve	21	Edit | Details | Delete
Bill	25	Edit | Details | Delete
Ram	20	Edit | Details | Delete
Chris	17	Edit | Details | Delete
Rob	19	Edit | Details | Delete
Ron	31	Edit | Details | Delete

Data is update

© 2020 - My ASP.NET Application

Learning Objectives

By the end of this session, the students have practised

- ✓ Separating Code between Controller and Views
- ✓ Controller Actions and Views
- ✓ Models in MVC

Conclusion & Q/A

See you tomorrow!