Module 1: Exploring ASP.NET MVC 4

Lesson 3: Introduction to ASP.NET MVC 4

Demonstration: How to Explore an MVC Application

Preparation Steps

1. Log on to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Start **Visual Studio 2012.**
3. Navigate to **Allfiles(D):01*, and then open the **PhotoSharingSample.sln** file.
4. Enable the **Allow NuGet to download missing packages during build** option, by performing the following steps:

   - On the **TOOLS** menu of the Microsoft Visual Studio window, click **Options**.
   - In the navigation pane of the **Options** dialog box, click **Package Manager**.
   - Under the Package Restore section, select the **Allow NuGet to download missing packages during build** checkbox, and then click **OK**.
5. Run the **PhotoSharingSample.sln** application.
6. In the Address bar of the Windows Internet Explorer window, note the port number that appears after “http://localhost:” You will use the port number during this demonstration.
7. Close the Windows Internet Explorer window.

   Note : In Hyper-V Manager, start the MSL-TMG1 virtual machine if it is not already running.

Demonstration Steps

1. In the Solution Explorer pane of the **PhotoSharingSample – Microsoft Visual Studio** window, expand **PhotoSharingSample**, and then note that the PhotoSharingSample application does not have the default.htm, the default.aspx, or the default.cshtml files to act as a home page.
2. In the Solution Explorer pane, under PhotoSharingSample, expand **Controllers**, and then click **HomeController.cs**.
3. In the HomeController.cs code window, locate the following code.

`cs        Public ActionResult Index()        {            return View();        }` >**Note:** This code block represents an action that will return a view called Index.

4. In the Solution Explorer pane, expand **Views**, and then expand **Photo**.
5. In the Solution Explorer pane, under Photo, click **Index.cshtml**.
6. In the Index.cshtml code window, locate the following code.

`cs        <h2>@ViewBag.Title</h2>        <p> @Html.ActionLink("Create New", "Create")        </p>` >**Note:** This code block represents the View that renders the home page.

7. On the toolbar of the **PhotoSharingSample – Microsoft Visual Studio** window, click **Internet Explorer**.
8. In the **http://localhost:***<yourportnumber>***/** window, note that the default home page is displayed.
9. On the taskbar, click the **Microsoft Visual Studio** icon.
10. In the **PhotoSharingSample – Microsoft Visual Studio** window, in the Solution Explorer pane, expand **App_Start**, and then click **RouteConfig.cs**.
11. In the RouteConfig.cs code window, locate the following code.

cs routes.MapRoute(name: "Default", url: "{controller}/{action}/{id}",) >**Note:** This code block represents the default route that forwards requests to the specified controller.

12. On the taskbar, click the **Internet Explorer** icon.
13. In the Address bar of the Windows Internet Explorer window, type the URL **http://localhost:***<yourportnumber>***/home/index**, and then click the **Go to** button.

    Note: The browser window displays the Home page of the http://localhost:*<yourportnumber>*/home/index web application.

14. On the taskbar, click the **Microsoft Visual Studio** icon.
15. In the **PhotoSharingSample – Microsoft Visual Studio** window, in the Solution Explorer pane, expand **Models**, and then click **Photo.cs**.
16. In the Photo.cs code window, locate the following code.

cs [Required] public string Title { get; set;} >**Note:** This code block represents the **Title** property for a photo stored in the application.

17. In the Solution Explorer pane, under Controllers, click **PhotoController.cs**.
18. In the PhotoController.cs code window, locate the following code.

cs public class PhotoController : Controller >**Note:** This code block represents that the **PhotoController** class inherits the System.Web.MVC.Controller base class.

19. In thePhotoController.cs code window, locate the following code.

cs public ActionResult Details(int id = 0) { Photo photo = db.Photos.Find(id); if (photo == null) { return HttpNotFound(); } return View("Details", photo); } >**Note:** This code block represents the **Details** action of the Photo Controller.

20. In the Solution Explorer pane, expand **Views**, expand **Photo**, and then click **Details.cshtml**.
21. In the Details.cshtml code window, locate the following code.

cs <h2>"@Model.Title"</h2> >**Note:** The Razor view engine runs this code and renders the Photo Title property that you viewed in the model.

22. On the taskbar, click the **Internet Explorer** icon.

23. In the Address bar of the Windows Internet Explorer window, type **http://localhost:***<yourportnumber>***/photo/details/2**, and then click the **Go to** button.

    Note: The photo with ID 2 is displayed in the browser window. Note that the title of the photo is rendered at the top.

24. In the Windows Internet Explorer window, click the **Close** button.
25. In the **PhotoSharingSample (Running) – Microsoft Visual Studio** window, click the **Close** button.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Module 3: Developing ASP.NET MVC 4 Models

Lesson 1: Creating MVC Models

Demonstration: How to Add a Model

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Open **Visual Studio 2012**.

Note: In Hyper-V Manager, start the **MSL-TMG1** virtual machine, if it is not already running.

Demonstration Steps

1. On the **File** menu of the **Start Page - Microsoft Visual Studio** window, point to **New,** and then click **Project.**
2. In the navigation pane of the **New Project** dialog box, expand **Installed**, expand **Templates**, and then expand **Visual C#**.
3. Under **Visual C#**, click **Web**, and then, in the result pane, click **ASP.NET MVC 4 Web Application**.
4. In the **Name** text box of the **New Project** dialog box, type **OperasWebSites**.
5. In the **New Project** dialog box, click **Browse**.
6. In the **Location** text box, go to **Allfiles(D):03*, and then click **Select Folder**.

7. In the **New Project** dialog box, click **OK**.
8. In the **Select a Template** list of the **New ASP.NET MVC 4 Project** dialog box, click **Empty**, and then click **OK**.
9. In the **Solution Explorer** pane of the **OperasWebSites - Microsoft Visual Studio** window, right-click **Models**, point to **Add**, and then click **Class**.
10. In the **Name** text box of the **Add New Item - OperasWebSites** dialog box, type **Opera.cs**, and then click **Add**.
11. In the **Opera** class of the **Opera.cs** code window, type the following code.

`cs         public int OperaID { get; set; }         public string Title { get; set; }         public int Year { get; set; }         public string Composer { get; set; }` 12. Place the mouse cursor at the end of the **OperaID** property code, press Enter, and then type the following code.

`cs          [Required]          [StringLength(200)]` 13. In the Required data annotation, right-click **Required**, point to **Resolve**, and then click **using System.ComponentModel.DataAnnotations**. 14. Place the mouse cursor at the end of the **Year** property, press Enter, and then type the following code.

`cs          [Required]` 15. Place the mouse cursor at the end of the Opera class, press Enter, and then type the following code.

`cs         public class CheckValidYear : ValidationAttribute {          }` 16. In the **CheckValidYear** class, type the following code.

`cs         public override bool IsValid(object value)         { int year = (int)value;             if (year < 1598)          { return false;              }             else            { return true;              }         }` 17. In the **CheckValidYear** class, type the following code.

`cs         public CheckValidYear()         { ErrorMessage = "The earliest opera is Daphne, 1598, by Corsi, Peri, and Rinuccini";          }` 18. In the **Opera** class, place the mouse cursor at the end of the **Title** property code, press Enter, and then type the following code.

`cs          [CheckValidYear]` 19. On the **Build** menu of the **OperasWebSites - Microsoft Visual Studio** window, click **Build Solution**, and then note that the application is being built. 20. In the **OperasWebSites - Microsoft Visual Studio** window, click the **Close** button.

Lesson 2: Working with Data

Demonstration: How to Use Entity Framework Code

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Start **File Explorer**.

3. Go to **AllFiles (D):03*.
4. Open the **OperasWebSite.sln** project.
5. Enable the **Allow NuGet to download missing packages during build** option, by performing the following steps:

- On the **TOOLS** menu of the Microsoft Visual Studio window, click **Options**.

- In the navigation pane of the **Options** dialog box, click **Package Manager**.

- Under the **Package Restore** section, select the **Allow NuGet to download missing packages during build** check box, and then click **OK**.

6. On the **Build** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Build Solution**, and then note that the application is built successfully.

Note: In Hyper-V Manager, start the MSL-TMG1 virtual machine, if it is not already running.

Demonstration Steps

1. On the **Tools** menu of the **OperasWebSite - Microsoft Visual Studio** window, point to **Library Package Manager**, and then click **Package Manager Console**.
2. In **Package Manager Console** window, type the following command, and then press Enter.

```
cs          install-package entityframework -version 5.0.0.0
```

3. In the **Solution Explorer** pane of the **OperasWebSite - Microsoft Visual Studio** window, click **web.config**.

4. In the **web.config** code window, place the mouse cursor at the end of the **</appsettings>** tag, press Enter, and then type the following code.

`cs <connectionStrings> <add name="OperasDB" connectionString="Data Source=(LocalDB)\v11.0;AttachDbFilename=|DataDirectory|\Operas.mdf;Integrated Security=True;" providerName="System.Data.SqlClient" /> </connectionStrings>` 5. In the **Solution Explorer** pane, right-click **Models**, point to **Add**, and then click **Class**. 6. In the **Name** text box of the **Add New Item - OperasWebSite** dialog box, type **OperasDB**, and then click **Add**. 7. In the **OperasDB.cs** code window, locate the following code.

`cs          using System.Web;` 8. Place the mouse cursor at the end of the located code, press Enter, and then type the following code.

`cs          using System.Data.Entity;` 9. In the **OperasDB.cs** code window, locate the following code.

`cs          public class OperasDB` 10. Append the following code to the existing line of code.

`cs          : DbContext` 11. In the **OperasDB** class, type the following code.

`cs              public DbSet<Opera> Operas{ get; set; }` 12. In the **Solution Explorer** pane, right-click **Models**, point to **Add**, and then click **Class**. 13. In the

Name text box of the **Add New Item - OperasWebSite** dialog box, type **OperasInitializer**, and then click **Add**. 14. In the **OperasInitializer.cs** code window, place the mouse cursor at the end of the **System.web** namespace code, press Enter, and then type the following code.

`cs          using System.Data.Entity;` 15. In the **OperasInitializer.cs** code window, locate the following code.

`cs          public class OperasInitializer` 16. Append the following code to the existing line of code.

`cs          : DropCreateDatabaseAlways<OperasDB>` 17. In the **OperasInitializer** class code block, type the following code, press Spacebar, and then click, **Seed(OperasDB context)**.

`cs          override` 18. In the **Seed** method, place the mouse cursor after the call to **base.Seed**, press Enter, and then type the following code.

```
cs          var operas = new List<Opera>         {              new
Opera {                  Title = "Cosi Fan Tutte",
Year = 1790,                  Composer = "Mozart"            }
};          operas.ForEach(s =>context.Operas.Add(s));
context.SaveChanges();
```

19. In the **Solution Explorer** pane, click **Global.asax**. 20. In the **Global.asax** code window, place the cursor at the end of the **System.Web.Routing** namespace, press Enter, and then type the following code.

```
cs        using System.Data.Entity;        using
OperasWebSite.Models;
```

21. In the **Application_Start** method code block, type the following code.

```
cs          Database.SetInitializer<OperasDB>(new
OperasInitializer());
```

22. On the **Build** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Build Solution**, and then note that the application is built successfully. 23. In the **Solution Explorer** pane, right-click **Controllers**, click **Add**, and then click **Controller**. 24. In the **Controller Name** box, type **OperaController**. 25. In the **Template** box, click **MVC controller with read/write actions and views, using Entity Framework**. 26. In the **Model Class** box, click **Opera (OperasWebSite.Models)**. 27. In the **Data context class** box, click **OperasDB (OperasWebSite.Models)**, and then click **Add**. 28. In the **Solution Explorer** pane, in the **Views/Operas** folder, double-click **Create.cshtml**. 29. In the **Create.cshtml** code window, locate and delete the following code.

`cs          @section Scripts {@Script.Render("~/bundles/jqueryval")}` 30. On the **DEBUG** menu of the **OperasWebSite - Microsoft Visual** Studio window, click **Start Debugging**

Note: An error message displays in the Internet Explorer window. The error message is expected because the home page view has not been added.

31. In the Address bar of the **Internet Explorer** window, append the existing URL with **opera/index**, and then click **Go to**.
32. On the **Index** page, click **Create New**.
33. In the **Title** text box of the result page, type **Carmen**, and then, in the **Year** text box, type **1475**.

34. In the **Composer** text box, type **Bizet**, and then click **Create**.

Note: An error message is displayed by the custom validator.

35. In the **Year** text box, type **1875**, and then click **Create**.
36. In the **Internet Explorer** window, click **Close**.
37. In the **OperasWebSite - Microsoft Visual Studio** window, click **Close**.

Module 4: Developing ASP.NET MVC 4 Controllers

Lesson 1:Writing Controllers and Actions

Demonstration: How to create a Controller

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Start **File Explorer**.
3. Go to **AllFiles (D):04*.
4. Open the **OperasWebSite.sln** project.
5. Enable the **Allow NuGet to download missing packages during build** option, by performing the following steps:

- On the **TOOLS** menu of the Microsoft Visual Studio window, click **Options**.

- In the navigation pane of the **Options** dialog box, click **Package Manager**.

- Under the **Package Restore** section, select the **Allow NuGet to download missing packages during build** check box, and then click **OK**.

6. On the **Build** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Build Solution**, and then note that the application is built successfully.

Note: In Hyper-V Manager, start the MSL-TMG1 virtual machine if it is not already running.

Demonstration Steps

1. In the Solution Explorer pane of the **OperasWebSite - Microsoft Visual Studio** window, right-click **Controllers**, point to **Add**, and then click **Controller**.
2. In the **Controller Name** text box of the **Add Controller** dialog box, type **OperaController**.
3. In the **Template** box, click **Empty MVC controller**, and then click **Add**.
4. In the OperaController.cs code window, locate the following code.

cs using System.Web.MVC; 5. Ensure that the cursor is at the end of the System.Web.MVC namespace, press Enter, and then type the following code.

cs using System.Data.Entity; using OperasWebSite.Models; 6. In the **OperaController** class code block, press Enter, type the following code, and then press Enter.

cs private OperasDB contextDB = new OperasDB(); 7. In the **Index** action code block, select the following code.

cs return View(); 8. Replace the selected code with the following code.

cs return View("Index", contextDB.Operas.ToList()); 9. Ensure that the cursor is at the end of the **Index** action code block, press Enter, and then type the following code.

cs public ActionResult Details(int id) { } 10. In the **Details** action code block, type the following code.

cs Opera opera = contextDB.Operas.Find(id); if (opera != null) { return View("Details", opera); } else { return HttpNotFound(); } 11. Place the mouse cursor at the end of the **Details** action code block, press Enter twice, and then type the following code.

cs public ActionResult Create() { } 12. In the **Create** action code block, type the following code.

cs Opera newOpera = new Opera(); return View("Create", newOpera); 13. Place the mouse cursor at the end of the **Create** action code block, press Enter twice, and then type the following code.

cs [HttpPost] public ActionResult Create(Opera newOpera) { } 14. Place the mouse cursor in the **Create** action code block with the HTTP verb **POST**, and then type the following code.

cs if (ModelState.IsValid) { contextDB.Operas.Add(newOpera); contextDB.SaveChanges(); return RedirectToAction("Index"); } else { return View("Create", newOpera); } 15. On the **FILE** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Save Controllers.cs**. 16. In the **OperasWebSite - Microsoft Visual Studio** window, click the **Close** button. 17. In the

Microsoft Visual Studio dialog box, note that the message, **Save changes to the following items?** is displayed, and then click **Yes**.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Module 5: Developing ASP.NET MVC 4 Views

Lesson 2: Using HTML Helpers

Demonstration: How to Use HTML Helpers

Preparation Steps

1. Log on to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Start **File Explorer**.
3. Navigate to **AllFiles (D):05*.
4. Open the **OperasWebSite.sln** project.
5. Enable the **Allow NuGet to download missing packages during build** option, by performing the following steps:

- On the **TOOLS** menu of the Microsoft Visual Studio window, click **Options**.
- In the navigation pane of the **Options** dialog box, click **Package Manager**.
- Under the Package Restore section, select the **Allow NuGet to download missing packages during build** checkbox, and then click **OK**.

6. On the **Build** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Build Solution**, and then note that the application is built successfully.

Note: In Hyper-V Manager, start the **MSL-TMG1** virtual machine if it is not already running.

Demonstration Steps

1. In the Solution Explorer pane of the **OperasWebSite - Microsoft Visual Studio** window, expand **Controllers** , and then click **OperaController.cs**.

2. In the **OperaController.cs** code window, locate the following code, right-click the code, and then click **Add View**.

`cs            public ActionResult Create()` 3. In the **View Name** box of the **Add View** dialog box, ensure that the name displayed is **Create**. 4. In the **Add View** dialog box, ensure that the **Create a strongly-typed view** check box is selected. 5. In the **Model class** box, ensure that the value is **Opera (OperasWebSite.Models)**. If not, in the **Model class** box, click **Opera (OperasWebSite.Models)**. 6. In the **Scaffold template** box, ensure that the value is **Empty**. 7. In the **Add View** dialog box, ensure that the **Use a layout or master page** check box is not selected, and then click **Add**. 8. In the **DIV** element of the **Create.cshtml** code window, type the following code.

```
<h2>Add an Opera</h2>
```

9. Place the mouse cursor at the end of the **</h2>** tag, press Enter twice, and then type the following code.

`cs      @using (Html.BeginForm("Create","Opera",FormMethod.Post)) {        }` 10. In the **using** code block, type the following code.

`cs      <p>           @Html.LabelFor(model =>model.Title): @Html.EditorFor(model =>model.Title) @Html.ValidationMessageFor(model =>model.Title)         </p>` 11. Place the mouse cursor at the end of the **</p>** tag corresponding to the **model.Title** property, press Enter twice, and then type the following code.

`cs         <p>             @Html.LabelFor(model =>model.Year): @Html.EditorFor(model =>model.Year) @Html.ValidationMessageFor(model => model.Year)                </p>` 12. Place the mouse cursor at the end of the **</p>** tag corresponding to the **model.Year** property, press Enter twice, and then type the following code.

`cs         <p>             @Html.LabelFor(model =>model.Composer): @Html.EditorFor(model =>model.Composer) @Html.ValidationMessageFor(model => model.Composer)         </p>` 13. Place the mouse cursor at the end of the **</p>** tag corresponding to the **model.Composer** property, press Enter twice, and then type the following code.

`cs         <input type="submit" value="Create"/>` 14. Place the mouse cursor at the end of the **<input>** tag, press Enter, and then type the following code.

`cs         @Html.ActionLink("Back to List", "Index")` 15. On the **DEBUG** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Start Debugging**.

`>**Note** : The Operas I Have Seen page is displayed.`

16. On the Operas I Have Seen page, click **operas I've seen**.

    Note : On the Index page, the list of Operas is displayed.

17. On the Index page, click **Create New**.

    Note : The Add an Opera page is displayed.

18. In the **Year** box of the Add an Opera page, type **1597** , and then click **Create**.

    Note : Messages corresponding to the Title , Year , and Composer boxes are displayed. The web application mandates you to enter values in all the boxes. Alerts are also displayed for any inappropriate entries, with relevant messages.

19. In the **Title** box of the Add an Opera page, type **Rigoletto**.
20. In the **Year** box of the Add an Opera page, type **1851**.
21. In the **Composer** box of the Add an Opera page, type **Verdi** , and then click **Create**.

    Note : The Opera is created with the mentioned values.

22. In the Windows Internet Explorer window, click the **Close** button.
23. In the **OperasWebSite - Microsoft Visual Studio** window, click the **Close** button.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Module 6: Testing and Debugging ASP.NET MVC 4 Web Applications

Lesson 1: Unit Testing MVC Components

Demonstration: How to run unit tests

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Start **File Explorer**.
3. Go to **AllFiles (D):06*.
4. Double-click **OperasWebSite.sln**.
5. Enable the **Allow NuGet to download missing packages during build** option, by performing the following steps:

- On the **TOOLS** menu of the Microsoft Visual Studio window, click **Options**.

- In the navigation pane of the **Options** dialog box, click **Package Manager**.

- Under the **Package Restore** section, select the **Allow NuGet to download missing packages during build** check box, and then click **OK**.

6. On the **Build** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Build Solution**, and then note that the application is built successfully.

Note: In Hyper-V Manager, start the **MSL-TMG1** virtual machine if it is not already running.

Demonstration Steps

1. In the Solution Explorer pane of the **OperasWebSite - Microsoft Visual Studio** window, right-click **Solution 'OperasWebSite' (1 project)**, point to **Add**, and then click **New Project**.
2. In the navigation pane of the **Add New Project** dialog box, under Installed, expand **Visual C#**, and then click **Test**.
3. In the result pane of the **Add New Project** dialog box, click **Unit Test Project**, in the **Name** box, type **OperasWebSiteTests**, and then click **OK**.
4. In the Solution Explorer pane, under OperasWebSiteTests, right-click **References**, and then click **Add Reference**.
5. In the navigation pane of the **Reference Manager - OperasWebSiteTests** dialog box, click **Solution**.
6. In the **Name** column of the result pane, click **OperasWebSite**, select the check box corresponding to OperasWebSite, and then click **OK**.
7. In the Solution Explorer pane, under **OperasWebSiteTests**, right-click **References**, and then click **Add Reference**.
8. In the navigation pane of the **Reference Manager - OperasWebSiteTests** dialog box, click **Assemblies**, and then click **Extensions**.
9. In the **Name** column of the result pane, click **System.Web.Mvc** with version number **4.0.0.0**, select the corresponding check box, and then click **OK**.
10. In the Solution Explorer pane, under **OperasWebSiteTests**, right-click **UnitTest1.cs**, and then click **Rename**.
11. In the Solution Explorer pane, replace **UnitTest1** with **HomeControllerTests.cs**, and then press Enter.
12. In the **Microsoft Visual Studio** dialog box, click **Yes**.
13. In the HomeControllerTests.cs code window, locate the following code.

`cs          public void TestMethod1()` 14. Replace the code with the following code.

`cs          public void Test_Index_Return_View()` 15. Ensure that the cursor is at the end of the Microsoft.VisualStudio.TestTools.UnitTesting namespace, press Enter, and then type the following code.

`cs          using System.Web.Mvc;          using OperasWebSite.Controllers;          using OperasWebSite.Models;` 16. In the **Test_Index_Return_View** code block, press Enter, and then type the following code.

`cs          HomeController controller = new HomeController();`
`var result = controller.Index() as ViewResult;`
`Assert.AreEqual("WrongName", result.ViewName);` >**Note** : This test is created to show the students a failing test.

17. On the **TEST** menu of the **OperasWebSite - Microsoft Visual Studio** window, point to **Run**, and then click **All Tests**.
18. In the Failed Tests (1) section of the Test Explorer pane, note that **Test_Index_Return_View** is listed.
19. In the Test Explorer pane, click **Test_Index_Return_View**.
20. At the bottom of the Test Explorer pane, drag the separator upward, and then view the test results.
21. In the Test Explorer pane, click the **Close** button.
22. In the Solution Explorer pane, under OperasWebSiteTests, click **HomeControllerTests.cs**.
23. In the HomeControllerTests.cs code window, locate the following code.

`cs          Assert.AreEqual("WrongName", result.ViewName);` 24. Replace the code with the following code.

`cs          Assert.AreEqual("Index", result.ViewName);` 25. On the **TEST** menu of the **OperasWebSite - Microsoft Visual Studio** window, point to **Run**, and then click **All Tests**. 26. In the Passed Tests (1) section of the Test Explorer pane, note that **Test_Index_Return_View** is listed. 27. In the Test Explorer pane, click **Test_Index_Return_View**, and then, in the lower part of the Test Explorer pane, view the test results. 28. In the Test Explorer pane, click the **Close** button. 29. In the **OperasWebSite - Microsoft Visual Studio** window, click the **Close** button.

This document is provided “as-is.” Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Version: 2017

Module 7: Structuring ASP.NET MVC 4 Web Applications

Lesson 2: Configuring Routes

Demonstration: How to Add Routes

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Open **File Explorer**.
3. Go to **AllFiles (D):07*.
4. Double-click **OperasWebSite.sln**.
5. Enable the **Allow NuGet to download missing packages during build** option, by performing the following steps:

a. On the **TOOLS** menu of the **Microsoft Visual Studio** window, click **Options**.

b. In the navigation pane of the **Options** dialog box, click **Package Manager**.

c. Under the **Package Restore** section, select the **Allow NuGet to download missing packages during build** checkbox, and then click **OK**.

6. On the **Build** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Build Solution**, and then note that the application is built successfully.

Note: In Hyper-V Manager, start the MSL-TMG1 virtual machine, if it is not already running.

Demonstration Steps

1. On the **DEBUG** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Start Debugging**.
2. On the **Operas I Have Seen** page, click the **operas I've seen** link.
3. On the **Index** page, click the **Details** link corresponding to **Cosi Fan Tutte**.
4. In the Address bar of the **Internet Explorer** window, note that the URL is **http://localhost:<portnumber>/Opera/Details/1**.

Note: This URL indicates that the controller is Opera, the action is Details, and the ID is 1.

5. In the **Internet Explorer** window, click the **Close** button.
6. In the **Solution Explorer** pane, expand **OperasWebSite**, expand **Controllers**, and then click **OperaController.cs**.
7. In the **OperaController.cs** code window, place the mouse cursor at the end of the **Details** action code block, press Enter twice, and then type the following code.

`cs public ActionResult DetailsByTitle(string title) { }` 8. In the **DetailsByTitle** action code block, type the following code, and then press Enter.

`cs Opera opera = (Opera)(from o in contextDB.Operas where o.Title == title select o).FirstOrDefault();` 9. In the **DetailsByTitle** action code block, after the code that you just typed, type the following code.

cs   if (opera == null)   {   return HttpNotFound();   }   return View("Details", opera); 10. In the **Solution Explorer** pane, under OperasWebSite, expand **App_Start**, and then click **RouteConfig.cs**. 11. In the **RouteConfig.cs** code window, locate the following code.

cs   routes.IgnoreRoute("{resource}.axd/{*pathInfo}"); 12. Place the mouse cursor at the end of the call to the **IgnoreRoute()** method, press Enter twice, and then type the following code.

cs   routes.MapRoute(name: "OperaTitleRoute",url: "opera/title/{title}",defaults: new { controller = "Opera", action ="DetailsByTitle" }); 13. On the **FILE** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Save All**. 14. On the **DEBUG** menu of **OperasWebSite - Microsoft Visual Studio** window, click **Start Debugging**. 15. On the Operas I Have Seen page, click the **operas I've seen** link. 16. In the Address bar of the **Internet Explorer** window, append the existing URL with **/title/rigoletto**, and then click **Go**.

Note: The details of the Rigoletto opera are displayed.

17. In the **Internet Explorer** window, click **Close**.
18. In the **OperasWebSite - Microsoft Visual Studio** window, click **Close**.

Lesson 3: Creating a Navigation Structure

Demonstration: How to Build Site Navigation

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Open **File Explorer**.
3. Go to **AllFiles (D):07*.
4. Double-click **OperasWebSite.sln**.
5. Enable the **Allow NuGet to download missing packages during build** option, by performing the following steps:

a. On the **TOOLS** menu of the **Microsoft Visual Studio** window, click **Options**.

b. In the navigation pane of the **Options** dialog box, click **Package Manager**.

c. Under the **Package Restore** section, select the **Allow NuGet to download missing packages during build** checkbox, and then click **OK**.

6. On the **Build** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Build Solution**, and then note that the application is built successfully.

Note: In Hyper-V Manager, start the MSL-TMG1 virtual machine, if it is not already running.

Demonstration Steps

1. On the **Tools** menu of the **OperasWebSite - Microsoft Visual Studio** window, point to **Library Package Manager** and then click **Package Manager Console**.

2. In **Package Manager Console** window, type the following command and then press Enter

`cs        install-package mvcsitemapprovider –version 3.3.4.0` 3. In the **Solution Explorer** pane of the **OperasWebSite - Microsoft Visual Studio** window, expand **OperasWebSite**, collapse **App_Start**, and then collapse **Controllers**. 4. In the **Solution Explorer** pane, under **Global.asax**, click **Mvc.sitemap**. 5. In the **Mvc.sitemap** code window, locate the following code.

`cs        <mvcSiteMapNode title="Home" controller="Home" action="Index">` 6. Place the mouse cursor at the end of the located code, press Enter, and then type the following code.

`cs        <mvcSiteMapNode title="All Operas" controller="Opera" action="Index" key="AllOperas" />` 7. On the **BUILD** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Build Solution**. 8. In the **Solution Explorer** pane, expand **Views**, expand **Home**, and then click **Index.cshtml**. 9. In the **Index.cshtml** code window, place the mouse cursor after the **<div>** tag, press Enter, and then type the following code.

`cs        Menu: @Html.MvcSiteMap().Menu(false, false, true)` 10. Place the mouse cursor at the end of the site map menu code block, press Enter, and then type the following code.

`cs         Breadcrumb Trail: @Html.MvcSiteMap().SiteMapPath()` 11. In the **Solution Explorer** pane, under **Views**, expand **Opera**, and then click **Index.cshtml**. 12. In the **Index.cshtml** code window, place the mouse cursor at the end of the **<body>** tag, press Enter, and then type the following code.

`cs         Menu: @Html.MvcSiteMap().Menu(false, false, true)` 13. Place the mouse cursor at the end of the site map menu code block, press Enter, and then type the following code.

`cs         Breadcrumb Trail: @Html.MvcSiteMap().SiteMapPath()` 14. On the **DEBUG** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Start Debugging**.

`>**Note:** On the **Operas I Have Seen** page, ensure that a menu is added.`

15. On the **Operas I Have Seen** page, under **Menu**, click the **All Operas** link.
16. On the **Index** page, note that the **Main Opera List** is displayed.

    Note: On the Index page, you can also view the menu.

17. In the **Breadcrumb Trail** section of the Index page, click the **Home** link.

    Note: The Operas I Have Seen page is displayed.

18. On the **Operas I Have Seen** page, under **Menu**, click the **About** link.

    Note: The About page of the web application is displayed.

19. In the **Internet Explorer** window, click the **Close** button.

20. In the **OperasWebSite - Microsoft Visual Studio** window, click **Close**.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Module 8: Applying Styles to ASP.NET MVC 4 Web Applications

Lesson 2: Applying CSS Styles to an MVC Application

Demonstration: How to Apply a Consistent Look and Feel

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name, **Admin**, and the password, **Pa$$w0rd**.
2. Open **File Explorer**.
3. Navigate to **Allfiles (D):08*.
4. Double-click **OperasWebSite.sln**.
5. Enable the **Allow NuGet to download missing packages during build** option, by performing the following steps:

a. On the **TOOLS** menu of the **Microsoft Visual Studio** window, click **Options**.

b. In the navigation pane of the **Options** dialog box, click **Package Manager**.

c. Under the **Package Restore** section, select the **Allow NuGet to download missing packages during build** check box, and then click **OK**.

Note: In Hyper-V Manager, start the MSL-TMG1 virtual machine if it is not already running.

Demonstration Steps

1. On the **DEBUG** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Start Debugging**.

Note: On the Operas I Have Seen page, note that the main heading, the menu list, and the breadcrumb control are displayed.

2. On the **Operas I Have Seen** page, click the **All Operas** link.

Note: On the localhost page, the main heading, the menu list, and the breadcrumb controls are not displayed.

3. On the localhost page, click the **Details** link corresponding to any opera.

Note: On the localhost page, the details of the opera are displayed. The main heading, the menu list, and the breadcrumb controls are not displayed.

4. In the Internet Explorer window, click the **Close** button.
5. In the Solution Explorer pane, expand **OperasWebSite**, and then expand **Views**.
6. In the Solution Explorer pane, under Views, right-click **Shared**, point to **Add**, and then click **View**.
7. In the **View name** box of the **Add View** dialog box, type **_SiteTemplate**.
8. In the **View engine** box, ensure that the value is **Razor (CSHTML)**, and then ensure that the **Create a strongly-typed view** check box is cleared.
9. In the **Add View** dialog box, clear the **Use a layout or master page** check box, and then click **Add**.
10. In the **_SiteTemplate.cshtml** code window, locate the following code, select the code, and then press Delete.

cs   @{   Layout = null;   } 11. In the **_SiteTemplate.cshtml** code window, locate the following code.

cs   <title>_SiteTemplate</title> 12. Replace the **TITLE** element with the following code.

cs   <title>@ViewBag.Title</title> 13. In the Solution Explorer pane, under Views, expand **Home**, and then click **Index.cshtml**. 14. In the **Index.cshtml** code window, locate the following code, and then select the code.

cs   <h1>Operas I Have Seen</h1>   <div class="topmenu"> @Html.MvcSiteMap().Menu(false, true, true)   </div> <div class="clear-floats" />   <div class="breadcrumb"> Breadcrumb Trail: @Html.MvcSiteMap().SiteMapPath()   </div> 15. On the **EDIT** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Cut**. 16. In the Solution Explorer pane, under Shared, click **_SiteTemplate.cshtml**. 17. In the _SiteTemplate.cshtml** code window, place the cursor in the **DIV** element. 18. On the **EDIT** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Paste**. 19. In the **_SiteTemplate.cshtml** code window, place the cursor at the end of the code you just pasted, press Enter, and then type the following code.

cs   <div>   @RenderBody()   </div> 20. Place the cursor after the **</title>** tag, press Enter, and then type the following code.

cs   <link type="text/css" rel="stylesheet" href="~/content/OperasStyles.css" /> 21. In the Solution Explorer pane, under Home, click **Index.cshtml**. 22. In the Razor code block of the **Index.cshtml** code window, locate the following code, select the code, and then press Delete.

cs   Layout = null; 23. In the Razor code block, type the following code.

cs   ViewBag.Title = "Operas I Have Seen"; 24. In the **Index.cshtml** code window, locate the following code, select the code, and then press Delete.

cs <!DOCTYPE html> <html> <head> <meta name="viewport" content="width=device-width" /> <title>Operas I Have Seen</title> </head> <body> <div> 25. In the **Index.cshtml** code window, locate the following code, select the code, and then press Delete.

cs </div> </body> </html> 26. In the Solution Explorer pane, right-click **Views**, point to **Add**, and then click **View**. 27. In the **View name** box of the **Add View** dialog box, type **_ViewStart**, and then, in the** View engine** box, ensure that the value is **Razor (CSHTML)**. 28. In the **Add View** dialog box, ensure that the **Create a strongly-typed view** and **Use a layout or master page** check boxes are cleared, and then click **Add**. 29. In the **_ViewStart.cshtml** code window, locate the following code.

cs Layout = null; 30. Replace the code with the following code.

cs Layout = "~/Views/Shared/_SiteTemplate.cshtml"; 31. In the **_ViewStart.cshtml** code window, locate the following code.

cs <!DOCTYPE html> 32. In the **_ViewStart.cshtml** code window, select from the code located to the end tag of the HTML element, and then press Delete 33. On the **DEBUG** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Start Debugging**. 34. On the **Operas I Have Seen** page, note the main heading, the menu list, and the breadcrumb control. 35. On the **Operas I Have Seen** page, click the **All Operas** link, and then, on the **Index of Operas** page, note that the main heading, the menu list, and the breadcrumb controls are displayed. 36. On the **Index of Operas** page, click the **Details** link corresponding to any opera, and then note that the main heading, the menu list, and the breadcrumb controls are displayed along with the details of the opera. 37. In the Internet Explorer window, click the **Close** button. 38. In the **OperasWebSite - Microsoft Visual Studio** window, click the **Close** button.

This document is provided “as-is.” Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Version: 2017

Module 9: Building Responsive Pages in ASP.NET MVC 4 Web Applications

Lesson 2: Implementing a Caching Strategy

Demonstration: How to Configure Caching

Preparation Steps

1. Log on to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Start **File Explorer**.
3. Navigate to **Allfiles (D):09*.
4. Double-click **OperasWebSite.sln**
5. Enable the **Allow NuGet to download missing packages during build** option, by performing the following steps:

a. On the **TOOLS** menu of the **Microsoft Visual Studio** window, click **Options**.

b. In the navigation pane of the **Options** dialog box, click **Package Manager**.

c. Under the **Package Restore** section, select the **Allow NuGet to download missing packages during build** checkbox, and then click **OK**.

Note: In Hyper-V Manager, start the **MSL-TMG1** virtual machine if it is not already running.

Demonstration Steps

1. On the **DEBUG** menu of the **OperasWebSite – Microsoft Visual Studio** window, click **Start Debugging**.
2. On the Operas I Have Seen page, click the **Tools** button, and then click **F12 developer tools**.
3. On the **Cache** menu of the developer window, click **Always refresh from server**.
4. On the **Network** tab of the developer window, click **Start capturing**.
5. On the Operas I Have Seen page, click the **All Operas** link.
6. When the page is fully loaded, in the developer window, click **Stop capturing**.
7. In the URL section of the developer window, click **http://localhost:**<*portnumber*>**/Opera**, and then click **Go to detailed view**.
8. On the **Timings** tab, click the **Request** entry.
9. In the **Duration** column, note the value displayed.
10. On the **Network** tab, click **Clear** , and then click **Start capturing**.
11. On the Operas I Have Seen page, click the **All Operas** link.
12. When the page is fully loaded, in the developer window, click **Stop capturing**.
13. In the URL section of the developer window, click **http://localhost:**<*portnumber*>**/Opera**, and then click **Go to detailed view**.
14. On the **Timings** tab, click the **Request** entry.

15. In the **Duration** column, note the value displayed.

Note: The time taken by the server to render the **/Opera** page and return the page to the browser is similar to the time taken by the server in the first instance. The page is not cached.

16. In the Windows Internet Explorer window, click the **Close** button.
17. In the Solution Explorer pane of the **OperasWebSite – Microsoft Visual Studio** window, expand **OperasWebSite** , expand **Controllers** , and then click **OperaController.cs**.
18. In the OperaController.cs code window, locate the following code.

`cs          using System.Web.Mvc;` 19. Place the mouse cursor at the end of the located code, press Enter, and then type the following code.

`cs          using System.Web.UI;` 20. In the OperaController.cs code window, locate the following code.

`cs          public ActionResult Index()` 21. Place the mouse cursor immediately before the located code, press Enter, and then type the following code.

`cs          [OutputCache(Duration=600, Location=OutputCacheLocation.Server, VaryByParam="none")]` 22. On the **DEBUG** menu of the **OperasWebSite – Microsoft Visual Studio** window, click **Start Debugging**. 23. On the **Cache** menu of the developer window, click **Always refresh from server**. 24. On the **Network** tab, click **Start capturing**. 25. On the Operas I Have Seen page, click the **All Operas** link. 26. When the page is fully loaded, in the developer window, click **Stop capturing**. 27. In the URL section of the developer window, click **http://localhost:***<portnumber>***/Opera**, and then click **Go to detailed view**. 28. On the **Timings** tab, click the **Request** entry. 29. In the **Duration** column, note the value displayed. 30. On the **Network** tab, click **Clear** , and then click **Start capturing**. 31. On the Operas I Have Seen page, click the **All Operas** link. 32. When the page is fully loaded, in the developer window, click **Stop capturing**. 33. In the URL section of the developer window, click **http://localhost:***<portnumber>***/Opera**, and then click **Go to detailed view**. 34. On the **Timings** tab, click the **Request** entry. 35. In the **Duration** column, note the value displayed.

```
>**Note** : Note that the time taken by the server to render the **/Opera*
* page and return the page to the browser is significantly less than the t
ime taken by the server in the first instance.
```

36. On the **File** menu of the developer window, click **Exit**.
37. In the Windows Internet Explorer window, click the **Close** button.
38. In the **OperasWebSite – Microsoft Visual Studio** window, click the **Close** button.

Version: 2017

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Module 10: Using JavaScript and jQuery for Responsive MVC 4 Web Applications

Lesson 1: Rendering and Executing JavaScript Code

Demonstration: How to Use NuGet to Add a JavaScript Library

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Open **File Explorer**.
3. Go to **Allfiles (D):10*.
4. Double-click **OperasWebSite.sln**.

Note: In Hyper-V Manager, start the **MSL-TMG1** virtual machine, if it is not already running.

Demonstration Steps

1. In the **Solution Explorer** pane of the **OperasWebSite - Microsoft Visual Studio** window, expand **OperasWebSite**.

Note: There is no folder named **Scripts** at the top level of the project.

2. In the **Solution Explorer** pane, expand **Content**.

Note: The **Content** folder has only one file named **OperaStyles.css** , and there are no sub-folders.

3. On the **Tools** menu of the **OperasWebSite - Microsoft Visual Studio** window, point to **Library Package Manager**, and then click **Package Manager Console**.
4. In **Package Manager Console** window, type the following command, and then press Enter

`cs        install-package jQuery.UI.Combined –version 1.8.17` 5. In the **Solution Explorer** pane, expand **Scripts**.

```
>**Note:** NuGet Package Manager has added five files for jquery and jquer
yUI to the application. Note the version number for jquery and jqueryUI.
```

6. In the **Solution Explorer** pane, under **Contents**, expand **themes**, expand **base**, and then click **jquery-ui.css**.

   Note: NuGet Package Manager has added style sheets to the **Content** folder. These styles are used to set the styles for jQueryUI widgets, and the most important of these style sheets is **jquery-ui.css**.

7. In the **Solution Explorer** pane, collapse **base**, expand **Views**, and then expand **Shared**.
8. In the **Solution Explorer** pane, under **Shared**, click **_SiteTemplate.cshtml**.

9. In the **_SiteTemplate.cshtml** code window, locate the following code.

cs `</head>` 10. Place the mouse cursor before the located code, type the following code, and then press Enter.

cs `<script type="text/javascript" src="@Url.Content("~/Scripts/jquery-ui-1.8.17.js")"></script>` 11. In the **_SiteTemplate.cshtml** code window, locate the following code.

cs `<title>@ViewBag.Title</title>` 12. Place the mouse cursor at the end of the located code, press Enter, and then type the following code.

cs `<link type="text/css" rel="stylesheet" href="@Url.Content("~/Content/themes/base/jquery-ui.css")" />` >**Note:** You can now use jQueryUI calls on any views in the application.

13. On the **FILE** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Save All**.
14. In the **OperasWebSite - Microsoft Visual Studio** window, click **Close**.

Lesson 2: Using jQuery and jQueryUI

Demonstration: How to Add a jQueryUI Widget

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Open **File Explorer**.
3. Go to **Allfiles (D):10*
4. Double-click **OperasWebSite.sln**.
5. Enable the **Allow NuGet to download missing packages during build** option, by performing the following steps:

a. On the **TOOLS** menu of the **Microsoft Visual Studio** window, click **Options**.

b. In the navigation pane of the **Options** dialog box, click **Package Manager**.

c. Under the **Package Restore** section, select the **Allow NuGet to download missing packages during build** checkbox, and then click **OK**.

Note: In Hyper-V Manager, start the MSL-TMG1 virtual machine, if it is not already running.

Demonstration Steps

1. On the **DEBUG** menu of the **OperasWebsite – Microsoft Visual Studio** window, click **Start Debugging**.
2. On the **Operas I Have Seen** page, click the **All Operas** link.
3. In the **Main Opera List** section, click the **Details** link corresponding to **Cosi Fan Tutte**.

4. Under **Reviews**, note that there are three opera reviews displayed for **Cosi Fan Tutte**, simultaneously.
5. In the **Internet Explorer** window, click **Close**.
6. In the **Solution Explorer** pane of the **OperasWebsite – Microsoft Visual Studio** window, under **Shared**, click **_ReviewsForOpera.cshtml**.
7. In the **_ReviewsForOpera.cshtml** code window, locate the following code.

`cs       <h3>Reviews</h3>` 8. Place the mouse cursor immediately before the located code, type the following code, and then press Enter.

`cs      <script>         $(function() {             $("#reviews-tool").accordion();          });      </script>` 9. On the **DEBUG** menu of the **OperasWebsite – Microsoft Visual Studio** window, click **Start Debugging**. 10. On the **Operas I Have Seen** page, click the **All Operas** link. 11. In the **Main Opera List** section, click the **Details** link corresponding to **Cosi Fan Tutte**. 12. Under **Reviews**, note that there are three expandable sections, and each section contains a review.

```
 >**Note:** You can expand each section and then read the review content.
```

13. In the **Internet Explorer** window, click **Close**.
14. In the **OperasWebSite – Microsoft Visual Studio** window, click the **Close** button.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Module 11: Controlling Access to ASP.NET MVC 4 Web Applications

Lesson 1: Implementing Authentication and Authorization

Demonstration: How to Authorize Access to Controller Actions

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Start **File Explorer**.
3. Go to **Allfiles (D):11*.

4. Double-click **OperasWebSite.sln**.
5. Enable the **Allow NuGet to download missing packages during build** option, by performing the following steps:

a. On the **TOOLS** menu of the **Microsoft Visual Studio** window, click **Options**.

b. In the navigation pane of the **Options** dialog box, click **Package Manager**.

c. Under the **Package Restore** section, select the **Allow NuGet to download missing packages during build** checkbox, and then click **OK**.

Note: In Hyper-V Manager, start the MSL-TMG1 virtual machine if it is not already running.

Demonstration Steps

1. On the **DEBUG** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Start Debugging**.
2. On the **Operas I Have Seen** page, click **All Operas**.
3. On the **Index of Operas** page, click the **Create New** link.

Note: The Create an Opera page is displayed without signing in to the application. This enables anonymous users to create new operas.

4. In the Windows Internet Explorer window, click the **Close** button.
5. In the Solution Explorer pane of the **OperasWebSite - Microsoft Visual Studio** window, expand **OperasWebSite**, expand **Controllers**, and then click **OperaController.cs**.
6. In the OperaController.cs code window, locate the following code.

`cs         public ActionResult Create()` 7. Place the mouse cursor before the located code, type the following code, and then press Enter.

`cs         [Authorize]` 8. In the OperaController.cs code window, locate the following code.

`cs         [HttpPost]         public ActionResult Create(Opera newOpera)` 9. Place the mouse cursor before the located code, type the following code, and then press Enter.

`cs         [Authorize]` 10. On the **FILE** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Save All**. 11. On the **DEBUG** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Start Debugging**. 12. On the **Operas I Have Seen** page, click **All Operas**. 13. On the **Index of Operas** page, click the **Create New** link.

Note: The Login view is now displayed and this prevents anonymous users from creating new operas.

14. On the **Index of Operas** page, click the **Register** link.
15. In the **User name** text box of the **Register** page, type **David Johnson**.
16. In the **Password** text box, type **Pa**$w0rd **, inthe ** Confirmpassword ** textbox, type ** Pa$**w0rd**, and then click **Register**.
17. On the **Operas I Have Seen** page, click **All Operas**.
18. On the **Index of Operas** page, click the **Create New** link.

Note: The Add an Opera page is displayed because the request is authenticated.

19. In the Windows Internet Explorer window, click the **Close** button.
20. In the **OperasWebSite - Microsoft Visual Studio** window, click the **Close** button.

Lesson 2: Assigning Roles and Membership

Demonstration: How to Reset a Password

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Start **File Explorer**.
3. Go to **Allfiles (D):11*.
4. Double-click **OperasWebSite.sln**.

Note: In Hyper-V Manager, start the **MSL-TMG1** virtual machine if it is not already running.

Demonstration Steps

1. In the Solution Explorer pane of the **OperasWebSite - Microsoft Visual Studio** window, under **Controllers**, click **AccountController.cs**.
2. In the AccountController.cs code window, locate the following comment.

`cs          //Add Reset Password Code Here` 3. In the AccountController.cs code window, replace the located comment with the following code.

`cs         try         {         }         catch (Exception)         {` } 4. In the **try** code block, type the following code.

`cs         changePasswordSucceeded = Membership.Provider.ChangePassword(User.Identity.Name, model.OldPassword, model.NewPassword);` 5. In the **catch** code block, type the following code.

`cs         changePasswordSucceeded = false;` 6. On the **FILE** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Save All**. 7. On the **DEBUG** menu of the **OperasWebSite - Microsoft Visual Studio** window, click **Start Debugging**. 8. On the **Operas I Have Seen** page, click the **Log In** link. 9. On the **Logon** page, in the **User name** text box, type **David Johnson**, in the **Password** box, type **Pa**$w0rd \ast\ast, andthenclick \ast\ast Login \ast\ast$ $.10. Onthe \ast\ast OperasIHaveSeen \ast\ast page, clickthe \ast\ast ResetPassword \ast\ast link. 11. Onthe \ast\ast ResetPassword \ast\ast page, inthe \ast\ast Currentpassword \ast\ast textbox, type \ast\ast Pa$**w0rd**, and then in the **New password** text box, type **Pa**$w0rd2 \ast\ast$ $.12. Inthe \ast\ast Confirmnewpassword \ast\ast textbox, type \ast\ast Pa$**w0rd2**, and then click **Change Password**.

Note: On the ResetPassword page, the message, **Your password has been changed.** is displayed.

13. In the Windows Internet Explorer window, click the **Close** button.
14. In the OperasWebSite – Microsoft Visual Studio window, click the **Close** button.
15. If the Microsoft Visual Studio warning message appears, click **Yes**.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Module 12: Building a Resilient ASP.NET MVC 4 Web Application

Lesson 2: State Management

Demonstration: How to Store and Retrieve State Information

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Open **File Explorer**.
3. Go to **Allfiles (D):12*.
4. Double-click **OperasWebSite.sln**.
5. Enable the **Allow NuGet to download missing packages during build** option, by performing the following steps:

a. On the **TOOLS** menu of the **Microsoft Visual Studio** window, click **Options**.

b. In the navigation pane of the **Options** dialog box, click **Package Manager**.

c. Under the **Package Restore** section, select the **Allow NuGet to download missing packages during build** checkbox, and then click **OK**.

Note: In Hyper-V Manager, start the **MSL-TMG1** virtual machine, if it is not already running.

Demonstration Steps

1. In the **Solution Explorer** pane of the **OperasWebSite – Microsoft Visual Studio** window, under **OperasWebSite**, expand **Controllers**, and then click **HomeController.cs**.
2. In the **HomeController.cs** code window, locate the following code.

`cs public ActionResult About() { return View(); }` 3. Place the mouse cursor at the end of the located code, press Enter twice, and then type the following code.

cs public ContentResult GetBackground() { } 4. Place the mouse cursor within the **GetBackground** action code block, and then type the following code.

cs string style; if (Session["BackgroundColor"] != null) { } else { } 5. Place the mouse cursor within the **if** statement code block you just added, and then type the following code.

cs style = String.Format("background-color: {0};", Session["BackgroundColor"]); 6. Place the mouse cursor within the **else** statement code block you just added, and then type the following code.

cs style = "background-color: #dc9797;"; 7. Place the mouse cursor at the end of the **GetBackground** action code block and outside the **if…else** statements, press Enter, and then type the following code.

cs return Content(style); 8. Place the mouse cursor outside any action code block but inside the **HomeController** class, and then type the following code.

cs public ActionResult SetBackground(string color) { } 9. Place the cursor within the **SetBackground** action code block, and then type the following code.

cs Session["BackgroundColor"] = color; return View("Index"); 10. In the **Solution Explorer** pane, expand **Views**, expand **Home**, and then click **Index.cshtml**. 11. In the **Index.cshtml** code window, place the mouse cursor at the end of the **P** element, press Enter twice, and then type the following code.

cs <p> Choose a background color: </p> 12. Place the mouse cursor at the end of the text in the **P** element, press Enter, and then type the following code.

cs @Html.ActionLink("Pink", "SetBackground", new { color = "#dc9797"}) 13. Place the mouse cursor at the end of the link you just created, press Enter, and then type the following code.

cs @Html.ActionLink("Blue", "SetBackground", new { color = "#82bbf2"}) 14. In the **Solution Explorer** pane, expand **Shared**, and then click **_SiteTemplate.cshtml. **15. In the** _SiteTemplate.cshtml** code window, locate the following code.

cs <body> 16. Replace the located code with the following code.

cs <body style="@Html.Action("GetBackground", "Home")"> 17. On the **DEBUG** menu of the **OperasWebSite – Microsoft Visual Studio** window, click **Start Debugging**. 18. On the **Operas I Have Seen** page, click the **Blue** link, and then note that the blue background color has been applied to the home page. 19. On the **Operas I Have Seen** page, click the **All Operas** link.

```
>**Note:** If the background color of the page is blue, click the **Refres
h** button.
```

20. On the **Index of Operas** page, click the **Details** link corresponding to the title, **Cosi Fan Tutte**.
21. On the **Cosi Fan Tutte** page, note that the blue background color has been applied to the page.

    Note: The blue background preference is applied to all pages of the application.

22. On the **Opera: Cosi Fan Tutte** page, click **Home**.
23. On the **Operas I Have Seen** page, click the **Pink** link, and then note that the pink background color has been applied to the home page.

    Note: If the background color of the page is blue, click **Refresh**.

24. On the **Operas I Have Seen** page, click **All Operas**.
25. On the **Index of Operas** page, click the **Details** link corresponding to the title, **Cosi Fan Tutte**.
26. On the **Cosi Fan Tutte** page, note that the pink background color has been applied to the page.

    Note: The pink background preference is applied to all pages of the application.

27. In the **Internet Explorer** window, click **Close**.
28. In the **OperasWebSite – Microsoft Visual Studio** window, click **Close**.
29. If the message **Do you want to stop debugging?** is displayed, click **Yes**.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Module 13: Using Windows Azure Web Services in ASP.NET MVC 4 Web Applications

Lesson 3: Consuming Windows Azure Services in a Web Application

Demonstration: How to Call a Windows Azure Service by Using jQuery

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name **admin**, and the password **Pa$$w0rd**.
2. Open **File Explorer**.

3. Go to **Allfiles (D):13*.
4. Double-click **OperasWebSite.sln**.
5. Enable the **Allow NuGet to download missing packages during build** option by performing the following steps:

a. On the **TOOLS** menu of the **Microsoft Visual Studio** window, click **Options**.

b. In the navigation pane of the **Options** dialog box, click **Package Manager**.

c. Under the **Package Restore** section, select the **Allow NuGet to download missing packages during build** check box, and then click **OK**.

Note: In Hyper-V Manager, start the MSL-TMG1 virtual machine if it is not already running.

Demonstration Steps

1. In the **Solution Explorer** pane of the **OperasWebSite – Microsoft Visual Studio** window, expand **OperasWebSite**, expand **Views**, expand **Home**, and then double-click **Index.cshtml**.

2. In the **Index.cshtml** code window, locate the following code.

`cs          @Html.ActionLink("operas I've seen.", "Index", "Opera") </p>` 3. Place the cursor after the located code, press Enter twice, and then type the following code.

`cs          <form>          </form>` 4. Place the cursor in the **FORM** element code block you just created, and then type the following code.

`cs          <input type="button" value="Get Latest Quote" name="GetLatestQuote" onclick="callWebService();" />` 5. Place the cursor at the end of the **INPUT** element, press Enter, and then type the following code.

`cs          <p id="quote-display"></p>` 6. In the **Index.cshtml** code window, locate the following code.

`cs          </form>` 7. Place the cursor at the end of the located code, press Enter twice, and then type the following code.

`cs          <script type="text/javascript">          </script>` 8. Place the cursor in the **SCRIPT** element code block that you just created, and then type the following code.

`cs          function callWebService()          {          }` 9. Place the cursor in the **callWebService** function code block, and then type the following code.

`cs          var serviceUrl = '@Url.Content("~/WebServices/QuotesService.asmx")';` 10. In the **callWebService** function code block, place the cursor at the end of the variable that you just created, press Enter, and then type the following code.

`cs          $.ajax({          type: "POST",          url: serviceUrl + "/LatestQuote",          data: {},          contentType: "application/json; charset=utf-8",          dataType:`

`"json",             success: OnSuccess,            error: OnError });` 11. Place the cursor at the end of the **callWebService** function code block, but within the **SCRIPT** element, press Enter twice, and then type the following code.

`cs          function OnSuccess(response)          {          }` 12. Place the cursor in the **OnSuccess** function code block, and then type the following code.

`cs          $('#quote-display').html(response.d);` >**Note: response.d** is the property you use to access JSON data from the server.

13. Place the cursor at the end of the **OnSuccess** function code block, but within the **SCRIPT** element, press Enter twice, and then type the following code.

`cs          function OnError(response)          {          }` 14. Place the cursor within the **OnError** function, and then type the following code.

`cs          $('#quote-display').html("Could not obtain the latest quote");` 15. On the **DEBUG** menu of the **OperasWebSite – Microsoft Visual Studio** window, click **Start Debugging**. 16. On the **Operas I Have Seen** page, click **Get Latest Quote**.

Note: jQuery calls the web service and displays a quote on the home page. Note that you need not reload the page to display the quote.

17. In the **Internet Explorer** window, click **Close**.
18. In the **OperasWebSite – Microsoft Visual Studio** window, click **Close**.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Version: 2017

Module 14: Implementing Web APIs in ASP.NET MVC 4 Web Applications

Lesson 1: Developing a Web API

Demonstration: How to Explore a Web API by Using Internet Explorer

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Start **File Explorer**.
3. Navigate to **Allfiles(D):14*.
4. Double-click **OperasWebSite.sln**.
5. Enable the **Allow NuGet to download missing packages during build** option, by performing the following steps:

a. On the **TOOLS** menu of the Microsoft Visual Studio window, click **Options**.

b. In the navigation pane of the **Options** dialog box, click **Package Manager**.

c. Under the **Package Restore** section, select the **Allow NuGet to download missing packages during build** check box, and then click **OK**.

Note: In Hyper-V Manager, start the MSL-TMG1 virtual machine if it is not already running.

Demonstration Steps

1. In the **Solution Explorer** pane, expand **OperasWebSite**.
2. In the **Solution Explorer** pane, under **OperasWebSite**, right-click **Controllers**, point to **Add**, and then click **Controller**.
3. In the **Controller name** box of the **Add Controller** dialog box, type **OperasApiController**, in the **Template** box, click **Empty API Controller**, and then click **Add**.
4. In the OperasApiController.cs code window, locate the following code.

`cs        using System.Web.Http;` 5. Place the cursor at the end of the located code, press Enter, and then type the following code.

`cs        using OperasWebSite.Models;` 6. Place the cursor in the **OperasApiController** class code block, press Enter, and then type the following code.

`cs        private OperasDB contextDB = new OperasDB();` 7. Place the cursor at the end of the code you just typed, press Enter twice, and then type the following code.

`cs        public IEnumerable<Opera> GetOperas()        {        }` 8. Place the cursor in the **GetOperas** action code block, and then type the following code.

cs   return contextDB.Operas.AsEnumerable(); 9. Place the cursor at the end of the **GetOperas** action code block, press Enter twice, and then type the following code.

cs   public Opera GetOperas(int id)   {   } 10. Place the cursor in the **GetOperas** action code block you just created, and then type the following code.

cs   Opera opera = contextDB.Operas.Find(id); 11. Place the cursor at the end of the code you just entered, press Enter, and then type the following code.

cs   if (opera == null)   {   throw new HttpResponseException(HttpStatusCode.NotFound);   } 12. Place the cursor at the end of the code you just entered, press Enter, and then type the following code.

cs   return opera; 13. On the **FILE** menu of the **OperasWebSite – Microsoft Visual Studio** window, click **Save All**. 14. On the **DEBUG** menu of the **OperasWebSite – Microsoft Visual Studio** window, click **Start Debugging**. 15. In the Address bar of the Internet Explorer window, type **http://localhost:<*yourPortNumber*>/api/OperasApi**, and then click **Go to**. 16. In the Navigation bar, click **Open**. 17. If the “How do you want to open this type of file (.json)?” message displays, click **More options**, and then click **Microsoft Visual Studio Version Selector**. 18. On the **EDIT** menu of the **OperasApi.json – Microsoft Visual Studio** window, point to **Find and Replace**, and then click **Quick Find**. 19. In the **Search Item** box of the **Quick Find** dialog box, type **Rigoletto**, and then click **Find Next**. 20. In the **Microsoft Visual Studio** dialog box, click **OK**. 21. In the **Quick Find** dialog box, click the **Close** button.

```
>**Note:** Visual Studio finds the JSON data for the **Rigoletto** opera.
Note that this is just one entry in the JSON data, which includes all oper
as in the web application.
```

22. In the **OperasApi.json – Microsoft Visual Studio** window, click the **Close** button.
23. In the Address bar of the Internet Explorer window, type **http://localhost:<*yourPortNumber*>/api/OperasApi/3**, and then click **Go to**.
24. In the Navigation bar, click **Open**.
25. If the “How do you want to open this type of file (.json)?” message displays, click **More options**, and then click **Microsoft Visual Studio Version Selector**.
26. In the **3.json - Microsoft Visual Studio** window, note that only the information relating to the **Nixon in China** opera is displayed.

    Note: The value for the **OperasID** parameter corresponding to the **Nixon in China** opera is **3**.

27. In the **3.json - Microsoft Visual Studio** window, click the **Close** button.
28. In the Internet Explorer window, click the **Close** button.
29. In the **OperasWebSite – Microsoft Visual Studio** window, click the **Close** button.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Module 15: Handling Requests in ASP.NET MVC 4 Web Applications

Lesson 2: Using Web Sockets

Demonstration: How to Add a Chat Room to a Web Application by using SignalR

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Start **File Explorer**.
3. Navigate to **Allfiles(D):15*.
4. Double-click **OperasWebSite.sln**.
5. Enable the **Allow NuGet to download missing packages during build** option by performing the following steps:

a. On the **TOOLS** menu of the Microsoft Visual Studio window, click **Options**.

b. In the navigation pane of the **Options** dialog box, click **Package Manager**.

c. Under the Package Restore section, select the **Allow NuGet to download missing packages during build** check box, and then click **OK**.

Note : In Hyper-V Manager, start the MSL-TMG1 virtual machine if it is not already running.

Demonstration Steps

1. In the **Solution Explorer** pane of the **OperasWebsite – Microsoft Visual Studio** window, right-click **OperasWebSite**, point to **Add**, and then click **Class**.
2. In the **Name** box of the **Add New Item – OperasWebSite** dialog box, type **ChatHub**, and then click **Add**.
3. In the ChatHub.cs code window, locate the following code.

cs `using System.Web;` 4. Place the cursor at the end of the located code, press Enter, and then type the following code.

cs `using Microsoft.AspNet.SignalR;` 5. In the ChatHub.cs code window, locate the following code.

cs public class ChatHub 6. Replace the located code with the following code.

cs public class ChatHub : Hub 7. In the **ChatHub** class code block, type the following code.

cs public void Send(string name, string message) { } 8. In the **Send** method code block, type the following code.

cs Clients.All.broadcastMessage(name, message); >**Note:** The **Send()** method sends any received message back to all the clients that are connected to the hub. You need to define the **broadcastMessage()** method in the client-side code to receive messages. The client-side code must also call the **Send()** method to broadcast messages.

9. In the **Solution Explorer** pane, expand **Views**, expand **Home**, and then click **Chat.cshtml**.
10. In the Chat.cshtml code window, within the final **<script>** element, type the following code.

cs $(function() { }); 11. Within the anonymous function you just created, type the following code.

cs var chat = $.connection.chatHub; 12. Place the cursor at the end of the variable you just created, press Enter, and then type the following code.

cs chat.client.broadcastMessage = function(name, message) { }; >**Note:** This function is the implementation of the **broadcastMessage()** function that you called in the Hub code.

13. Within the anonymous function you just created, type the following code.

cs var listItem = '<li>' + name + ': ' + message + '</li>'; 14. Place the cursor at the end of the variable you just created, press Enter, and then type the following code.

cs $('#discussion').append(listItem); 15. Place the cursor at the end of the **broadcastMessage** function code block, press Enter, and then type the following code.

cs var displayname = prompt('Enter your name:', ''); 16. Place the cursor at the end of the **displayname** variable code block you just created, press Enter, and then type the following code.

cs $('#chat-message').focus(); 17. Place the cursor at the end of the code block you just created, press Enter, and then type the following code.

cs $.connection.hub.start().done(function() { }); 18. Within the anonymous function code block you just created, type the following code.

cs $('#sendmessage').click(function() { }); 19. Within the new anonymous function code block you just created, type the following code.

```
cs          chat.server.send(displayname, $('#chat-message').val());
```

20. Place the cursor at the end of the code block you just created, press Enter, and then type the following code.

```
cs          $('#chat-message').val('').focus();
```

21. On the **DEBUG** menu of the **OperasWebSite – Microsoft Visual Studio** window, click **Start Debugging**. 22. On the **Operas I Have Seen** page, click the **Enter the Operas Chat Room** link. 23. In the **Enter your name** box of the **localhost needs some information** dialog box, type **Rebecca Laszlo** , and then click **OK**. 24. In the **Message** box of the **Operas I Have Seen** page, type a message of your choice, and then click **Send**.

Note: SignalR sends the message you typed to the hub. The hub broadcasts the message to all connected clients.

25. On the taskbar, right-click the **Internet Explorer** icon, and then click **Internet Explorer**.
26. In the Address bar of the Internet Explorer window, type **http://localhost:<** ***portnumber*** **>**,and then press Enter.
27. On the **Operas I Have Seen** page, click the **Enter the Operas Chat Room** link.
28. In the **Enter your name** box of the **localhost needs some information** dialog box, type **Elisa Graceffo**, and then click **OK**.
29. In the **Message** box of the **Operas I Have Seen** page, type a message of your choice, and then click **Send**.
30. On the taskbar, click the first instance of the Internet Explorer window. Note that the message from **Elisa Graceffo** is displayed because both users are connected to the same hub.
31. Close all the Internet Explorer windows.
32. In the **OperasWebSite – Microsoft Visual Studio** window, click the **Close** button.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Version: 2017

Module 16: Deploying ASP.NET MVC 4 Web Applications

Lesson 1: Deploying a Web Application

Demonstration: How to Create a Microsoft Azure Web App

Preparation Steps

Sign in to the **20486B-SEA-DEV11** virtual machine with the user name **admin** and the password **Pa$$w0rd**.

Note : In Hyper-V Manager, start the **MSL-TMG1** virtual machine if it is not already running.

Demonstration Steps

1. On the taskbar, click the **Internet Explorer** icon.
2. In the Address bar of the Internet Explorer window, type **https://manage.windowsazure.com**, and then press Enter.
3. If a page appears, prompting you to enter your email address, type your email address, and then click **Continue**. Wait for the **Sign In** page to appear, type your email address and password, and then click **Sign In**.

Note: During the sign-in process, if a page appears prompting you to choose from a list of previously used accounts, select the account that you previously used, and then continue to type your credentials.

4. In the left pane of the Microsoft Azure page, click **WEB APPS**.
5. In the lower-left pane of the Microsoft Azure page, click **NEW**, and then click **CUSTOM CREATE**.
6. In the **URL** box of the **Create Web App** page, type *<your username>* **operas**.
7. From the **APP SERVICE PLAN** drop-down list on the **Create Web App** page, select **Create new App Service plan**, and then in the **REGION** box, click *<A region near you>*.
8. In the **DATABASE** box of the **Create Web App** page, click **Create a new SQL database**, in the **DB CONNECTION STRING NAME** box, type **OperasDB**, and then click the **Next** button.
9. In the **NAME** box of the **Specify database settings** page, type **OperasDB**, in the **SERVER** box, click **New SQL database server**, and then, in the **SERVER LOGIN NAME** box, type *<your first name>*.
10. In the **SERVER LOGIN PASSWORD** box of the **Specify database settings** page, type **Pa**$w0rd \ast\ast, inthe \ast\ast CONFIRMPASSWORD \ast\ast box, type \ast\ast Pa$**w0rd**, and then click the **Complete** button.

Note: Microsoft Azure creates the new web app and database to support the Operas web app.

11. In the Internet Explorer window, click the **Close** button.

Lesson 2: Deploying an ASP.NET MVC 4 Web Application

Demonstration: How to Deploy a Website to Microsoft Azure

Preparation Steps

1. Sign in to the virtual machine, **20486B-SEA-DEV11**, with the user name, **admin**, and the password, **Pa$$w0rd**.
2. Start File Explorer.
3. Navigate to **Allfiles(D):16*.
4. Open the **OperasWebSite.sln** project.
5. In the Solution Explorer pane, under **OperasWebSite_Data**, delete the **aspnetdb.mdf** file. Also, delete the **aspnetdb_log.ldf** file, if present.
6. Enable the **Allow NuGet to download missing packages during build** option, by performing the following steps:

a. On the **TOOLS** menu of the Microsoft Visual Studio window, click **Options**.

b. In the navigation pane of the **Options** dialog box, click **Package Manager**.

c. Under the Package Restore section, select the **Allow NuGet to download missing packages during build** check box, and then click **OK**.

Note : In Hyper-V Manager, start the MSL-TMG1 virtual machine if it is not already running.

Demonstration Steps

1. On the taskbar, click the **Internet Explorer** icon.
2. In the Address bar of the Internet Explorer window, type **https://manage.windowsazure.com**, and then press Enter.
3. If a page appears, prompting you to enter your email address, type your email address, and then click **Continue**. Wait for the **Sign In** page to appear, type your email address and password, and then click **Sign In**.

Note: During the sign-in process, if a page appears prompting you to choose from a list of previously used accounts, select the account that you previously used, and then continue to type your credentials.

4. In the left pane of the **Microsoft Azure** page, click **WEB APPS**.
5. In the **NAME** column of the **web apps – Microsoft Azure** page, click *<your username>* **operas**.
6. On the Microsoft Azure page, click **DASHBOARD**.

Note: If the quick start page appears, repeat steps 5 and 6 before proceeding to the next step.

7. In the quick glance section of the Microsoft Azure page, click the **Download the publish profile** link.
8. In the navigation bar, click **Save**, and then click the **Close** button.
9. On the taskbar, click the **Microsoft Visual Studio** icon.
10. In the Solution Explorer pane of the OperasWebSite - Microsoft Visual Studio window, right-click **OperasWebSite**, and then click **Publish**.
11. On the **Profile** page of the Publish Web wizard, click **Import**.

12. In the **Import Publish Settings** dialog box, click *<your username>* **operas.azurewebsites.net.PublishSettings**, and then click **Open**.
13. On the **Connection** page of the Publish Web wizard, click **Validate Connection**.
14. If the **Certificate Error** dialog box appears, click **Accept**.
15. On the **Connection** page, click **Next**.
16. On the **Settings** page, click **Next**.
17. On the **Preview** page, click **Start Preview**.
18. On the **Preview** page, click **Publish**.

Note: Visual Studio publishes the website. This process can take several minutes. When the publish operation is complete, the website is displayed in the Internet Explorer window.

19. If the **Certificate Error** dialog box appears, click **Accept**.
20. In the Internet Explorer window, if the **Server Error in '/' Application** error displays, click the **Refresh** button.

Note: If the Operas I Have Seen page does not appear, you need to re-publish the OperasWebSite project.

21. On the **Operas I Have Seen** page, click **All Operas**.
22. In the Navigation bar, if the message **Intranet settings are turned off by default.** is displayed, click **Turn on Intranet settings**.
23. If the message **Are you sure you want to turn on intranet-level security settings?** appears, click **Yes**.
24. On the **Index of Operas** page, click the **Details** link corresponding to **Cosi Fan Tutte**.
25. On the **Opera: Cosi Fan Tutte** page, click the **Back to List** link.
26. On the **Index of Operas** page, click the **Details** link corresponding to **Nixon in China**.
27. In the Internet Explorer window, click the **Close** button.
28. In the **OperasWebSite - Microsoft Visual Studio** window, click the **Close** button.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.