

Vendor: Microsoft

Exam Code: 070-486

Exam Name: Developing ASP.NET MVC 4 Web Applications

Version: Demo

Topic 1, Olympic Marathon

Background

You are developing an ASP.NET MVC application in Visual Studio 2012 that will be used by Olympic marathon runners to log data about training runs.

Business Requirements

The application stores date, distance, and duration information about a user's training runs. The user can view, insert, edit, and delete records.

The application must be optimized for accessibility.

All times must be displayed in the user's local time.

Technical Requirements

Data Access:

Database access is handled by a public class named RunnerLog.DataAccess.RunnerLogDb. All data retrieval must be done by HTTP GET and all data updates must be done by HTTP POST.

Layout:

All pages in the application use a master layout file named \Views\Shared_Layout.cshtml.

Models:

The application uses the \Models\LogModel.cs model.

Views:

All views in the application use the Razor view engine.

Four views located in \Views\RunLog are named:

- CalculatePace.cshtml
- EditLog.cshtml
- GetLog.cshtml
- InsertLog.cshtml

The application also contains a \Views\Home\Index.cshtml view.

Controllers:

The application contains a \Controllers\RunLogController.cs controller.

Images:

A stopwatch.png image is located in the \Images folder.

Videos:

A map of a runner's path is available when a user views a run log. The map is implemented as an Adobe Flash application and video. The browser should display the video natively if possible, using H264, Ogg, or WebM formats, in that order. If the video cannot be displayed, then the Flash application should be used.

Security:

You have the following security requirements:

- The application is configured to use forms authentication.
- Users must be logged on to insert runner data.

Guaranteed Success with EnsurePass VCE Software & PDF File

- Users must be members of the Admin role to edit or delete runner data.
- There are no security requirements for viewing runner data.
- You need to protect the application against cross-site request forgery.
- Passwords are hashed by using the SHA1 algorithm.

RunnerLog.Providers.RunLogRoleProvider.es contains a custom role provider.

Relevant portions of the application files follow. (Line numbers are included for reference only.)

Application Structure

Controllers\RunLogController.cs

```
RC01 public class RunLogController : Controller
RC02 {
RC03
      public ActionResult GetLog()
RC04
         List<LogModel> log = RunnerLogDb.GetLogsFromDatabase();
RC05
RC06
         return View(log);
RC07
        }
RC08
RC09
      public ActionResult InsertLog()
RC10
RC11
        LogModel log = new LogModel();
RC12
         log.RunDate = DateTime.Now;
RC13
         return View(log);
RC14
        3
RC15
RC16
       [HttpPost]
RC17
      public ActionResult InsertLog(LogModel log)
RC18
       - {
RC19
         RunnerLogDb.InsertLog(log);
RC20
         return RedirectToAction("GetLog");
RC21
       }
RC22
RC23
      public ActionResult DeleteLog(int id)
RC24
       - {
RC25
         RunnerLogDb.DeleteLog(id);
RC26
          return RedirectToAction("GetLog");
RC27
        3
RC28
RC29
      public ActionResult EditLog(int id)
RC30
       - 4
         LogModel log = RunnerLogDb.GetRunnerLog(id);
RC31
RC32
          return View(log);
RC33
        3
RC34 }
```

Models\LogModel.cs

LM01	public class LogModel		
LM02	{		
LM03	[Required]		
LM04	<pre>public int Id { get; set; }</pre>		
LM05			
LM06	[Required]		
LM07	<pre>public DateTime RunDate { get; set; }</pre>		
LM08			
LM09	[Required]		
LM10	[Range (0.01, 1000.00)]		
LM11	<pre>public double Distance { get; set; }</pre>		
LM12			
LM13	[Required]		
LM14	<pre>public TimeSpan Time { get; set; }</pre>		
LM15			
LM16	public string ShortDate		
LM17	{		
LM18	get		
LM19	{		
LM20	return RunDate.ToLocalTime().ToShortDateString();		
LM21	}		
LM22	}		
LM23	}		

Views\RunLog_CalculatePace.cshtml

CP01 @model RunnerLog.Models.LogModel CP02 @(Convert.ToInt32(Model.Time.TotalMinutes / Model.Distance)) Min CP03 @(Convert.ToInt32(Model.Time.TotalSeconds % 60 / Model.Distance)) Seconds

Views\RunLog\EditLog.cshtml

```
EL01 @model RunnerLog.Models.LogModel
EL02 <h2>Edit Log Item</h2>
EL03 <script src="@Url.Content("~/Scripts/jquery.validate.min.js")"></script>
EL04 <script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"></
script>
EL05 Gusing (Html.BeginForm()) {
EL06
       @Html.AntiForgeryToken()
EL07
       @Html.ValidationSummary(true)
EL08
       <fieldset>
EL09
         <legend>LogModel</legend>
EL10
         <h3>
           Log Id: @Model.Id
EL11
EL12
         </h3>
EL13
         <div>
EL14
           @Html.LabelFor(model => model.Distance)
EL15
         </div>
EL16
         <div>
           @Html.EditorFor(model => model.Distance)
EL17
EL18
           @Html.ValidationMessageFor(model => model.Distance)
EL19
        </div>
EL20
         <div>
EL21
           @Html.LabelFor(model => model.Time)
EL22
         </div>
EL23
         <div>
           @Html.EditorFor(model => model.Time)
EL24
EL25
           @Html.ValidationMessageFor(model => model.Time)
EL26
        </div>
EL27
        <input type="submit" value="Save" />
EL28
         EL29
EL30
       </fieldset>
EL31 }
```

Views\RunLog\GetLog.cshtml

GT.01	<pre>@model List<runnerlog.models.logmodel></runnerlog.models.logmodel></pre>
	<h2>View Runs </h2>
GL05	
GL06	
GL00	
GLOS	Duration
GL00	Avg Mile Pace
1111	
	Gforeach (RunnerLog.Models.LogModel log in Model)
GL12	{
GL13	>
GL14	
GL15	<pre>@Html.DisplayFor(model => log.Id)</pre>
GL15	
GL17	
GL18	
GL19	
GL20	
GL21	<pre>@Html.DisplayFor(model => log.Distance)</pre>
GL22	
GL23	
GL24	<pre>@Html.DisplayFor(model => log.Time)</pre>
GL25	
GL26	
GL27	
GL28	
GL29	
GL30	<pre>@Html.ActionLink("Edit", "EditLog", new { id = log.Id })</pre>
GL31	
GL32	
GL33	<pre>@Html.ActionLink("Delete", "DeleteLog", new { id = log.Id })</pre>
GL34	
GL35	
GL36	
GL37	

Views\RunLog\InsertLog.cshtml

```
IL01 @model RunnerLog.Models.LogModel
IL02 <script src="@Url.Content("~/Scripts/jquery.validate.min.js")"></script>
IL03 <script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"></
script>
IL04
     @using (Html.BeginForm())
IL05 {
       @Html.ValidationSummary(true)
IL06
IL07
      <fieldset>
IL08
         <legend>LogModel</legend>
IL09
IL10
         <div>
           @Html.LabelFor(model => model.RunDate)
IL11
IL12
         </div>
IL13
         <div>
IL14
           @Html.EditorFor(model => model.RunDate)
IL15
           @Html.ValidationMessageFor(model => model.RunDate)
         </div>
IL16
IL17
         <div>
IL18
           @Html.LabelFor(model => model.Distance)
IL19
         </div>
IL20
        <div>
IL21
           @Html.EditorFor(model => model.Distance)
IL22
           @Html.ValidationMessageFor(model => model.Distance)
         </div>
IL23
IL24
         <div>
           @Html.LabelFor(model => model.Time) HH:MM:SS
TL25
IL26
         </div>
TT.27
         <div>
IL28
           @Html.EditorFor(model => model.Time)
IL29
           @Html.ValidationMessageFor(model => model.Time)
IL30
         </div>
IL31
         IL32
           <input type="submit" value="Create" />
IL33
         IL34
        </fieldset>
IL35 }
```

Views\Shared_Layout.cshtml

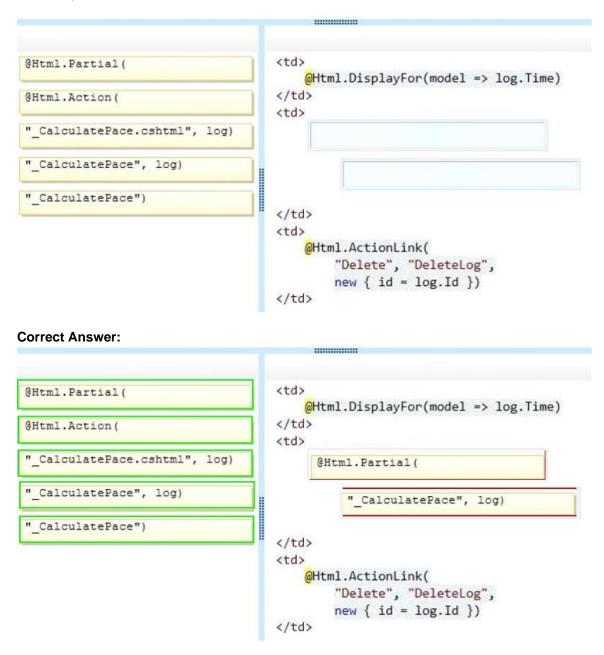
```
LOO1 <! DOCTYPE html>
L002
      <html lang="en">
LO03
     <head>
L004
        . . .
L005
      </head>
L006
      <body>
L007
       . . .
L008
       <footer>
1009
         <script type="text/javascript">
L010
L011
           var c = document.getElementById('myCanvas');
           var ctx = c.getContext('2d');
L012
           ctx.font = '30pt Calibri';
L013
L014
           ctx.strokeStyle = 'gray';
L015
           ctx.lineWidth = 3;
L016
            ctx.strokeText('London 2012', 80, 30);
L017
         </script>
L018
       </footer>
L019
      </body>
L020
      </html>
```

DRAG DROP

You need to implement the Views\RunLog_CalculatePace.cshtml partial view from Views\Runlog \GetLog.cshtml to display the runner's average mile pace.

How should you implement the view?

To answer, drag the appropriate code segments to the correct location or locations. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.



DRAG DROP

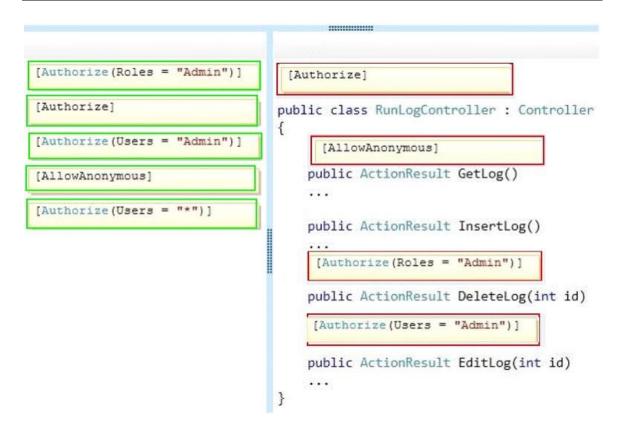
You need to implement security according to the business requirements.

How should you modify RunLogController?

To answer, drag the appropriate code segment to the correct location or locations. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

[Authorize(Roles = "Admin")]	
[Authorize]	public class RunLogController : Controller
[Authorize(Users = "Admin")]	1
[AllowAnonymous]	<pre>public ActionResult GetLog()</pre>
[Authorize(Users = "*")]	public ActionResult InsertLog()
	<pre>public ActionResult DeleteLog(int id)</pre>
	<pre>public ActionResult EditLog(int id) }</pre>

Correct Answer:



You need to make the "Distance" header of the table bold in the Views/RunLog/GetLog.cshtml view. Which code segment should you use?

- A. table>tr{ font-weight: bold; }
- B. table>th:last-child{ font-weight: bold; }
- C. table+first-child{ font-weight: bold; }
- D. table>tr>th:nth-child (2) { font-weight: bold; }

Correct Answer: D

You need to extend the edit functionality of RunLogController. Which code segment should you use?

```
C A. [HttpGet]
       [ActionName("EditLog")]
      [ValidateAntiForgeryToken]
      public ActionResult EditLog(LogModel log)
       1
         . . .
       3
C B. [HttpPost]
      [ActionName("EditLog")]
      public ActionResult EditLogValidated(LogModel log)
       ł
         ...
       3
C C. [HttpPost]
      [ActionName("EditLog")]
       [ValidateAntiForgeryToken]
      public ActionResult EditLogValidated(LogModel log)
       4
         . . .
       }
C D. [HttpPost]
      [ActionName("EditLog")]
       [RequireHttps]
      public ActionResult EditLogValidated(LogModel log)
       Ł
         . . .
       }
A. Option A
B. Option B
C. Option C
D. Option D
```

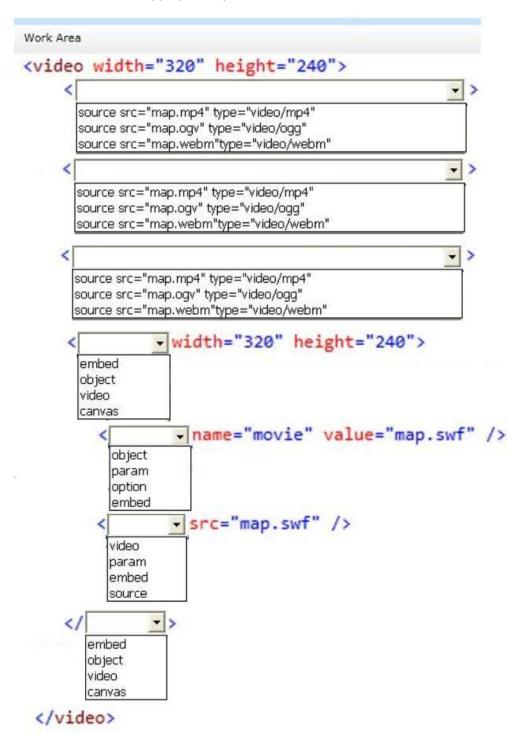
Correct Answer: C

HOTSPOT

You need to implement the map of the runners' paths.

How should you build the video viewer?

To answer, select the appropriate options in the answer area.



Correct Answer:

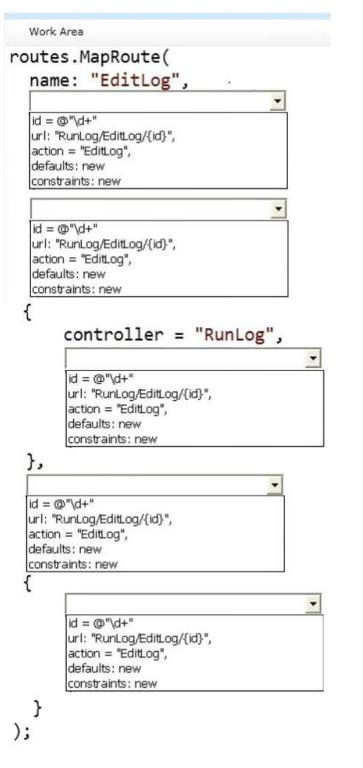


HOTSPOT

You need to ensure that only valid parameters are passed to the EditLog action.

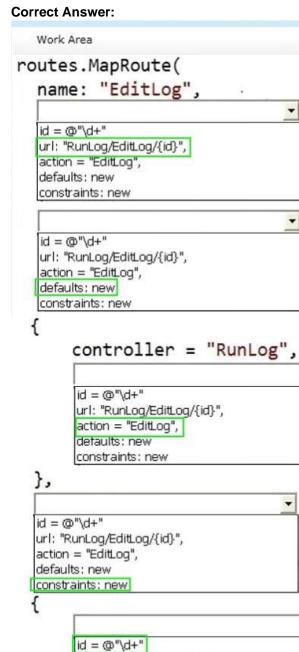
How should you build the route?

To answer, select the appropriate options in the answer area.



-

-



id = @"\d+" url: "RunLog/EditLog/{id}", action = "EditLog", defaults: new constraints: new

});

DRAG DROP

You need to ensure that the application uses RunLogRoleProvider custom role provider.

How should you modify the web.config file?

To answer, drag the appropriate line of code to the correct location or locations. Each line of code may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.



DRAG DROP

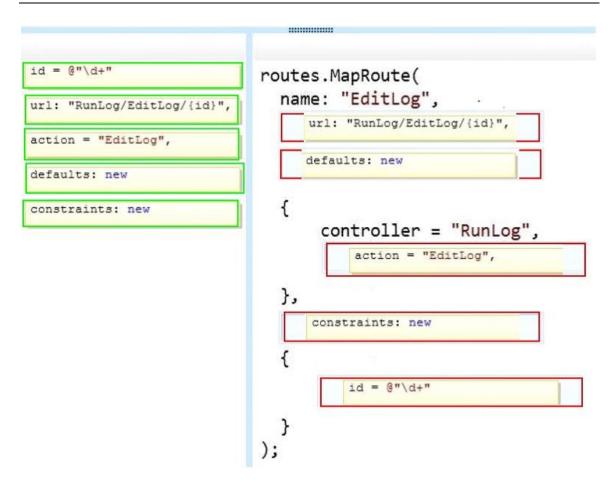
You need to ensure that only valid parameters are passed to the EditLog action.

How should you build the route?

To answer, drag the appropriate code segments to the correct location or locations. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

id = @"\d+"	
10 - 6 (0+	routes.MapRoute(
url: "RunLog/EditLog/{id}",	name: "EditLog",
action = "EditLog",	
defaults: new	
constraints: new	<pre>{ controller = "RunLog",</pre>
	},
	{
	}
);

Correct Answer:



If the canvas element is supported by the client browser, the application must display "London 2012" in the footer as text formatted by JavaScript at the end of the _Layout.cshtml file. You need to modify the layout to ensure that "London 2012" is displayed as either formatted text or as plain text, depending on what the client browser supports. Which code segment should you add?

- A. <canvas id="myFooter">
 @(Request,Browser.JavaApplets ? new HtmlString("London 2012") : null)
 </canvas>
- B. <canvas id="myFooter"London 2012</canvas>
- C. <canvas id="myCanvas">London 2012</canvas>
- D. <canvas id="myCanvas"><canvas> London 2012

Correct Answer: C

You need to add an action to RunLogController to validate the users' passwords. Which code segment should you use?

```
C A public ActionResult Login(string username, string password)
        byte[] buffer = Encoding.UTF8.GetBytes(password + username);
        byte[] hash = MD5.Create().ComputeHash(buffer);
        ComparePassword (username, hash);
        return ContextDependentView();
      }
C B. [RequireHttps]
      public ActionResult Login (string username, string password)
      -{
        byte[] buffer = Encoding.UTF8.GetBytes(password + username);
        byte[] hash = SHA1.Create().ComputeHash(buffer);
        ComparePassword (username, hash);
       return ContextDependentView();
      3
C C. public ActionResult Login(string username, string password)
        byte[] buffer = Encoding.UTF8.GetBytes(password + username);
        byte[] hash = SHA1.Create().ComputeHash(buffer);
        ComparePassword (username, hash);
        return ContextDependentView();
      }
C D. [RequireHttps]
      public ActionResult Login(string username, string password)
      -{
        byte[] buffer = Encoding.UTF8.GetBytes(password + username);
       byte[] hash = MD5.Create().ComputeHash(buffer);
       ComparePassword (username, hash);
       return ContextDependentView();
      3
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Correct Answer: B