urexam



Higher Quality

Better Service!

We offer free update service for one year Http://www.ourexam.com **Exam** : 70-517

Title: Recertification for MCSD:

SharePoint Applications

Version: DEMO

1.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.
- 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

```
public class RunLogController : Controller
RC01
RC02
RC03
        public ActionResult GetLog()
RC04
RC05
          List<LogModel> log = RunnerLogDb.GetLogsFromDatabase();
RC06
          return View(log);
RC07
RC08
RC09
       public ActionResult InsertLog()
RC10
RC11
          LogModel log = new LogModel();
          log.RunDate = DateTime.Now;
RC12
RC13
          return View(log);
RC14
        }
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
        public ActionResult EditLog(int id)
RC29
RC30
RC31
          LogModel log = RunnerLogDb.GetRunnerLog(id);
RC32
          return View(log);
RC33
RC34
     }
```

Models\LogModel.cs

```
LM01
     public class LogModel
LM02
LM03
        [Required]
LM04
        public int Id { get; set; }
LM05
LM06
        [Required]
LM07
        public DateTime RunDate { get; set; }
LM08
LM09
       [Required]
        [Range (0.01, 1000.00)]
LM10
LM11
        public double Distance { get; set; }
LM12
LM13
        [Required]
LM14
        public TimeSpan Time { get; set; }
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

```
EL-01
     @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 @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()
EL06
EL07
        @Html.ValidationSummary(true)
EL08
       <fieldset>
EL09
         <legend>LogModel</legend>
EL10
         <h3>
FT.11
            Log Id: @Model.Id
EL12
        </h3>
EL13
         <div>
EL14
            @Html.LabelFor(model => model.Distance)
EL15
        </div>
         <div>
EL16
EL17
            @Html.EditorFor(model => model.Distance)
EL18
            @Html.ValidationMessageFor(model => model.Distance)
         </div>
EL19
         <div>
EL20
EL21
            @Html.LabelFor(model => model.Time)
EL22
         </div>
EL23
         <div>
EL24
            @Html.EditorFor(model => model.Time)
            @Html.ValidationMessageFor(model => model.Time)
EL25
EL26
         </div>
EL27
         >
EL28
           <input type="submit" value="Save" />
EL29
         EL30
       </fieldset>
EL31 }
```

Views\RunLog\GetLog.cshtml

```
GL01 @model List<RunnerLog.Models.LogModel>
GL02 <h2>View Runs </h2>
GL03 
GL04
      Id 
GL05
GL06
       Date 
GL07
       Distance 
GL08
        Duration 
GL09
        Avg Mile Pace 
GL10
      @foreach (RunnerLog.Models.LogModel log in Model)
GL11
GL12
GL13
       GL14
         GL15
           @Html.DisplayFor(model => log.Id)
GL16
         GL17
         GL18
GL19
         GL20
         GL21
           @Html.DisplayFor(model => log.Distance)
GL22
         GL23
         GL24
           @Html.DisplayFor(model => log.Time)
         GL25
GL26
         GL27
GL28
         GL29
         @Html.ActionLink("Edit", "EditLog", new { id = log.Id })
GL30
GL31
         GL32
         GL33
           @Html.ActionLink("Delete", "DeleteLog", new { id = log.Id })
GL34
         GL35
        GL36
GL37
```

Views\RunLog\InsertLog.cshtml

```
IL01 @model RunnerLog.Models.LogModel
IL02 <script src="@Url.Content("~/Scripts/jquery.validate.min.js")"></script>
ILO3 <script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"></
script>
IL04 @using (Html.BeginForm())
IL05 {
IL06
       @Html.ValidationSummary(true)
IL07
       <fieldset>
IL08
         <legend>LogModel</legend>
IL09
IL10
        <div>
IL11
            GHtml.LabelFor(model => model.RunDate)
         </div>
IL12
         <div>
IL13
IL14
            @Html.EditorFor(model => model.RunDate)
IL15
           @Html.ValidationMessageFor(model => model.RunDate)
IL16
         </div>
IL17
        <div>
IL18
            @Html.LabelFor(model => model.Distance)
IL19
         </div>
IL20
        <div>
           @Html.EditorFor(model => model.Distance)
IL21
IL22
           @Html.ValidationMessageFor(model => model.Distance)
IL23
         </div>
IL24
         <div>
IL25
            @Html.LabelFor(model => model.Time) HH:MM:SS
IL26
        </div>
IL27
         <div>
            @Html.EditorFor(model => model.Time)
TT.28
TT.29
            @Html.ValidationMessageFor(model => model.Time)
IL30
        </div>
IL31
         <0>>
IL32
           <input type="submit" value="Create" />
IL33
         IL34
       </fieldset>
IL35 }
```

Views\Shared_Layout.cshtml

```
L001
     <! DOCTYPE html>
L002
       <html lang="en">
L003
     <head>
L004
        . . .
L005
     </head>
L006
     <body>
L007
L008
        <footer>
L009
L010
          <script type="text/javascript">
L011
            var c = document.getElementById('myCanvas');
L012
           var ctx = c.getContext('2d');
            ctx.font = '30pt Calibri';
L013
L014
            ctx.strokeStyle = 'gray';
            ctx.lineWidth = 3;
L015
L016
            ctx.strokeText('London 2012', 80, 30);
L017
          </script>
L018
        </footer>
L019
       </body>
LO20
       </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.)

```
.....
                               @Html.Partial(
                                  @Html.DisplayFor(model => log.Time)
                               @Html.Action (
                               " CalculatePace.cshtml", log)
" CalculatePace", log)
" CalculatePace")
                               @Html.ActionLink(
                                      "Delete", "DeleteLog",
                                      new { id = log.Id })
```

Answer:

```
.....
                              @Html.DisplayFor(model => log.Time)
                              @Html.Action (
                              " CalculatePace.cshtml", log)
                                    @Html.Partial(
                                        " CalculatePace", log)
" CalculatePace")
                              @Html.ActionLink(
                                      "Delete", "DeleteLog",
                                      new { id = log.Id })
```

2.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(Users = "Admin")]

[AllowAnonymous]

[Authorize(Users = "*")]

[Authorize(Users = "*")]

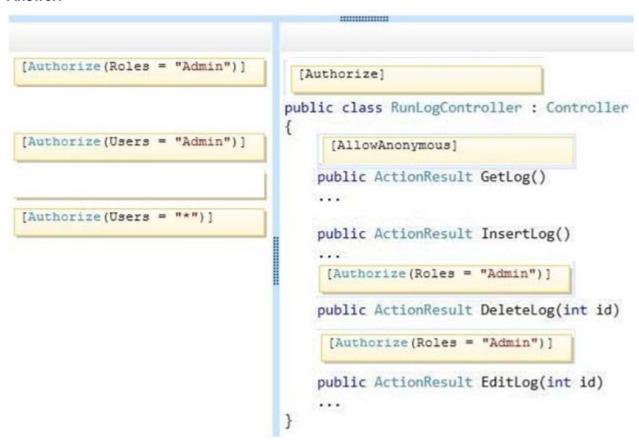
public ActionResult GetLog()
...

public ActionResult InsertLog()
...

public ActionResult DeleteLog(int id)

public ActionResult EditLog(int id)
...
}
```

Answer:



3. 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; }

Answer: D

4. You need to extend the edit functionality of RunLogController.

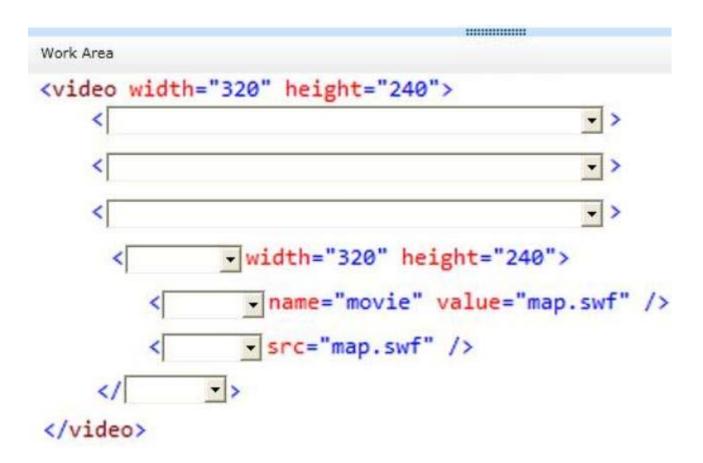
Which code segment should you use?

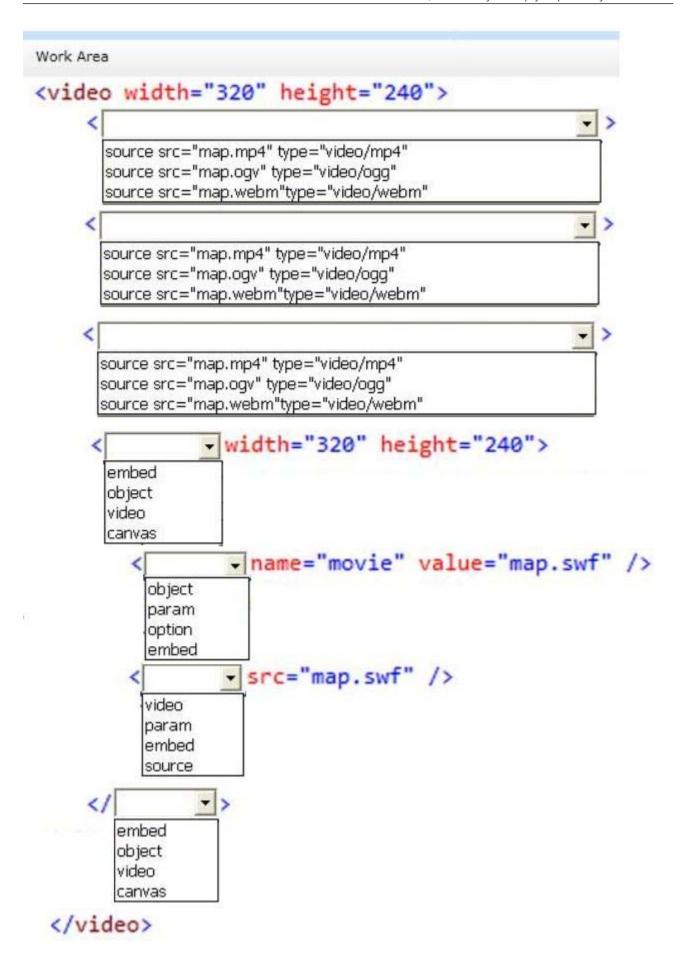
```
C A. [HttpGet]
       [ActionName ("EditLog")]
       [ValidateAntiForgeryToken]
       public ActionResult EditLog(LogModel log)
CB.
       [HttpPost]
       [ActionName ("EditLog")]
       public ActionResult EditLogValidated(LogModel log)
       {
C C. [HttpPost]
       [ActionName ("EditLog")]
       [ValidateAntiForgeryToken]
       public ActionResult EditLogValidated(LogModel log)
       }
C D. [HttpPost]
       [ActionName ("EditLog")]
       [RequireHttps]
       public ActionResult EditLogValidated(LogModel log)
       7
A. Option A
B. Option B
C. Option C
D. Option D
Answer: C
```

5.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.)





Answer:

