

TESTPASSPORT

練習問題



高品質な学習リソース

<https://www.testpassport.jp>

1年間無料アップデート

Exam : 70-486

**Title : Developing ASP.NET MVC 4
Web Applications**

Version : DEMO

1. トピック 1、オリンピックマラソン

バックグラウンド

Visual Studio 2012で ASP.NET MVC アプリケーションを開発しています。このアプリケーションは、オリンピックマラソンランナーがトレーニングの実行に関するデータを記録するために使用されます。

ビジネス要件

アプリケーションには、ユーザーのトレーニングの実行に関する日付、距離、および期間情報が格納されます。 ユーザーは、レコードの表示、挿入、編集、および削除を行うことができます。

アプリケーションは、アクセシビリティのために最適化する必要があります。

すべての時間は、ユーザーの現地時間で表示する必要があります。

技術要件

データアクセス:

データベースへのアクセスは、`RunnerLog.DataAccess.RunnerLogDb` という名前のパブリッククラスによって処理されます。

すべてのデータ検索は HTTP GET で行い、すべてのデータ更新は HTTP POST で行う必要があります。

レイアウト:

アプリケーションのすべてのページは、`\Views\Shared\_Layout.cshtml` という名前のマスターレイアウトファイルを使用します。

モデル:

アプリケーションは `\Models\LogModel.cs` モデルを使用します。

再生回数:

アプリケーションのすべてのビューは、**Razor** ビューエンジンを使用します。

`\Views\RunLog` にある4つのビューの名前は次のとおりです。

- `_CalculatePace.cshtml`
- `EditLog.cshtml`
- `GetLog.cshtml`
- `InsertLog.cshtml`

このアプリケーションには、`\Views\Home\Index.cshtml` ビューも含まれています。

コントローラ:

アプリケーションには `\Controllers\RunLogController.cs` コントローラが含まれています。

画像:

`stopwatch.png` イメージは `\Images` フォルダーにあります。

ビデオ:

ユーザーが実行ログを表示すると、ランナーのパスのマップが使用可能になります。 このマップは、Adobe Flash アプリケーションおよびビデオとして実装されています。 H264、Ogg、または WebM 形

式を使用して、可能であればブラウザーにネイティブでビデオを表示する必要があります。 ビデオを表示できない場合は、**Flash** アプリケーションを使用する必要があります。

セキュリティ:

あなたには、次のセキュリティ要件があります。

- アプリケーションがフォーム認証を使用するように設定されている。
- ユーザーはランナーデータを挿入するためにログオンする必要があります。
- ランナーデータを編集または削除するには、ユーザーが管理者ロールのメンバーである必要があります。
- ランナーデータを表示するためのセキュリティ要件はありません。
- クロスサイトリクエスト偽造からアプリケーションを保護する必要があります。
- パスワードは、**SHA1**アルゴリズムを使用してハッシュされます。

`RunnerLog.Providers.RunLogRoleProvider.cs` にはカスタムロールプロバイダが含まれています。

アプリケーションファイルの関連部分が続きます。（行番号は参照用にのみ記載されています）。

アプリケーション構造

Controllers\ RunLogController.cs

```
RC01 public class RunLogController : Controller
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;
RC12         log.RunDate = DateTime.Now;
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         RunnertogDb.DeleteLog(id);
RC26         return RedirectToAction("GetLog");
RC27     }
RC28
RC29     public ActionResult EditLog(int id)
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]
LM10     [Range (0.01, 1000.00)]
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

```
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 @using (Html.BeginForm()) {
EL06     @Html.AntiForgeryToken()
EL07     @Html.ValidationSummary(true)
EL08     <fieldset>
EL09         <legend>LogModel</legend>
EL10         <h3>
EL11             Log Id: @Model.Id
EL12         </h3>
EL13         <div>
EL14             @Html.LabelFor(model => model.Distance)
EL15         </div>
EL16         <div>
EL17             @Html.EditorFor(model => model.Distance)
EL18             @Html.ValidationMessageFor(model => model.Distance)
EL19         </div>
EL20         <div>
EL21             @Html.LabelFor(model => model.Time)
EL22         </div>
EL23         <div>
EL24             @Html.EditorFor(model => model.Time)
EL25             @Html.ValidationMessageFor(model => model.Time)
EL26         </div>
EL27         <p>
EL28             <input type="submit" value="Save" />
EL29         </p>
EL30     </fieldset>
EL31 }
```

Views\RunLog\GetLog.cshtml

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

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 {
IL06     @Html.ValidationSummary(true)
IL07     <fieldset>
IL08         <legend>LogModel</legend>
IL09
IL10         <div>
IL11             @Html.LabelFor(model => model.RunDate)
IL12         </div>
IL13         <div>
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>
IL21             @Html.EditorFor(model => model.Distance)
IL22             @Html.ValidationMessageFor(model => model.Distance)
IL23         </div>
IL24         <div>
IL25             @Html.LabelFor(model => model.Time) HH:MM:SS
IL26         </div>
IL27         <div>
IL28             @Html.EditorFor(model => model.Time)
IL29             @Html.ValidationMessageFor(model => model.Time)
IL30         </div>
IL31         <p>
IL32             <input type="submit" value="Create" />
IL33         </p>
IL34     </fieldset>
IL35 }
```

Views\Shared_Layout.cshtml

```
LO01 <!DOCTYPE html>
LO02 <html lang="en">
LO03 <head>
LO04 ...
LO05 </head>
LO06 <body>
LO07 ...
LO08 <footer>
LO09
LO10     <script type="text/javascript">
LO11         var c = document.getElementById('myCanvas');
LO12         var ctx = c.getContext('2d');
LO13         ctx.font = '30pt Calibri';
LO14         ctx.strokeStyle = 'gray';
LO15         ctx.lineWidth = 3;
LO16         ctx.strokeText('London 2012', 80, 30);
LO17     </script>
LO18 </footer>
LO19 </body>
LO20 </html>
```

ドラッグドロップ

ランナーの平均マイルペースを表示するには、Views\RunLog_CalculatePace.cshtml Views\Runlog\GetLog.cshtml の部分ビューを実装する必要があります。

どのようにビューを実装する必要がありますか？（答えを出すには、適切なコードセグメントを正しい場所にドラッグします。各コードセグメントは、1回、複数回、またはまったく使用されません。コンテンツを表示するには、分割バーをペインまたはドラッグしてドラッグする必要があります。

<pre>@Html.Partial(@Html.Action("_CalculatePace.cshtml", log) "_CalculatePace", log) "_CalculatePace")</pre>	<pre><td> @Html.DisplayFor(model => log.Time) </td> <td> <div> </div> </td> <td> @Html.ActionLink("Delete", "DeleteLog", new { id = log.Id }) </td></pre>
--	---

Answer:

<pre>@Html.Action("_CalculatePace.cshtml", log) "_CalculatePace")</pre>	<pre><td> @Html.DisplayFor(model => log.Time) </td> <td> @Html.Partial("_CalculatePace", log) </td> <td> @Html.ActionLink("Delete", "DeleteLog", new { id = log.Id }) </td></pre>
--	--

2. ドラッグドロップ

ビジネス要件に従ってセキュリティを実装する必要があります。

RunLogController をどのように変更する必要がありますか？（答えを出すには、該当するコードセグメントを正しい場所にドラッグします。各コードセグメントは、1回、複数回、またはまったく使用できません。コンテンツを表示するには、分割バーをペインまたはドラッグしてドラッグする必要があります。）

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

Answer:

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

3. Views / RunLog / GetLog.cshtml ビューで太字の表の「距離」ヘッダーを作成する必要があります。

どのコードセグメントを使用しますか？

- 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. RunLogController の編集機能を拡張する必要があります。

どのコードセグメントを使用しますか？

- ☐ A.

```
[HttpGet]
[ActionName("EditLog")]
[ValidateAntiForgeryToken]
public ActionResult EditLog(LogModel log)
{
    ...
}
```
- ☐ B.

```
[HttpPost]
[ActionName("EditLog")]
public ActionResult EditLogValidated(LogModel log)
{
    ...
}
```
- ☐ C.

```
[HttpPost]
[ActionName("EditLog")]
[ValidateAntiForgeryToken]
public ActionResult EditLogValidated(LogModel log)
{
    ...
}
```
- ☐ D.

```
[HttpPost]
[ActionName("EditLog")]
[RequireHttps]
public ActionResult EditLogValidated(LogModel log)
{
    ...
}
```

- A. オプション A
B. オプション B
C. オプション C
D. オプション D

Answer: C

5. ホットスポット

ランナーのパスのマップを実装する必要があります。

あなたはどのようにビデオビューアを構築する必要がありますか？ 回答するには、回答エリアで適切なオプションを選択します。

Work Area

```
<video width="320" height="240">  
  < >  
  < >  
  < >  
  < width="320" height="240">  
    < name="movie" value="map.swf" />  
    < src="map.swf" />  
  </ >  
</video>
```

Work Area

```
<video width="320" height="240">
```

```
< [v] >
```

```
source src="map.mp4" type="video/mp4"
source src="map.ogv" type="video/ogg"
source src="map.webm" type="video/webm"
```

```
< [v] >
```

```
source src="map.mp4" type="video/mp4"
source src="map.ogv" type="video/ogg"
source src="map.webm" type="video/webm"
```

```
< [v] >
```

```
source src="map.mp4" type="video/mp4"
source src="map.ogv" type="video/ogg"
source src="map.webm" type="video/webm"
```

```
< [v] width="320" height="240">
```

```
embed
object
video
canvas
```

```
< [v] name="movie" value="map.swf" />
```

```
object
param
option
embed
```

```
< [v] src="map.swf" />
```

```
video
param
embed
source
```

```
</ [v] >
```

```
embed
object
video
canvas
```

```
</video>
```

Answer:

Work Area

`<video width="320" height="240">``<`  `>`

source src="map.mp4" type="video/mp4"
 source src="map.ogv" type="video/ogg"
 source src="map.webm" type="video/webm"

`<` `>`

source src="map.mp4" type="video/mp4"
 source src="map.ogv" type="video/ogg"
 source src="map.webm" type="video/webm"

`<` `>`

source src="map.mp4" type="video/mp4"
 source src="map.ogv" type="video/ogg"
 source src="map.webm" type="video/webm"

`<` `width="320" height="240">`

embed
 object
 video
 canvas

`<` `name="movie" value="map.swf" />`

object
 param
 option
 embed

`<` `src="map.swf" />`

video
 param
 embed
 source

`</` `>`

embed
 object
 video
 canvas

`</video>`

