Lab 3 – Razor and Request/Session

# Description

This lab is designed to give you a basic understanding of the Razor Rendering Engine and an overview of MVC’s routing feature.

# Estimated Time

This lab will take an estimated 4 hours to complete

# Deliverable

Deploy your website to Windows Azure and submit the link to Brightspace.

# Notes

* Be sure Visual Studio is up to date.
* Follow along closely to the instructions!
* When using the <input> tag be sure to set the type to collect the proper values
* Use the example code ‘IntroductionToASP.NETMVCCore’ for help and tips.
* Demo can be found at: <http://cst8359.hopto.org/lab3>

# Step 1: Create a new MVC Core project called ‘Lab3’

1. Open Visual Studio 2015
2. Click: File -> New -> Project
3. Click: Templates -> Visual C# -> Web, select ‘ASP.NET Core Web Application (.NET Core)’
4. Name the application ‘Lab3’, save the project in your desired location and click the button ‘OK’
5. Select the ‘Empty’ ASP.NET Core Template
6. Be sure to uncheck ‘Host in the cloud’
7. Click the ‘OK’ button.

# Step 2: Configure your new Web Application

1. Modify the file ‘project.json’. Replace the section:

"dependencies": {

"Microsoft.NETCore.App": {

"version": "1.0.1",

"type": "platform"

},

"Microsoft.AspNetCore.Diagnostics": "1.0.0",

"Microsoft.AspNetCore.Server.IISIntegration": "1.0.0",

"Microsoft.AspNetCore.Server.Kestrel": "1.0.1",

"Microsoft.Extensions.Logging.Console": "1.0.0"

},

with:

"dependencies": {

"Microsoft.NETCore.App": {

"version": "1.0.1",

"type": "platform"

},

"Microsoft.AspNetCore.Diagnostics": "1.0.0",

"Microsoft.AspNetCore.Server.IISIntegration": "1.0.0",

"Microsoft.AspNetCore.Server.Kestrel": "1.0.1",

"Microsoft.AspNetCore.Mvc": "1.0.1",

"Microsoft.AspNetCore.Session": "1.0.0",

"Microsoft.AspNetCore.StaticFiles": "1.0.0",

"Microsoft.Extensions.Logging.Console": "1.0.0",

"Microsoft.Extensions.Caching.Memory": "1.0.0"

},

1. Visual Studio should now update itself with the packages you need for this application to run.
2. Modify Startup.cs

Add the following lines to the method ‘ConfigureServices(IServiceCollection services)’”

services.AddMvc();

services.AddMemoryCache();

services.AddSession();

Replace the contents of ‘Configure(IApplicationBuilder app, IHostingEnvironment env, ILoggerFactory loggerFactory)’ with:

loggerFactory.AddConsole();

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

app.UseStaticFiles();

app.UseSession();

app.UseMvc(routes =>

{

routes.MapRoute(

name: "default",

template: "{controller=Home}/{action=Index}/{id?}");

});

# Step 3: Create the ‘Controllers’ and ‘Views’

1. Create a folder in your project called ‘Controllers’
2. Create a new Controller in this folder called ‘Home’
   1. NOTE: if you use ‘MVC Controller Class’ template from visual studio it will create all of the basic code you need to have a valid controller
   2. To do so right click on the ‘Controllers’ folder, click ‘Add’, click ‘New Item’ and select the ‘MVC Controller Class’
3. Create a folder in your project called ‘Views’
4. Add the root of this new ‘Views’ folder create a filed called ‘\_ViewImports.cshtml’
5. Add the following lines of code to the file ‘\_ViewImports.cshtml’

@using Lab3

@addTagHelper "\*, Microsoft.AspNetCore.Mvc.TagHelpers"

# Step 4: You’re on your own (Part 1)

1. Create a View in your home controller called ‘Razor.cshtml’
2. Use the Razor Rendering Engine to draw the following grid. Use nested for loops and corresponding if-else statements (if needed)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |

1. After the grid, use Razor to draw the lyrics to the children’s song ’99 Bottles of Beer’
   1. Each time the number of beer is divisible by 20 write ‘WOOO!’
   2. If the number of beer falls below 50 write ‘awwww :(’

# Step 5: You’re on your own (Part 2)

1. Create a few called ‘CreatePerson.cshtml’
2. Collect the following information from the user
   1. First Name
   2. Last Name
   3. Age
   4. Email Address
   5. Date of Birth
   6. Password
   7. Description of person
3. Once collected, direct the user to the ‘DisplayPerson.cshtml’. You can either store the information in the Session to be retrieved later or you can access the Request to display the results.
4. Display the results collected on this new view.