W3D2 - Into to MVC (part 1).

Hello World Again.

1. Create a new project. (ASP.Net Core app)
   1. Call it MVCStuff, and select “web Application (Model View Controller)”
2. Right click the controller folder the add, and “new item”
   1. Select MVC controller and call it “HelloWorldController.cs”
3. Delete the internals only function in the file, then add 2 functions that return a “public string” called “Index” and “welcome”
4. Return a string in each of the functions you created.
5. Start the project then navigate to “[..]/helloworld” and “[…]/helloworld/welcome”
   1. What do you see?
   2. Something about navigating to these pages and them working. What do you think about this?
   3. Was this easier then with Razor pages.
6. Make welcome private, that is change “public” to “private” save and reload the welcome page.
   1. If you got a 404, then this the correct action.
7. Add welcome to the “index” function see what happens. (you will need a + to concatenate it onto your string)
   1. Navigate to the index page, (remove welcome from the url)
   2. What does this page look like now?
   3. Why do you think that you can access that code via the index, but not directly?

Views with controllers.

1. Go back to the hello world controller and replace the “index” with one that returns an “IActionResult” using the function “View ()”
   1. Controllers typically have methods that return IActionsResults rather than strings.
2. In the Solution folder add a new folder to “view” and call it “HelloWorld”
3. Add a new MVC View Page to this new folder called index.cshtml.
   1. Hint do this by add-> new item then selecting MVC View Page.
4. Add some HTML to it such that it looks like a somewhat presentable “hello world.”
5. Start the page.
6. Now in that section of code at the top, (the `@{ \n }`)that we should have left along, add this code `ViewData[“title”] = “something”` where something is something creative or descriptive.

Messing with the Layout

1. Open \_layout.cshtml and find the “renderbody()” function call. This is a place holder. What is does is it when the page loads it will apply this file then render your view at that location In the file.
2. Change the copyright to your name. you can find that directly under the “RenderBody” function call.
3. Lets also add a link to students.
   1. To do this find the header part, this will be above the “renderbody” function call.
   2. In the <div> that’s class is ”Navbar-header”, duplicate <a> element contained within.
   3. Name change the controller to “Student” and then rename the link to “StudentDB.”
4. Reload the page and see your changes.

Passing data between a View and a Controller.

1. Change the “Welcome” function back to public
2. Add 2 variables to pass it, call them “name” and “numTimes”
3. Set the default “numTimes” to 5.
   1. Hint: (you can default it by adding an “=” sign in the function call.)
4. Set the ViewData index named “message” to a message that welcomes the “name” to the page.
5. Set the ViewData index named, “numTimes” to “numTimes”
6. Then create an MVC web Page that is titled “welcome”
7. Create a fore loop that iterates through and add a <ul> element containing the “message” for “numTimes’
8. Save and load the page, passing it the variable information. i.e. load “/helloworld/welcome?name=Galigus&numtimes=1”
9. Now that we have done that, try passing the page 100, then 10k, then try 2147483647 (A.k.a. max\_int)
   1. what happened each time?
   2. What error did you get on the last one?
   3. Open task manager, what is your memory like?
   4. What about CPU?
   5. Kill the “dotnet” process under “visual studio” <- you will have to do this.
10. What limitations does typing in a number that is TOO BIG bring?
11. How would you fix it?
    1. Hint: you can use “if” statements and “switches” to achieve this.
12. Go test your fix.