Documentation for Senior Design: Web Lecture Notes

This report is to help anyone who works on this in the future figure out what’s going on, so they can work off the existing code more efficiently. It will be much easier to work with this for students who have taken Modern Software Development (CSCI 213). This is a required class for CSCI minors, so I’d advise getting as many CSCI minors as possible involved in this project.

In order to run and edit code, you’ll need VisualStudio Community. This is free and easy-to-use. Instructions for installing it below:

* VisualStudio Community can be downloaded here: <https://visualstudio.microsoft.com/vs/community/>

Then, you’ll need to install a workload called “ASP.NET and web development.” Here’s how you can do this:

* File > New > Project > Open Visual Studio Installer
* Check the box with “ASP.NET and web development”
* Click the “Modify” button

After having VisualStudio Community with the ASP.NET workspace, you can open the project either from VisualStudio or automatically via the .sln file. Instructions below for how to do this:

* Open the design2 folder and double click on “design2.sln.”

OR, from VisualStudio already opened

* File > Open > Project/Solution > design2 folder > design2.sln

Now, for running the code:

* There is a run button at the top of VisualStudio (has a picture of a green arrow pointing right)
* If you want to choose the web browser you want to run it on, click the arrow on the right of the button and change the selected web browser. In my experience, Firefox is faster for testing than Chrome.

Notes about the coded website

* The actual note pages aren’t the important part. Dr. Glower can use his own PDFs for that. However, additional notes or back ups are okay, so writing can be helpful.
* In regards to code, focus on example problems and practice quizzes.
* When you go under the “Notes,” “Quiz,” and “Examples” folders, you’ll find the existing classes.
  + When you do a single click on each class, it’ll take you to a designer page for it. It’s much easier to edit the design using this than the .html.
  + You can access the .html page by clicking “Source” at the bottom.
  + If you click the arrow under the class in Solution Explorer, it’ll show you the .cs for the class and the designer. The aspx.cs is the background c# code for each page. Go to this page to change how buttons, textboxes, etc. all work.

Example Problems

* The example problem sets are all in a folder titled “Examples.”
* Since Dr. Glower already has a lot of his own worked out examples, I tried to focus on interactive examples with buttons and text boxes instead (something we didn’t have yet).
* In the C# code, to access the text in a text box use *textboxname*.Text.
* To change what happens when a button is pressed, double click the button and it should automatically take you to the function for it.

Quizzes

* The quizzes are all in a folder titled “Quiz.”
* IMPORTANT: These also use classes from the folder titled “Classes.”
* The C# classes from the Classes folder are important for calculating answers and saving randomly generated values. Without these classes, these values can be either reset or lost upon page refreshes.
* Regarding randomly generated values in the questions: **this makes the coding a lot harder.** It’s a really good feature, but it’s likely not worthwhile. It requires having ranges for answers, deciding how many decimal places the correct answer should have, and code that’s at least 100 lines longer per page to support it.

Different CSS and modes (eg: Light Mode/Dark Mode)

* Include this piece of code on every page that has the option

public void Page\_PreInit()

{

if (Styles.DarkModeIsOff) this.Theme = "DarkMode";

else this.Theme = "LightMode";

}

* To include CSS on pages without darkmode, you still want a theme, so use this:

public void Page\_PreInit()

{

this.Theme = "LightMode";

}

* Dark Mode may cause problems on some pages. It’s easier to disable it than fix it sometimes. Otherwise, the fixes usually require setting textboxes, buttons, etc. to a class with attributes defined in the .CSS.

Hopefully, the comments on the apex.cs pages are good enough to follow along. The comments on the first quiz and examples page are probably the most clear and detailed.