Midterm Part 2

Level 0.

On your computer, in the CSC209 folder, make a new folder MidtermPart2 (do NOT uploaded to github). In it, cre Level1, Level2 etc. as you start working on each level of Part 2 of the midterm. Each such folder should have in it the full code needed to solve the problem, specified for that level. The goal is to create a comprehensive web site (let's refer to it as the **Midterm Portfolio**) that *integrates* previous homeworks (modified appropriately) and combines them with new features from w3schools.

For each level, do the best you can. In the *home.html* file of the web site (created at Level 3 and carried on to the upper levels), add a sentence or two to describe the status of the level, in case you could not finish it, and then move on to the next level.

If you find the strict sequence of levels too slow for your taste, and if you want and can implement several levels at once, you are allowed to jump directly to a higher level. Be advised, though, that:

1. **Your grade will reflect the highest level that works.** In other words, if you jump from Level 3 to Level 8, but Level 8 is not done correctly, your grade will reflect Level 3, not Level 8 (and not even Level 4).
2. It is OK to use features that we did not cover in class (e.g. things that you figured out on your own from W3schools tutorials or other on-line resources, or that you may have already known). However, in such a case, you may be asked by the professor to describe the logic and techiques used in your code, in a one-on-one zoom or in-person meeting, prior to being assigned a grade for the exam.

Level 1

From W3schools, HowTo: read and try three examples:

1. Example showing how to make an Icon Bar. Copy the code in a file *iconBar.html* in the Level1 folder.
2. Example showing how to make a collection of sections with **collapsible content**, called an Accordion. You can work either with the first example on this page, or with the example at the bottom of the page (animated accordion): they differ only in the css. Copy the code in a file *accordion.html* in the Level1 folder.
3. Example showing how to make Tabs. Copy the code in a file *tabs.html* in the Level1 folder.

Check that you can see all these examples on your computer, while running MAMP.
NOTE: *From this level on, I will not repeat this last step and assume that you have checked that each submitted level shows up correctly in a browser.*

Level 2

Copy the three html files from Level 1 into Level 2. For each file, extract its css into a separate file. They will be have the same name as the html file, but with a css file extension. For instance, the css file for *accordion.html* will be named *accordion.css*, etc. These three files should be located in a folder named css in Level2.
NOTE: *From this level on, make sure you have all the necessary css style files in this separate folder.*

Level 3

Copy the Level2 folder, rename it as Level3 and modify the files as described below.
NOTE:*From this level on, I will not repeat these instructions and assume that you have copied a previously finished level in which you will work out the required modifications for each specific level.*

At this level you will work primarily with the IconBar page. *The goal is to demonstrate that you can read and understand an example from w3schools, and modify it based on some independent explorations using resources available on the internet.*

1. In the level folder, create an "entry point" html page, name it home.html (not index.html).
2. Create a folder named **Pages**. Copy in it all the html pages you currently have in your level folder, making sure you update all the paths to other files (such as css files).
3. In the iconBar.html file, add a link to the home.html page for the Home icon.
4. On the home.html page, add a sentence or two explaining (the best you can) the role of the `link` tag in the head of the iconBar.html file, what kind of information you think it contributes to the html document and which part of the html document might be using information from the externally linked file.
5. Open the externally linked file from the previous step and do a search in it for the names of the five icons: if you figured out where they are reflected in the iconBar.html file, then you will find them (just to convince yourself that they come from this external file). Then, scan this external file for some other possible icons that might be useful. I suggest: fa-photo, fa-movie, fa-download, fa-print. Then, edit the iconBar.html file and delete the icons for email and globe and add these new icons (for photos, movies, download and print) to the icon bar.
6. Notice the change in the browser: the bar extends on a second row. Your task is to make the appropriate modification in the css file, so that all these icons show on one row.
7. In the Pages folder, create dummy html files (with suggestive names) for all the icons, so that when you click on an icon, it takes you to a separate file.

Level 4

Extend the previous level so that when you click on the Photo icon it will take you to the slide show from your Hw4, properly integrated, as described below. *The goal is to demonstrate that you know how to work with relative file paths. If you have difficulty with some of the paths, mention the problem in the home.html file and move on to the next step.*

1. Copy the html of your Slide Show homework (Hw4) into the Pages folder, rename it photos.html.
2. Copy its css file into the css folder, name it photos.css and modify its path in the photos.html file.
3. Copy the pictures you used in the slide show into a folder named **images**, located directly under the current Level folder (i.e. where the css and Pages folders reside). Modify the photos.html file to reflect the corresponding paths to the images.
4. Similarly, create a folder **js** for the javascript code used by the slide show, rename it as photos.js (to distinguish from future javascript files that will be placed in this folder) and update the path to the photos.js file in the photos.html file.

Level 5

Integrate one of the animations created in previous labs and homework and link it from the *Movies* icon. Compared to the previous level, here you will have fewer file paths to modify. *The goal is to demonstrate that you can integrate previous code into a required organization, with separate css and js folders. If you have difficulties, mention the problem in the home.html file and move on to the next step.*

Level 6

At this level, you start working with the new accordion.html file from w3schools. The first task is to understand what it is doing and isolate the javascript code present in this file. *The goal is to demonstrate that you understand how to define a function in javascript. If you have difficulties, mention the problem in the home.html file and move on to the next step.*

1. Identify the part of the existing javascript code that starts with **function**. This is an example of an anonymous function in javascript (you can read about this aspect in w3schools functions).
2. Copy the anonymous function (making sure that the parentheses close properly) and define an actual named function, calling it, say, *toggleFunction*, because it is used to toggle the text in each section of the accordion.
3. Replace the entire definition of the anonymous function with the name of the (now named) function, i.e. with *toggleFunction*. Notice that you do not have to call it, i.e. you do not follow it with the parentheses that indicate calling a function.

Level 7

Extend the previous level so that the javascript *toggleFunction* function is placed in an external file (call it *accordion.js*), placed in the **js** folder, and which is imported into the html using a script tag. *The goal is to demonstrate that you understand how to use external javascript files and use proper paths to locate them. If you have difficulties, mention the problem in the home.html file and move on to the next step.*

Level 8

1. Link the "download" icon from iconBar.html to the highest-level "accordion" html that works for you (either from Level 1, 6 or 7).
2. Replace the dummy text in each collapsible section of the accordion with the text: "Download image 1" (resp. 2, 3, etc. depending on how many images you have in your slide show).
3. In each collapsible section, include a link to the corresponding image from your slideshow (not the image itself).

This level should be straightforward.

Level 9: A- level

Extend the previous "accordion" page with javascript code that creates dynamically any number of collapsible sections (rather than having a fixed number, which is 3 in the example from w3schools). The number of sections is given in a global javascript variable NRIMAGES, and the sections are created by javascript code in a for loop. *This part is very similar to the extension you did in Hw4 for the slide show. It is also similar in spirit to the extension you did in Part 1 of the Midterm, for the table.* Clean up and, if necessary, fix the organization of the code, remove debugging statements, make sure you have properly named variables and function names and all the code is readable and tidy.

Level 10: A level

For top credit, I leave it to you to figure out the Tabs from w3schools and integrate it in some *creative* manner in this midterm portfolio. For instance, you could do something similar to Level 9 but with tabs instead of accordion, and have it linked from the Print or some other suitable icon in the icon bar.