**IMS 322 Team Project**

**Milestone#3 (80pts) Dynamic Web Page Using JQuery & JQuery Mobile**

**Due Date: May 1**

Milestone#3 is to enhance your Milestone#2 pages using JQuery, make it production-ready, and maybe even publish it on the web if you do the bonus part. The detailed requirements for the site are as follows.

**Detailed Requirements:**

# jQuery (40 pts)

* **FAQs**: Replace the current javaScript code of FAQs with Accordion jQuery widget. Create a custom theme using a Theme Roller and use it for your Accordion widget.
* **Image Swap (or Slide Show)**: Replace the current javaScript code for Image Swap (or Slide Show) with jQuery code. While replacing the code, you must add one effect or animation to either Image Swap (or Slide Show) page.
* **Form Validation**: Add (or replace) the current javaScript code of Contact page with jQuery form validation plug-in.

**Make Responsive & Mobile-Friendly (20pts)**

* Make your entire site responsive on various screen sizes. Please follow the guidelines in Module 3: Responsive Web. One of the last slides in the powerpoint gives tips for effective mobile pages.
* It doesn’t need to be perfect, but your website pages should look good on a mobile-size, tablet-size, and desktop-size devices without any overflows.
* You don’t need to create separate pages for this, I’d recommend using CSS media queries to adjust the styles as needed depending on the screen size.
* I will be checking the page designs at 375px wide, 768px wide, and 1280px wide. These are the standard screen sizes used in industry (mobile, tablet, and desktop respectively).
* The pages have to include the same content on all screen sizes.
* You can preview how things look on different screen sizes using Chrome dev tools and clicking the phone icon on the top-left to toggle between the different sizes.
* Points will be allocated for use of fluid images, media queries, %-based widths, and overall page visuals.

**Code Structure (10 pts):**

Ideally the following is done by one person when all the previous steps are done and the code for them is committed and pushed to the repository. The person structuring the code will run git pull to pull everyone’s code changes and then restructure the code as follows:

* The code for production websites often follow the same structure. Structure your code as follows:
* Your homepage should be at the base level of your repository (not in any folders) and named index.html
* Your other pages should also be at the base level of your repository and use camelCase for their name. For example, your contact page could be contact.html or contactUs.html
* Your images should be in a folder called images.
* Your css should be all in one file called style.css or you can have a separate css file for each page in which case you’d name them pageName.css
* We are now going to format your code so that it’s more easily readable.
  + Install the “Prettier - Code Formatter” extension via the Extensions sidebar icon in Codespaces.
  + Add a file named settings.json in a folder called .vscode. This file should have the following code in it:

{

"editor.formatOnSave": true,

"editor.defaultFormatter": "esbenp.prettier-vscode",

"[javascript]": {

"editor.defaultFormatter": "esbenp.prettier-vscode"

}

}

* + Open each of the files in your codebase (excluding image files) and manually save them (using CMND+S on Mac and CTRL+S on Windows). You’ll notice the spacing and alignment changes a bit. Make sure you commit and push these code changes.

**Analyze Website Accessibility & Performance (10 pts):**

* Lighthouse is a tool that you can use to audit a webpage and get automated suggestions on how to improve it in the areas of accessibility, performance, and more.
* You are going to run two Lighthouse audits for your website homepage (one for Mobile and one for Desktop) on your site and upload the results to Canvas. Following [this guide](https://developer.chrome.com/docs/lighthouse/overview/), run an audit in Chrome DevTools for all categories and toggle Mobile and Desktop as needed.
* If you’re having trouble with running Lighthouse through DevTools, you may install the Lighthouse Chrome Extension and run the audit that way.
* Save the two report files as HTML files and upload them to your repository in a folder called reports. Name them as follows: mobile.html and desktop.html.

**Deploy Website Bonus (5 pts):**

* GitHub Pages allows you to easily publish your website so anyone in the world can see it. You may publish your website and have a live link of your work to share on a portfolio.
* Note: only one team member needs to do this.
* To do so, go to your project repository’s homepage, click the “Settings” tab, click “Pages” in the sidebar, under “Branch change the “None” dropdown to “main”, don’t change “/ (root)” and click save.
* If you go back to your home page, you’ll see a yellow dot below the gray "Go to File” dropdown. This means your site is building. When this changes to a green checkmark, your site has built successfully.
* Go back to “Settings” -> “Pages” and you will see a link to your live website and a “Visit Site” button. Feel free to visit your site and share it with the world!
* Copy and add this link to your repository homepage under About -> Website so that I can see that you completed this bonus part.
* Note: some of the links on your site or in your nav menu may now be broken. You can fix these by removing the forward slash at the beginning of the href paths. For instance, a link to the contact page should be <a href=“contact.html” /> and not <a href=“/contact.html” />
* Optional: By default, your project repository is private. You may also make your repository public if you would like to share the code on a portfolio for others to see. Do this under Settings -> General -> Change repository visibility -> Change to public. This will make it public for everyone.

**SUBMISSION GUIDELINE**

Before deadline, everyone should commit and push their file changes and a project team leader should upload the repository link to the Canvas-Assignments-Team Project – Milestone#3.