Web Development 2

Project 01: Build a Responsive Web Page from a Pre-Designed Mock-Up

Course Value: 60%

Instructions:

1.) You will be randomly assigned to a group of 3-4 classmates
    a. Working in your assigned group is mandatory for this project.
    b. You can NOT work on your own for this project.
2.) With your teammates, select one of the pre-designed mock-ups located in the mock-ups folder
3.) Create a fully working responsive web page using HTML/CSS (Sass) and JavaScript that matches as closely as possible to the design of the mock-up that was selected in step 1
4.) For this project you must write your CSS using Sass
5.) Create a GitHub repository for this project for you and your teammates
6.) All your code should be version controlled and stored in your group's repository on GitHub

***** IMPORTANT ***:** Follow the project Setup instructions (listed below) as closely as possible for developing this project. When you have completed the project follow the project submission instructions (listed below) to build your project for submission. Failure to read and follow these instructions will lead to development delays, project submission problems, and loss of marks for not following instructions.

Important Project Notes:

- All team members will receive the same mark
- All your code and assets (images, fonts) should be stored on GitHub
- The web site should have a working mobile menu system
- ***IMPORTANT*** The web site should be coded using a mobile-first methodology
    - Small mobile screen layout should be the default, then use "min-width" media queries for larger screen layouts)
- Inside the individual mock-up folders there are animated gifs and/or videos which show how the mobile menu should function.
- Inside the individual mock-up folders there are animated gifs and/or videos which show how the web site re-flows from mobile to desktop. Use these animated gifs and/or videos as a reference.
- Groups that build responsive menus that closely match the responsive menu systems shown in the gifs and/or videos will score higher
- The more closely your web site matches the supplied mock-ups the higher your mark
- Inside some of the mock-ups folders for the individual sites you will also find some text copy that is to be used on the web site. For Lorem Ipsum text content, any lorem ipsum text will do.
- Some of the mock-up folders contain additional layout information, such as information on any required slideshows. Give this information a read to maximize your mark.
- The individual mock-up folders also contain a fonts folder with information on the fonts used on each mock-up

- Each individual mock-up folder contains an images folder with individual images that can be used in the finished web site. Some images have different size versions, use responsive image techniques for these images

Important Project Files

- Notice that the initial project folder includes five important files which should be left in the root directory of your project. Do not move these files. They can be left as they are
    - README.md
        - This file is used to describe the project or web site
        - This file is written in a language called Markdown
    - .gitignore
        - **Note:** If on a Mac this file may be hidden and will not show up in a finder window. To see this file in finder, open the folder and press "Cmd+Shift+.". Press "Cmd+Shift+." to re-hide the files
        - This file is used to tell Git to ignore certain files and folders
        - If you open up this file (it's a plain text file and can be opened in your text editor of choice) you may notice that this file lists the "dev/styles/*.css" as an ignored file. This is by design as we do not want to version control our development CSS as this will cause merge conflicts.
        - This file is written in plain text
    - gulpfile.js
        - This file contains the task running code that will automate the testing and development of your project. All the tasks have been created for you. There is no need to edit this file
    - package.json
        - This is the file that contains information about your project. It also contains a listing of all the npm dev dependency files that are used by Gulp.
    - package-lock.json
        - This file is used to lock the versions of the npm dependency programs to certain versions. This is helpful when sharing your project with multiple developers over a longer period of time as it helps prevent new team members from inadvertently installing newer and possibly incompatible versions of npm dev dependency programs.

Project Folder Structure

- In addition to the four files listed above, every project repository will have two additional folders:
  - "dev" folder (short for "development" folder)
    - This is where you will write your HTML, Sass and JavaScript. Gulp will watch this folder for any changes.
    - Most of your work will be done in the "dev" folder
  - "dist" folder (short for "distribution" folder)
    - This is the directory where you will build your final version of the project.
    - The final build is generated automatically by running the "npx gulp build" task in the command line. Gulp will handle compressing all the files and moving all the files to this directory for you. All you need to do is run the "gulp build" command at the command line and Gulp will handle the rest.

node_modules Folder

- When you install your node dev dependencies on your initial project install (instructions below) you will notice the creation of a node_modules folder. This folder can be left alone. The folder is not version controlled, so don't worry if you do not see it on GitHub. This is by design as the node_modules folder can easily be recreated by running "npm install" from the command line

Project Setup

- Follow the below steps to get your project up and running. Failure to follow these steps will result in your project not working correctly
  a. One member of your team should create an empty GitHub repository on github.com and add all your teammates as collaborators.
  b. Clone the GitHub repo created in step (a) to your local machine.
     i. *** Do NOT clone into a Dropbox, Google Drive, BCIT ShareFile or One Drive file syncing folder. These file syncing services cause problems with version control systems. You will be syncing your project up to the cloud on GitHub's servers, so your project is safe if your computer breaks down or something else happens to your local data.
  c. One team member should do the following:
     i. Open your cloned repo from step (b) and the "html-project-master" folder on your computer and copy all the contents of the "html-project-master-folder" into your cloned repo folder
        1. *** IMPORTANT *** Make sure you copy the hidden ".gitignore" file. You must make sure that you have hidden files set to visible in your operating system in order to copy the hidden ".gitignore" file to your repo folder.
           a. macOS users:

            i. Open the "html-project-master" folder in Finder and press "CMD+SHIFT+." (Command Key + Shift Key + the period character)

        b. Window users:

            i. Open the "html-project-master" folder in File Explorer and select the "View" tab and check the "Hidden items"

    d. The same team member who performed step (c) should do the following

        i. Commit the newly added files and push the new files up to your repo on github.com

    e. The other team members who did NOT perform step (c) and step (d) should do the following:

        i. Clone your team's repo from github.com

    f. Make sure NodeJS is installed on your system. If it is not, head over to http://nodejs.org and follow the install instructions

    g. Launch the Node.js Command Prompt (Windows) or Terminal (Mac)

    h. From the command prompt or terminal CD (change directory) into the directory where you cloned your team's repo

    i. From the command line run the command "npm install" (don't include the quotes). This step may take a minute or two as node will download all the development dependencies from npm that are required for this project

    j. Type "npx gulp" at the command line. This command will fire up Browsersync for auto refreshing in the browser. It will also fire up a watch command for your Sass files so that your Sass files will auto compile to CSS on save

        i. Do NOT run a "sass" command in the terminal. The Gulp file will handle the compiling of Sass for you

        ii. The "x" in "npx" is NOT a typo…it is "npx"

    k. Open your project folder in your text editor of choice and start editing the HTML, Sass and JavaScript files. If your command prompt or terminal window is open, Gulp will compile all your files and refresh the browser on save automatically.

- Remember to add and commit your changes to Git and push up your changes to GitHub as you make changes to your code

Special Notes on Editing the index.html file

- The “index.html” file has been specially setup to work correctly with this development environment. To avoid any issues avoid editing the following lines:
    - Lines 14 (the link to the CSS file)
        - This line includes the “link” tag that attaches the compiled CSS file
    - Line 23 (optional)
        - Leave this line alone if you wish to use jQuery
        - If you do not want to use jQuery, then you can delete this line
            - For ultimate optimization you can make a simple edit to the Gulp file to save Gulp from copying over the jQuery library to the “dist” folder
            - To stop Gulp from copying the jQuery library, edit the following line in the “gulpfile.js” file, so that the variable “useJQuery” has a value of false (see below)
                - gulpfile.js (line 39)
                    - const useJQuery = false
    - Line 25
        - This line includes the script tag that links to the compiled JavaScript file
- Don’t forget to edit the text between the “title” tag to reflect the title of your project

Including Google Fonts

- For this project, the most trouble-free way of including Google Fonts in this project is to place the Google Fonts “link” around line 10 of the “index.html” file

Including Icons

- Some of the mock-ups require icons. I recommend using Font Awesome for your icons.
- **Note:** You do not need the paid version of Font Awesome for this project. The free, self-hosted version is fine for this project. Click the link below for instructions on how to install Font Awesome locally for your project
    - Hosting Font Awesome Yourself
- See your instructor for help if you are stuck trying to get Font Awesome icons to work with your project

Including JavaScript

- **For jQuery**
    - A version of jQuery is automatically added to this project. You do NOT have to add the jQuery library to this project
- **For jQuery or other Plugins**
    - Place the plugin script inside the “libs” folder located in the “scripts folder of the ”dev” folder
        - Gulp will automatically add it to the compiled script folder
- **For your own scripts**
    - Write your own JavaScript in the “scripts.js” file inside the the “scripts” folder located in the “dev”
        - Gulp will automatically add your code to the compiled JavaScript file
        - The compiled script file is already referenced in the HTML file. Do NOT attach the compiled JavaScript file to the HTML file
        - If you wish to include additional script files, simply save them to the “scripts” folder and Gulp will automatically add them to the compiled JavaScript file
            - Do NOT attach your additional script files to the HTML file

Recommended Development Strategies

- Communicate often with your teammate(s) (within reason)
    - Never assume they magically know what you know. Keep your teammate(s) in the loop.
    - Talk to your teammate often about the status of the project
    - If you are going to start editing file “x” tell them that, then your teammate(s) won’t “accidentally” start also editing file “x”, which as I stated earlier is a recipe for disaster because this will inevitably cause merge conflicts with version control
    - Working remotely is not an excuse for not communicating with your teammate(s). We have text messaging, slack, social media and email. Make sure you leverage these modern communication tools to keep everyone on the team up to date with the project
- Avoid working on the same code file at the same time as your teammates
    - Working on the same file at the same time can cause merge conflict errors and general confusion when you have to figure out who’s version of the file is the latest.
    - This is the most common problem to people new to working on coding projects as part of a team.
    - The solution is making your code as modular as possible.
        - For this project most of your work will be with Sass. To make your Sass code modular, break up your Sass files into smaller “partial” files and import them into the master “styles.scss” file. This way team member 1 can work on “_header.scss” file while team member 2 can work on the “_footer.scss” file etc.
        - Since we cannot easily break up our HTML file into modular parts I would recommend sitting down as a team together and building the HTML file first.
-

FAQs

Our CSS Files Are Not Going Up to GitHub?

- Your compiled CSS files located in your “dev” folder are NOT version controlled and you will not see them in your GitHub repository (this is controlled by the “.gitignore” file). This is by design, as version controlling compiled files of any kind will lead to constant merge conflicts and other issues
- But how do we share the CSS between teammates?
    - Remember CSS is generated by Sass. So, to re-generate the CSS simply run the “gulp” command in the terminal and your CSS will re-generate

Our Concatenated JavaScript File Is Not Going Up to GitHub?

- The concatenated JavaScript file (script.min.js) is a computer-generated file created by the Gulp file, and should not be version controlled
- When you run the “gulp” command it will automatically recreate this file for you based on the script files stored in the “dev/scripts” folder.
- All the separate scripts that you write and store in the “scripts” folder will be version controlled

Project Submission

- Follow the below listed steps to properly submit your project
    1. Launch Node.js Command Prompt (Windows) or Terminal (Mac) and CD (change directory) into the directory where your group project is stored
    2. Run the command "npx gulp build". This will run the Gulp build task that is written in the "gulpfile.js" file. This command will compress all the HTML, CSS and JS files and move them to the "dist" folder. Any image files, font and media files will also be copied over to the "dist" folder. The entire process is automated and does not require you to move any files. Just run the "npx gulp build" command
    3. Test your project from the "dist" directory to make sure all is well
    4. Add and commit all the newly created "dist" files to Git and push the latest version up to your project stored on GitHub.
    5. One team member should download the finished project from GitHub
    6. With the downloaded copy from GitHub, one team member should submit the zipped copy of the project to the drop box for this project on the Learning Hub
        - **a.** *** IMPORTANT ***:
            - **i.** In your submission notes please include the names of your teammates so that your teammates can receive a grade for this project
        - **b.** *** IMPORTANT ***:
            - **i.** Make sure you submit the entire zip folder from GitHub. I need to look at both your dev folder and the dist folder

Technical Mark Deductions

- Major differences between the mock-ups and your submitted project will receive a *2 mark deduction for each major difference* from the "Responsive web page that matches the mock-up" marking category
- Slight minor differences will not cause mark deductions
    - Examples of minor differences include:
        - minor colour differences
        - minor font-size differences
        - minor spacing differences

Marking Criteria:

- All instructions followed	10 marks
- Proper semantic tags used	5 marks
- Functioning responsive menu system that matches the mock-up	5 marks
- Responsive web page that matches the mock-up	40 marks
- **Total**	**60 marks**