ECE444-PRA1: Front-End Design

Goal:

After this PRA, the students should be able to do the following:

1. Understand HTML, CSS and basic UI principles.
2. Add their own CSS stylesheets to improve the UI.
3. Use bootstrap components in their project.
4. Develop front-end prototypes based on user stories.
5. Implement dynamic changes in the DOM using JavaScript.

In this PRA, you will be developing a personal website. We use the code from this repository as a starting point:
https://github.com/varadbhogayata/varadbhogayata.github.io

Please clone the code from this repository to your local machine.

Material:

1. CSS Selectors - https://www.w3schools.com/cssref/css_selectors.asp
2. CSS Styling Rules - https://www.school-for-champions.com/web/css_rule_types.htm#.YVnwiprMI2w
3. All CSS Properties for reference - https://www.w3schools.com/cssref/
4. Javascript Guide https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide

Plan:

Students should watch the video (video link), which introduces good UI practices, HTML and CSS, then TAs will help students during the in-person PRA sessions.

Requirement & Evaluation:

Students should follow the activities below. Once you are done, please submit your GitHub repo URL in Assignment/PRA-1 and visit the TA during the PRA session for grading. Your grade depends on the completeness of the task.

In your readme file, we expect to see 5 sections (use markdown)

- Activity 1 (1pt): a screenshot
- Activity 2 (1pt): two screenshots (repo and the deployed website)
- Activity 3 (1pt): two screenshots (repo and the deployed website)
- Activity 4 (1pt): two screenshots (repo and the "places travelled" page in deployed website)
- Activity 5 (1pt): a screenshot of top three projects

*For adding images in readme file, you can create a subdirectory in your repo and provide the relative path in the readme.

Activity-1: Cloning personal website repo into your own GitHub account.

Create an empty repo on GitHub under your own account, and then commit code from the personal website repo mentioned above into the empty repo.

1. Create an empty public repository, name format: "<your GitHub username>.github.io"
2. Upload the cloned code base to the repo you just created.
3. Update the readme file with your name and mention: this repo is a clone of https://github.com/varadbhogayata/varadbhogayata.github.io (Because your repo is public on GitHub, and mentioning the source of the code is a good practice of reusing open-source code).
4. Commit the updated readme file and take a screenshot of the repo to add them to your readme file.

Activity-2: Adding your personal information.

1. Replace the information in all sections (about, experience, projects, skills, education, contact and resume) present on the website with your information.

2. Follow the instructions here, to deploy your repo using GitHub pages.

If successful, your personal website will be deployed at "<your GitHub username>.github.io

3. Commit all your changes and take screenshots of the repo and the home page of your deployed website. Add the screenshots to your readme file.

Activity-3: Changing the Theme Color of Your Personal Website

1. Replace the current theme of green in the personal website and change it to any other color of your choice ensuring color scheme is aesthetic.

2. Commit all your changes and take screenshots of the repo and the home page of your deployed website. Add the screenshots to your readme file.

Activity-4: Creating new section and embedding google maps

1. Create a new section called "places travelled" and add it to the navigation bar on the left.

2. In the "places travelled" section, embed a google map that shows all the places you have visited so far.

Here are some tutorials on how to embed google maps on your website and you are encouraged to refer to them.

For example:

    2.1. https://vibrantwebs.medium.com/google-maps-html-embed-with-multiple-locations-and-custom-markers-free-eab8fbce7f9a

    2.2. https://support.google.com/maps/answer/144361?hl=en&co=GENIE.Platform%3DDesktop

3. Commit all your changes and take screenshots of the repo and the "places travelled" section of your deployed website. Add the screenshots to your readme file.

Activity-5: Adding Dynamic Content with JavaScript

1. Update the section "Projects" to include your most recent projects.

2. You should use JavaScript to dynamically display only the latest project in the section.

3. Add a new <div> element in the HTML for the "Recent Projects" section. Ensure that this <div> has an ID so that JavaScript can inject the project content into this section.

4. Write a JavaScript function that loops through the array of projects and generates the HTML content for each project. Use DOM manipulation methods to insert this content into the "Recent Projects" section of your website.

5. Implement a "Load More" button at the bottom of the "Recent Projects" section that, when clicked, dynamically loads additional projects from your array using JavaScript.

6. Once the dynamic content is working, commit all your changes and take screenshots of the repo and the "Recent Projects" section of your deployed website. Add the screenshots to your readme file.

Here are some useful tutorials to DOM manipulations:

1. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Manipulating_documents
2. https://www.freecodecamp.org/news/the-javascript-dom-manipulation-handbook/