Project 1 - Personal Portfolio

Aim:

To produce a personal portfolio website.

Overview

A personal online portfolio is a website created to showcase your coding/programming skills, experiences, and personal projects. It is an online resume of your professional accomplishments and abilities which provides your prospective client or employer with an overview of the projects you have worked on and your skill level.

Key Elements

- **Home Page/Section:**
  This is the landing page of your portfolio. It should be inviting and provide a clear introduction to who you are.

- **About Page/Section:**
  This section offers more detailed information about you. It may include your professional background, skills, education, and employment.

- **Contact Page/Section:**
  This page should include your professional contact information, such as your email address, phone number, and possibly links to professional social media profiles or LinkedIn.

- **Projects Page/Section:**
  This is where you display your work, describe the projects, the technologies used, and your role in each project. Placeholder contents can be provided if no available projects.

Assessment Criteria

Usability Impact

- Design a web application that meets accessibility guidelines, follows the principles of UX design and presents a structured layout and navigation model, and meets its given purpose.
- Design the organisation of information on the page following the principles of user experience design (headers are used to convey structure, information is easy to find due to being presented and categorised in terms of priority).
- Write custom HTML and CSS code to create a responsive front-end web application consisting of one or more HTML pages with significant interactive functionality.
- Implement an interactive web application that incorporates images or graphics of usable resolution, consistent styling, undistracted foregrounds.
- If used, implement clear navigation to allow users to find resources on the site intuitively.

Layout and Visual Impact

- Design interactivity for a web application that lets the user initiate and control actions and gives feedback.
- Design a web application that meets accessibility guidelines (e.g., contrast between background and foreground colours to cater for the visually impaired) Optionally, add alt text for non-text elements.
- Include graphics that are consistent in style and colour.
- Ensure that foreground information is never distracted by backgrounds.
- Ensure that you use up to a maximum of three colours in the web design.

Code Quality

- Organise HTML, CSS into well-defined and commented sections.
- Clearly separate and identify code written for the website and code from external sources (e.g., libraries or tutorials)
- Attribute all code from external sources to its original source via comments above the code and (for larger dependencies) in the README.
- Organise code and assets files in directories by file type.
- Group files in directories by file type (e.g., an assets directory will contain all static files and may be organised into sub-directories such as CSS, images, etc.)
- Write a README.md file for the web application that explains its purpose, the value that it provides to its users, and the deployment procedure.
- Name files consistently and descriptively, without spaces or capitalisation to allow for cross-platform compatibility.
- Insert screenshots of the project features, give a brief description of what each feature does and explain its value to the user.
- Use Git & GitHub for version control of an interactive web application up to deployment.

Getting Started

In this section, it covers the necessary resources and requirements that are needed to complete the project and ensure that you satisfy all requirements.

Session with your Mentor:

Before commencing the project, you will need to book a session with your Mentor to plan how you will go about completing the project, understanding the project brief, planning, and staging, and finally how to go about submitting the project.

Documentation:

Write a README.md file for your project that explains what the project does and the value that it provides to its users. Attribute any code from other sources. Document all testing procedures and findings.

Version Control:

Use Git & GitHub for version control in making regular commits to your GitHub repo.

- Make sure you are adding, committing, and pushing your code regularly throughout creating your project.
- When committing your code to GitHub, ensure you commit messages describing what code you have changed since the last commit.

Deployment:

Deploy the final version of your code to a hosting platform such as GitHub Pages. You can follow the steps below to deploy it:

- Open your project repository on GitHub, Go to the 'Settings' tab.
- Scroll down to 'GitHub Pages'
- Under 'Source', click on the 'None' dropdown and choose 'Master'.
- Make sure the Root Folder is selected and click 'Save'.
- Your page will reload, scroll back down to GitHub Pages and you will now see a URL. This is a link to a live version of your site.

Testing:

Make sure to run your code through code validation & quality tools for example:

- HTML - https://validator.w3.org/#validate_by_input
- CSS - http://jigsaw.w3.org/css-validator/
- Document all errors and warnings found, make sure you document your process of fixing these errors.

Attribution:

Maintain clear separation between code written by you and code from external sources using clear comments. Include the URL of where you found the code (e.g., libraries or tutorials).

Resources

Main Technologies Required: HTML, CSS

Optional: Bootstrap and/or other CSS libraries/frameworks/any other JavaScript libraries like jQuery.

GitHub Documentation: https://er-bharat1992.medium.com/writing-readme-md-markdown-file-file-bd711d1afbfa

GitHub Commits: https://github.com/sw1ckham/i-brary/commits/master

Readme Project example: https://github.com/Allwrightben/starquiz

Mentor Support: Via your dedicated mentor