- ACTION PLAN -

Building a Design System to Strengthen Web Development Skills

Executive Summary

This Action Plan aims at addressing the urgent need to strengthen full-stack web development skills. It has been determined that building a design system with both front- and back-end features is the best solution to achieve this strategic goal.

The Web remains the prime destination for people worldwide to access information and knowledge. Countless digital footprints are generated everyday, and this massive market deserves investment.

Since taking that web-based applications course in school, I've designed around 10 websites, most of which are static - websites that run without a database system and server-side programs. Whilst I managed to make websites and web applications responsive, it took me a lot of work to build these sites due to the absence of a reusable, extensible, and accessible design system with tools readily usable out of the box.

A design system is a set of guidelines and made-ready components that enable developers to collaborate and build consistent and recognizable digital experiences fast. A typical system usually include a set of guidelines explaining the best practices in web design, a documentation detailing how the design system can help build web-based experiences, and a range of pre-built components with source code and illustrative examples allowing developers to put into action the system in their own projects.

Building a design system will expose me to opportunities to dig deeper into some of the most popular components of websites and web-based applications and master the development techniques of these components such as navigation bars, data entry forms, and slideshows, etc. This learning experience will ensure that I have a chance to encounter challenges from every aspect of the Web, thereby enabling the improvement of my full-stack web development skills.

Overview

This is a web development project. You will be tasked to build a full-stack web development from scratch using HTML, CSS, JavaScript, PHP and Python. This framework should include a front-end and a back-end component. Once it's released, it should be used in future web projects. The websites and applications that run on this framework should be responsive, mobile-first, accessible and visually appealing to site visitors and customers.

This framework should include everything from a responsive grid layout to data submission forms supported by server-side programs. The first edition of the framework should include just the basic web components. It should be a minimum viable product (MVP) with all the components necessary to construct a fully functional website or web application.

Breakdown

The framework will have a front and back end:

- Front-End: The front should include all the commonly used web components on websites and web applications such as a12-column grid layout, drop-downs, navigation bars, footer menus, and data submission forms, etc. It should also include animation that makes the site move smoothly and visually attractive.

- Back-End: The back should include the code for supporting the website to function, to collect and store data, and to perform data processing and analysis. It should provide features such as data collection forms, login and logout, and shopping cart, etc. as stand-alone plugins.

- Animation: Animation should make websites and web applications appealing. Transition animation is commonly used on modern websites and applications, and this framework should provide animation out of the box.

Deliverables

A framework should be usable by not only one but a group of web developers. Developers should be able to collaborate on the framework to build out stunning websites and applications. To make this possible, this project should include three components: 1) documentation, 2) examples,

and 3) code.

It should include a detailed documentation instructing users on how to utilize the tools that the framework offers in their projects. It should include examples for users to assess whether the framework fits their needs. It should also include code for users to easily implement the framework in their projects.

- Documentation: The documentation should include texts, examples, and code. It should be served as a PDF file in companion with a web-based HTML document. The documentation's overarching purpose is to get users onboard the framework, to help them understand what the framework is capable of, and to support them in their development projects.

- Examples: The examples should help users see clearly the effectiveness of the framework. It should include examples of each component that the framework includes. It should as well present some pages or even websites and web applications built using the framework.

- Code: Of course, the framework should include code snippets to help developers understand how to implement the framework in their projects. This code should help them build with the framework effortlessly.

Submission

All the materials related to this project shall be submitted in three places. There's a dedicated code repository on Gitee to store the source code of the website, grouped by iterations. There's a website for hosting the online documentation, examples, and downloadable files. And there're local folders on this machine to store these submitted files.

Timeline

This project may take a month or so to complete. Please set out a timeline to ensure that the project will be delivered on time. The deadline is 1 May 2022.

Resources

- Twitter Bootstrap (https://www.getbootstrap.com): Bootstrap is a popular front-end framework from Twitter, Inc. engineers. Refer to the Bootstrap

documentation for a list of components and utilities that should be included on the front-end of the web development framework.

- Mozilla Developer Network (https://developer.mozilla.org): The Mozilla Developer Network, or MDN, is the world's largest online documentation repository for web developers. It's offered in multiple languages covering techniques to build accessible websites and web-based applications.