DA558A Lab 2 Reflections
Fredrik Halsius

As a general note, I added some features during the course of the implementation to get some more practice. This includes custom form validation and an API call to Open Trivia database, which provide quizzes for free.

By doing so, I introduced some new **challenges** as I had to normalize data between different input sources:

- My own pre-defined quiz stored as json
- Quizzes from Open Trivia database
- User defined quizzes stored in local storage

Moreover, I found it challenging to keep the styles and CSS files organized. This includes, for instance, finding unique class names to avoid global scope pollution. While working with vanilla CSS, changes in one place tend to trigger the need to update something elsewhere.

To organize the project, I divided the logic into different pages, with reusable templates and classes to form a **solution**:

Pages:

- The user contact form
- Overview (where all quizzes live)
- The quiz page itself, with a Create Quiz overhead.

Templates:

- Reusable components were identified and tagged <template> in the HTML-docs

Classes to support different game features:

- Quiz, responsible for playing a quiz
- Create Quiz, responsible for user-defined quizzes
- ScoreTracker, keeping track of required answers and the game score
- LocalStorage, responsible for storing user defined quizzes
- Validator, responsible for all custom validation logic

Styles CSS:

- One main global stylesheet for CSS variables and global styles
- One or several page-specific stylesheets to separate concerns

As the application is divided into several pages and classes I did not use my own name to label files as name.html and name.js, as proposed.

Vanilla JS, HTML, and CSS pose some challenges as the projects grow in size, and it is also here we might find the **improvement areas**. First, before inventing the wheel again, one might consider using third-party libraries. This project could have benefited from implementing a JS-validation library. Second, to keep things organized - a CSS preprocessor might help out, I like Tailwind and SCSS. Third, a project like this may benefit from a front-end library using a bundler, for instance, React or Vue.

Reflections

Despite the challenges encountered during the project implementation, some solutions were found and implemented, resulting in a kind of organized and functional application. The identified areas for improvement offer opportunities to enhance the project's organization, efficiency and maintainability.

At first glance, this project may seem straightforward, but as this reflection outlines, there are many intricacies and areas that pose challenges. Overall, it has been an educational and fun assignment.