DECO3801 - Tech Spike Functional Coverage Document

THEM - Typed HTML5 Evaluation Machine
Carl Hattenfels, Scott Heiner, Shen Yong Lau, Robert Meyer, Brendan Miller, David Uebergang

The following is a comprehensive list of functional specifications our HTML5 Evaluator will have implemented by the end of this course. Those that are have yet to be implemented are greyed out.

- Users can type HTML code into a direct input field, which is sent to the back-end parser for verification.
- Users can upload one or multiple HTML files, which are saved to the database and sent to the parser for verification.
- Users can upload a zip file, which is unzipped and added as a set to the database. These files are sent to the parser for verification, along with information about the file structure.
- The help page should provide the user with information as to how to navigate the application.
- The Single File (or Direct Input) Error Page displays an error bar showing the relative amount of each type of error the code contains. On this page, the user's code is displayed with text highlighting the sections of code marked as erroneous by the parser. The user can click on highlighted text to show the respective error message relating to that error. This is shown in a sidebar.
- The Multiple File Error Page displays a list of files the user has uploaded. Clicking a file name takes you to its individual page. Error bars are included next to each file representing that file's error count.
- The File Structure Error Page displays the user's file structure visually, with each individual HTML file having an error bar indicating the types of error that document contains. It should also have a display that shows any errors in file structure that are not tied to specific files, e.g. having no `index.html`. Errors in file structure and file linking returned by the parser should be displayed too. Placing the mouse over such a file error will display the error message relating to that particular error in a sidebar structure. Clicking a file should take you to the file's individual error page.
- The back-end Python program should parse errors passed to it from the website in the four key areas of syntactical / semantic considerations, accessibility requirements, deprecated tags, and poor practice. These should all have associated error messages and, where appropriate, possible corrections. It should take a file structure passed to it from the website and provide information about incorrect file links, with possible suggestions. Currently, we are parsing two errors.