Testing Notes:

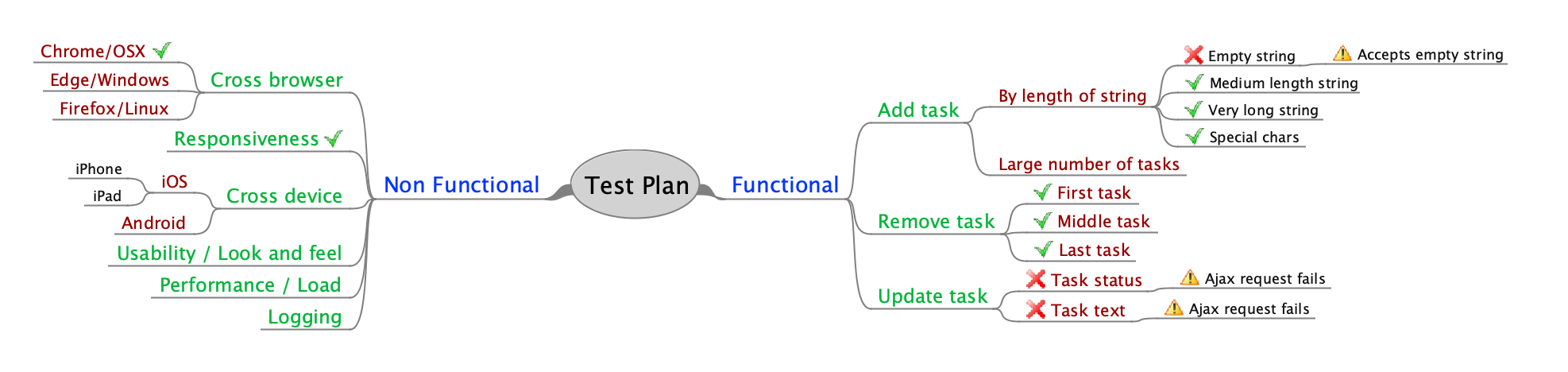
* The app currently has no unit tests. Need to work with the developers to add unit tests (and have this adopted as a practice), and use any code coverage tool to ensure build fails if coverage drops below a pre-set number
* Use a linting tool (like eslint) and add this step to the build to ensure consistency in code style and to enforce following coding best practices

Test Automation notes:

* Have followed the practice of having the automated tests as part of the same code repo as the app. The tests are in the `*./tests*` folder. Instructions are in `***./tests/README.md***`
* Automated tests for the app are built using the same language ecosystem as the app under test. This is to facilitate having developers contribute to the tests too and fostering shared ownership.
* Have followed the Page Object Model to organise tests. This may be unnecessary if the app is only ever going to be a single page.
* Have followed the practice of user-journey based tests that cover a journey rather than separate BDD style acceptance tests for each functionality. This helps to keep the test run time low, test count low, but with the drawback that the tests aren’t independent. For e.g., if the task add functionality isn’t working, the test will fail, even if there’s nothing wrong with task delete. However, the upside is this will help prevent a maintenance nightmare of too many tests.
* In addition to the files under the `*./tests*` folder, I've also modified the below files of the app:
* `*docker-compose.yml*`: Added selenium container & read environment variables from a file `*.env*`
* Added task `*test*`, and webdriverIO dependencies in `*package.json*`. Currently the dependencies are added to the main dependencies block, as I couldn’t figure out why they are not installing when I had them in the devdependencies block.
* Added a route `*/reset*` which deletes all existing tasks. This is to enable the tests to bring the app to a known state before each test. This route is enabled only if ENVIRONMENT is set to development.
* Currently the test framework doesn’t handle updating tasks or deleting specific tasks. Will need to add these abilities in when future tests need it. As of now, it can only delete the first task.

Manual Exploratory testing notes:

Areas explored:



Legend:

-> Test Passed



 -> Test Failed

No icon -> Test not performed yet

Issues:

|  |  |
| --- | --- |
|  |  |
| 1 | Am able to add zero length or all whitespace strings: |
| 2 | Occasional HTTP 418 response if I add items while previous add is still being processed: |
| 3 | On updating status or text of a task, the status/text **does** get updated, however the ajax update request `*POST* [*http://localhost:8080/api/update/<task*](http://localhost:8080/api/update/%3ctask) *id>*` gets no response:    Server log showing no response code or response time for this request: |
| 4 | Alignment of the checkbox, textbox and the `X` not quite right: |
| 5 | The `*GET /api/all*` request seems quite slow, even with zero todo items. (I understand this is in my local environment, but still looks concerning enough to test this out in a real prod-like environment): |
| 6 | Intermittent issue where occasionally when I remove a task, the resultant page still shows the task, and I am able to remove it again. See network traffic below. Task `*5c3ca0c5742aa7e58a4c451e*` was removed, but the `*all*` request after that came back with a *304*, indicating no change, and I was able to delete it again, and this time the delete response was *304*, indicating it was already deleted: |