## FLOWARD – QA AUTOMATION ENGINEER ASSIGNMENT

**Criteria for finalizing technologies**

* Cypress with JavaScript has been selected as the technology
* Because of the following features that Cypress provides for UI and API Automation:
  + Easy setup and maintenance
  + Mocha framework and Chai assertion library
  + Cypress test runner facilitating faster test runs and debugging
  + Various plugins assisting with different automation requirements
  + Automatic wait for pages to load
  + Intercept Network requests and ability to mock API response
  + Easy data handling through fixtures

**Framework Architecture**

The following diagram represents the architecture implemented:



1. Chai Assertion library
2. Cypress commands facilitating interaction with web elements in Actions class
3. Mocha framework providing the hooks, describe and it blocks
4. API intercepts and response/mock
5. Page object files having web element locators and member functions are created utilizing the reusable customized functions from Action class
6. Cypress test cases are created utilizing the functions in respective page objects or functions from Actions class
7. Code is pushed to GitHub or any equivalent centralized repository
8. (8- 10) Code from the master/develop branch will be utilized by GitLab (or any CI/CD tool) to execute the tests on Floward on multiple browsers
9. HTML reports will be published on the CI/CD tool

The framework is implemented in such a way, that anyone with much Cypress or JavaScript expertise can look through the code easily and understand

**CI/CD integration aspect**

.yml or .yaml files will be created to integrate the tests with CI/CD. The code from master/develop branch will be fetched from GitHub by the CI/CD tool. Parallel tests can be conducted and published to Cypress Dashboard. HTML reports can be published in the CI/CD tool after every run by storing the artefacts. The build outcome can be shared with stakeholders over Slack or MS Teams by using appropriate web hooks. Added a sample. yml file for the assignment in the repository.

**Test Result Reporting**

HTML reports will be published in the CI/CD tool after every run by storing the artefacts. The report will be having embedded screenshots to observe the outcome of every test. The report can be shared with stakeholders over email (post build step) or messaging applications (via web hooks)