**Unit, UI, and API Test Automation Framework Requirements**

1. You shall unit test your daily code before creating pull/merge requests.
2. You shall get your pull requests reviewed by at least 2 senior developers.
3. You shall incorporate GitHub Actions to implement a CI/CD pipeline.
4. You shall adopt an API Test Automation tool to test the Business Layer of the application.
5. You shall use headless browser mode to test the User Interface of the application.
6. You shall use Synthetic as well as Production data to test the application.
7. You shall perform Discovery as well as Usability Testing.
8. The test automation framework must distinguish between the role of test files, test cases, test results and provide respective behavior.
9. The test automation framework must include critical components such as a test runner and test driver.
10. The test automation framework shall provide coverage information for each test case
    1. The coverage report shall
11. The test automation framework shall allow a test to report its result automatically

**Process Requirements**

1. You shall use the AGILE process to work on this project.
   1. You shall use SCRUM methodology to complete this project.
   2. You shall hold daily meetings with the stakeholders.
   3. You shall report the progress to the stakeholders daily.
   4. You shall maintain minimum documentation and focus your efforts on development.
2. You shall use GitHub to host your project.
3. You shall use 5 branches to maintain this project.
   1. First branch shall be Master, that hosts the most stable release source code.
   2. Second branch shall be Beta, that shall be made available to the testers and insiders who shall use this for testing.
   3. Third Branch shall be called Hotfix, which will be used to quickly patch critical issues.
   4. Fourth branch shall be called Features, which shall contain the code with newly introduced features.
   5. Fifth branch shall be Releases, which shall be used to issue releases to the users.
4. You shall use GitHub Issues to assign tasks as well as manage schedule of the project.

4.1 You shall use 6 types of Issue labels to differentiate them

4.1.1 Urgent, for critical issues

4.1.2 Documentation, to document progress

4.1.3 Duplicate, to prevent redundancy

4.1.4 Enhancement, to mark a feature that, while not necessary, would be a good addition to the project

4.1.5 Question, to request for further information regarding the issue

4.1.6 Wontfix, to mark an issue that won’t be worked upon for the time being