BIKERR

TESTING PLAN

Version *1.0*
02/03/2021

TABLE OF CONTENTS

1. Testing Procedure

1.1 Unit Testing

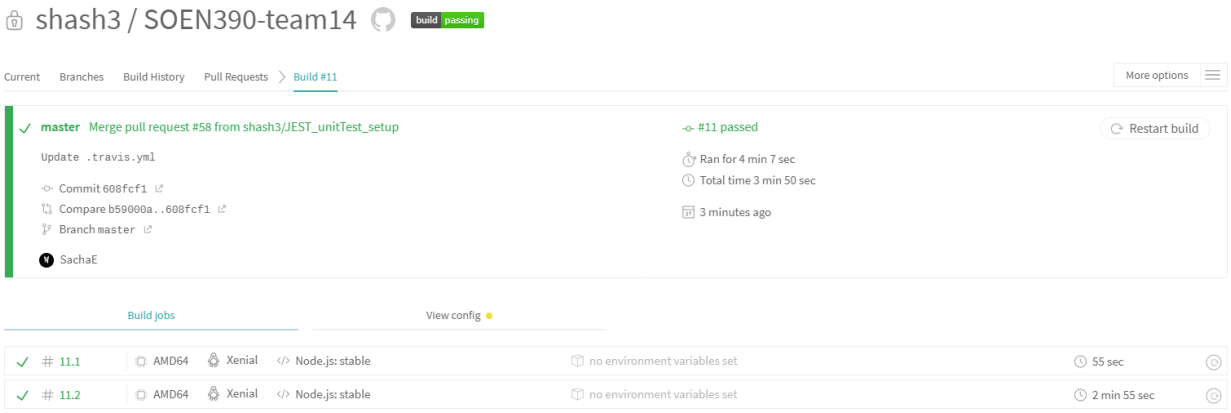
We will be using Jest as our testing framework for our Unit tests. We chose to use Jest since it is one of the most popular frameworks for JavaScript projects. Another reason we chose this framework is because the unit tests are easy to write and execute and the learning curve is not too steep. We will be using version 26.6 of Jest since it is the latest and most up to date stable version. It can be installed by following the instruction given on the official Jest webpage at the following link: https://jestjs.io/docs/en/getting-started
We have already configured this into our code and have a dummy test set up in the file "sum.test.js" as seen below:


```
C:\Users\Shashank\Documents\GitHub\SOEN390-team14>npm run test

> soen390-team14@1.0.0 test C:\Users\Shashank\Documents\GitHub\SOEN390-team14
> jest

 PASS  ./sum.test.js
  √ adds 1 + 2 to equal 3 (2 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.025 s
Ran all test suites.
```

1.2 Test Execution on Pull Requests

We will be using TravisCi for our code analysis which is a continuous integration service used to build and test projects hosted on GitHub. TravisCi will automatically detect when a commit is made and pushed to a GitHub repository and will try to build and run tests. We already have TravisCi configured with our repository on GitHub to build and test our back-end and front-end. We are looking into adding

SonarCloud/SonarJs for continuous inspection of code quality. Sonar does static code analysis, which provides a detailed report of bugs, code smells, vulnerabilities, code duplications.



1.3 Code coverage

We will be using Istanbul for our code coverage. Istanbul is a Javascript code coverage tool that is built into jest, and since we will be using Jest for our unit testing, this implementation is convenient for our project. The following document details how to configure jest coverage:

https://github.com/facebook/jest/blob/master/docs/Configuration.md

2. Weekly activities

2.1 Weekly Tests

i. Build and run the most up to date version of the project. A specific set of activities should be performed in order to maintain consistency. For the same reason, the same machine should be used to perform the weekly tests.

ii. Weekly review of bugs:

- Verify that “fixed” bugs are really fixed and do not persist.
- Rank bugs relative to their urgency as well as the progress of the project as a whole.
- Generate a weekly report of fixed bugs.
- Verify that “fixed” bugs did not generate new bugs.