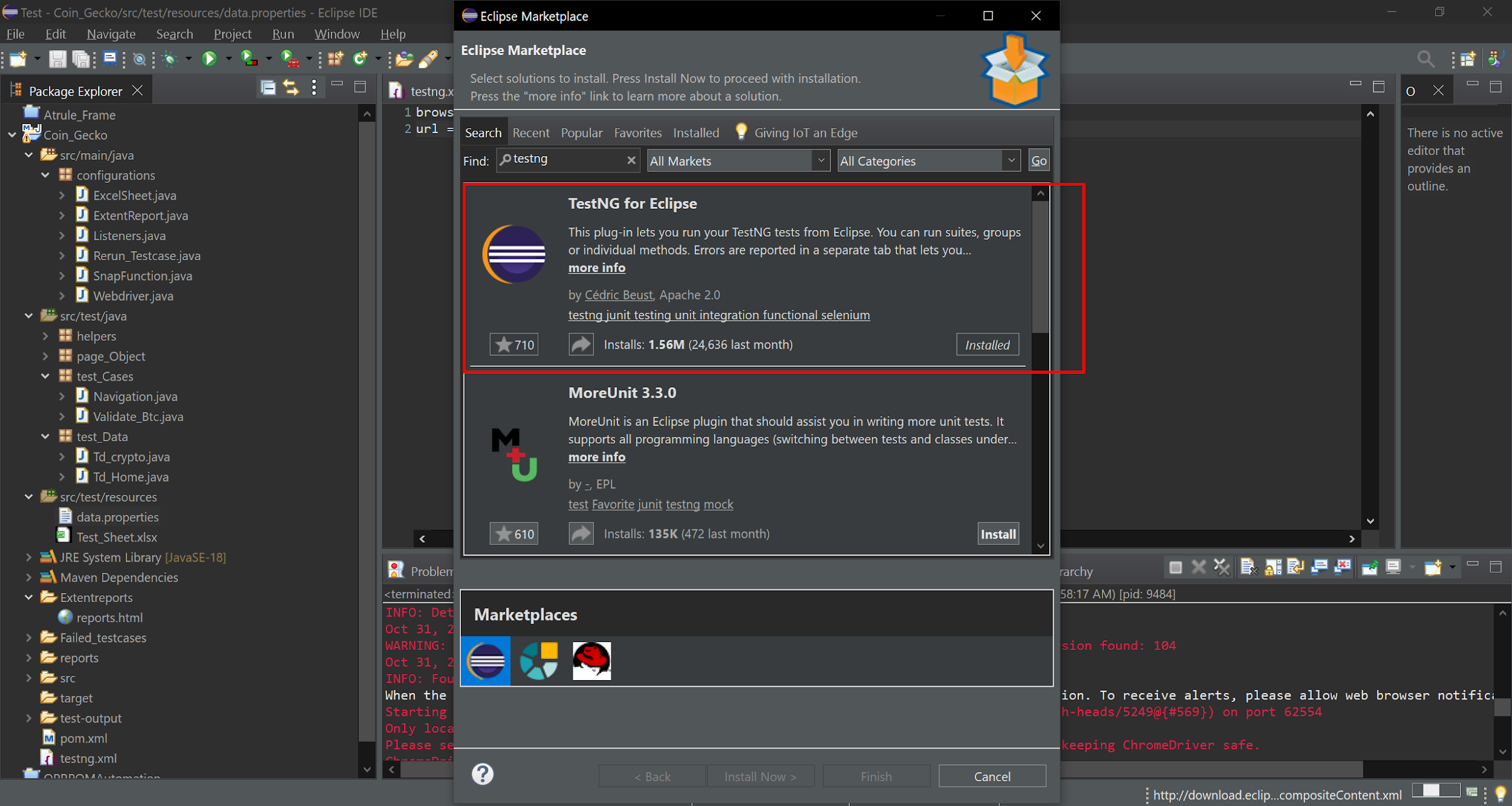
# Automation Framework

# Coin Gecko

Setup

1. Make sure Java is installed in your machine.
2. Setup eclipse, and install TestNg tool using Eclipse Marketplace



1. Now click on File > Open Projects from File System > Click Directory button > navigate to your project > Select Folder > Finish.
2. Maven will install the requirements of the automation framework.

Assertions

Assertions is applied throughout the framework.

Note:- Assertions are applied mostly on static text. We couldn’t apply assertions on the Numbers Because they were not static. In order to apply assertions on them we needed some api’s so that before each validation we hit the api and get some fresh input and test it against the actual one.

Locators

The first priority in implementing locators was to ID’s, but in most of the senarios we had to make custome css selector and xpath as there were most classnames available.

Which Language and Tool used?

Selenium with Java. Along with the implementation of TestNg,Spark Reporter and other 3rd party tools.

Which Project Management tool used?

Maven is being used as a Project/dependency management tool.

Which design Pattern is used?

Page Object Pattern is being used.

Hierarchy

*src/main/java:*

*Configurations:-*

*ExcelSheet: Contains methods to update test status in Excel Sheet.*

*ExtentReports: Contains directory and implementation of spark reports.*

*Listeners: This class listens to all the execution of our test cases.*

*Rerun\_Testcase: This class will rerun the failed test cases 3 times to check whether they are failing due to any other reason.*

*Snap\_Function: It has screenshot methods.*

*Webdriver: Contains webdriver implementation.*

*src/test/java:*

*Helpers:*

*Helpers\_Crypto : This class contains shared function limited to a webpage.*

*Helpers: Class contains shared functions that are being used throughout the project.*

*Page\_Object:*

*Homepage: Contains locator of homepage page.*

*AccessCryptoData: Contains all the locators of a specific page.*

*test\_Cases:*

*Navigation:* Contains test cases of redirection/Navigation from homepage.

*Validate\_btc: Contains test cases of share, add to portfolio,etc scenarios.*

*test\_Data(Using Data Providers):*

*Td\_Home: Contains test data for homepage.*

*Td\_crypto: Contains test cases of page showing specific crypto details.*

*src/test/resources:*

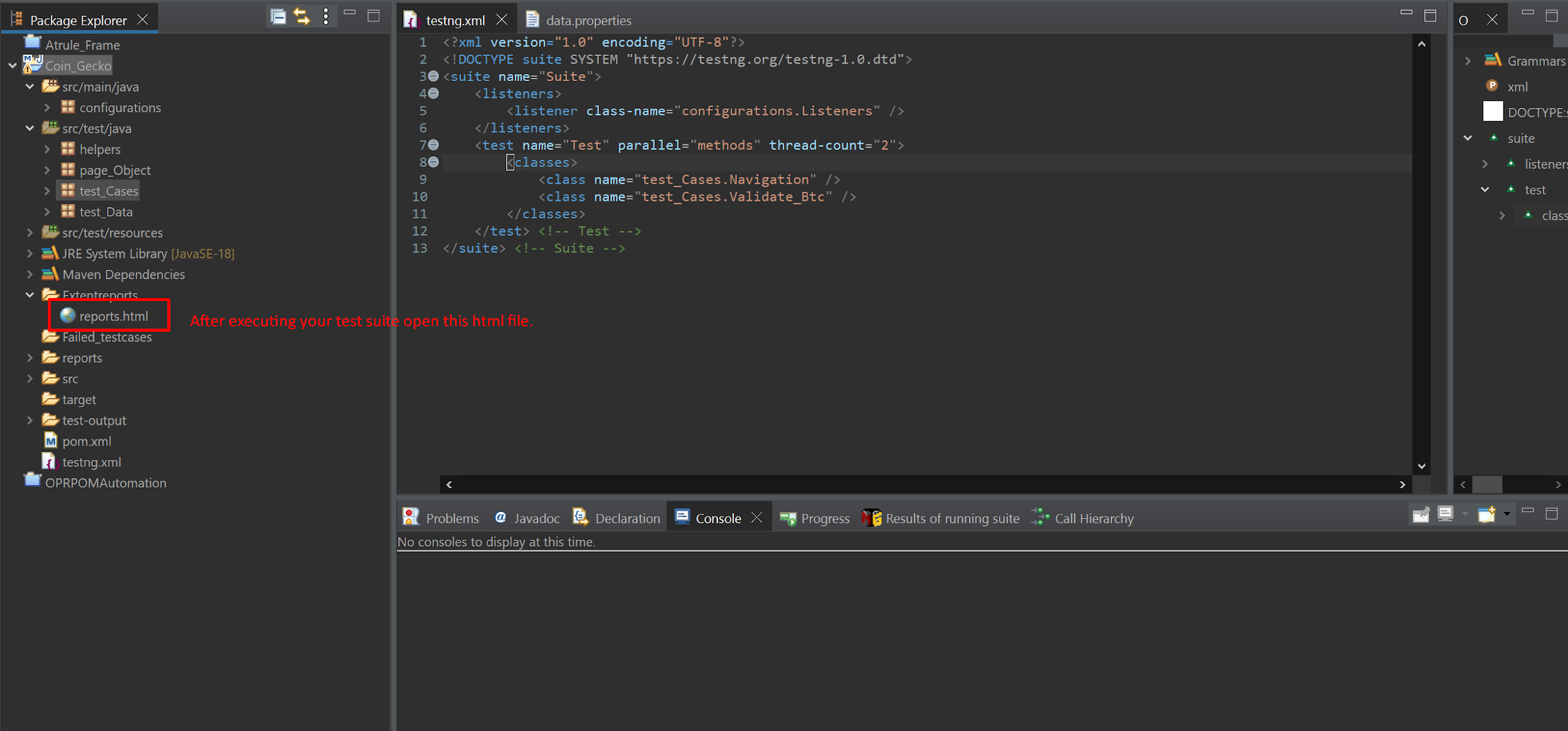
*Data.properties: Contains Url and browser name.*

*Test\_Sheet.xlxs : Contains Method name and execution result.*

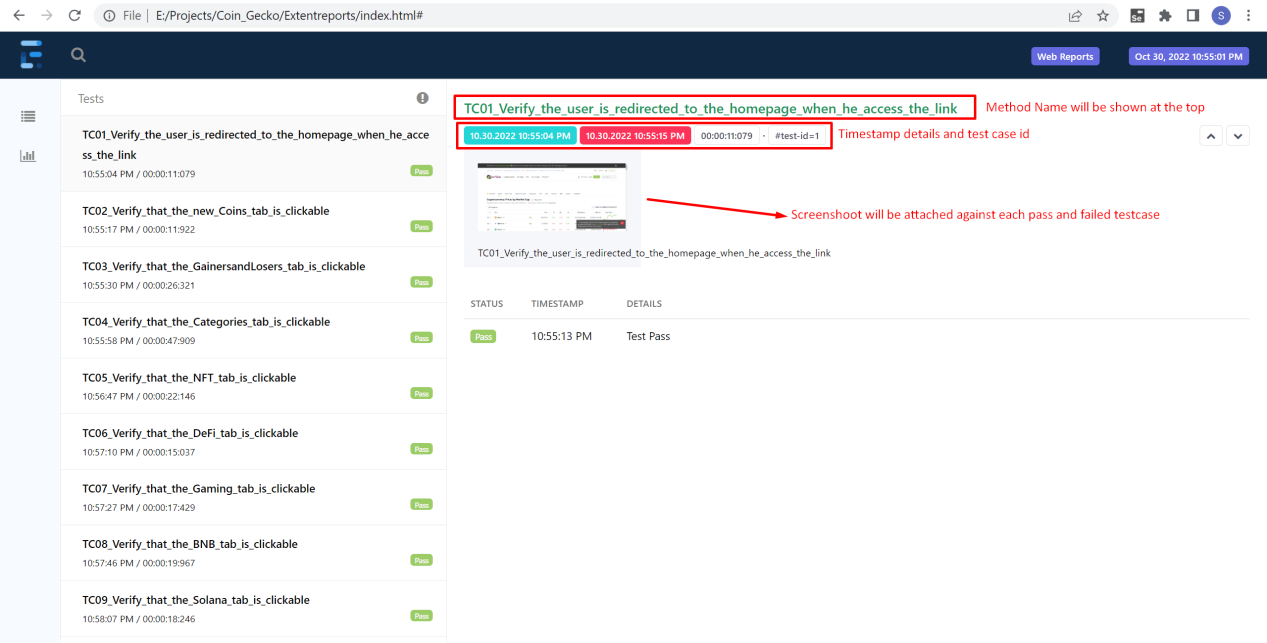
Implementation of Reporting:-

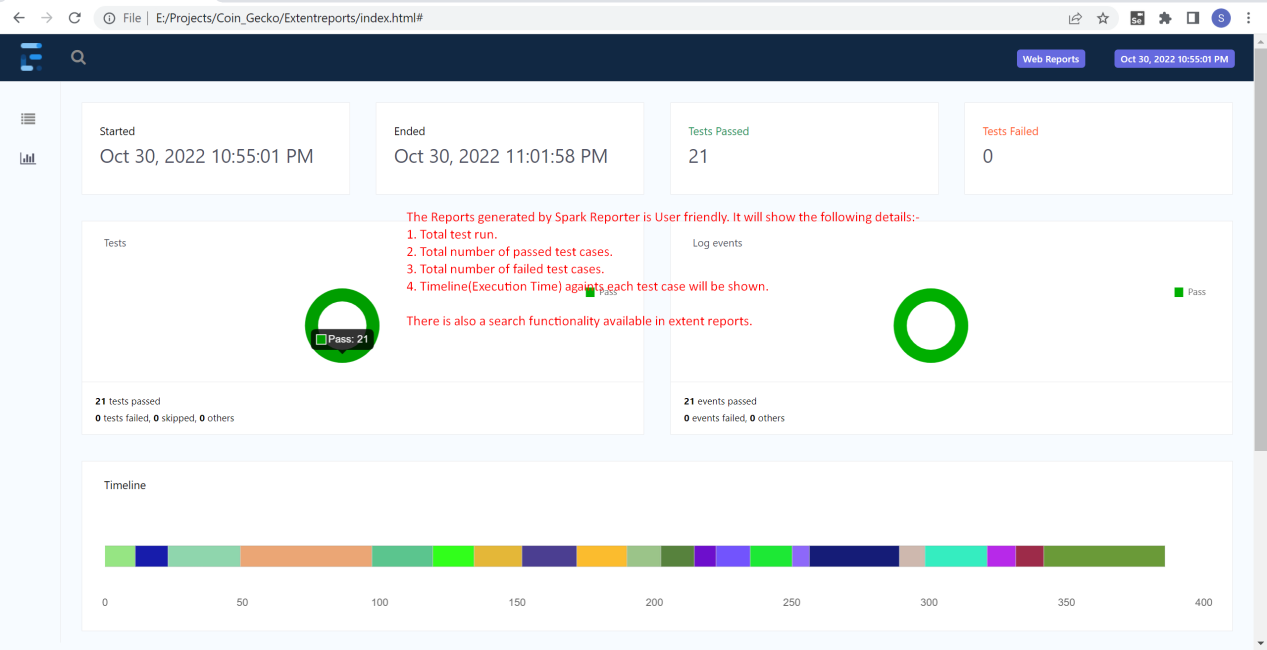
Currently the framework is generating three different types of reports Spark Reports, TestNg reports and Output of the result in excel sheet.

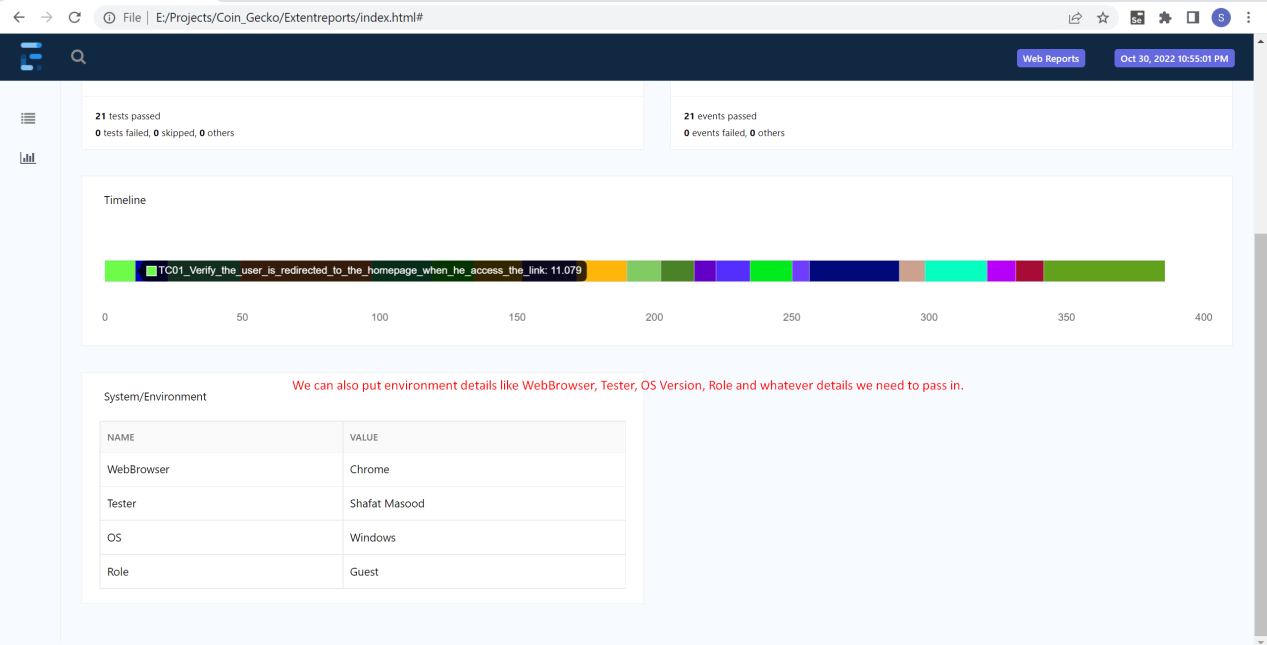
Note:- In order to view Spark reports click the following html File.

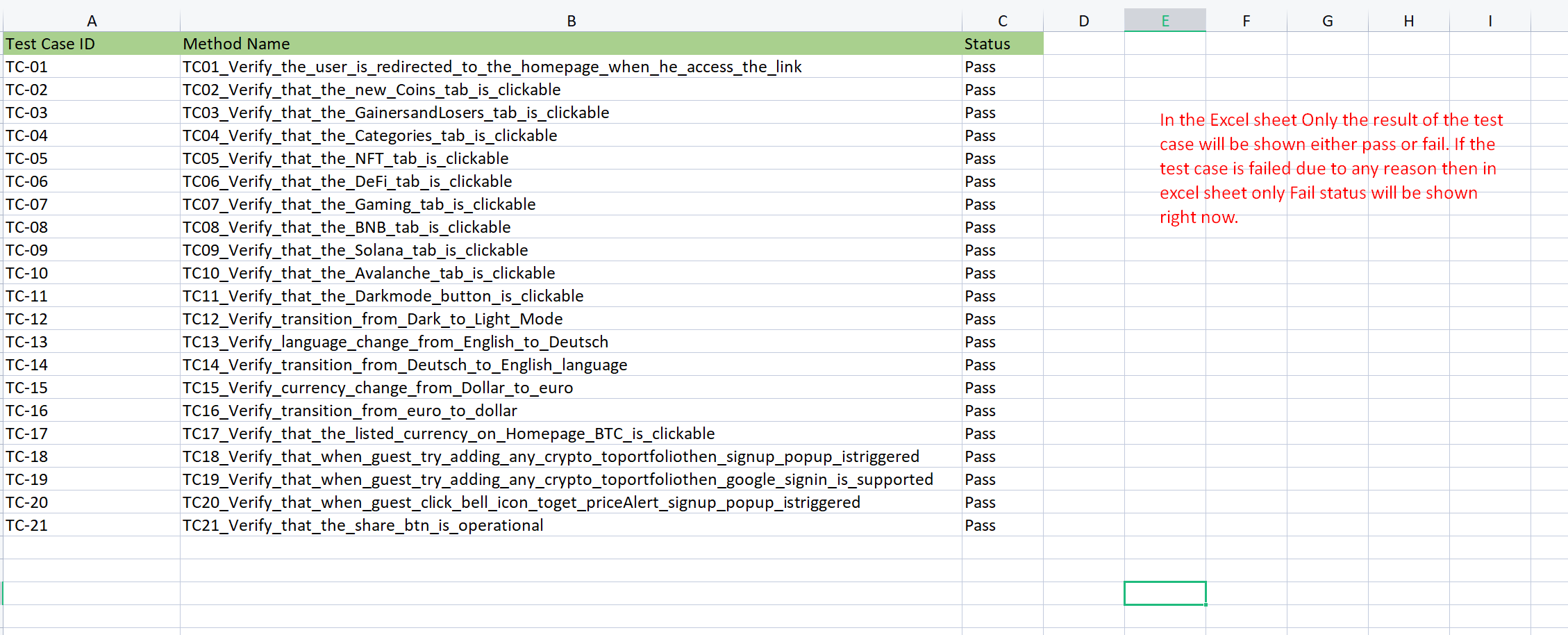


Following are the screenshots attached, the detail is mentioned in the screenshot.





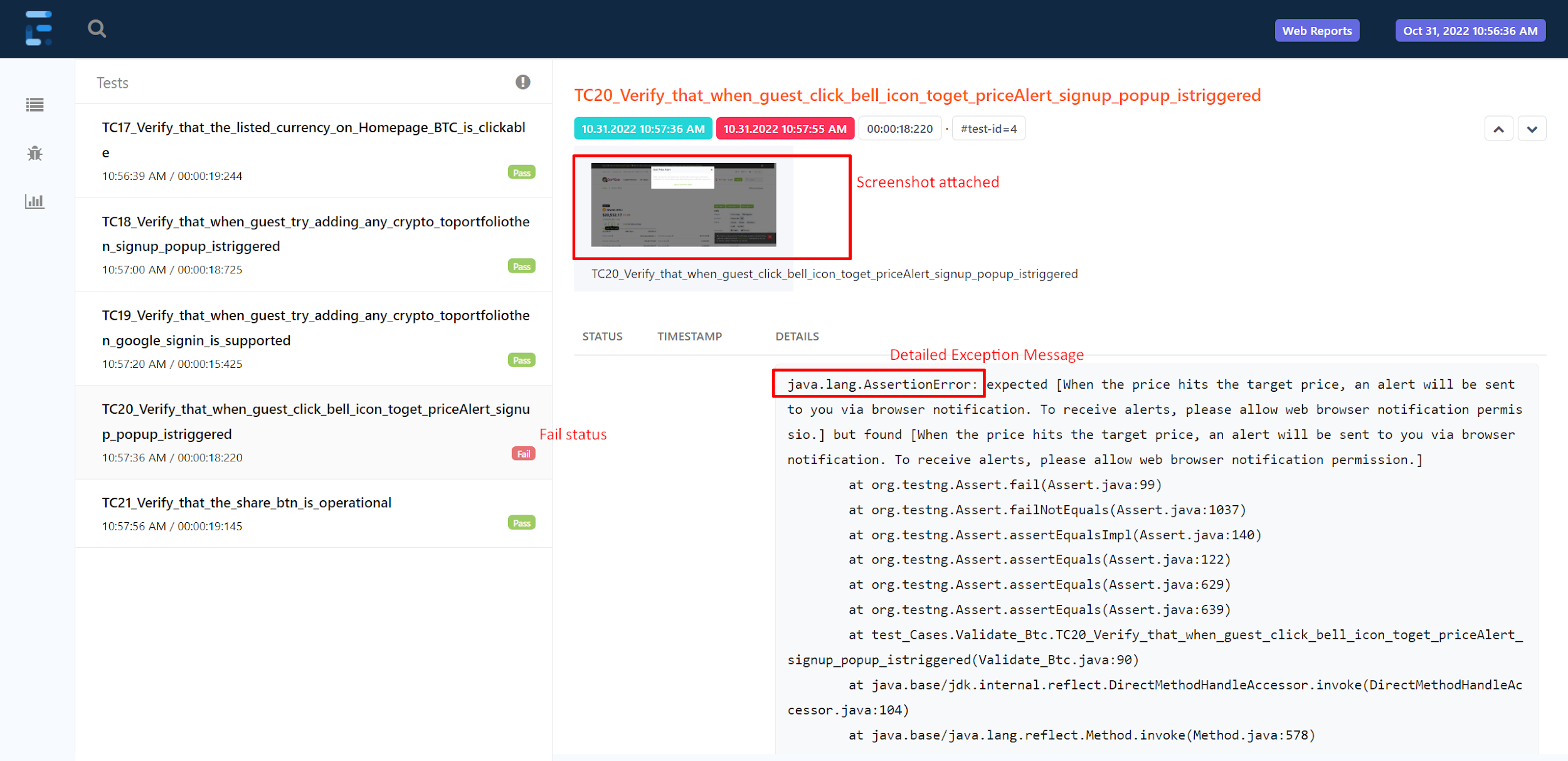




Error Handling:

If any test case fail due to any reason then the exception and the screenshot will be attached in the spark reporter.But first the automation framework will rerun that test case 3 times to check whether its failing due to flakiness or not.

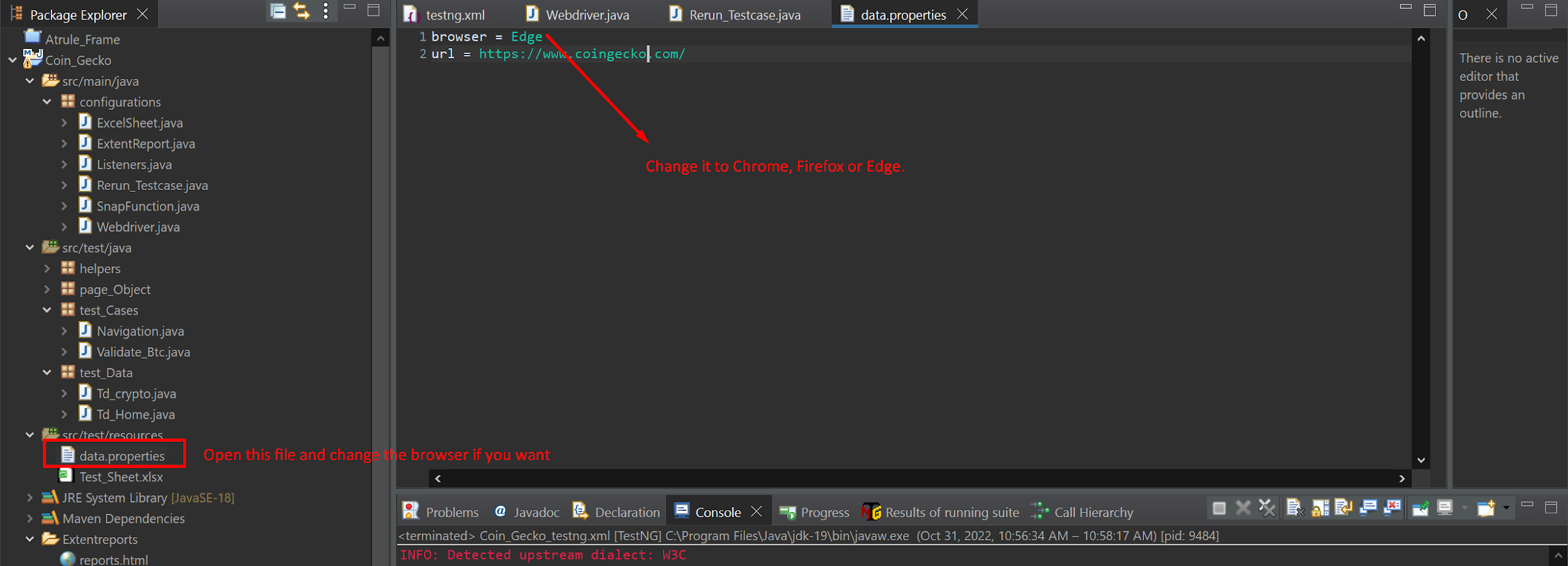
The bug report by extent reports will be as:-



Which Browsers does this framework Support?

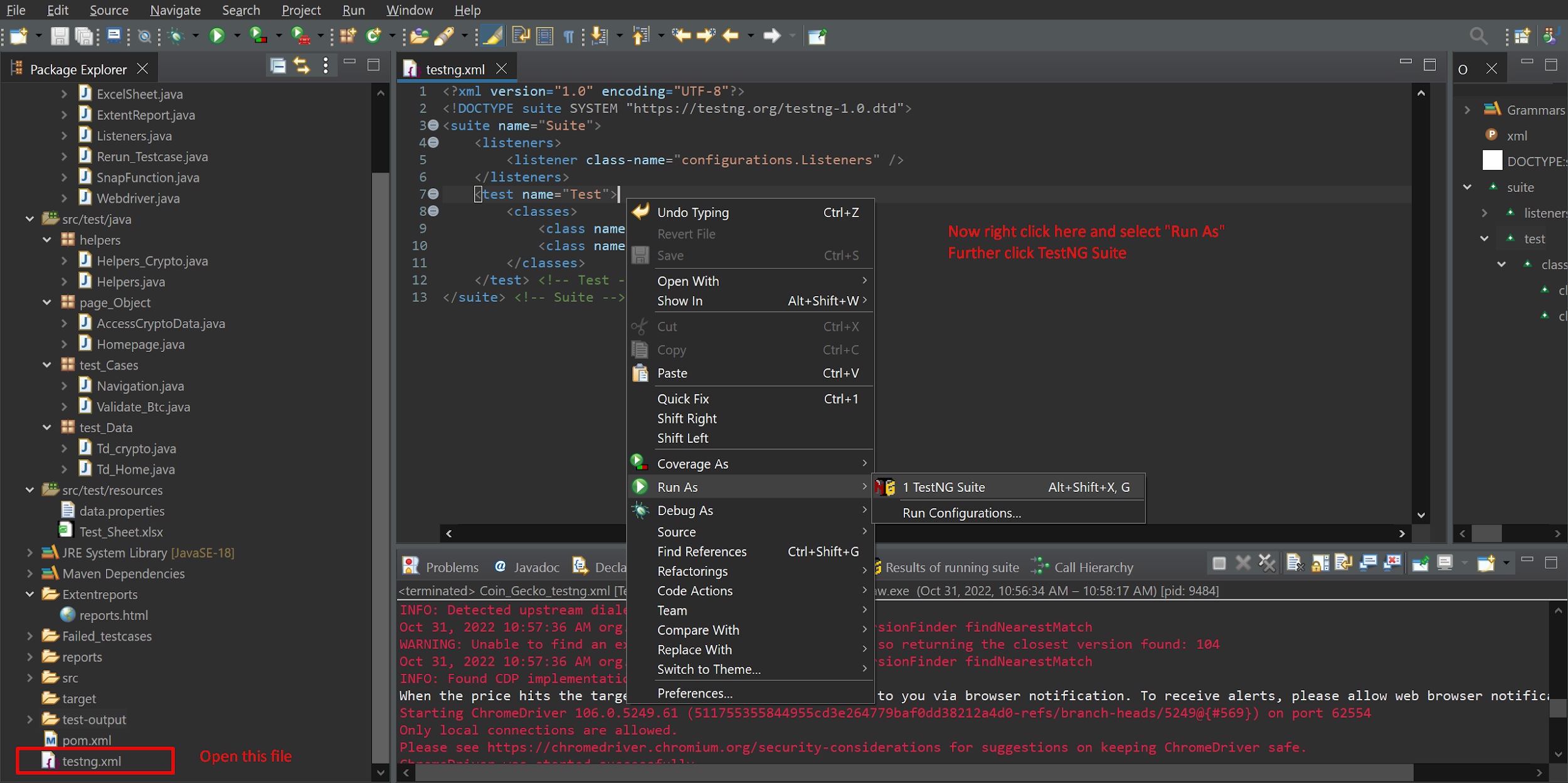
The framework supports 3 different browsers Chrome, Edge and firefox.

The Default one is Microsoft Edge, if you want to change it you can do so in the data.properties file under src/test/resources.



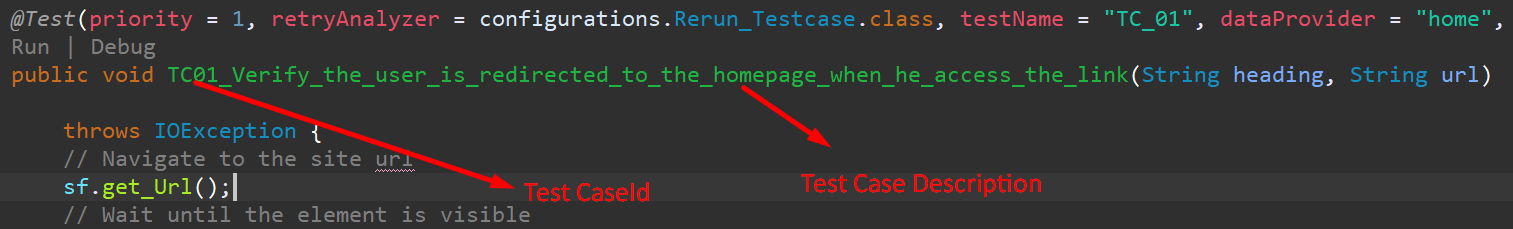
How to Run framework?

Individual Test cases can be run. But if you want to run the complete test suite then open the testng.xml file and right click then run using testng. This execution will generate reports.



Naming Convention

Method Names are detailed hence improving readability of the code.



Comments

Throughout the project Comments have been placed to increase code readability.

Automated Test Cases

Following are the test cases that have been automated:

| TC01\_Verify\_the\_user\_is\_redirected\_to\_the\_homepage\_when\_he\_access\_the\_link |
| --- |
| TC02\_Verify\_that\_the\_new\_Coins\_tab\_is\_clickable |
| TC03\_Verify\_that\_the\_GainersandLosers\_tab\_is\_clickable |
| TC04\_Verify\_that\_the\_Categories\_tab\_is\_clickable |
| TC05\_Verify\_that\_the\_NFT\_tab\_is\_clickable |
| TC06\_Verify\_that\_the\_DeFi\_tab\_is\_clickable |
| TC07\_Verify\_that\_the\_Gaming\_tab\_is\_clickable |
| TC08\_Verify\_that\_the\_BNB\_tab\_is\_clickable |
| TC09\_Verify\_that\_the\_Solana\_tab\_is\_clickable |
| TC10\_Verify\_that\_the\_Avalanche\_tab\_is\_clickable |
| TC11\_Verify\_that\_the\_Darkmode\_button\_is\_clickable |
| TC12\_Verify\_transition\_from\_Dark\_to\_Light\_Mode |
| TC13\_Verify\_language\_change\_from\_English\_to\_Deutsch |
| TC14\_Verify\_transition\_from\_Deutsch\_to\_English\_language |
| TC15\_Verify\_currency\_change\_from\_Dollar\_to\_euro |
| TC16\_Verify\_transition\_from\_euro\_to\_dollar |
| TC17\_Verify\_that\_the\_listed\_currency\_on\_Homepage\_BTC\_is\_clickable |
| TC18\_Verify\_that\_when\_guest\_try\_adding\_any\_crypto\_toportfoliothen\_signup\_popup\_istriggered |
| TC19\_Verify\_that\_when\_guest\_try\_adding\_any\_crypto\_toportfoliothen\_google\_signin\_is\_supported |
| TC20\_Verify\_that\_when\_guest\_click\_bell\_icon\_toget\_priceAlert\_signup\_popup\_istriggered |
| TC21\_Verify\_that\_the\_share\_btn\_is\_operational |