**MOCHA FRAMEWORK**

* In my project we have used **JavaScript** as programming language
* And the Framework was **Mocha Framework** with **BDD** approach
* Mocha is a testing framework for Java Script running on node.js
* This framework helped us to handle the asynchronous behavior of java script concept

Since it default uses **async** and **await** keyword

That means functionality of mocha framework is synchronized with async and await keyword

* And it provides describe and it blocks for creating of testScripts

Hook of mocha

* Before
* After
* before Each
* afterEach
* And we have also used webdriverIO framework, it provides some inbuilt libraries for testing

Libraries like

* Node\_modules
* Test
* Pageobjects
* Specs
* Package-lock.json
* Package.json
* Wdio.config.js
* **Node\_modules** contains all the executable file which are used for execute the testScripts
* **Test** contains all the POM classes in **Pageobjects,** And all the testScripts are written in **Specs**
* **Package-lock.json** holdsdetailed information on all the dependencies installed based on the package
* **Package.json** defines the rules required to run your application and install dependencies
* **Wdio.config.js** perfect place to define custom commands
* runner :local
* specs :Is used to run the script based on relative path(Batch Execution)
* suite: this will help us for (group execution)
* capabilities: we can specify browser and its instance(cross browser execution)
* services: we can provide browser driver here
* Base Url:we can specify the baseurl here
* Bail: If bail is (1), it will stop the execution if one test case fail
* connectionRetryCount: 3 (it will retry)
* reporters: we use allure report (these will have screenshot attached also)

Hooks – are nothing but we can interfere with test scripts

1. onprepare : it will execute first (before worker) we can specify service
2. onworkerstart: it will execute (After worker)
3. onworkerEnd: Just before initializing the webdriver session
4. beforeSession: it will execute before test execution happen
5. **before:** **here we makes expect as global for chai assertion usage**
6. AfterSession: it will execute while shut Downing
7. beforeCommand: it will execute before suite
8. afterCommand: it will execute after test
9. beforeSuite it will execute before test
10. AfterSuite :it will execute after command
11. beforeTest it will execute before hook start
12. **afterTest: here we write code for taking screenshot**
13. beforeHook it will execute before Suite Start
14. AfterHook: it will execute After Suite Start
15. onComplete: t will execute after refresh
16. onReload:

Dependencies used in framework

* mocha framework dependency
* selenium standalone dependency
* @wdio/allure-reporter. dependency
* Execljs dependency
* chai dependency

Working principle

* Since the mocha framework is giving describe and it block

(We use testsuitename in describe block)(We use testScriptname in it block)

* First it block will execute which contain test script code
* Since the browser and browser driver is specified in capabilities and service config respectively, it will open the specific browser,
* And URL is taking from Base Url from config file and end point from the excel file (in order to fetch the data from excel file we use exceljs dependency in package.json file)
* While executing the testscript it will take data from the excel file with the help of excel.js dependency and performs action on GUI application by calling the necessary methods present in POM classes present in Page Object folder
* Likewise it will execute all the other testcases which are present in it block

Once after execution it will automatically close the browser

* Since the execution is monitor by the Jenkins (CD/CI tool) it provides three execution platforms.

1. On **Demand**
2. Build **Periodically**
3. Poll **SCM** (Source Control Management)

Apart from that Jenkins will also facilitate the pipeline execution

* Jenkins will execute all the test cases by fetching data from **GITHUB**
* Since we have used allure report, report will be generated we can see passed testcases, failed testcases, skipped testcases etc..,
* Since we have written code for taking screenshot in afterTest hook in config file, it will take the screenshot and attached it to allure reports.