Name : Sakshi Mohan Kamble PRN:2020BTEIT00205

Course:Agile software tools and practices

Assignment 6

# Use cypress tool for QA/ automation testing, compare cypress tool with selenium tool

Cypress Tools

[Cypress framework](https://www.browserstack.com/guide/cypress-framework-tutorial) is a NodeJS-based modern automation tool that supports JavaScript/Typescript as the programing language. There are many advantages and disadvantages of Cypress. However, despite its recency, it is the most popular and easy-to-use tool in the automation world, quickly becoming a favorite of devs and testers.

Understanding Folder Structures in Cypress

By default, Cypress comes with a folder structure. The main folder is **cypress,** within which there are subfolders.

* **Integration:** This folder contains the actual test scripts.
* **Fixtures:** If you are using external data inside your tests, your data can be organized inside the Fixtures folder.
* **Plugins:** The Plugins folder contains the special files that can execute the code before the project loads. If your project needs pre-processors, include them in this folder and configure them accordingly. By default, the plugins folder contains the index.js file, which can be customized to create your tasks.
* **Support:** The Support folder contains utilities, global commands, frequently used codes, etc. By default, this folder comes with two files

– **commands.js** and **index.js**. Additional files and folders can be added as required.

* **Assets**: A folder called **downloads** will be created after the test run, including screenshots, videos, etc.

Writing your First Test Case for Cypress Automation The **cypressdemo** folder contains

* 1. node\_modules folder
  2. cypress folder
  3. cypress.json file
  4. package.json file
  5. package-lock.json file.

To create your tests, navigate to **cypress/integration** and create a fresh new folder (eg: **demo**).

Inside the demo folder, create the test file (ex: **firsttest.js**) using the code below:

//firsttest.js

describe('My First Test', () => {

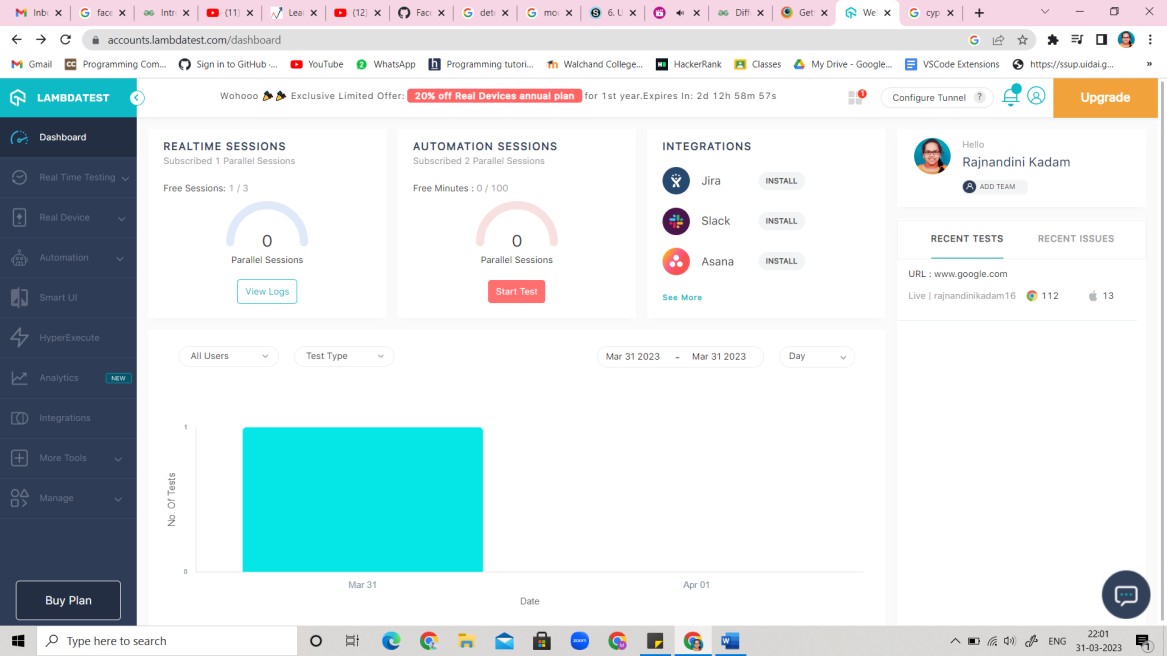
it('Launch Browser and Navigate', () => {

cy.visit('https:/[/www.bro](http://www.browserstack.com/%27)%3B)w[serstack.com/');](http://www.browserstack.com/%27)%3B)

cy.title().should('eq', 'Most Reliable App & Cross Browser Testing Platform | BrowserStack')

})

})



# Difference between Cypress and Selenium :

|  |  |  |
| --- | --- | --- |
| **S.No**  **.** | **Cypress** | **Selenium** |
| 1. | This is a framework that is robust in nature. | It is a library that needs a unit testing framework to process. |
| 2. | To setup Cypress is easier as compared to Selenium. | Its setup is quite complex. |
| 3. | It supports Js only. | It supports different programming languages like java, python. |
| 4. | It supports only one browser i.e. Chrome. | It can support various browsers. |
| 5. | To perform tasks, it uses DOM manipulation techniques. | It does not use DOM manipulation techniques. |
| 6. | It is robust in nature. | It is not robust in nature. |
| 7. | There is no support for multiple tabs. | There is support for multiple tabs. |
| 8. | Limited support for OS. | Many OS can be used to process tasks. |

Reference:

* <https://www.browserstack.com/guide/cypress-automation-tutorial>
* <https://www.geeksforgeeks.org/difference-between-cypress-and-selenium/>