**Cypress Automation**

# Day 1:

**What is Cypress?**

· Cypress is a next generation front end Automation testing tool built for the modern web applications

**Cypress Architecture.**

Most testing tools (like Selenium) operate by running outside of the browser and executing remote commands across the network. But Cypress engine directly operates inside the browser. In other words, It is the browser that is executing your test code

This enables Cypress to listen and modify the browser behavior at run time by manipulating DOM and altering Network requests and responses on the fly

Cypress open doors to New Kind of testing with Having ultimate control over your application (front and back)

Cypress Browser Support:  
Chrome  
Electron  
Firefox & IE

Cypress Components:

Test Runner  
Dash Board Service

**How Cypress is Unique from other Automation tools?**

Cypress automatically waits for commands and assertions before moving on. No more async hell.

Ability to test Edge Testcases by Mocking the server response

Cypress takes snapshots as your tests run. We can hover over each command in the Command Log to see exactly what happened at each step.

Because of its Architectural design, Cypress delivers fast, consistent and reliable test execution compared to other Automation tools

View videos of your entire tests execution when run from the Cypress Dashboard.

Understand the basic frontend development and testing.

1. HTML to define the content of web pages
2. CSS to specify the layout of web pages
3. JavaScript/TypeScript to program the behavior of web pages

Understand web testing concepts

1. E2E
2. Unit
3. Integration

Understand purpose of testing framework

//tagname [attribute = ‘value’]

# Day 2:

Steps to install and setup a cypress

* Installing Node.Js

For install Node JS 🡪 Follow the steps in the docs

* Set up folder for working
* Install Cypress via npm (Node Package Manager)

npm install cypress –save-dev

* Open a cypress for 1st time

npx cypress open

Understanding the cypress dashboard

1. Understand the cypress test runner interface
2. Understand the folder structure
   * Cypress/e2e 🡪 Contains test files
   * Cypress/fixture 🡪 Store the test data (JSON)
   * Cypress/support 🡪 Holds reusable command and global settings

Write a first test in Cypress

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

it('Visits a web page', () => {

cy.visit('https://example.com');

cy.contains('Example Domain').should('be.visible');

});

});

Save the file in cypress/e2e and run it in test run

# Day 3:

Exploring the cypress Commands

Get comfortable with basic commands

1. Navigation

cy.visit(), cy.go(), cy.reload()

1. Interacting with elements

cy.get(), cy.contains(), cy.type(), cy.click(), cy.select()

1. Assertion

.should(), .and()

1. Debugging

cy.debug(), .log()