**Cypress**

**How will cypress run in a system, if the project is open in a new system rather than the one it got implemented?**

**Ans. Cypress** is an end-to-end testing framework for web test automation. Cypress runs test cases directly inside the browser.

To run Cypress in a system, firstly we need to install Node.js, once we install node we can install cypress with command **npm install cypress**. npm stands for non package manager.

Once we install the cypress, we can open cypress using **npx cypress open** command and execute the tests by clicking on test file name.npx stands for **non package executor**

First time when we execute **npm cypress open** command, cypress prepares the framework ready for you. This involves creating folder structure such as cypress, support, plugin, integration etc. This happens only when we execute **npx cypress open** command for the first time, subsequent execution of this command only launches the cypress window since set up is done already.

**If the project is open in new system rather than the one it got implemented: -->**

To run the Cypress project on another system, you need to install Node.js, after installing the node then we have to install Cypress on the system by using the commands npm i.e **npm install cypress**. After installing these, we can run the project on our system by using the command i.e **npx cypress open**.

If the project is not running then there is an issue of commands or other.

When we run **npx cypress open** command in terminal. Then it opens the cypress and creates the cypress folder in the project. In the cypress folder, there are 4 packages:

1. Fixture
2. Integration
3. Plugins
4. Support

**Fixture**: We use fixtures for storing the test data into the json file and to avoid hard code.

**Integration**: This contains our all-test cases that we created.

**Plugins**: we put custom plugin code in this folder. It has its own file i.e index.js

**Support**:

**Fixture**: In the fixture file we store our test data to avoid the hardcode.

Syntex:

1. **before(function() {**
2. **cy.fixture("user").then(function(data){**
3. **this.data=data;**
4. **})**

**Page Object Model:** We can use POM in cypress. POM is used to avoid duplication in code and to improve the test cases maintenance.

**Q.1 cypress run --headed**

Cypress runs the tests in Electron heedlessly by default.

When you pass –headed, it will force Electron to be shown. This will match how you run Electron via cypress open.

Q.2. **cypress run --headless**

Cypress will run the tests in Chrome and Firefox headed by default. Whenever you pass the --headless option, it will force the browser to be hidden.

**Q.3 Allure report:**

"**scripts**": {

**"cy:clientLogin"**: "npx cypress open && cypress run --spec \"cypress/integration/tests/POM.js\" --reporter mocha-allure-reporter", // Open test runner and run test case

**"report:allure"**: "allure generate allure-results --clean -o allure-report && allure open allure-report", // generate test case report and open

**"test:allure"**: "npm run cy:clientLogin || npm run report:allure" // just a name in which we can execute both above commands

},

for executing this we can use npm run test: **npm run test:allure** // command use to execute test case and generate reports

**Q.4. Difference between npx cypress open and headed**.

--- By using headed, we can directly open the default browser and runner and in npx cypress open it can open test runner in which we can run test cases on multiple browsers.

**Q.5 What is Iframe?**

Ans. Iframe stands for Inline frame.Inline Frame is an HTML document embedded inside another HTML document on a website. It is used to insert content from other sources, such as an advertisement, into a Web page.

Cypress can't access iframes directly.

**Q.6 Why we need?**

Iframes are used to load external content such as videos or images easily to websites. When embed or load an iframe, this is loaded separately from basic HTML document markup(index.html).

**<iframe**

**src="https://opensource-demo.orangehrmlive.com/"**

**width="1000"**

**height="1200"**

**data-cy="test-iframe"**

**>**

**</iframe>**

The advantage of using iframes is that users will still be able to see external content on the website even if the rest of the web content has not loaded yet. It's also a great way to keep users in the website directly because they don't have to visit other websites to view the external content.

Disadvantages of using iframes include issues around security and it can sometimes slow down your page if the iframe content is slow to load.

for handling this in chrome we can use security handler in cypress.json:

**{**

**"chromewebsecurity": false**

**}**

**index.html : put the website link in the iframe :**

**<iframe**

**src="https://opensource-demo.orangehrmlive.com/"**

**width="1000"**

**height="1200"**

**data-cy="test-iframe">**

**</iframe>**

**Q.7. How to handle iframe in cypress:**

**cy.get('iframe').then($iframe => {**

**const $body = $iframe.contents().find('body')[1]**

**cy.wrap($body)**

**.find('#id')**

**.clear()**

**.type("anything")**

**cy.route()** to manage the behavior of network requests**.**

**cy.server()** responses to **cy.route()** and to change the behavior of network requests.

**Intercept:** Spy and stub network requests and responses.

**cy.intercept(URL)**

**cy.intercept are:**

-- Reply to a request with a static JSON file defined in the fixtures/ folder (i.e., we can mock the response)

-- Handle the request ourselves

-- Change the request's statusCode to simulate a server failure

-- Simulate a network failure

-- Simulate a delay in the request

If for some reason you want to intercept any JavaScript method, so whenever this method is called,

you want to inject a piece of code to be executed either before or after it.

Interceptors are code blocks that you can use to preprocess or post-process HTTP calls,

helping with global error handling, authentication, logging, and more.

There are two types of events for which you may want to intercept HTTP calls, request and response events.

The request interceptor should be executed before the actual HTTP request is sent,

whereas the response interceptor should be executed before it reaches the application code that made the call.

**TFORM: TFORM is a web-application i.e., Works as a data center together the information of devices such as windows,linux,centos,ios.(only admin has access to this website who can perform actions)**