Automation Testing

E2E Testing with Cypress Phase-End Project Problem Statement

Phase-End Project

Automate and Test Angular Web Application

Project Agenda: To perform E2E testing on a web application using the test automation tool Cypress

Description: As part of this project, go ahead and deploy the below-given Angular application on your local:
https://github.com/Simplilearn-Edu/ATE_PEP1_Testing_Using_Cypress.git

This application is based on the most popular Angular dashboard template (ngx-admin) and uses the Nebular module set.
Once the application is deployed, we will use Cypress to perform E2E testing.
You will install Cypress and start the Test Runner to perform testing.
You will compile the cypress.config.js file and create the Cypress project structure.
In the Cypress folder, create an E2E testing project structure.
Validate various forms, layouts, modals, overlays, and other components on the application.
Use different locators to test elements like textboxes, buttons, checkboxes, and radio buttons on the application.
Use Cypress assertions to validate the code wherever required. Add Cypress wait to the code for synchronization.

Tools Required: Nodejs v14.8.0, Cypress, and Visual Studio Code

Hint: Since the lab comes with node v17 by default, use NVM manager to change the version to 14.8.0; for more information, see the project's README.md file

Set up Scenario:

- Use Visual Studio and clone the repository: **https://github.com/Simplilearn-Edu/ATE_PEP1_Testing_Using_Cypress.git**
- Compile the code repository and run the code using npm
- The application will be available at localhost: 4200
- In the same folder, install Cypress and start the Cypress test runner
- Update the cypress.config.js file with details of specPattern and baseURL
- Create a file with the name Test.spec.js and describe your first project

Detailed Scenario 1:

- Create test cases to open the website on localhost: 4200
- In the same test, write code to click on forms and the forms' layout
- Go to the code of the application ngx-admin in vscode, and edit the attributes of Sign In button, add an attribute and value as data-cy="signInButton"
- The application will deploy automatically
- Inspect the button using the new attribute and write Cypress code for the Button element
- Under the horizontal form, inspect the email and type the text. Inspect the password, type the text, and click on the checkbox
- Use Cypress Assertion to validate if the form contains a button with the name Sign in

Detailed Scenario 2:

- Create a test case to open the website on localhost:4200. Under the form layout, write code to find all three radio buttons
- Inspect the first radio button, check if it is enabled or not, and then click on it
- Now, inspect radio button 2, check if it is enabled or not, and then click on it
- Validate that radio button 1 should not be selected. Use cypress assertion
- Validate whether radio button 3 is disabled or not. If radio button 3 is disabled, then use Cypress Assertion

Detailed Scenario 3:

- Create a test case to open the website on localhost: 4200
- Write test code to navigate to modals and overlays, then click on them
- Inspect the toaster and click on it in the application
- There are three checkboxes under this. Use the check method to select the checkboxes
- Again, inspect all three checkboxes and click on only the second checkbox using the click() method