1. build pj folder and start

-------------------------------------------

mkdir Register-selenium-cucumber

cd Register-selenium-cucumber

npm init -y

-------------------------------------------

2. install all use library

-------------------------------------------

npm install selenium-webdriver cucumber cucumber-html-reporter chai

npm install lru-cache glob@latest uuid@latest @cucumber/cucumber-expressions @cucumber/cucumber @cucumber/gherkin

npm audit fix --force

-------------------------------------------

3.build importance folder

-------------------------------------------

mkdir features

mkdir features\step\_definitions

mkdir features\support

-------------------------------------------

4.Create the feature file: Create a file named registration.feature

in the features directory with the following content:

-------------------------------------------

Feature: User Registration

Scenario: Successful user registration

Given I am on the registration page

When I fill in the registration form

And I submit the registration form

Then I should see a success message

-------------------------------------------

5.Create step definitions: Create a file named registrationSteps.js

in the features\step\_definitions directory with the following content:

-------------------------------------------

const { Given, When, Then, setDefaultTimeout } = require('@cucumber/cucumber');

const { Builder, By, until } = require('selenium-webdriver');

// Set default timeout for steps

setDefaultTimeout(60000);

let driver;

// Step definition: Given I am on the registration page

Given('I am on the registration page', async function () {

// Initialize Selenium WebDriver

driver = await new Builder().forBrowser('firefox').build();

// Navigate to the registration page URL

await driver.get('http://example.com/register');

});

// Step definition: When I fill in the registration form

When('I fill in the registration form', async function () {

// Find form elements and fill them with test data

await driver.findElement(By.id('username')).sendKeys('testuser');

await driver.findElement(By.id('email')).sendKeys('testuser@example.com');

await driver.findElement(By.id('password')).sendKeys('password123');

await driver.findElement(By.id('confirmPassword')).sendKeys('password123');

});

// Step definition: When I submit the registration form

When('I submit the registration form', async function () {

// Find and click on the registration button

await driver.findElement(By.id('register')).click();

});

// Step definition: Then I should see a success message

Then('I should see a success message', async function () {

// Wait for the success message element to appear

let successMessage = await driver.wait(

until.elementLocated(By.id('successMessage')),

10000

);

// Get the text of the success message

let messageText = await successMessage.getText();

// Use chai assertion to check if the message matches the expected text

expect(messageText).to.equal('Registration successful!');

// Close the WebDriver session

await driver.quit();

});

-------------------------------------------

6.Create a Cucumber configuration file: Create a file named cucumber.js

in the root of your project with the following content:

-------------------------------------------

module.exports = {

default: `--require features/\*\*/\*.js --format progress --format json:reports/cucumber\_report.json`

};

-------------------------------------------

7.Generate HTML report: Create a file named generateReport.js

in the root of your project with the following content:

-------------------------------------------

const reporter = require('cucumber-html-reporter');

const options = {

theme: 'bootstrap',

jsonFile: 'reports/cucumber\_report.json',

output: 'reports/cucumber\_report.html',

reportSuiteAsScenarios: true,

launchReport: true,

metadata: {

"App Version": "0.1.0",

"Test Environment": "STAGING",

"Browser": "Firefox",

"Platform": "Windows 10",

"Parallel": "Scenarios",

"Executed": "Remote"

}

};

reporter.generate(options);

-------------------------------------------

8.Run the tests:

-------------------------------------------

npx cucumber-js

-------------------------------------------

9.Generate the report:

-------------------------------------------

node generateReport.js

-------------------------------------------