Український державний університет імені Михайла Драгоманова

Факультет математики, інформатики та фізики

Звіт про виконану роботу до лабораторного заняття № 6

*з курсу*

*Автоматизація тестування*

***Виконав/ла:*** студентка групи41ІПЗ

спеціальності «Інженерія програмного

забезпечення»

***Антосевич К.О.***

***Викладач:*** Кархут В.Я.

1. Створіть 6 тестів для тестування стартової сторінки Google. Налаштуйте очукування для двох з них так щоб тести пройшли з негативним результатом. 2. Додайте jest-html-reporter до проекту та сформулюйте базовий html репорт для даного набору тестів. 3. Створіть розширену версію репорт у де: a. Тести посортовані відповідно до часу виконання b. Будуть включені консоль логи c. Будуть включені розширені описи помилок отриманих в результаті тестів 4. Підготуйте репорт формату xml з допомогою jest-junit бібліотеки для тестів з пункту 1. 5. Підготуйте репорт з діаграмами з допомогою jest-stare бібліотеки для тестів з пункту 1. 6. Підготуйте репорт з діаграмами з допомогою jest-allure бібліотеки для тестів з пункту 1. Додаткове завдання 1: Знайдіть та налаштуйте бібліотеку яка включатиме в репорт скріншоти створені під час виконання даних тестів

Json

"scripts": {

"test": "jest",

"report": "jest --reporters=default --reporters=jest-html-reporter --reporters=jest-stare --reporters=jest-allure-reporter"

},

"jest": {

"testEnvironment": "node",

"reporters": [

"default",

["jest-junit", { "outputDirectory": "./reports", "outputName": "junit.xml" }]

]

}

Js

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

const chrome = require('selenium-webdriver/chrome');

const path = require('chromedriver').path;

const service = new chrome.ServiceBuilder(path).build();

chrome.setDefaultService(service);

describe('Google Page Tests', () => {

let driver;

beforeAll(async () => {

driver = new Builder().forBrowser('chrome').build();

await driver.get('https://www.google.com');

}, 30000);

afterAll(async () => {

await driver.quit();

}, 15000);

test('Page title is correct', async () => {

const title = await driver.getTitle();

expect(title).toEqual('Google');

});

test('Search box is present', async () => {

const searchBox = await driver.findElement(By.name('q'));

expect(searchBox).toBeDefined();

});

test('Google Search button is present', async () => {

const searchButton = await driver.findElement(By.name('btnK'));

expect(searchButton).toBeDefined();

});

test('I\'m Feeling Lucky button is present', async () => {

const luckyButton = await driver.findElement(By.name('btnI'));

expect(luckyButton).toBeDefined();

});

// Негативні тести

test('Page title is incorrect', async () => {

const title = await driver.getTitle();

expect(title).toEqual('Not Google');

});

test('Non-existent element is not found', async () => {

await driver.wait(until.elementLocated(By.id('non-existent')), 5000);

}, 10000);

});

Bash

npm run report

додаткове завдання

Цей код виконує базове тестування стартової сторінки Google, генерує декілька видів репортів, та демонструє як додавати скріншоти до звіту за допомогою Allure.

afterEach(async () => {

let screenshot = await driver.takeScreenshot();

reporter.addAttachment("Screenshot", Buffer.from(screenshot, 'base64'), 'image/png');

});