МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Дисципліна «Автоматизоване тестування»

Лабораторна робота № 5 та 6

Виконав:

студент гр. 6.1213

Явдощен Яків Валерійович

Перевірив:

Викладач

Кудін О.В.

Запоріжжя

2023

Завдання для самостійної роботи

1. Реалізувати скрипт для автоматизації наступних дій Cypress:

* Відкрити сторінку <http://suninjuly.github.io/math.html> .
* Прочитати значення змінної x.
* Обчислити математичну функцію від x.
* Ввести відповідь в текстове поле.
* Вибрати checkbox "I'm the robot".
* Вибрати radiobutton "Robots rule!".
* Натиснути кнопку Submit.

1. Реалізувати скрипт тестування наступного функціонала ресурсу http://demo-store.seleniumacademy.com:

* Пошук довільного товару.
* Тестування роботи корзини.
* Довільний функціонал за вибором студента

3. Ознайомитись з переліком API для тестування <https://gorest.co.in/>

4. Виконати тестування основних методів вибраного API.

Створити github репозиторій з тестами.

**https://github.com/Kallaban/automated-testing**

Контрольні запитання

1. Види тестування: Тестування - це процес перевірки програмного коду з метою виявлення помилок та підтвердження правильності роботи програми. Існує кілька видів тестування, включаючи:

a. Модульне тестування: Тестування окремих модулів, функцій або компонентів програми для перевірки їх правильності та незалежності від інших частин програми.

b. Функціональне тестування: Перевірка, чи виконує програма задані функціональні вимоги і чи працює вона відповідно до специфікацій.

c. Інтеграційне тестування: Перевірка взаємодії між різними модулями або компонентами програми.

d. Приймальне тестування: Тестування, яке виконується клієнтом або користувачем для підтвердження того, що програма відповідає вимогам та вимогам специфікацій.

e. Навантажувальне тестування: Тестування, що перевіряє, як програма працює при різних рівнях навантаження.

2. Особливості модульного та приймального тестування: a. Модульне тестування:

Фокусується на ізоляції окремих модулів, функцій або класів програми.

Тести створюються розробниками для перевірки правильності конкретних частин коду.

Може бути використане для раннього виявлення помилок і їх виправлення.

Зазвичай використовується інструменти, такі як фреймворки для модульного тестування (наприклад, Mocha).

b. Приймальне тестування:

Проводиться з точки зору користувача або клієнта програми.

Ціль - підтвердити відповідність програми вимогам та специфікаціям.

Може включати тестування в реальних умовах або на виробничих серверах.

Зазвичай виконується після інших видів тестування.

3. Призначення бібліотек Mocha та Chai:

Mocha: Mocha - це популярний фреймворк для тестування JavaScript, який використовується для створення та запуску модульних тестів. Він надає зручний інтерфейс для створення та організації тестів, підтримує асинхронні тести, і має багато додаткових плагінів для розширення функціональності. Mocha дозволяє використовувати різні асерти та стилі опису тестів.

Chai: Chai - це бібліотека для тестування, яка дозволяє виразно виражати твердження в тестах на JavaScript. Вона пропонує різні стилі асертів (наприклад, assert, expect, should), які дають можливість виразно описувати очікувані результати в тестах. Chai допомагає зробити тести більш читабельними та легшими для розуміння.