CSE 307

Assignment on

Automated Functional Testing Using Cucumber

Submitted By
1705002 - Sheikh Azizul Hakim
1705045 - Iftekhar Hakim Kaowsar

Submitted To
Professor Dr. Anindya Iqbal
Department of CSE, BUET

July 31, 2021

1 Introduction

Functional Testing is a type of software testing that validates the software system against the functional requirements/specifications. The testing is mainly done as black box, since the underlying code is not much of a concern to the testing processes. We wanted to do an automated web functional testing with a tool named Cucumber [1].

2 Testing Tool

Cucumber is based upon a software development methodology named **Behaviour Driven Development (BDD)**. BDD aims to bridge the gaps between software engineers and their nontechnical clients by producing system documentation that is automatically checked against the system's behaviour. The documentation needs to be written in a medium understandable by both the computer and the nontechnical clients.

Cucumber reads executable specifications written in plain text and validates that the software does what those specifications say. The specifications consists of multiple examples, or scenarios. For example:

```
Background: Going to Login Page
  Given Browser is opened and maximized
  When Go to goodreads login page

Scenario: Checking incorrect login
  When Incorrect credentials placed
  And Error message visible and browser still at login page
  Then End
```

The abovementioned snippet is written in Gherkin, which is a set of grammar tools that makes plain text interpretable by Cucumber. Since the snippet is also self-explanatory even to a non-technical client, Gherkin code serves three purposes at once: living documentation, automated testing and executable specifications. Step definitions connect Gherkin code to programming code. In our case, we performed the background testing with Selenium on Java 15.

Selenium is a set of tools that support browser automation. We used Selenium's webdriver (Chromedriver) to perform the actual testing.

3 Methodology

We performed our testing on `https://www.goodreads.com`. We searched for the relevant HTML tag by inspecting on Chrome. Then we fetched that WebElement using Selenium and performed other relevant operations. Some of the examples we tested include:

1. Correct Login
2. Incorrect Login
3. Proper Logout After Login

[1] `https://cucumber.io`

4. Upload/Update Valid Image File as Profile Picture
5. Upload/Update A Non Image File as Profile Picture
6. Search Operations Without Login
    - Prominent book names (e. g., 1984) should match exactly.
    - Keywords (e. g., Quantum Computing, Software) should appear in the search.

Same Search Operations After Login.

4 Code

Our testing code is publicly available at `https://github.com/IftekharHakimK/AI-Testing-307`.

5 Some shots

```
Scenario Outline: Upload "<file>" as Profile Picture
  When Correct credentials placed
  Then Upload "<file>" as Profile Picture
  Then End
  Examples:
    | file          |
    | BUET_LOGO.png |
    | Flag_Bangladesh.jpg|
```

Figure 1: Parameterized Input to work with several inputs

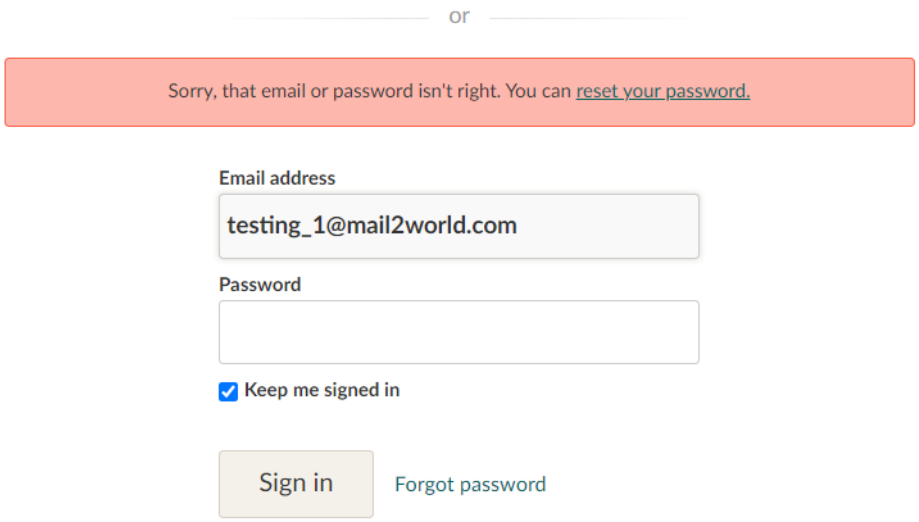


Figure 2: Checking the incoming error message after invalid login

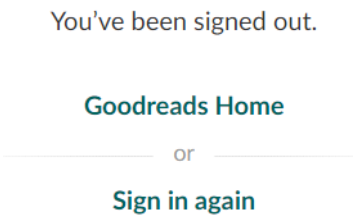
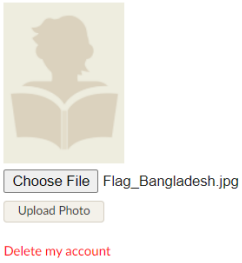
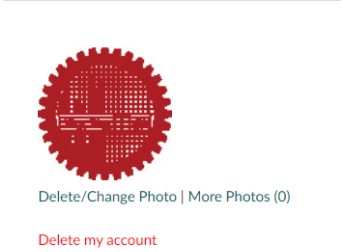


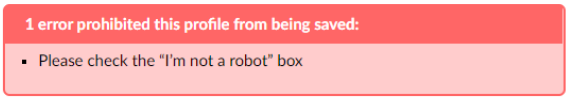
Figure 3: Checking successful logout by reading prompt message



(a) File upload in input box

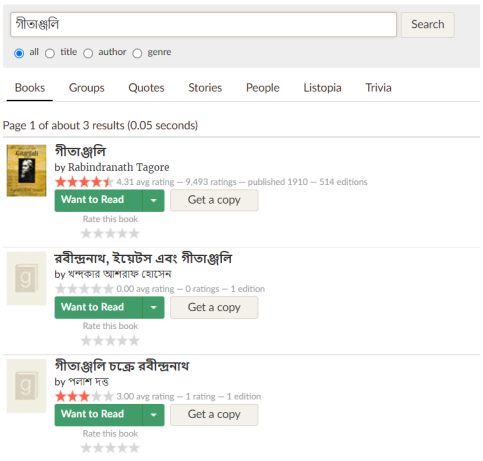


(b) Successful upload of PNG file

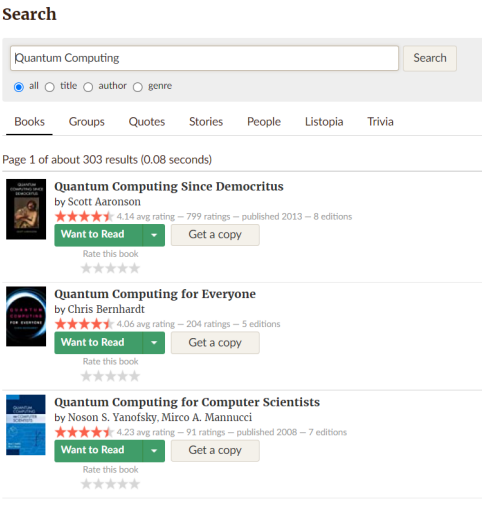


(c) Error message checking after uploading txt file in profile picture

Figure 4: Checking file upload in profile picture editing



(a) Matching exact input with results



(b) Checking keyword in results

Figure 5: Testing search box

6 Discussion

It would be better if we could perform a test on a software developed by our own, so that proper vulnerabilities could be exposed. Testing on a well deployed website as Goodreads, on the other hand, helped us to understand how the automation actually works.

Both the strength and weakness of Cucumber come from Gherkin. Since Gherkin wants to emulate natural language by matching regular expressions, the complex mathematical operations are difficult to be expressed in Gherkin.

Although we used Selenium underneath Cucumber, Cucumber is not limited to one browser automation framework. For example, Cucumber can also do API testing using tools like RestAssured[2].

Using Selenium webdrivers, we cannot download file. Otherwise, we could have downloaded the profile picture to match it with the uploaded one.

[2] `https://rest-assured.io/`