CSE 307 Assignment

Selenium and JUnit based UI Testing of Django Project

Prepared By
Nazmul Takbir - 1705103
Saif Ahmed Khan - 1705110

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology

Contents

List of Figures

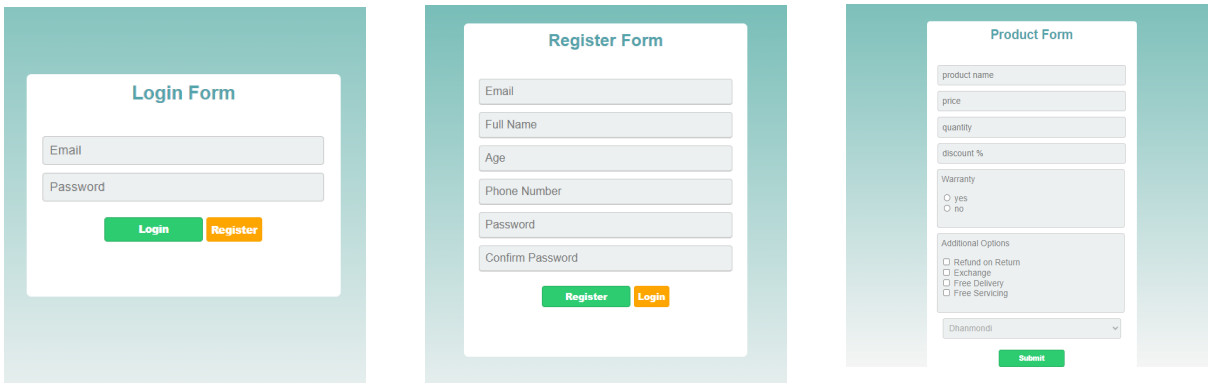
1 Introduction

In our assignment, we used Selenium Webdriver 3.141.59 which is a tool that automates web browsers. It is a web framework that comes in a jar format that allows us to execute cross-browser tests. Selenium WebDriver allows us to choose any programming language to create test scripts that run independently of the source code of the project being tested. We can use it to automate web-based application testing to ensure that the application performs the expected tasks without any hiccups.

2 Goals

In any web application, forms need to be user friendly, that is, it must always prompt correct error messages to the user if wrong data is given in the input. The app would only perform the intended tasks when correct format/type of data is provided in the forms.

Our goal was to test web forms by using Selenium Webdriver to automate both user input and verification of correct response from web app. We constructed our own web app using Django 3.1.7 as the backend and PostgreSQL 10.17 as the database.



(a) Login Form (b) Register Form (c) Product Form

Figure 1: The Forms of our Web App

Our web app has three forms (Fig.4):

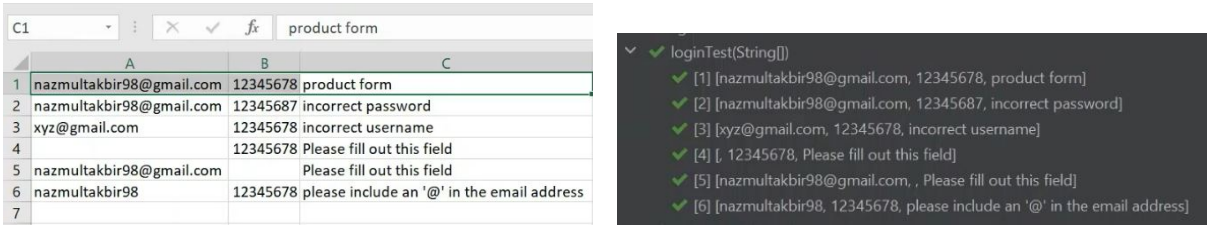
- **Login Page:** the user can login
- **Register Page:** new users can register
- **Product Page:** logged in users can add new product information to the database

We used selenium to automate the process of all three tasks by reading test inputs from given csv files, fill the forms with the data read and match the expected output in the csv file against the application's validation message. Each type of form was tested independently using JUnit 5.

3 Results

Click here to see the detailed demonstration of our project in YouTube. Below we have added pictures showing the test input and test output for all the forms.

Login Page:



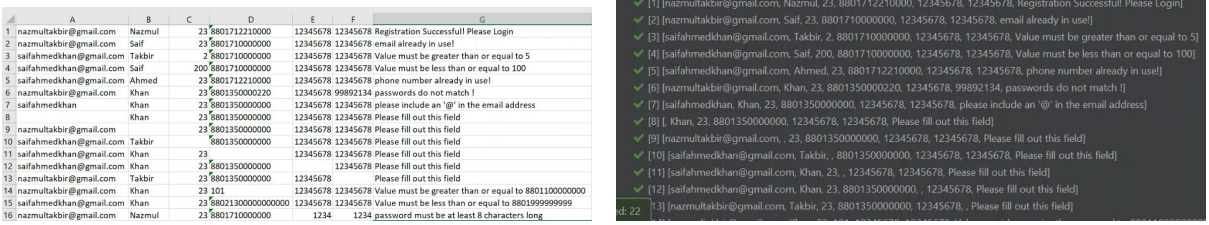
	A	B	C
1	nazmultakbir98@gmail.com	12345678	product form
2	nazmultakbir98@gmail.com	12345687	incorrect password
3	xyz@gmail.com	12345678	incorrect username
4		12345678	Please fill out this field
5	nazmultakbir98@gmail.com		Please fill out this field
6	nazmultakbir98	12345678	please include an '@' in the email address
7			

loginTest(String[])
- [1] [nazmultakbir98@gmail.com, 12345678, product form]
- [2] [nazmultakbir98@gmail.com, 12345687, incorrect password]
- [3] [xyz@gmail.com, 12345678, incorrect username]
- [4] [, 12345678, Please fill out this field]
- [5] [nazmultakbir98@gmail.com, , Please fill out this field]
- [6] [nazmultakbir98, 12345678, please include an '@' in the email address]

(a) Data (b) Result

Figure 2: Test for Login Page

Register Page:



	A	B	C	D	E	F	G
1	nazmultakbir@gmail.com	Nazmul	23	8801712210000	12345678	12345678	Registration Successful! Please Login
2	nazmultakbir@gmail.com	Saif	23	8801710000000	12345678	12345678	email already in use!
3	saifahmedkhan@gmail.com	Takbir	2	8801710000000	12345678	12345678	Value must be greater than or equal to 5
4	saifahmedkhan@gmail.com	Saif	200	8801710000000	12345678	12345678	Value must be less than or equal to 100
5	saifahmedkhan@gmail.com	Ahmed	23	8801712210000	12345678	12345678	phone number already in use!
6	nazmultakbir@gmail.com	Khan	23	8801350000220	12345678	99892134	passwords do not match !
7	saifahmedkhan	Khan	23	8801350000000	12345678	12345678	please include an '@' in the email address
8		Khan	23	8801350000000	12345678	12345678	Please fill out this field
9	nazmultakbir@gmail.com		23	8801350000000	12345678	12345678	Please fill out this field
10	saifahmedkhan@gmail.com	Takbir		8801350000000	12345678	12345678	Please fill out this field
11	saifahmedkhan@gmail.com	Khan	23		12345678	12345678	Please fill out this field
12	saifahmedkhan@gmail.com	Khan	23	8801350000000		12345678	Please fill out this field
13	nazmultakbir@gmail.com	Takbir	23	8801350000000	12345678		Please fill out this field
14	nazmultakbir@gmail.com	Khan	23	101	12345678	12345678	Value must be greater than or equal to 8801100000000
15	saifahmedkhan@gmail.com	Khan	23	88021300000000000	12345678	12345678	Value must be less than or equal to 8801999999999
16	nazmultakbir@gmail.com	Nazmul	23	8801710000000	1234	1234	password must be at least 8 characters long

registerTest(String[])
- [1] [nazmultakbir@gmail.com, Nazmul, 23, 8801712210000, 12345678, 12345678, Registration Successful! Please Login]
- [2] [nazmultakbir@gmail.com, Saif, 23, 8801710000000, 12345678, 12345678, email already in use!]
- [3] [saifahmedkhan@gmail.com, Takbir, 2, 8801710000000, 12345678, 12345678, Value must be greater than or equal to 5]
- [4] [saifahmedkhan@gmail.com, Saif, 200, 8801710000000, 12345678, 12345678, Value must be less than or equal to 100]
- [5] [saifahmedkhan@gmail.com, Ahmed, 23, 8801712210000, 12345678, 12345678, phone number already in use!]
- [6] [nazmultakbir@gmail.com, Khan, 23, 8801350000220, 12345678, 99892134, passwords do not match !]
- [7] [saifahmedkhan, Khan, 23, 8801350000000, 12345678, 12345678, please include an '@' in the email address]
- [8] [, Khan, 23, 8801350000000, 12345678, 12345678, Please fill out this field]
- [9] [nazmultakbir@gmail.com, , 23, 8801350000000, 12345678, 12345678, Please fill out this field]
- [10] [saifahmedkhan@gmail.com, Takbir, , 8801350000000, 12345678, 12345678, Please fill out this field]
- [11] [saifahmedkhan@gmail.com, Khan, 23, , 12345678, 12345678, Please fill out this field]
- [12] [saifahmedkhan@gmail.com, Khan, 23, 8801350000000, , 12345678, Please fill out this field]
- [13] [nazmultakbir@gmail.com, Takbir, 23, 8801350000000, 12345678, , Please fill out this field]

(a) Data (b) Result

Figure 3: Test for Register Page

Product Page:

	A	B	C	D	E	F	G	H
	Chinigura	12.2	10	5	1	110	Banani	Product Added Successfully! Check History for more
	White Bre	12.2	1	5	2	1001	Mohamm	Please Select One of These Options
	White Bre	12.2	1.5	5	0	1011	Dhanmon	Please enter a valid value
	White Bre	-1	1.5	5	1	11	Uttara	Please enter a valid value

productAddTest(String[])
- [1] [Chinigura Rice, 12.2, 10, 5, 1, 0110, Banani, Product Added Successfully! Check History for more]
- [2] [White Bread, 12.2, 1, 5, 2, 1001, Mohammadpur, Please Select One of These Options]
- [3] [White Bread, 12.2, 1.5, 5, 0, 1011, Dhanmondi, Please enter a valid value]
- [4] [White Bread, -1, 1.5, 5, 1, 0011, Uttara, Please enter a valid value]

(a) Login Form (b) Register Form

Figure 4: Test for Product Page

4 Improvements

In our project we tested text field, email field, password field, number field, radio buttons, dropdown menu, checked boxes and button clicks. However, there are many more things that could have been tested such sliders, file submissions, audio/video playbacks, etc.

5 Limitations

Selenium is a greatly way to quickly test an UI for a lot of test cases in s short time. In this project we used this capability of Selenium to test if web forms are user friendly. However, just showing the correct message or performing the correct task is not enough to make a form user friendly. The visual design of the UI plays a big role in making the form user friendly. This cannot be tested using automation tools such as Selenium because it requires human judgement.

6 Resources

- Video Demonstration
- Selenium Project Code
- Django Project Code