

A Project Report On

Software Testing and Quality Assurance
(Mini Project II)

CLASS: BE-2

GUIDED BY
Prof. Snehal Shintre



DEPARTMENT OF COMPUTER ENGINEERING
PUNE INSTITUTE OF COMPUTER TECHNOLOGY
DHANKAWADI, PUNE-43

SAVITRIBAI PHULE PUNE UNIVERSITY
2020-21

Abhishek Saware-41269
Tejas Yadav-41282
Sharad Ugalmugle - 41280

Title:

Create a small web-based application by selecting relevant system environment/platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Narrate scripts in order to perform regression tests. Identify the bugs using Selenium WebDriver and IDE and generate test reports encompassing exploratory testing

Problem Definition:

Perform Web testing and identify the bugs using Selenium WebDriver and IDE and generate test reports encompassing exploratory testing on a self developed web app.

Objective

Perform testing on a blogging site and write test cases.

Test Environment:

An Ubuntu 20.04 environment

Django 2.0

Selenium web-driver

Selenium IDE

Google Chrome

Theory:

Selenium:

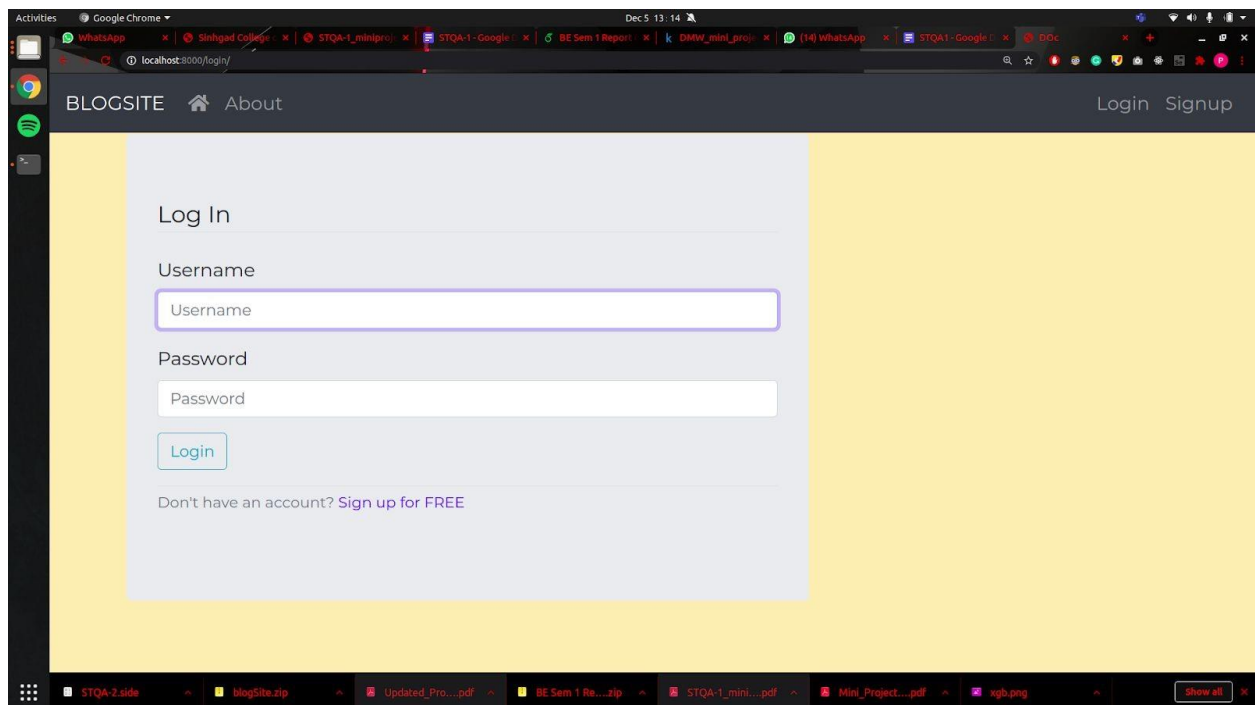
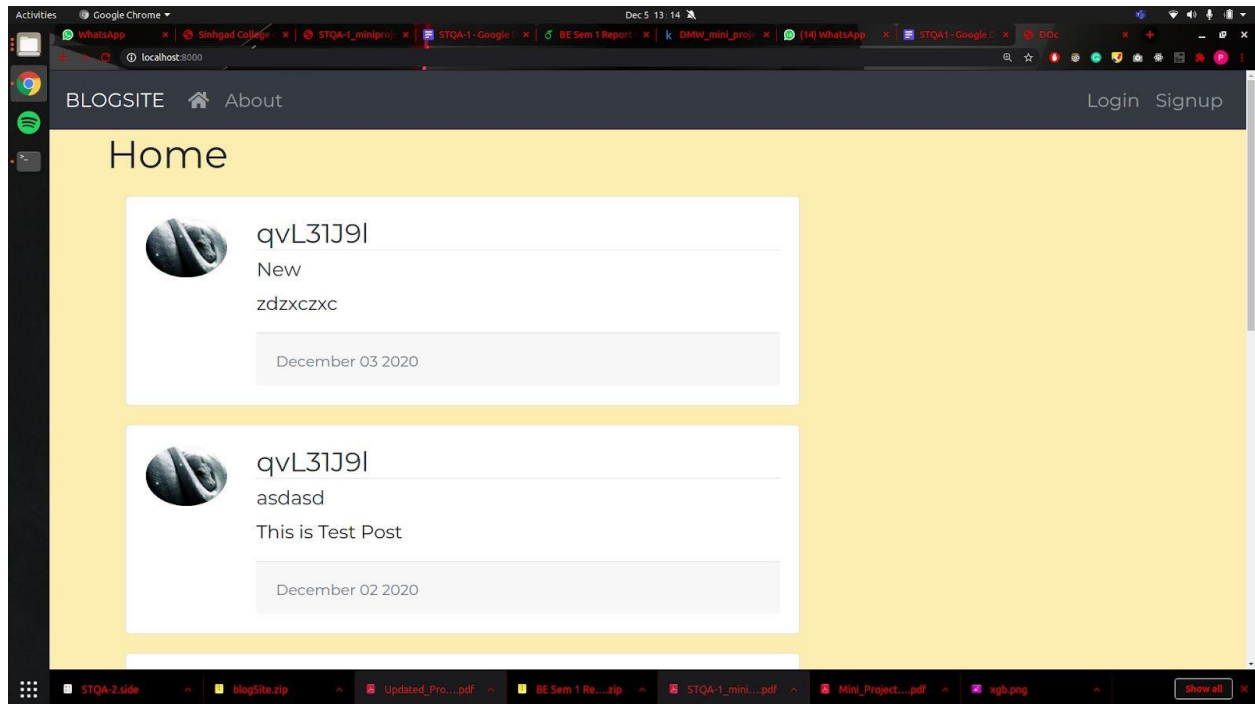
Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. Selenium is a suite of software tools to automate Web Browsers. It is an Open source suite of tools mainly used for Functional and Regression Test Automation. Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. It is quite similar to HP Quick Test Pro (QTP now UFT) only that Selenium focuses on automating web-based applications. Testing done using a Selenium tool is usually referred as Selenium Testing.

Selenium IDE:

Selenium IDE (Integrated Development Environment) is primarily a record/run tool that a test case developer uses to develop Selenium Test cases. Selenium IDE is an easy to use tool from the [Selenium Test Suite](#) and can even be used by someone new to developing automated test cases for their web applications. One does not require any special setup to get started with Selenium IDE. You just need to add the extension of your specific browser. Selenium IDE provides you with a GUI (Graphical User Interface) for easily recording your interactions with the website.

Selenium IDE allows a user or a test case developer to create the test cases and test suites and edit it later as per their requirements. The development environment also provides the capability of converting test cases to different programming languages, which makes it easier for the user and does not mandate the need for knowing a specific programming language.

Sample Screenshots of application



Output logs of sample tests

1. Create Post

The screenshot shows the Selenium IDE interface for a project named 'STQA-2*'. The test 'create_post*' is selected, and its execution is shown in the 'Log' panel. The test is running on the URL 'http://localhost:8000'.

Command	Target	Value
type	id=id_386	addadd
click	id=id_content	
type	id=id_content	This is Test Post
click	css=btn	
assertText	css=row > .col-md-8	test_user_3inaddsdnThis is Test PostnCreated on : December 05 2020 Publish
click	css=.card-body	
click	linkText=Publish	
assertText	css=small:nth-child(1)	Posted on : December 05 2020

The 'Log' panel shows the following steps:

2. click on css=.btn OK
3. click on linkText=Create New Post OK
4. click on id=id_title OK
5. type on id=id_title with value addadd OK
6. click on id=id_content OK
7. type on id=id_content with value This is Test Post OK
8. click on css=btn OK
9. assertText on css=row > .col-md-8 with value test_user_3inaddsdnThis is Test PostnCreated on : December 05 2020 Publish OK
10. click on css=.card-body OK
11. click on linkText=Publish OK
12. assertText on css=small:nth-child(1) with value Posted on : December 05 2020 OK

The test 'create_post' completed successfully.

2. Delete Post

The screenshot shows the Selenium IDE interface for a project named 'STQA-2*'. The test 'delete_post*' is selected, and its execution is shown in the 'Log' panel. The test is running on the URL 'http://localhost:8000'.

Command	Target	Value
open	/	
set window size	1317x741	
click	linkText=test_user_3	
click	css=.card:nth-child(3) h3	
double click	css=.card:nth-child(3) h3	
click	linkText=asdasd	

The 'Log' panel shows the following steps:

1. open on / OK
2. setWindowSize on 1317x741 OK
3. click on linkText=test_user_3 OK
4. click on css=.card:nth-child(3) h3 OK
5. doubleClick on css=.card:nth-child(3) h3 OK
6. click on linkText=asdasd OK
7. click on css=.fa-trash-alt OK
8. click on css=btn-lg OK
9. assertNotText on css=.card:nth-child(3) > .card-body with value test_user_3inaddsdnThis is Test PostnDecember 05 2020 OK

The test 'delete_post' completed successfully.

3. Login

The screenshot shows the Selenium IDE interface for a project named 'STQA-2*'. The 'Tests' panel on the left lists several test cases, with 'valid_credentials1' selected. The main panel displays the configuration for the 'login_test' command, which is part of the 'valid_credentials1' suite. The configuration table is as follows:

Command	Target	Value
1 store	test_user_3	username
2 store	testpwd3	pwd
3 run	login_test	

Below the configuration table, there are input fields for 'Command', 'Target', 'Value', and 'Description'. The 'Log' panel at the bottom shows the execution steps for 'login_test', including opening the browser, setting window size, clicking the login link, and typing the username and password. The log indicates that the test was successful.

4. Signup

The screenshot shows the Selenium IDE interface for the same project 'STQA-2*'. The 'Executing' panel on the left shows the 'signup_test' command. The main panel displays the configuration for the 'signup_test' command, which is part of the 'valid_credentials1' suite. The configuration table is as follows:

Command	Target	Value
1 open	/	
2 set window size	1317x741	
3 click	linkText=Signup	
4 type	id=td_username	test_user_3
5 click	id=td_email	
6 type	id=td_email	test_user_3@email.com
7 click	id=td_firstname	
8 type	id=td_firstname	test_user3_firstname
9 click	id=td_lastname	

Below the configuration table, there are input fields for 'Command', 'Target', 'Value', and 'Description'. The 'Log' panel at the bottom shows the execution steps for 'signup_test', including opening the browser, setting window size, clicking the signup link, and typing the email, first name, and last name. The log indicates that the test was successful.

Source code/ Functions of the application

Create Post

Generated by Selenium IDE

```
import pytest
```

```
import time
```

```
import json
```

```
from selenium import webdriver
```

```
from selenium.webdriver.common.by import By
```

```
from selenium.webdriver.common.action_chains import ActionChains
```

```
from selenium.webdriver.support import expected_conditions
```

```
from selenium.webdriver.support.wait import WebDriverWait
```

```
from selenium.webdriver.common.keys import Keys
```

```
from selenium.webdriver.common.desired_capabilities import  
DesiredCapabilities
```

```
class TestCreatepost():
```

```
    def setup_method(self, method):
```

```
        self.driver = webdriver.Chrome()
```

```
        self.vars = { }
```

```
    def teardown_method(self, method):
```

```
        self.driver.quit()
```

```
    def test_createpost(self):
```

```
        self.driver.get("http://localhost:8000/")
```

```
        self.driver.find_element(By.CSS_SELECTOR, ".btn").click()
```

```
        self.driver.find_element(By.LINK_TEXT, "Create New Post").click()
```

```
        self.driver.find_element(By.ID, "id_title").click()
```

```
        self.driver.find_element(By.ID, "id_title").send_keys("asdasd")
```

```

self.driver.find_element(By.ID, "id_content").click()

self.driver.find_element(By.ID, "id_content").send_keys("This is Test
Post")

self.driver.find_element(By.CSS_SELECTOR, ".btn").click()

assert self.driver.find_element(By.CSS_SELECTOR, ".row >
.col-md-8").text == "test_user_3\\\\nasdasd\\\\nThis is Test
Post\\\\nCreated on : December 05 2020 Publish"

self.driver.find_element(By.CSS_SELECTOR, ".card-body").click()

self.driver.find_element(By.LINK_TEXT, "Publish").click()

assert self.driver.find_element(By.CSS_SELECTOR,
"small:nth-child(1)").text == "Posted on : December 05 2020"

```

Login

Generated by Selenium IDE

```

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action_chains import ActionChains

from selenium.webdriver.support import expected_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.common.desired_capabilities import
DesiredCapabilities

class TestValidcredentials1():

    def setup_method(self, method):

```



```

self.driver = webdriver.Chrome()
self.vars = {}
def teardown_method(self, method):
    self.driver.quit()
def logintest(self):
    self.driver.get("http://localhost:8000/")
    self.driver.set_window_size(1317, 741)
    self.driver.find_element(By.LINK_TEXT, "Login").click()
    self.driver.find_element(By.ID, "id_username").click()
    self.driver.find_element(By.ID,
"id_username").send_keys(self.vars["username"])
    self.driver.find_element(By.ID, "id_password").click()
    self.driver.find_element(By.ID,
"id_password").send_keys(self.vars["pwd"])
    self.driver.find_element(By.CSS_SELECTOR, ".btn").click()
    [object Object]
def test_validcredentials1(self):
    self.vars["username"] = "test_user_3"
    self.vars["pwd"] = "testpwd3"
    self.logintest()

```

Signup

Generated by Selenium IDE

```
import pytest
```

```
import time
```

```
import json
```

```
from selenium import webdriver
```

```
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import
DesiredCapabilities
class TestSignuptest():
    def setup_method(self, method):
        self.driver = webdriver.Chrome()
        self.vars = { }
    def teardown_method(self, method):
        self.driver.quit()
    def test_signuptest(self):
        self.driver.get("http://localhost:8000/")
        self.driver.set_window_size(1317, 741)
        self.driver.find_element(By.LINK_TEXT, "Signup").click()
        self.driver.find_element(By.ID,
"id_username").send_keys("test_user_3")
        self.driver.find_element(By.ID, "id_email").click()
        self.driver.find_element(By.ID,
"id_email").send_keys("test_user_3@email.com")
        self.driver.find_element(By.ID, "id_firstname").click()
        self.driver.find_element(By.ID,
"id_firstname").send_keys("test_user3_fname")
        self.driver.find_element(By.ID, "id_lastname").click()
```

```
self.driver.find_element(By.ID,
"id_lastname").send_keys("test_user3_lname")
self.driver.find_element(By.ID, "id_password1").click()
self.driver.find_element(By.ID,
"id_password1").send_keys("testpwd3")
self.driver.find_element(By.ID, "id_password2").click()
self.driver.find_element(By.ID,
"id_password2").send_keys("testpwd3")
self.driver.find_element(By.CSS_SELECTOR, ".btn").click()
assert self.driver.find_element(By.CSS_SELECTOR, ".row >
.col-md-8").text == "Account created for test_user_3 !. You can now
login.\\\\nLog In\\\\nUsername\\\\nPassword\\\\nDon't have an
account? Sign up for FREE"
```

Conclusion :

Performed automation testing on a self developed blogging site and verified that no bugs or defects were found.