A Project Report ​ On​

**Software Testing and Quality Assurance**

**(Mini Project)**

SUBMITTED BY

|  |  |
| --- | --- |
| **Pranav Rathi** | **Roll No: 41265** |
| **Samarsinh Salunkhe** | **Roll No: 41267** |
| **Shivendu Shukre** | **Roll No: 41273** |

**CLASS: BE-2**

GUIDED BY

# Prof. Snehal Shintre



**DEPARTMENT OF COMPUTER ENGINEERING**

PUNE INSTITUTE OF COMPUTER TECHNOLOGY

DHANKAWADI, PUNE-43

SAVITRIBAI PHULE PUNE UNIVERSITY

2022-23

**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.
* Understand the use of Selenium WebDriver and IDE to generate test reports.

**Test Environment:**

64-bit OS

Django 3.0

Selenium web-driver

Selenium IDE

Google Chrome

**Theory:**

**Selenium:**

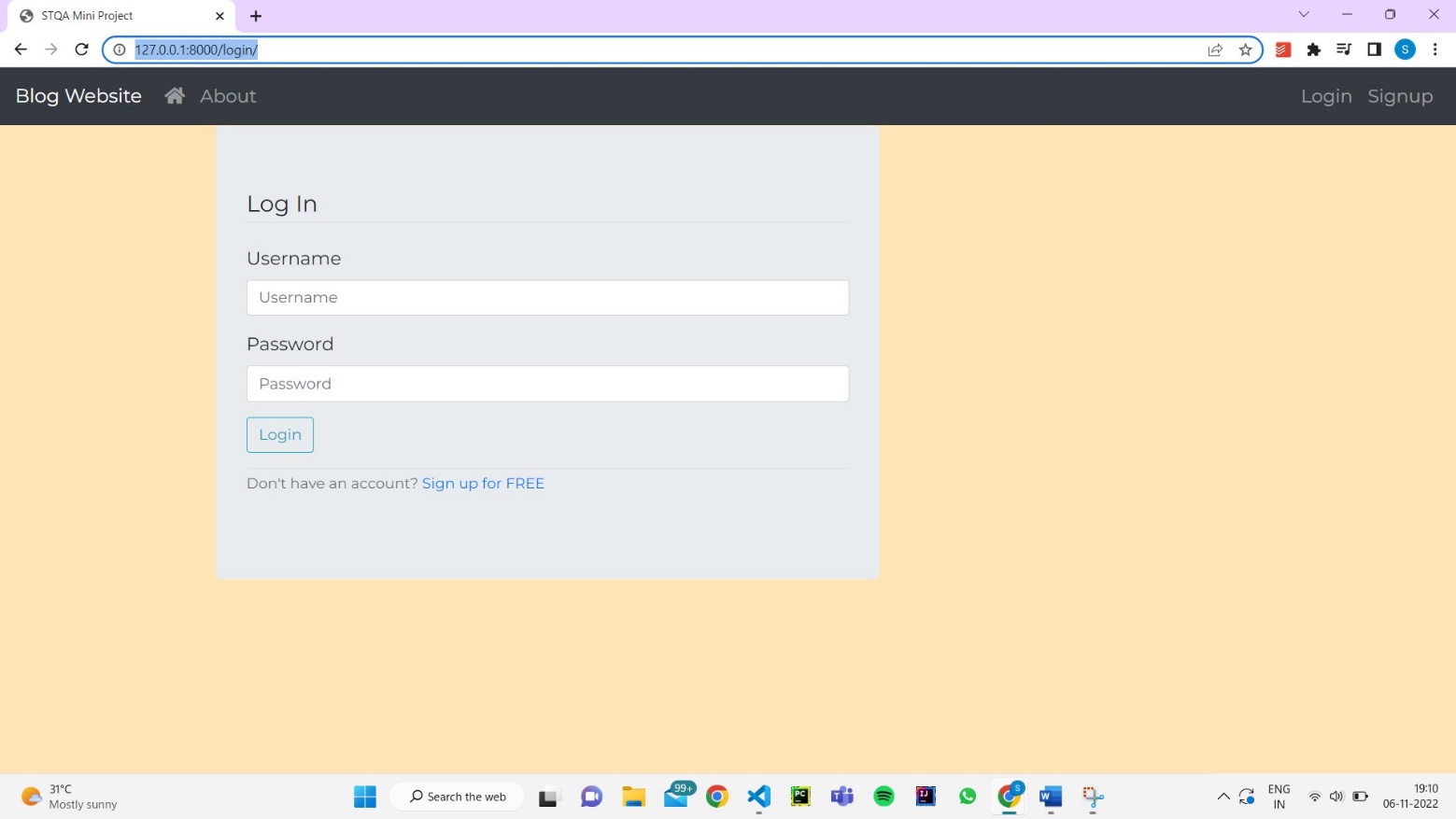
Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. Selenium is a suite of software tools to automate Web Browsers. It is an Open-source suite of tools mainly used for Functional and Regression Test Automation. Selenium is a free (open 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.

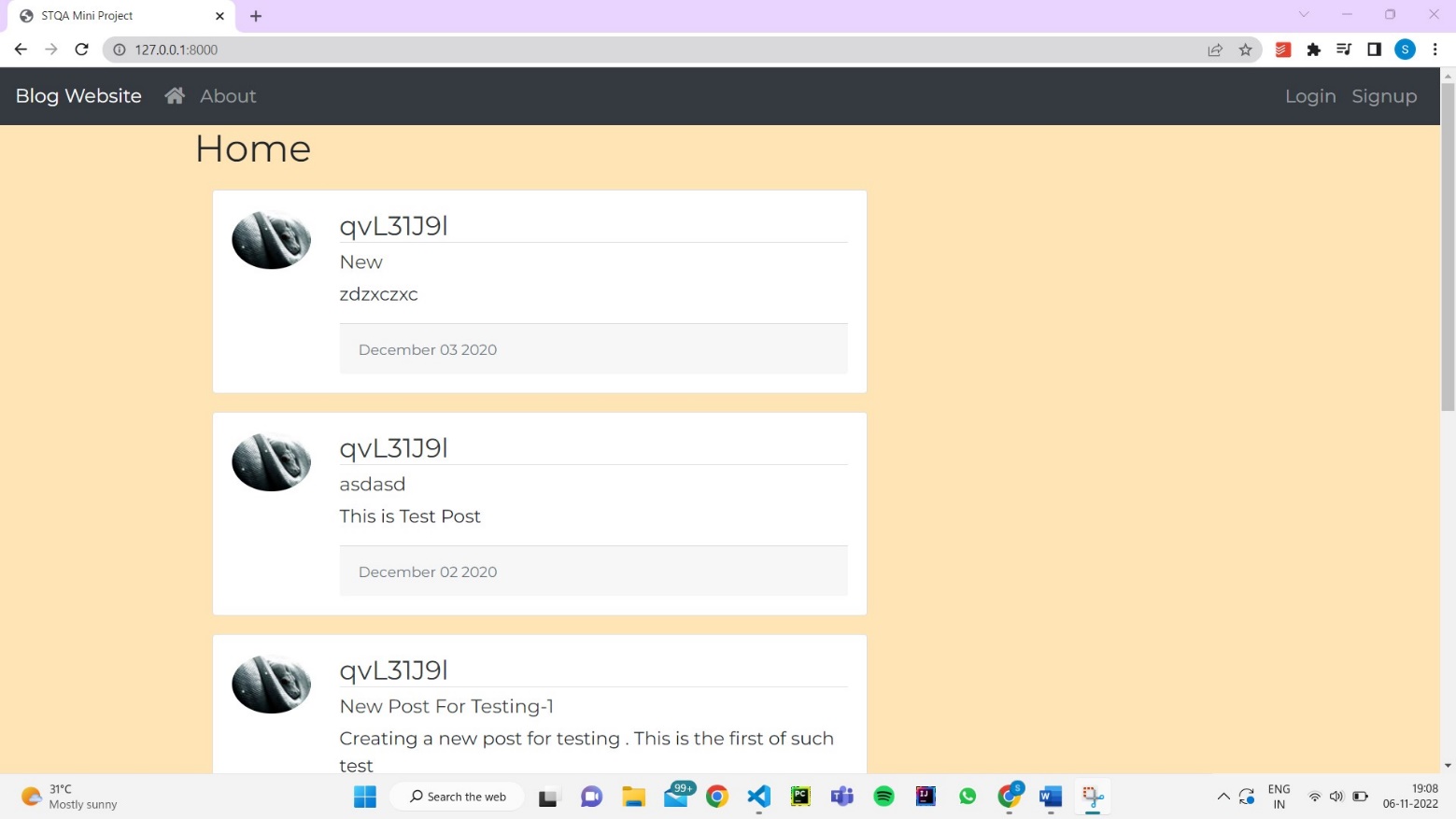
**Selenium IDE:**

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 Suit​[e](https://www.browserstack.com/selenium) 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.

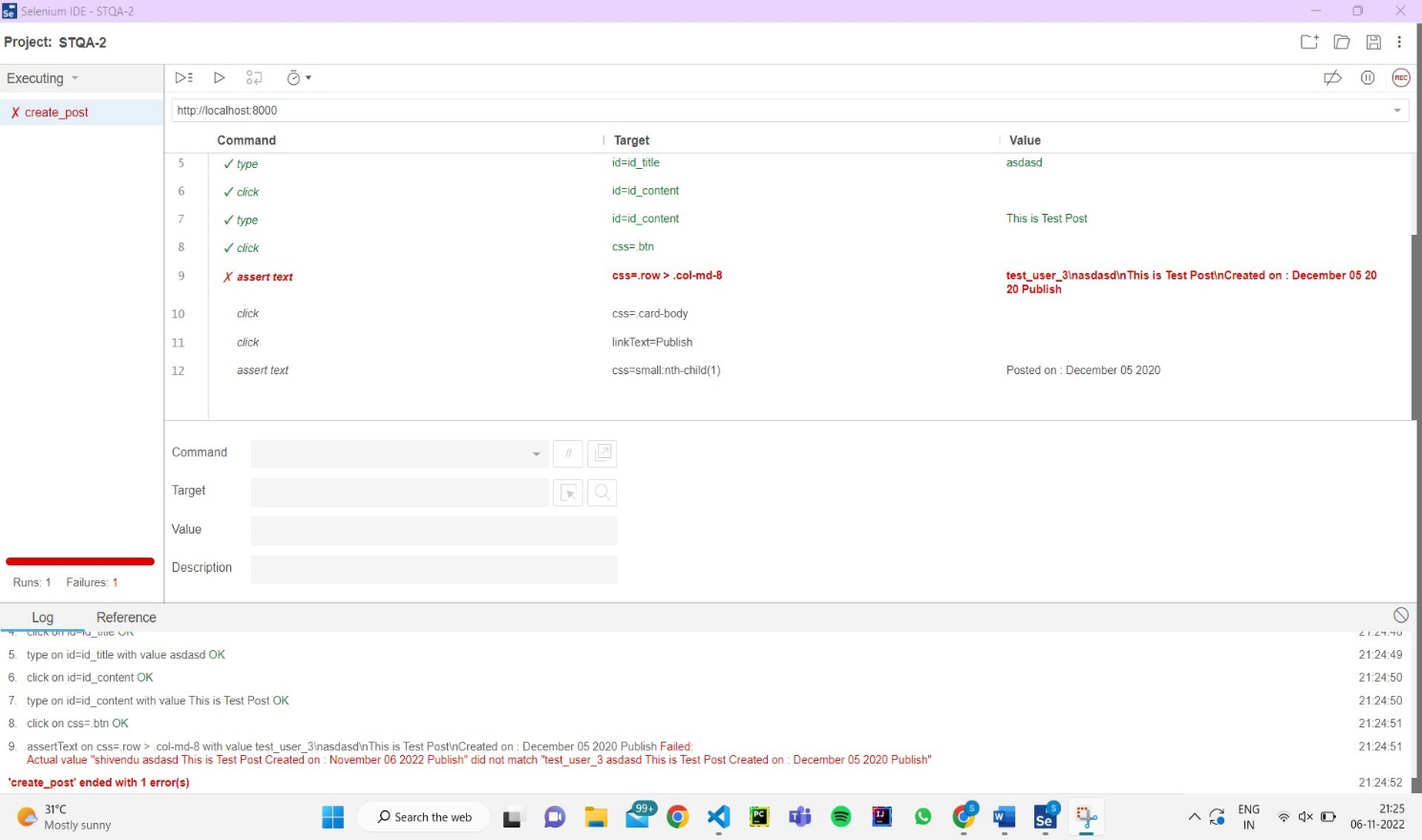
# Sample Screenshots of application



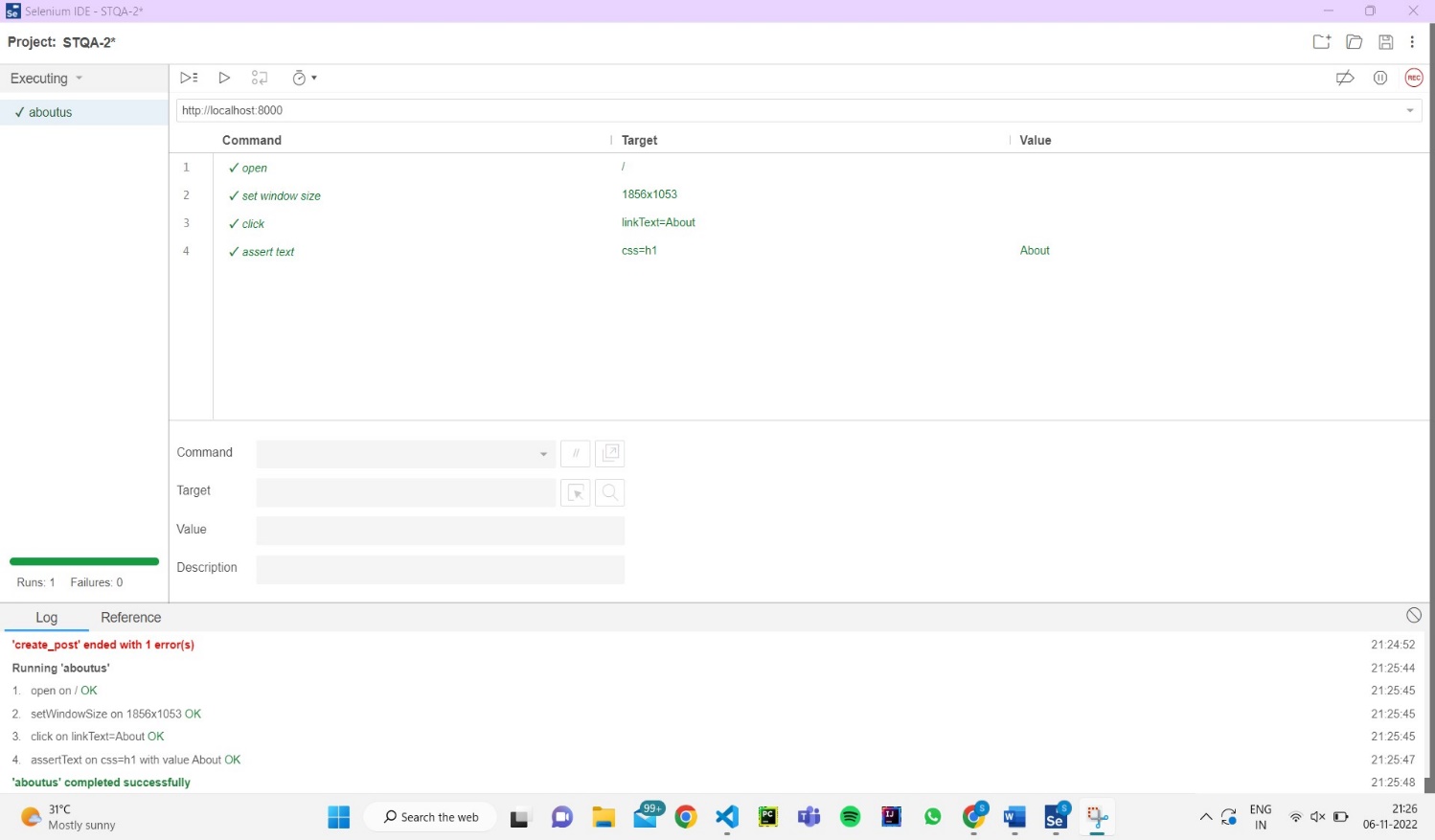


# Output logs of sample tests

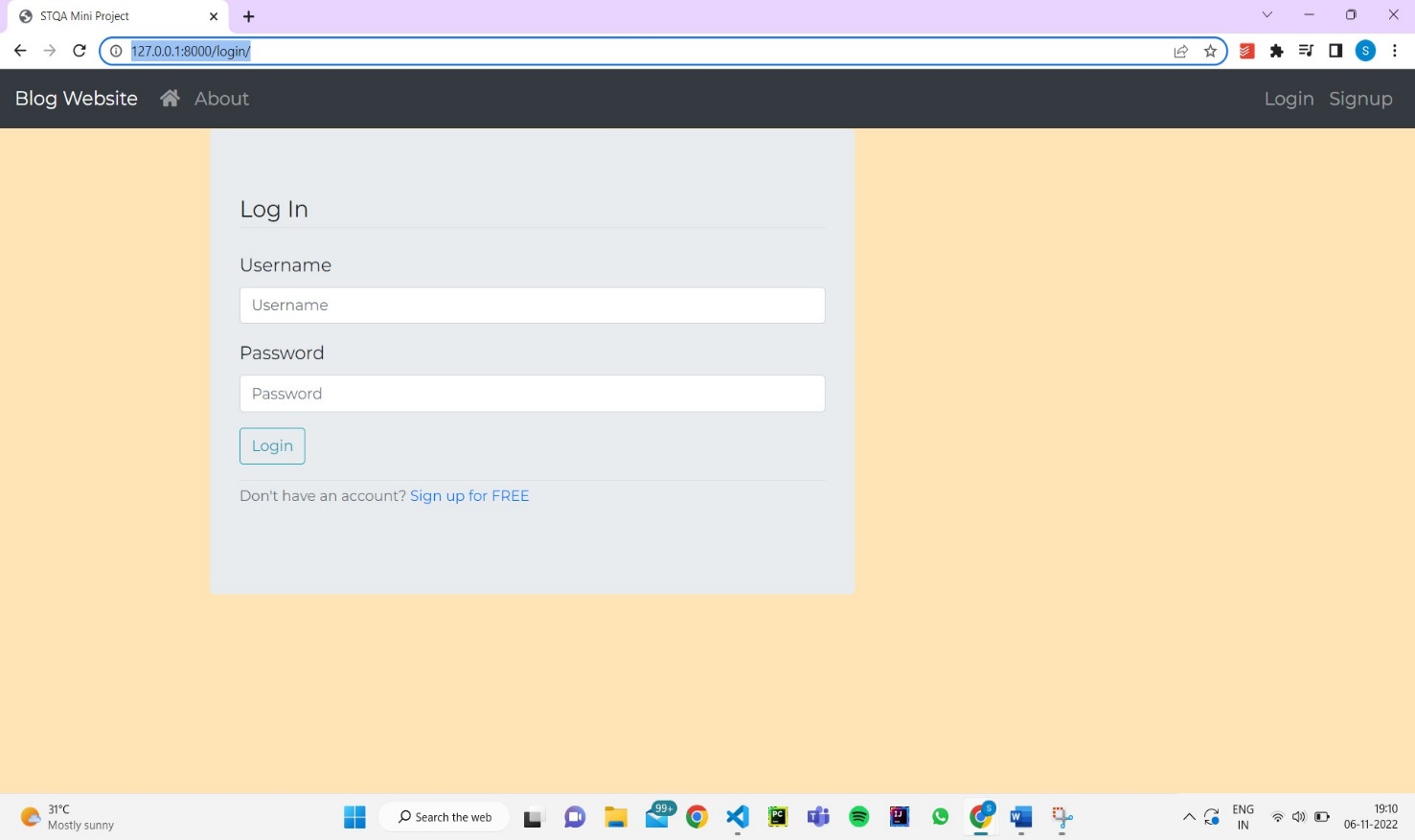
1. Create Post



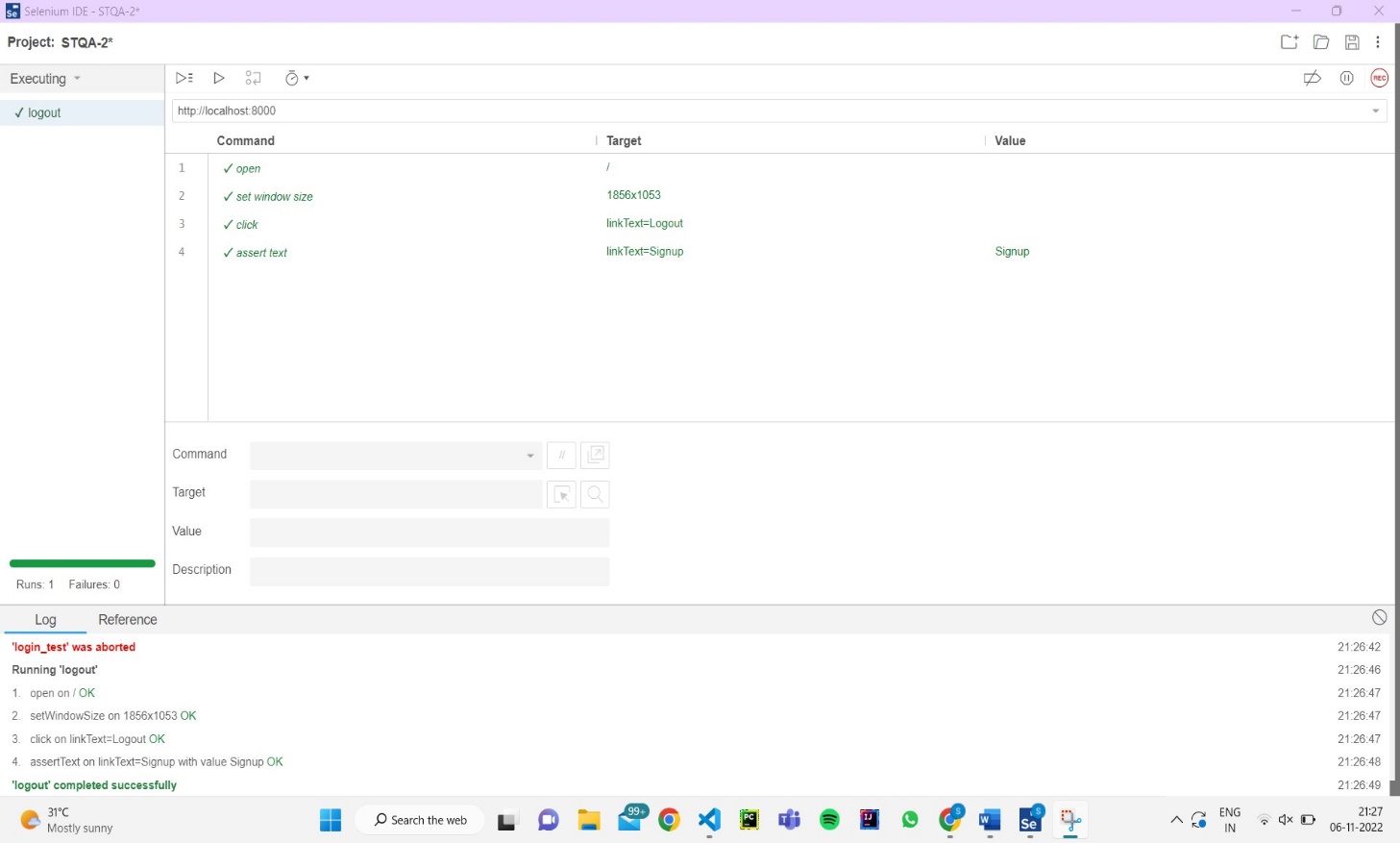
1. About Us



1. Login



1. Logout



# 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.