A PROJECT REPORT ON

Software Testing and Quality Assurance (Mini Project II)

SUBMITTED BY

Aditya Jadhav	**Roll No: 41332**
Karan Kangude	**Roll No: 41340**
Kaustubh Odak	**Roll No: 41341**

CLASS: BE-3

GUIDED BY
Prof. S. S. Suaradkar

DEPARTMENT OF COMPUTER ENGINEERING

PUNE INSTITUTE OF COMPUTER TECHNOLOGY

DHANKAWADI, PUNE – 43

SAVITRIBAI PHULE PUNE UNIVERSITY
2021 -2022

Title:

Create a small web-based application by selecting relevant system environments/platforms and programming languages. Narrate concise Test Plan consisting of 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 deployed web app.

Objective:

Perform testing on a blogging site and write test cases

Test Environment:

- 64-bit Linux/Windows OS
- Node.js 14 LTS or above
- NPM and Yarn package managers
- Selenium WebDriver and IDE
- Firefox or Chromium-based browser

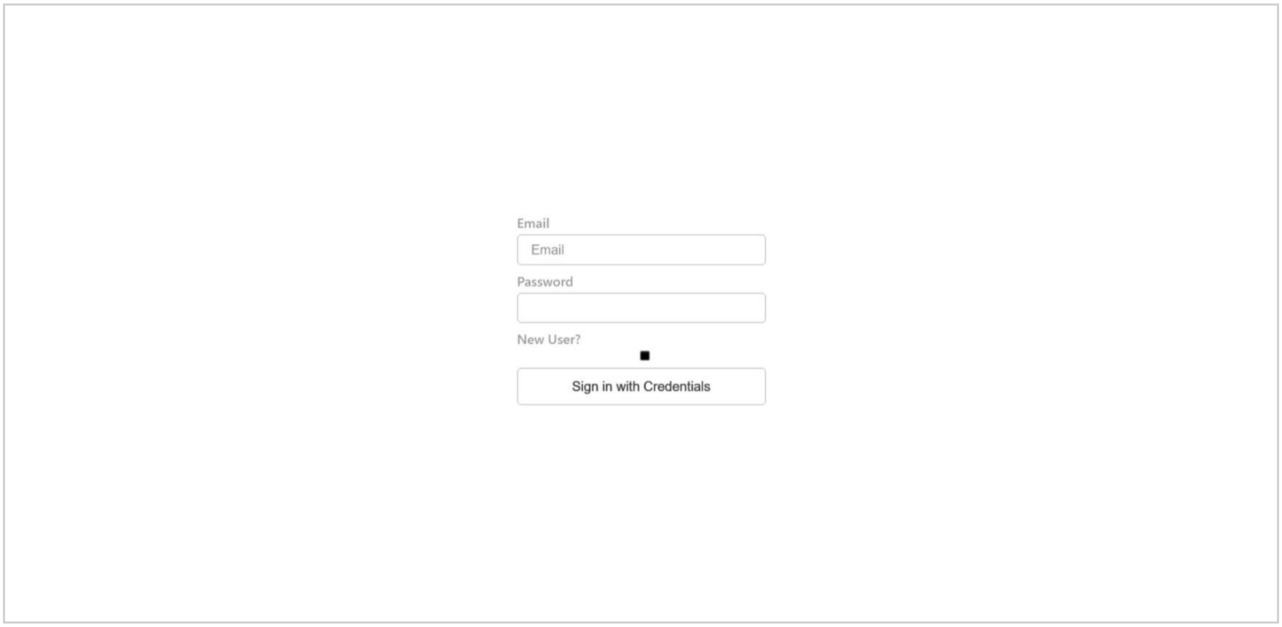
Theory:

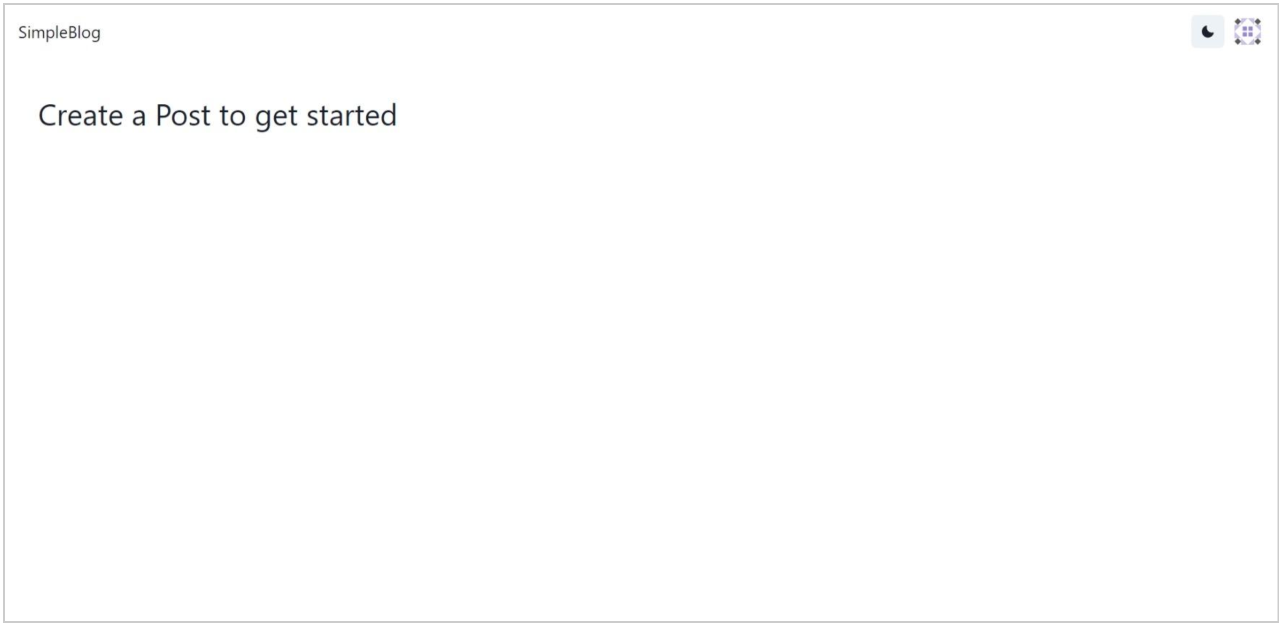
Selenium:
Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. Selenium is a suite of software tools to automate web browsers. It is an open source suite of tools mainly used for functional and regression test automation. It is quite similar 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 to 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.

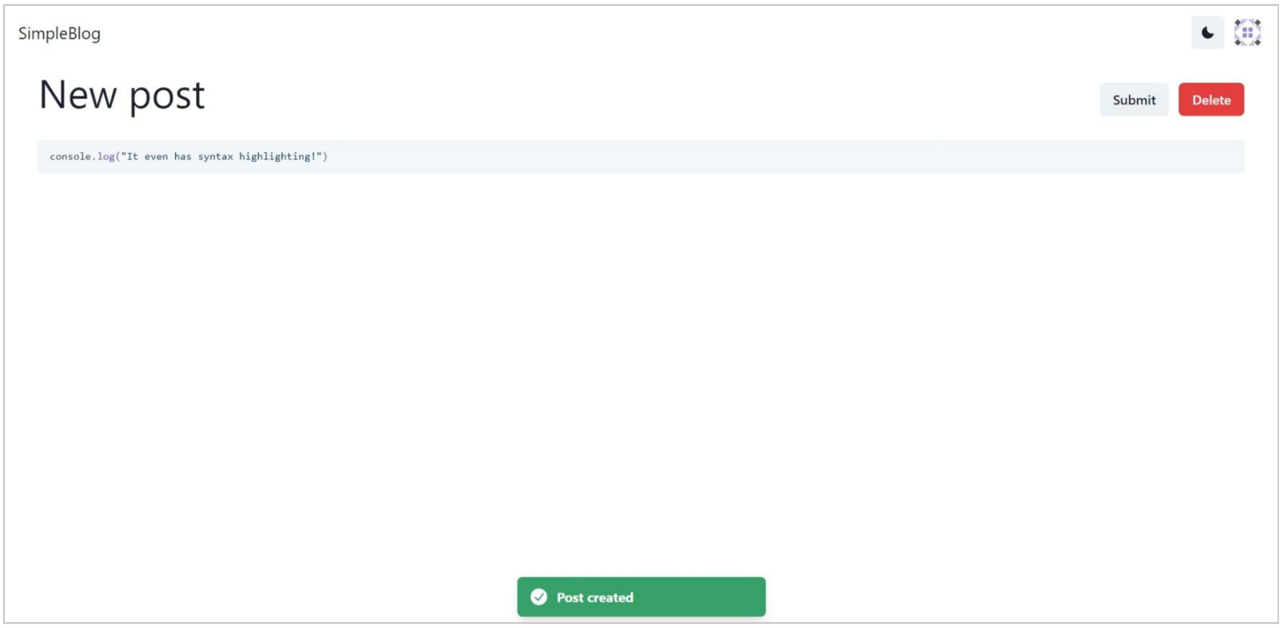
Selenium IDE allows a user or a test case developer to create the test cases and test suites and edit it later as per their requirements. The development environment also provides the capability of converting test cases to different programming languages, which makes it easier for the user and does not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

Screenshots of the application

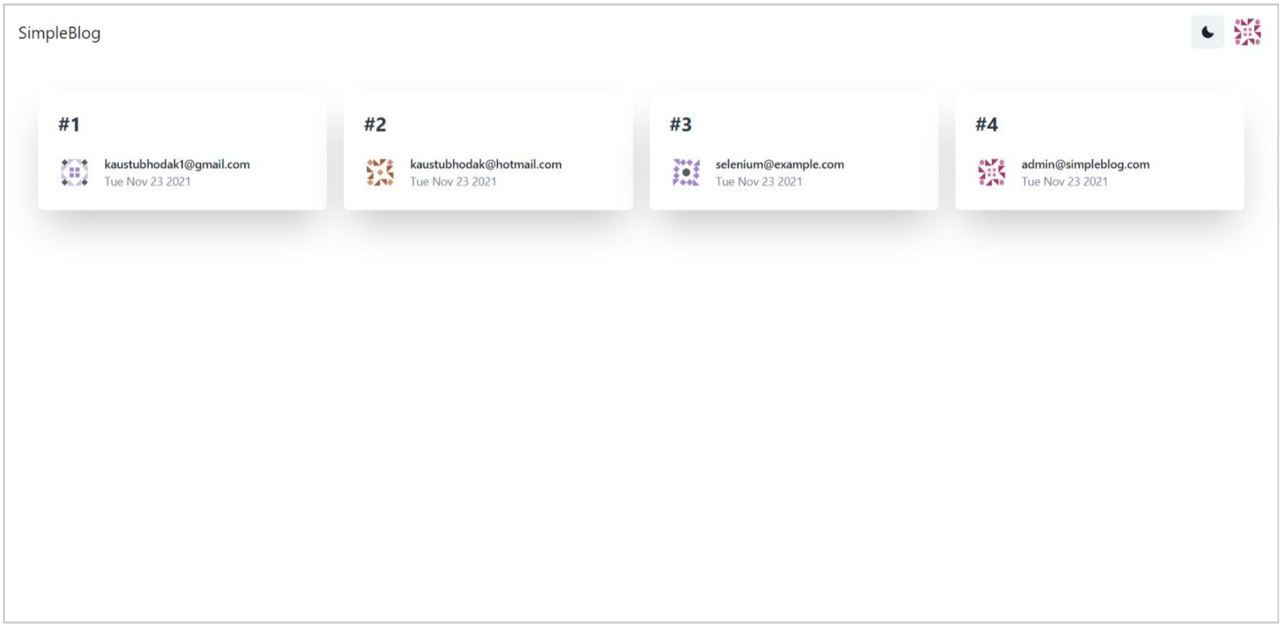




Create a Post to get started



```
console.log("It even has syntax highlighting!")
```



#1
kaustubhodak1@gmail.com
Tue Nov 23 2021

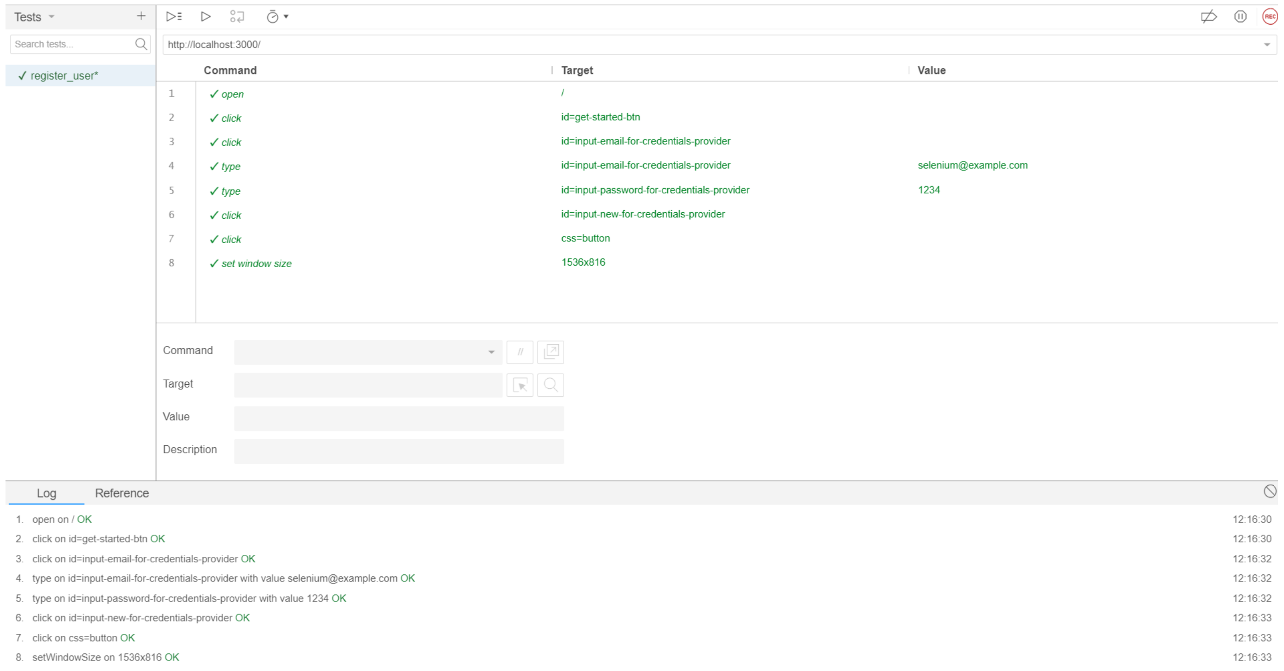
#2
kaustubhodak@hotmail.com
Tue Nov 23 2021

#3
selenium@example.com
Tue Nov 23 2021

#4
admin@simpleblog.com
Tue Nov 23 2021

Output logs:

1. Register user



http://localhost:3000/

	Command	Target	Value
1	✓ open	/	
2	✓ click	id=get-started-btn	
3	✓ click	id=input-email-for-credentials-provider	
4	✓ type	id=input-email-for-credentials-provider	selenium@example.com
5	✓ type	id=input-password-for-credentials-provider	1234
6	✓ click	id=input-new-for-credentials-provider	
7	✓ click	css=button	
8	✓ set window size	1536x816	

Command

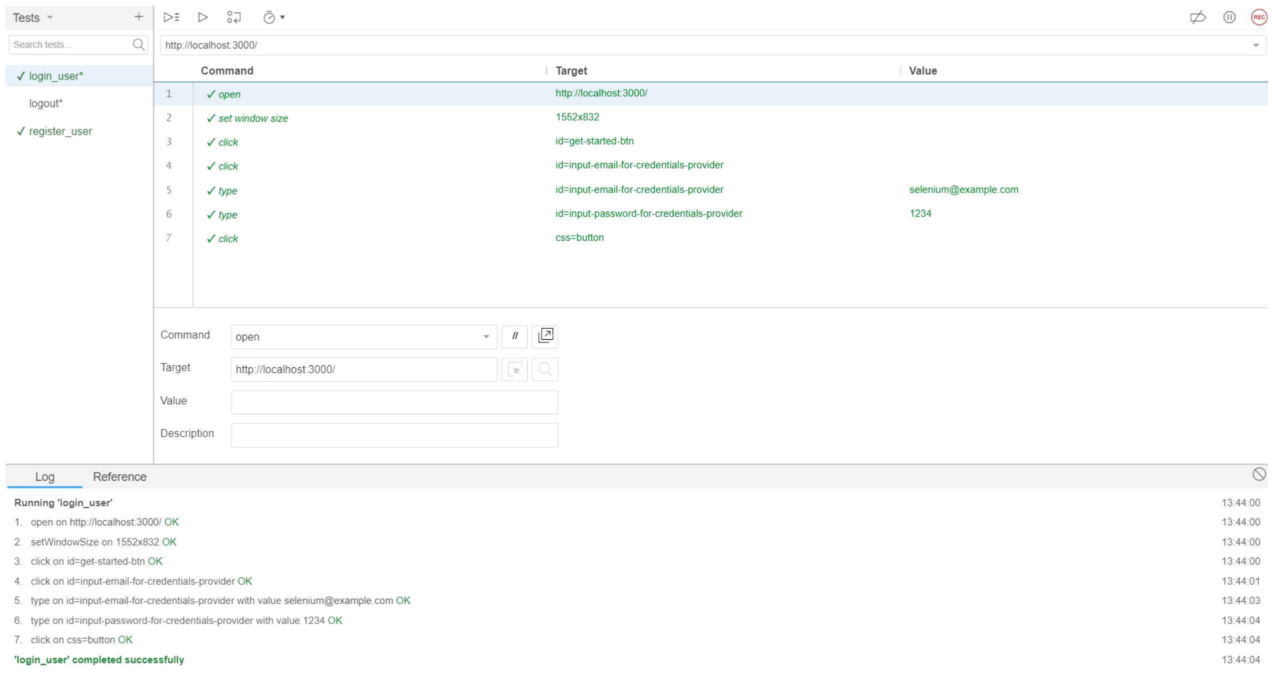
Target

Value

Description

```
1. open on / OK                                                                                   12:16:30
2. click on id=get-started-btn OK                                                                 12:16:30
3. click on id=input-email-for-credentials-provider OK                                            12:16:32
4. type on id=input-email-for-credentials-provider with value selenium@example.com OK             12:16:32
5. type on id=input-password-for-credentials-provider with value 1234 OK                          12:16:32
6. click on id=input-new-for-credentials-provider OK                                              12:16:33
7. click on css=button OK                                                                         12:16:33
8. setWindowSize on 1536x816 OK                                                                   12:16:33
```

2. Login user



http://localhost:3000/

	Command	Target	Value
1	✓ open	http://localhost:3000/	
2	✓ set window size	1552x832	
3	✓ click	id=get-started-btn	
4	✓ click	id=input-email-for-credentials-provider	
5	✓ type	id=input-email-for-credentials-provider	selenium@example.com
6	✓ type	id=input-password-for-credentials-provider	1234
7	✓ click	css=button	

Command: open

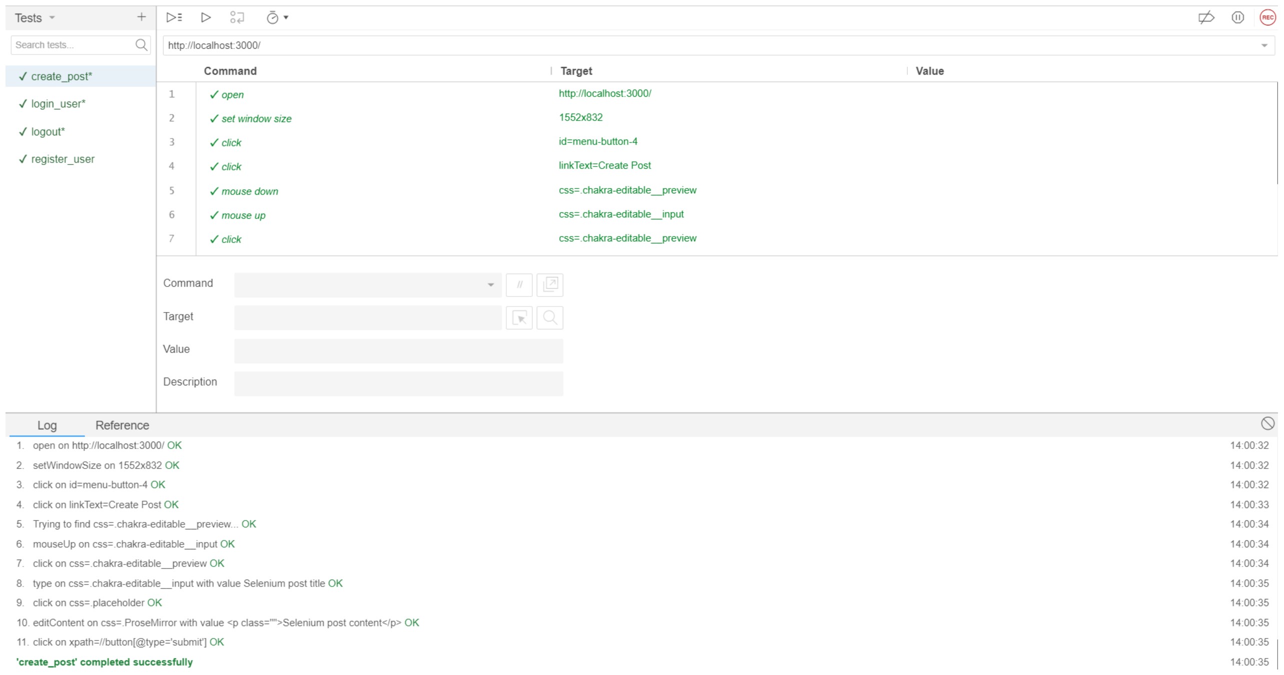
Target: http://localhost:3000/

Value

Description

```
Running 'login_user'                                                                              13:44:00
1. open on http://localhost:3000/ OK                                                              13:44:00
2. setWindowSize on 1552x832 OK                                                                   13:44:00
3. click on id=get-started-btn OK                                                                 13:44:00
4. click on id=input-email-for-credentials-provider OK                                            13:44:01
5. type on id=input-email-for-credentials-provider with value selenium@example.com OK             13:44:03
6. type on id=input-password-for-credentials-provider with value 1234 OK                          13:44:04
7. click on css=button OK                                                                         13:44:04
'login_user' completed successfully                                                               13:44:04
```

3. Create post



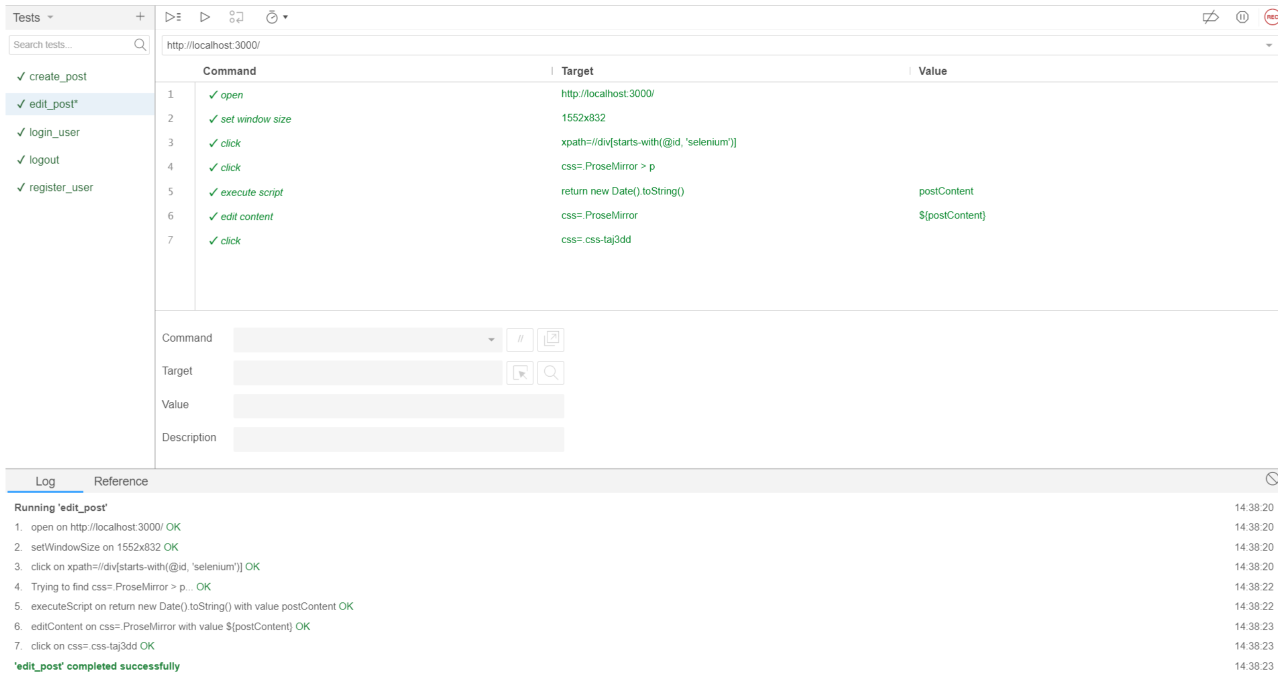
Tests: ✓ create_post*, ✓ login_user*, ✓ logout*, ✓ register_user

http://localhost:3000/

	Command	Target	Value
1	✓ open	http://localhost:3000/	
2	✓ set window size	1552x832	
3	✓ click	id=menu-button-4	
4	✓ click	linkText=Create Post	
5	✓ mouse down	css=.chakra-editable__preview	
6	✓ mouse up	css=.chakra-editable__input	
7	✓ click	css=.chakra-editable__preview	

```
1.  open on http://localhost:3000/ OK                                                                   14:00:32
2.  setWindowSize on 1552x832 OK                                                                        14:00:32
3.  click on id=menu-button-4 OK                                                                        14:00:32
4.  click on linkText=Create Post OK                                                                    14:00:33
5.  Trying to find css=.chakra-editable__preview... OK                                                  14:00:34
6.  mouseUp on css=.chakra-editable__input OK                                                           14:00:34
7.  click on css=.chakra-editable__preview OK                                                           14:00:34
8.  type on css=.chakra-editable__input with value Selenium post title OK                               14:00:35
9.  click on css=.placeholder OK                                                                        14:00:35
10. editContent on css=.ProseMirror with value <p class="">Selenium post content</p> OK                 14:00:35
11. click on xpath=//button[@type='submit'] OK                                                          14:00:35
'create_post' completed successfully                                                                    14:00:35
```

4. Edit post



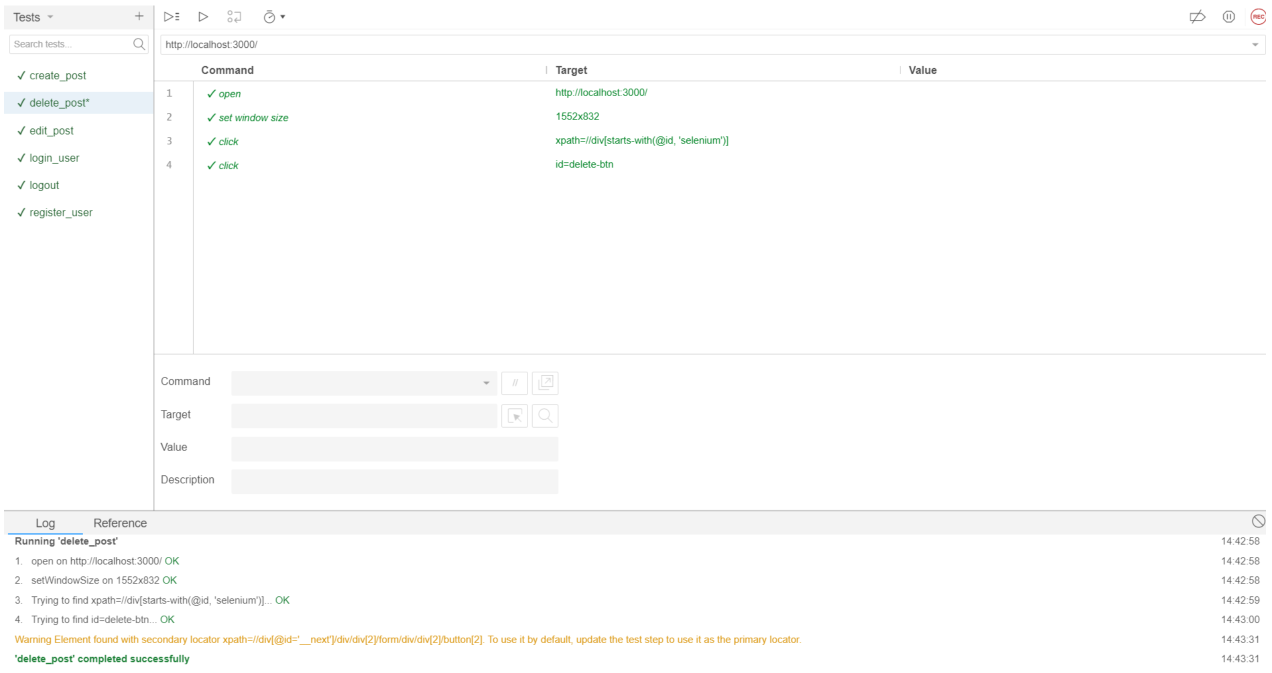
Tests: ✓ create_post, ✓ edit_post*, ✓ login_user, ✓ logout, ✓ register_user

http://localhost:3000/

	Command	Target	Value
1	✓ open	http://localhost:3000/	
2	✓ set window size	1552x832	
3	✓ click	xpath=//div[starts-with(@id, 'selenium')]	
4	✓ click	css=.ProseMirror > p	
5	✓ execute script	return new Date().toString()	postContent
6	✓ edit content	css=.ProseMirror	${postContent}
7	✓ click	css=.css-taj3dd	

```
Running 'edit_post'                                                                                     14:38:20
1.  open on http://localhost:3000/ OK                                                                   14:38:20
2.  setWindowSize on 1552x832 OK                                                                        14:38:20
3.  click on xpath=//div[starts-with(@id, 'selenium')] OK                                               14:38:20
4.  Trying to find css=.ProseMirror > p... OK                                                           14:38:22
5.  executeScript on return new Date().toString() with value postContent OK                             14:38:22
6.  editContent on css=.ProseMirror with value ${postContent} OK                                        14:38:23
7.  click on css=.css-taj3dd OK                                                                         14:38:23
'edit_post' completed successfully                                                                      14:38:23
```

5. Delete post



Tests
- ✓ create_post
- ✓ delete_post*
- ✓ edit_post
- ✓ login_user
- ✓ logout
- ✓ register_user

http://localhost:3000/

	Command	Target	Value
1	✓ open	http://localhost:3000/	
2	✓ set window size	1552x832	
3	✓ click	xpath=//div[starts-with(@id, 'selenium')]	
4	✓ click	id=delete-btn	

```
Running 'delete_post'                                                                                              14:42:58
1. open on http://localhost:3000/ OK                                                                               14:42:58
2. setWindowSize on 1552x832 OK                                                                                    14:42:58
3. Trying to find xpath=//div[starts-with(@id, 'selenium')]... OK                                                  14:42:59
4. Trying to find id=delete-btn... OK                                                                              14:43:00
Warning Element found with secondary locator xpath=//div[@id='__next']/div/div[2]/form/div/div[2]/button[2]. To use it by default, update the test step to use it as the primary locator.   14:43:31
'delete_post' completed successfully                                                                               14:43:31
```

Source code/functions

```
// Generated by Selenium IDE
const { Builder, By, Key, until } = require('selenium-webdriver')
const assert = require('assert')
const { describe, it, beforeEach, afterEach } = require('mocha')

describe('test_suite', function() {
  this.timeout(30000)
  let driver
  let vars

  beforeEach(async function() {
    driver = await new Builder().forBrowser('chrome').build()
    vars = {}
  })

  afterEach(async function() {
    await driver.quit();
  })

  it('register_user', async function() {
    await driver.get("http://localhost:3000//")
    await driver.manage().window().setRect({ width: 1536, height: 816 })
    await driver.findElement(By.id("get-started-btn")).click()
    await driver.findElement(By.id("input-email-for-credentials-provider")).click()
    await
driver.findElement(By.id("input-email-for-credentials-provider")).sendKeys("selenium
@example.com")
    await
driver.findElement(By.id("input-password-for-credentials-provider")).sendKeys("1234"
)
    await driver.findElement(By.id("input-new-for-credentials-provider")).click()
    await driver.findElement(By.css("button")).click()
  })

  it('login_user', async function() {
    await driver.get("http://localhost:3000/")
    await driver.manage().window().setRect({ width: 1552, height: 832 })
    await driver.findElement(By.id("get-started-btn")).click()
    await driver.findElement(By.id("input-email-for-credentials-provider")).click()
    await
driver.findElement(By.id("input-email-for-credentials-provider")).sendKeys("selenium
@example.com")
    await
driver.findElement(By.id("input-password-for-credentials-provider")).sendKeys("1234"
)
    await driver.findElement(By.css("button")).click()
  })

  it('create_post', async function() {
    await driver.get("http://localhost:3000/")
    await driver.manage().window().setRect({ width: 1552, height: 832 })
    await driver.findElement(By.id("menu-button-4")).click()
    await driver.findElement(By.linkText("Create Post")).click()
    {
      const element = await driver.findElement(By.css(".chakra-editable__preview"))
      await driver.actions({ bridge: true
}).moveToElement(element).clickAndHold().perform()
    }
    {
      const element = await driver.findElement(By.css(".chakra-editable__input"))
      await driver.actions({ bridge: true
}).moveToElement(element).release().perform()
    }
    await driver.findElement(By.css(".chakra-editable__preview")).click()
    await driver.findElement(By.css(".chakra-editable__input")).sendKeys("Selenium
post title")
    await driver.findElement(By.css(".placeholder")).click()
    {
      const element = await driver.findElement(By.css(".ProseMirror"))
      await driver.executeScript("if(arguments[0].contentEditable === 'true')
{arguments[0].innerText = '<p class=\"\">Selenium post content</p>'}", element)
    }
    await driver.findElement(By.xpath("//button[@type=\'submit\']")).click()
  })
  it('edit_post', async function() {
    await driver.get("http://localhost:3000/")
    await driver.manage().window().setRect({ width: 1552, height: 832 })
    await driver.findElement(By.xpath("//div[starts-with(@id,
\'selenium\')]")).click()
    await driver.findElement(By.css(".ProseMirror > p")).click()
    vars["postContent"] = await driver.executeScript("return new Date().toString()")
    {
      const element = await driver.findElement(By.css(".ProseMirror"))
      await driver.executeScript("if(arguments[0].contentEditable === 'true')
{arguments[0].innerText = 'vars["postContent"]'}", element)
    }
    await driver.findElement(By.css(".css-taj3dd")).click()
  })
  it('delete_post', async function() {
    await driver.get("http://localhost:3000/")
    await driver.manage().window().setRect({ width: 1552, height: 832 })
    await driver.findElement(By.xpath("//div[starts-with(@id,
\'selenium\')]")).click()
    await driver.findElement(By.id("delete-btn")).click()
  })
})
```

Conclusion:

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