ASSIGNMENT

Submitted By,

Mathew Peter

Roll No:43

INTMCAS6

20INTMCA302


SkillUP
by Simplilearn
Declaration of Completion
Mathew Peter
has successfully completed the online course:
Introduction to Selenium
This professional has demonstrated initiative and a commitment to deepening their skills and advancing their career. Well done!
12th Feb 2023
Certificate code : 4158398
Krishna Kumar
CEO

Selenium Testing

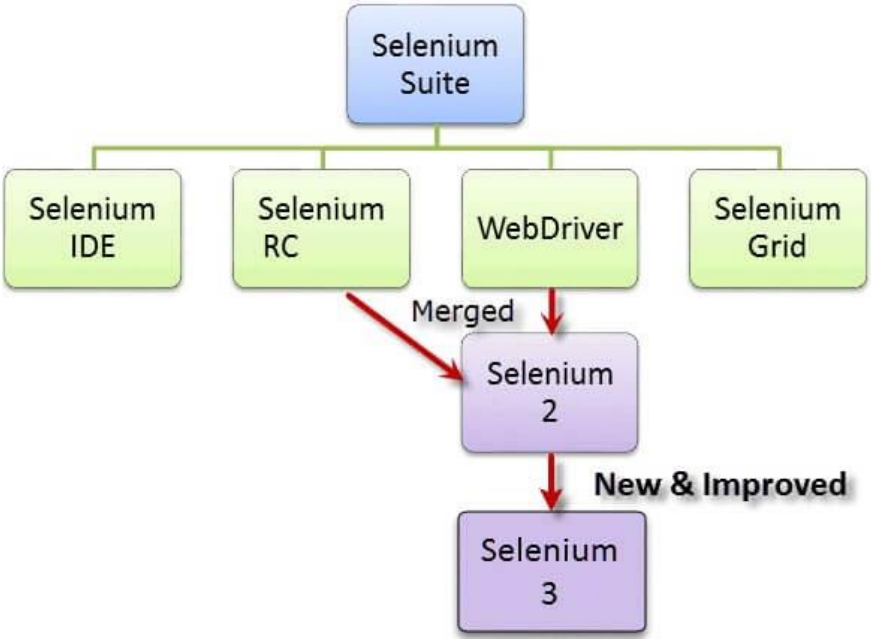
Selenium is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python, etc. to create Selenium Test Scripts. Testing done using the Selenium testing tool is usually referred to as Selenium Testing.

Selenium Tool Suite

Selenium Software is not just a single tool but a suite of software, each piece catering to different Selenium QA testing needs of an organization.

Here is the list of tools

- Selenium Integrated Development Environment (IDE)
- Selenium Remote Control (RC)
- WebDriver
- Selenium Grid

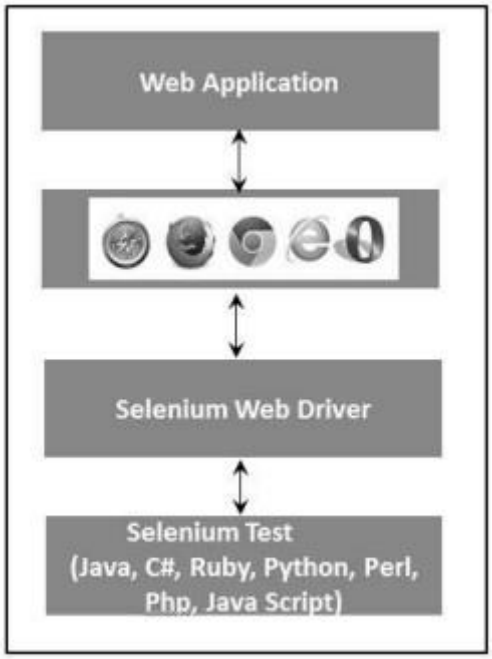


Selenium – Web Driver

WebDriver is a tool for automating testing web applications. It is popularly known as Selenium 2.0. WebDriver uses a different underlying framework, while Selenium RC uses JavaScript Selenium-Core embedded within the browser which has got some limitations. WebDriver interacts directly with the browser without any intermediary, unlike Selenium RC that depends on a server. It is used in the following context −

- Multi-browser testing including improved functionality for browsers which is not well-supported by Selenium RC (Selenium 1.0).
- Handling multiple frames, multiple browser windows, popups, and alerts.
- Complex page navigation.
- Advanced user navigation such as drag-and-drop.
- AJAX-based UI elements.

Architecture



Need for Selenium WebDriver

To understand why WebDriver was introduced, let's look at the shortcomings of Selenium RC.

Selenium Remote Control (RC) is a test tool that allows you to write automated web application UI tests in any programming language against any HTTP website using any mainstream JavaScript-enabled browser.

Selenium RC Server receives Selenium commands from your test program, interprets them, and reports the results back to the program. The Web browser is injected with Selenium core, which interprets and executes the Selenese commands used in the test script. The web browser now interacts with the web server accordingly. This setup, however, complicates the architecture and takes additional time for execution.

Selenium WebDriver helped overcome these drawbacks. The WebDriver does not use an additional server, instead of making direct calls to the browser using each browser's native support for automation.

Selenium Automation Testing

Manual testing, a vital part of the application development process, unfortunately, has many shortcomings, chief of them being that the process is monotonous and repetitive. To

overcome these obstacles, Jason Huggins, an engineer at Thought works, decided to automate the testing process. He developed a JavaScript program the called JavaScriptTestRunner that automated web application testing. This program was renamed Selenium in 2004.

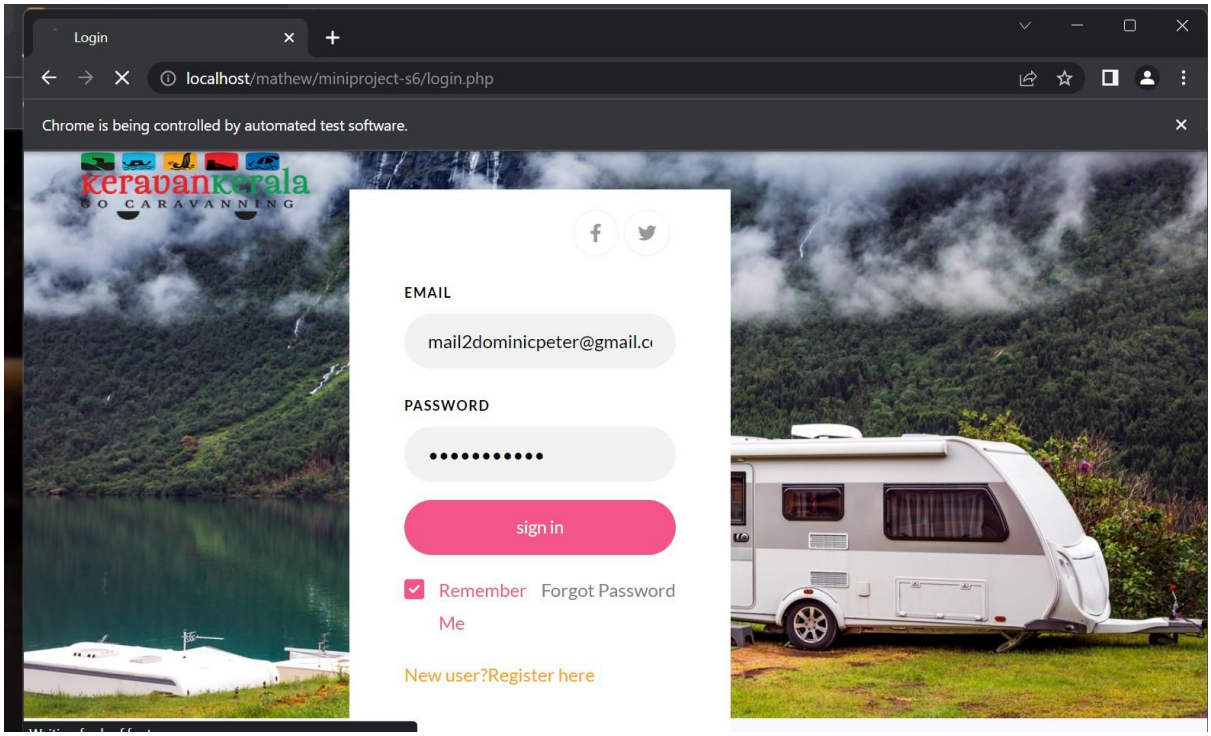
Selenium Automation Testing Tools

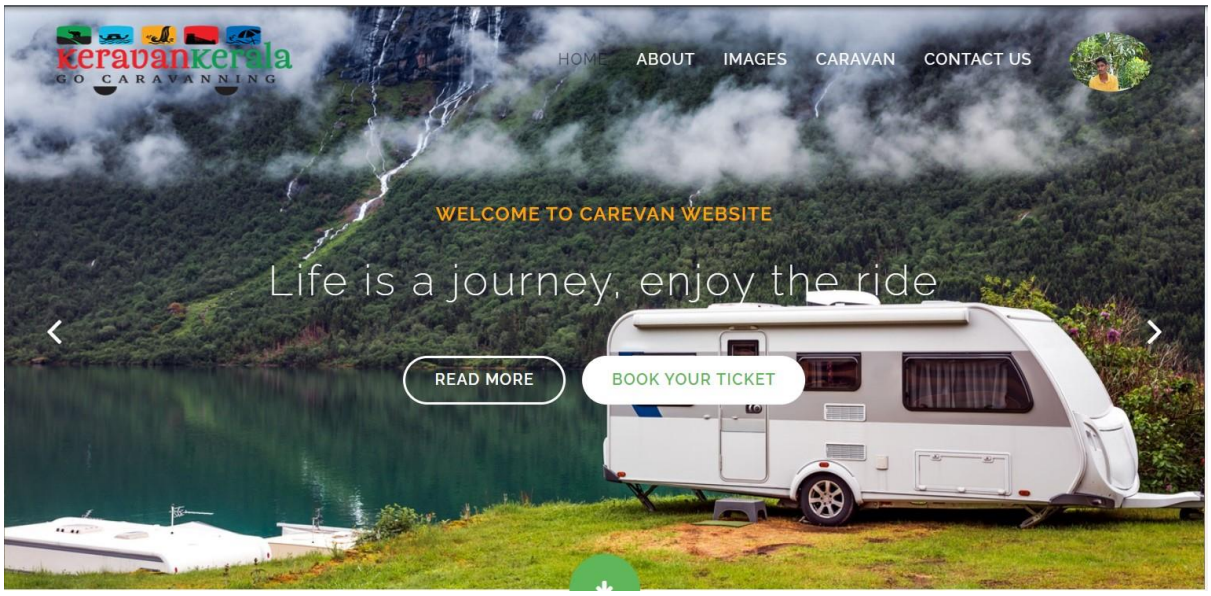
Selenium consists of a set of tools that facilitate the testing process.

Code For Login Form

```
from selenium import webdriver
import time
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys

# set up the driver
driver = webdriver.Chrome()

# navigate to the login page
driver.get("http://localhost/mathew/miniproject-s6/login.php")

# enter the username
username_input = driver.find_element(By.ID, "email")
username_input.send_keys("mail2dominicpeter@gmail.com")

# enter the password
password_input = driver.find_element(By.ID, "password")
password_input.send_keys("Mathew@2001")

# submit the form
submit_button = driver.find_element(By.ID, "submit")
submit_button.click()
time.sleep(10)
# close the browser window
driver.quit()
```

Output



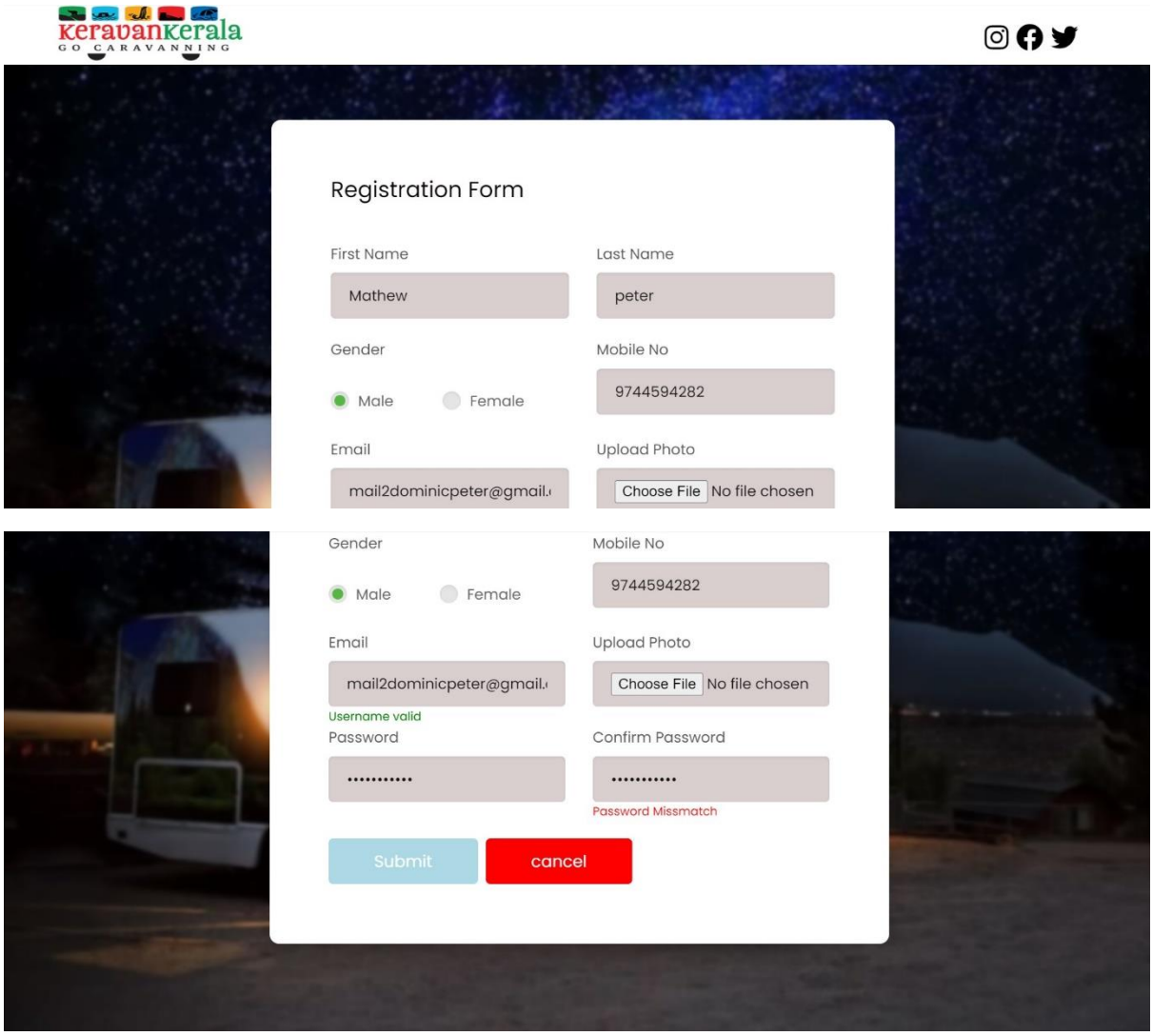


Code For Registration Form

```python
from selenium import webdriver
import time
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys

# set up the driver
driver = webdriver.Chrome()

# navigate to the login page
driver.get("http://localhost/mathew/miniproject-s6/registeruser1.php")

# enter the first
password_input = driver.find_element(By.ID, "txt1")
password_input.send_keys("Mathew")

# enter the last
password_input = driver.find_element(By.ID, "txt2")
password_input.send_keys("peter")


# enter the username
username_input = driver.find_element(By.ID, "phone1")
username_input.send_keys("9744594282")


# enter the confirm password
username_input = driver.find_element(By.ID, "cpassword")
username_input.send_keys("Mathew@2002")


# enter the username
username_input = driver.find_element(By.ID, "email1")
username_input.send_keys("mail2dominicpeter@gmail.com")



# enter the password
password_input = driver.find_element(By.ID, "password")
password_input.send_keys("Mathew@2002")

# submit the form
```

```python
# enter the username
username_input = driver.find_element(By.ID, "email1")
username_input.send_keys("mail2dominicpeter@gmail.com")



# enter the password
password_input = driver.find_element(By.ID, "password")
password_input.send_keys("Mathew@2002")

# submit the form
submit_button = driver.find_element(By.ID, "submit")
submit_button.click()
time.sleep(1)
# close the browser window
driver.quit()
```

Output


keravankerala
GO CARAVANNING
Registration Form
First Name
Mathew
Last Name
peter
Gender
Male
Female
Mobile No
9744594282
Email
mail2dominicpeter@gmail.
Upload Photo
Choose File
No file chosen
Gender
Male
Female
Mobile No
9744594282
Email
mail2dominicpeter@gmail.
Upload Photo
Choose File
No file chosen
Username valid
Password
Confirm Password
Password Missmatch
Submit
cancel