

Software Engineering Task-10

Kavya Aggarwal

HU22CSEN0100288

Introduction to Selenium

Selenium is an open-source automation tool used for testing web applications across different browsers and platforms. It provides a suite of tools that help testers and developers automate browser actions, validate web applications, and perform functional and regression testing.

Components of Selenium

Selenium consists of several components that work together to facilitate browser automation:

1. Selenium WebDriver

- Directly interacts with web browsers to execute automation scripts.
- Supports multiple programming languages, including Java, Python, C#, and JavaScript.
- Works with various browsers such as Chrome, Firefox, Edge, and Safari.

2. Selenium IDE (Integrated Development Environment)

- A browser extension that allows recording and playback of test cases.
- Suitable for beginners as it requires no programming knowledge.

- Supports exporting test scripts to different programming languages for advanced testing.

3. Selenium Grid

- Enables parallel test execution across multiple machines and browsers.
- Enhances test efficiency by reducing execution time.
- Useful for large-scale web applications with extensive test scenarios.

4. Selenium RC (Remote Control) [Deprecated]

- An older component that required a server to interact with web browsers.
- Replaced by WebDriver due to its more efficient and direct browser interaction.

Features of Selenium

- **Cross-Browser Compatibility:** Supports all major web browsers.
- **Multiple Programming Language Support:** Works with Java, Python, JavaScript, C#, Ruby, and more.
- **Platform Independence:** Runs on Windows, macOS, and Linux.
- **Integration with Testing Frameworks:** Compatible with frameworks like JUnit, TestNG, and PyTest.
- **Parallel Test Execution:** Selenium Grid allows tests to be executed simultaneously across multiple environments.
- **Headless Browser Testing:** Allows running tests without a graphical interface, increasing speed and efficiency.
- **Support for CI/CD Pipelines:** Can be integrated with Jenkins, GitHub Actions, and other automation tools.

Common Selenium Commands

To work on any application using Selenium, we generally use the following commands:

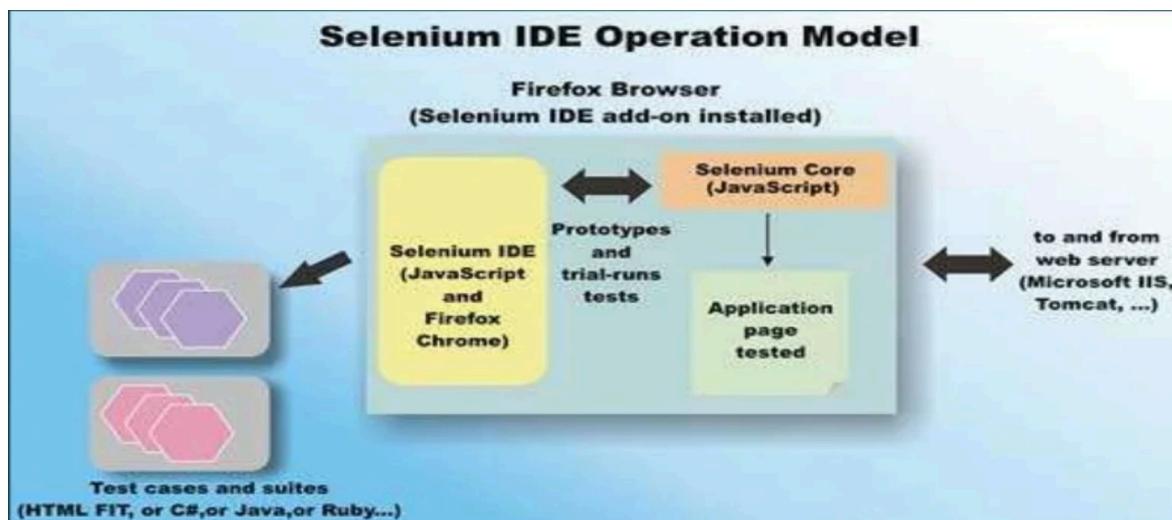
- **Start()**: To launch the browser.
- **Open()**: To open the URL.
- **Close()**: To kill or close the browser.
- **windowMaximize()**: To maximize the window.
- **Type()**: To enter some text into a text box.
- **Click()**: To click on a button, radio button, or link.
- **Select()**: To select a value or label from a combo box, list box, or dropdown.
- **Check()**: To check the checkbox.
- **selectPopUp()**: To identify the popup window.
- **selectWindow()**: To identify the child window.
- **selectFrame()**: To identify the frame.
- **getAlert()**: To click OK on an alert box.
- **getConfirmation()**: To click OK on a confirmation message.
- **chooseCancelOnNextConfirmation()**: To click Cancel on the next displayed confirmation message.
- **chooseOkOnNextConfirmation()**: To click OK on the next displayed confirmation message.

Use Cases of Selenium

- **Functional Testing:** Validates user interactions with a web application.
- **Regression Testing:** Ensures new updates do not break existing functionality.
- **Performance Testing:** Measures application response times under different conditions.
- **Web Scraping:** Extracts data from web pages for analysis.
- **Automated UI Testing:** Verifies UI elements and interactions automatically.

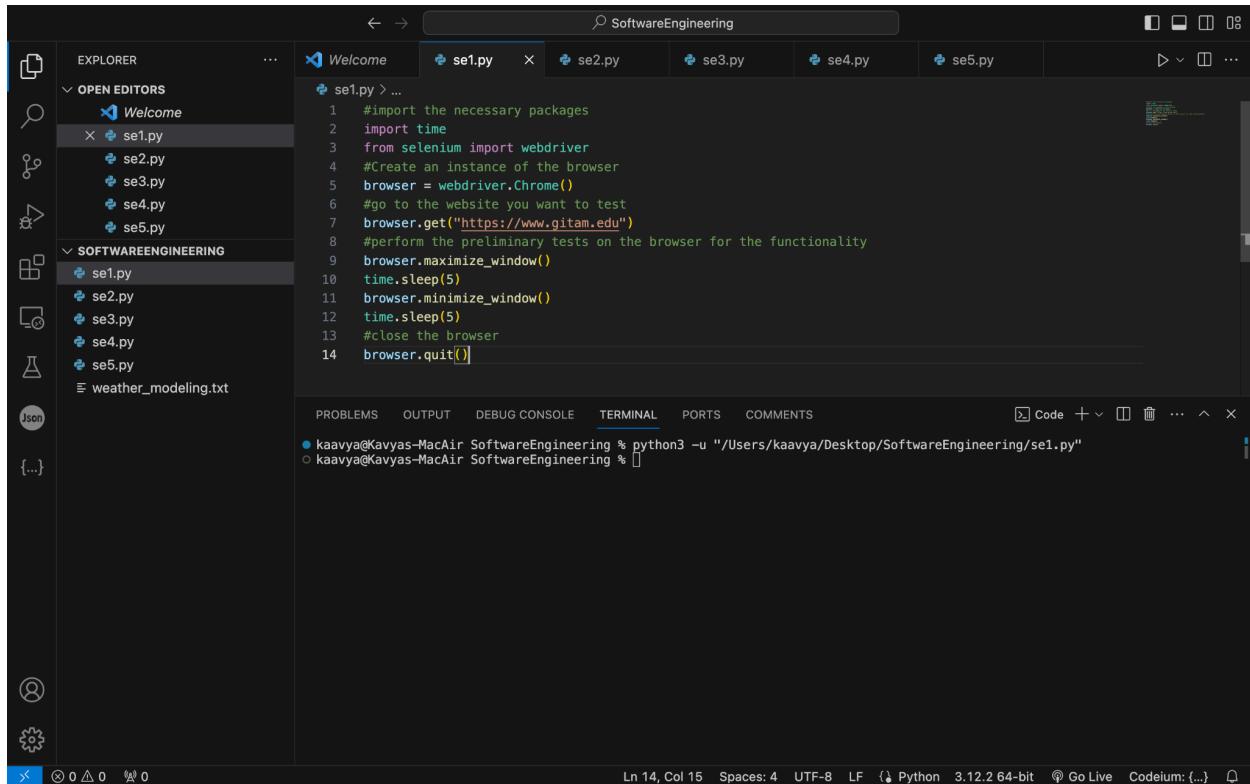
Limitations of Selenium

- **Limited Support for Desktop Applications:** Primarily designed for web applications.
- **No Built-in Reporting:** Requires third-party tools like TestNG, Extent Reports, or Allure.
- **Maintenance Overhead:** Test scripts may require frequent updates due to UI changes.
- **Difficult to Handle Captchas & OTPs:** Requires external tools or manual intervention.



CODE SNIPPETS

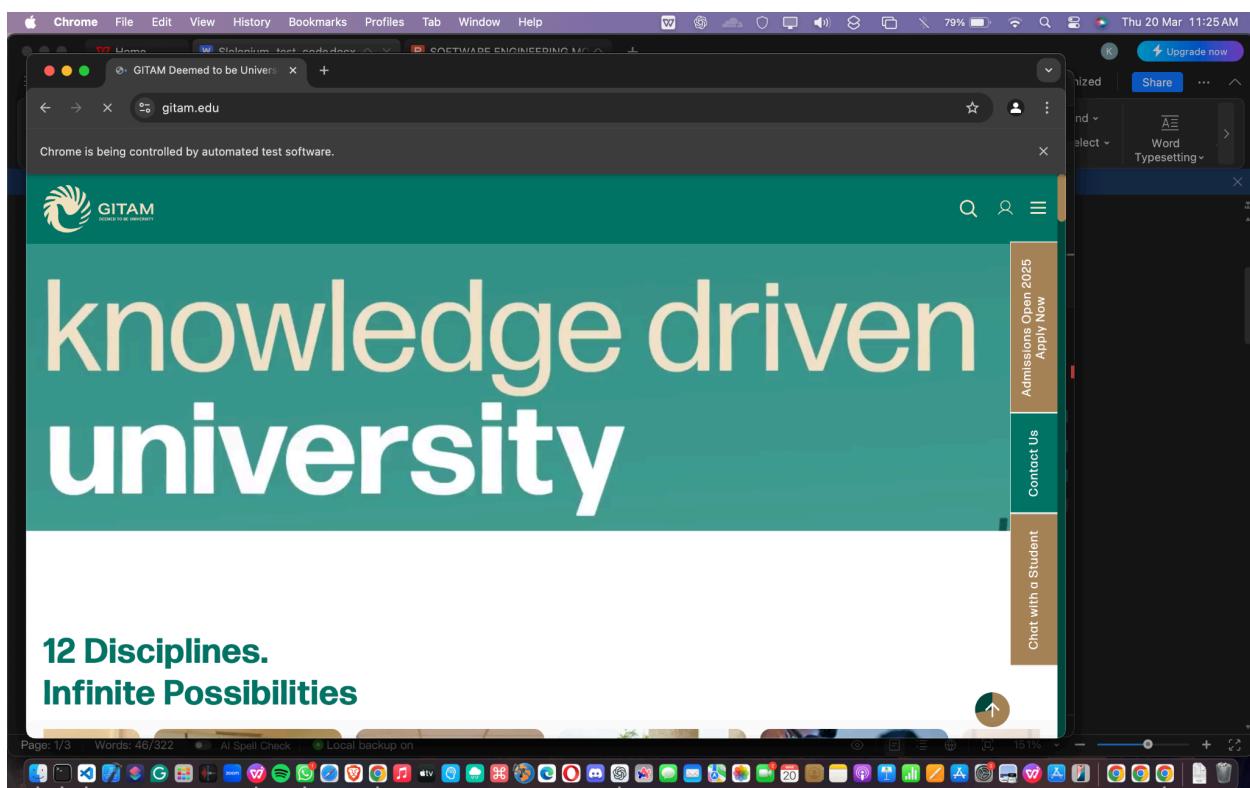
1)



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists files in the 'SoftwareEngineering' folder: 'se1.py', 'se2.py', 'se3.py', 'se4.py', 'se5.py', and 'weather_modeling.txt'. The 'OPEN EDITORS' section shows 'Welcome' and 'se1.py' is the active editor. The code in 'se1.py' is:

```
1 #import the necessary packages
2 import time
3 from selenium import webdriver
4 #Create an instance of the browser
5 browser = webdriver.Chrome()
6 #go to the website you want to test
7 browser.get("https://www.gitam.edu")
8 #perform the preliminary tests on the browser for the functionality
9 browser.maximize_window()
10 time.sleep(5)
11 browser.minimize_window()
12 time.sleep(5)
13 #close the browser
14 browser.quit()
```

The Terminal tab at the bottom shows the command 'python3 -u "/Users/kaavya/Desktop/SoftwareEngineering/se1.py"' being run.

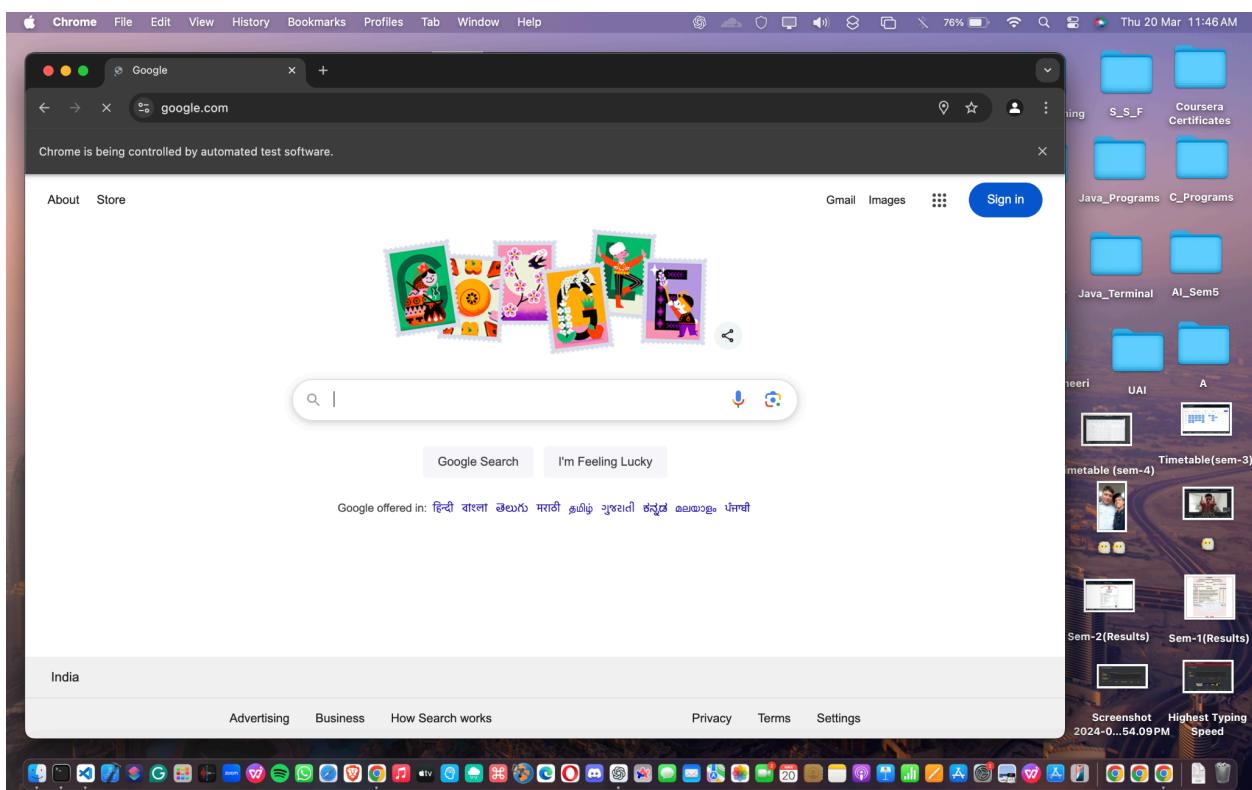


2)

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists files: Welcome, se1.py, se2.py, se3.py, se4.py, se5.py, and weather_modeling.txt. The se2.py file is open in the editor, displaying the following Python code:

```
#Locating Elements on Webpage
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium import webdriver
#find element by id
driver = webdriver.Chrome()
driver.get('https://www.google.com')
element = driver.find_element(By.NAME,"q")
element.send_keys("Selenium!")
element.send_keys(Keys.ENTER)
driver.quit()
```

The terminal at the bottom shows the command: `python3 -u "/Users/kaavya/Desktop/SoftwareEngineering/se2.py"`. The status bar indicates the code is in Python 3.12.2 64-bit.



3)

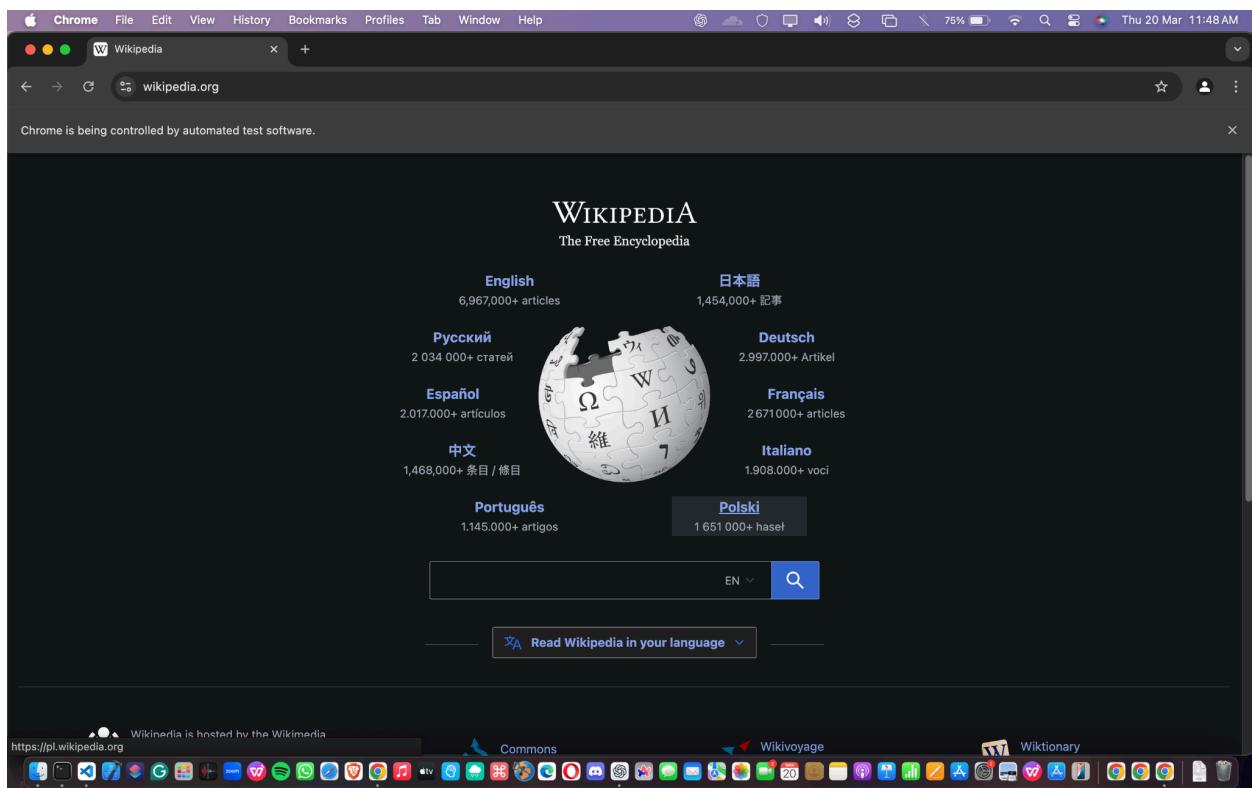
The screenshot shows the Visual Studio Code (VS Code) interface. The Explorer sidebar on the left lists files under 'OPEN EDITORS' and 'SOFTWAREENGINEERING'. The 'se3.py' file is selected in the Explorer and is open in the main editor area. The code uses Selenium to open Wikipedia and perform basic interactions. The terminal at the bottom shows command-line history for running the script.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time
# Initialize Chrome WebDriver
driver = webdriver.Chrome()
# Open Google
driver.get('https://www.wikipedia.org/')
title = driver.title
print(title)
driver.minimize_window()
time.sleep(5)
driver.maximize_window()
# Close Browser
time.sleep(8)
driver.quit()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

kaavya@Kavyas-MacAir SoftwareEngineering % python3 -u "/Users/kaavya/Desktop/SoftwareEngineering/se2.py"
kaavya@Kavyas-MacAir SoftwareEngineering % python3 -u "/Users/kaavya/Desktop/SoftwareEngineering/se3.py"
Wikipedia
Search Completed!
kaavya@Kavyas-MacAir SoftwareEngineering %

Ln 17, Col 27 Spaces: 4 UTF-8 LF ↵ Python 3.12.2 64-bit ⓘ Go Live Codeium: (...) ⌂



4)

```

from selenium import webdriver
import time
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
# Use Chrome WebDriver (Make sure you have chromedriver installed)
driver = webdriver.Chrome()
# Set timeout
driver.set_page_load_timeout(10)
# Open webpage
driver.get("https://www.wikipedia.org/")
# Find the search box and enter a search query
search_box = driver.find_element(By.NAME, "search")
search_box.send_keys("Selenium Python")
search_box.send_keys(Keys.ENTER) # Press Enter
# Wait for results to load
time.sleep(5)

```

kaavya@Kavyas-MacAir SoftwareEngineering % python3 -u "/Users/kaavya/Desktop/SoftwareEngineering/se2.py"
 kaavya@Kavyas-MacAir SoftwareEngineering % python3 -u "/Users/kaavya/Desktop/SoftwareEngineering/se3.py"
 Wikipedia
 Search Completed!
 kaavya@Kavyas-MacAir SoftwareEngineering % python3 -u "/Users/kaavya/Desktop/SoftwareEngineering/se4.py"
 Wikipedia
 Search Completed!
 kaavya@Kavyas-MacAir SoftwareEngineering % python3 -u "/Users/kaavya/Desktop/SoftwareEngineering/se4.py"
 o kaavya@Kavyas-MacAir SoftwareEngineering %

Chrome is being controlled by automated test software.

WIKIPEDIA The Free Encyclopedia

Search Wikipedia

Search

Donate Create account Log in

Help

Tools

Appearance hide

Text

- Small
- Standard
- Large

This page always uses small font size

Width

- Standard
- Wide

Color (beta)

- Automatic
- Light
- Dark

Results 1 – 20 of 28

The page "[Selenium Python](#)" does not exist. You can [create a draft and submit it for review](#) or request that a redirect be created, but consider checking the search results below to see whether the topic is already covered.

View (previous 20 | next 20) (20 | 50 | 100 | 250 | 500)

Selenium (software)

PHP, Python, Ruby and Scala. Selenium runs on Windows, Linux, and macOS. It is open-source software released under the Apache License 2.0. Selenium is an...
 19 KB (2,020 words) - 03:11, 22 December 2024

Playwright (software)

to Cypress. Playwright supports programming languages like JavaScript, Python, C# and Java, though its main API was

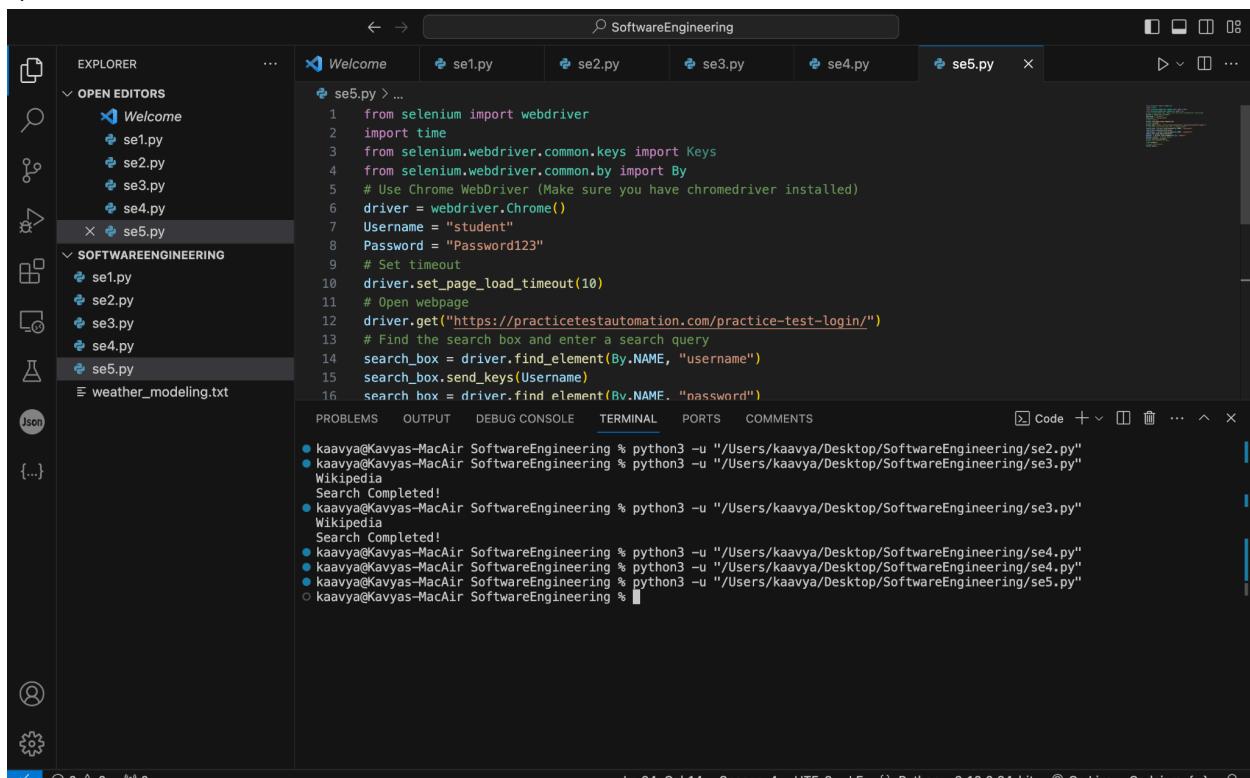
Textbooks from Wikibooks

[Android/Printable version](#)

and mocks present. The following tools may prove useful: Robotium - like Selenium for Android Sikuli - a screen-scraping execution system Monkeyrunner - See all results

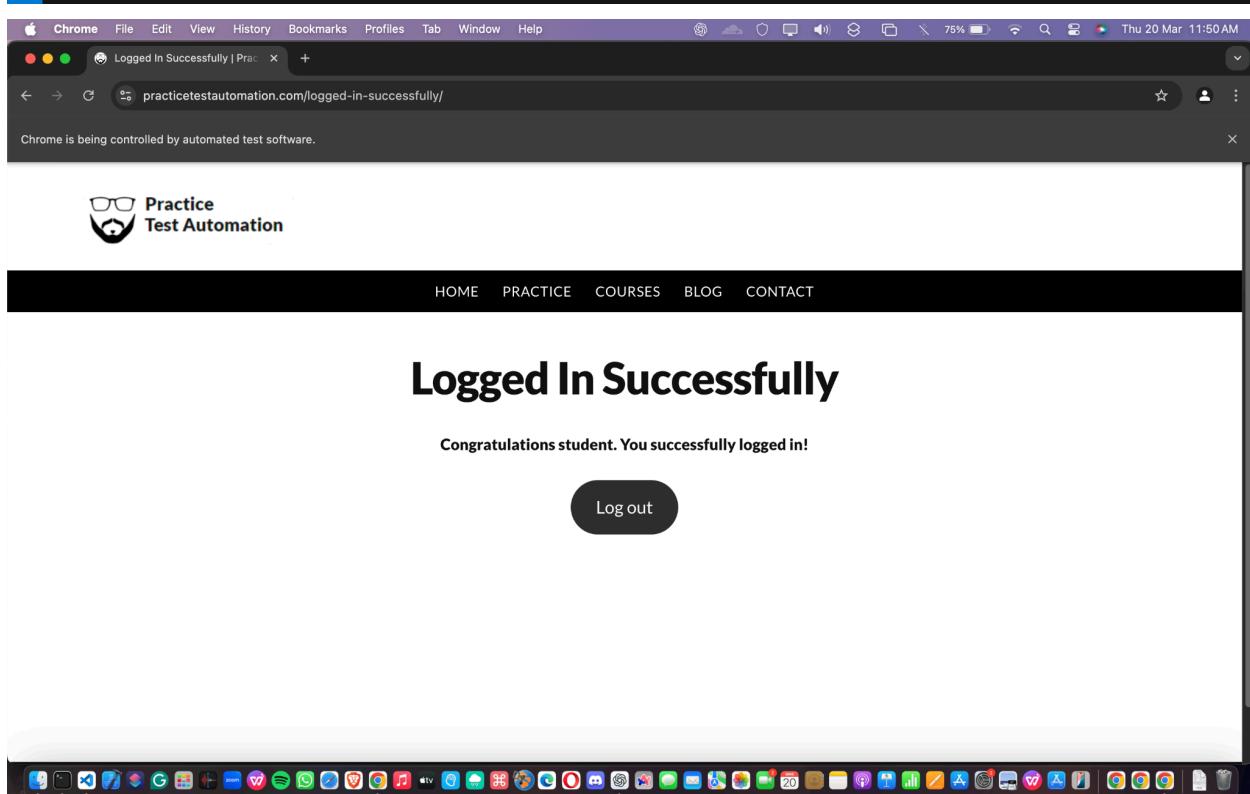
Timetable (sem-3)
 timetable (sem-4)
 Sem-2(Results) Sem-1(Results)
 Screenshot 2024-0...54.09PM Highest Typing Speed

5)



The terminal window shows the execution of several Python scripts:

```
kaavya@Kavyas-MacAir SoftwareEngineering % python3 -u "/Users/kaavya/Desktop/SoftwareEngineering/se2.py"
kaavya@Kavyas-MacAir SoftwareEngineering % python3 -u "/Users/kaavya/Desktop/SoftwareEngineering/se3.py"
Wikipedia
Search Completed!
kaavya@Kavyas-MacAir SoftwareEngineering % python3 -u "/Users/kaavya/Desktop/SoftwareEngineering/se3.py"
Wikipedia
Search Completed!
kaavya@Kavyas-MacAir SoftwareEngineering % python3 -u "/Users/kaavya/Desktop/SoftwareEngineering/se4.py"
kaavya@Kavyas-MacAir SoftwareEngineering % python3 -u "/Users/kaavya/Desktop/SoftwareEngineering/se4.py"
kaavya@Kavyas-MacAir SoftwareEngineering % python3 -u "/Users/kaavya/Desktop/SoftwareEngineering/se5.py"
kaavya@Kavyas-MacAir SoftwareEngineering %
```



The browser window displays the URL practicetestautomation.com/logged-in-successfully/. The page content includes:

Logged In Successfully | Practice Test Automation

HOME PRACTICE COURSES BLOG CONTACT

Logged In Successfully

Congratulations student. You successfully logged in!

Log out