



---

Software Testing Laboratory  
(CS6474)  
Assignment 03 :Selenium WebDriver

---

**Tapas Manna**  
**223CS3152**  
Master of Technology  
223cs3152@nitrkl.ac.in

**Department of Computer Science Engineering**  
**NIT, Rourkela**  
January 25, 2025

## Contents

<b>1</b>	<b>NITRIS Login and download result</b>	<b>3</b>
1.1	Selenium WebDriver Screenshot of code. . . . .	3
1.2	IRCTC : Selenium WebDriver python code . . . . .	3
<b>2</b>	<b>Website 1:IRCTC <a href="https://www.irctc.co.in/">https://www.irctc.co.in/</a></b>	<b>6</b>
2.1	Selenium WebDriver Screenshot IRCTC website. . . . .	6
2.2	IRCTC : Selenium WebDriver python code . . . . .	6
<b>3</b>	<b>Website 2:<a href="https://www.amazon.in/">https://www.amazon.in/</a></b>	<b>8</b>
3.1	Selenium WebDriver Screenshot amazon website. . . . .	8
3.2	Amazon : Selenium WebDriver python code . . . . .	8
<b>4</b>	<b>Website 3:MyGov <a href="https://www.mygov.in/">https://www.mygov.in/</a></b>	<b>9</b>
4.1	Selenium WebDriver Screenshot MyGov website. . . . .	9
4.2	MyGov : Selenium WebDriver python code . . . . .	9
<b>5</b>	<b>Website 4:<a href="https://vedabase.io/">https://vedabase.io/</a></b>	<b>11</b>
5.1	Selenium WebDriver Screenshot vedaBase website. . . . .	11
5.2	VedaBase : Selenium WebDriver python code . . . . .	11
<b>6</b>	<b>Website 5:<a href="https://www.coursera.org/">https://www.coursera.org/</a></b>	<b>13</b>
6.1	Selenium WebDriver Screenshot coursera website. . . . .	13
6.2	Coursera : Selenium WebDriver python code . . . . .	13
<b>7</b>	<b>Website 6:<a href="https://stackoverflow.com/">https://stackoverflow.com/</a></b>	<b>15</b>
7.1	Selenium WebDriver Screenshot stackOverFlow website. . . . .	15
7.2	stackOverFlow : Selenium WebDriver python code . . . . .	15
<b>8</b>	<b>Website 7:<a href="https://takeuforward.org/">https://takeuforward.org/</a></b>	<b>17</b>
8.1	Selenium WebDriver Screenshot takeuforward website. . . . .	17
8.2	Takeuforward : Selenium WebDriver python code . . . . .	17
<b>9</b>	<b>Website 8:<a href="https://google.com/">https://google.com/</a></b>	<b>19</b>
9.1	Selenium WebDriver Screenshot google classroom website. . . . .	19
9.2	Geeks for Geeks Classroom : Selenium WebDriver python code . . . . .	19
<b>10</b>	<b>Website 9:<a href="https://www.geeksforgeeks.org/">https://www.geeksforgeeks.org/</a></b>	<b>21</b>
10.1	Selenium WebDriver Screenshot gfg website. . . . .	21
10.2	Gfg : Selenium WebDriver python code . . . . .	21
<b>11</b>	<b>Website 13:<a href="https://www.linkedin.com/">https://www.linkedin.com/</a></b>	<b>23</b>
11.1	Selenium WebDriver Screenshot linkdin website. . . . .	23
11.2	Takeuforward : Selenium WebDriver python code . . . . .	23
<b>12</b>	<b>Website 10:<a href="https://www.flipkart.com/">https://www.flipkart.com/</a></b>	<b>24</b>
12.1	Selenium WebDriver Screenshot Flipkart website. . . . .	24
12.2	Flipkart : Selenium WebDriver python code . . . . .	24

## 1 NITRIS Login and download result

## 1.1 Selenium WebDriver Screenshot of code.

Figure 1: IRCTC Website

## 1.2 IRCTC : Selenium WebDriver python code

```
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.support.ui import WebDriverWait as wait
4 from selenium.webdriver.support import expected_conditions
5 import base64
6
7 url = "https://eapplication.nitrkl.ac.in/nitris/Login.aspx"
8 path = "/Users/tapasmanna/Downloads/"
9
10 def getBrowser():
11     options = webdriver.ChromeOptions()
12     options.use_chromium = True
13
14     # options.add_argument('--headless') # Run Edge in headless mode
15     return webdriver.Chrome(options=options)
16
17
18 def login(browser, username, password):
19     try:
20         # enter username
21         browser.find_element(By.ID, "txtUserName").send_keys(username)
22
23         #enter password
24         browser.find_element(By.ID, "txtPassword").send_keys(password)
25
26         # click login
27         browser.find_element(By.ID, "btnLogin").click()
28
29     except Exception as e:
30         print(f"An error occurred: {e}")
31
32     finally:
33         browser.quit()
```

```

27     browser.find_element(By.ID, "btnLogin").click()
28
29     #wait for login to complete
30     wait(browser, 30).until(
31         expected_conditions.title_contains("NITRIS")
32     )
33 except:
34     return False
35
36 return True
37
38
39
40 def downloadResult(username, password):
41     try:
42         browser = getBrowser()
43         print("Browser completed")
44         browser.get(url)
45
46         # if not already login
47         if url == browser.current_url:
48             if not login(browser, username, password):
49                 return False
50
51         print("login complete")
52
53         #time.sleep(20)
54
55         # click on academic
56         browser.find_element(By.ID, "Academic").click()
57
58         # wait for examination button to be interactable
59         wait(browser, 30).until(
60             expected_conditions.element_to_be_clickable((By.XPATH, "//header/nav[1]/div[2]/ul[1]/li[2]/div[1]/div[1]/a[5]"))
61         )
62
63         # click on examination button
64         browser.find_element(By.XPATH, "//header/nav[1]/div[2]/ul[1]/li[2]/div[1]/div[1]/a[5]").click()
65
66         # wait for new page to open by waiting for examination result element to
67         # be present in DOM
68         wait(browser, 30).until(
69             expected_conditions.presence_of_element_located((By.XPATH, "//body/form[@id='form1']/div[@id='main-wrapper']/aside[1]/div[1]/nav[1]/ul[1]/li[3]/a[1]"))
70         )
71
72         # click on examination result
73         browser.find_element(By.XPATH, "//body/form[@id='form1']/div[@id='main-wrapper']/aside[1]/div[1]/nav[1]/ul[1]/li[3]/a[1]").click()
74
75         # wait for view grade to be interactable
76         wait(browser, 30).until(
77             expected_conditions.element_to_be_clickable((By.XPATH, "//body/form[@id='form1']/div[@id='main-wrapper']/aside[1]/div[1]/nav[1]/ul[1]/li[3]/ul[1]/li[2]/a[1]"))
78     )

```

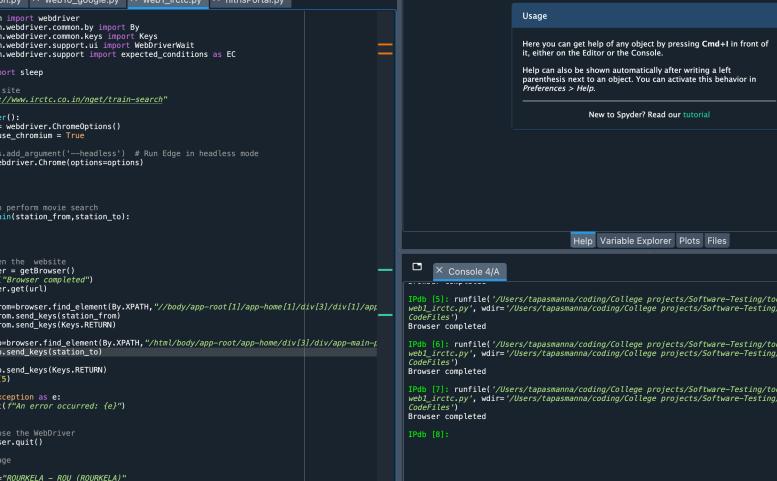
```

78
79     # click on grade card
80     browser.find_element(By.XPATH, "//body/form[@id='form1']/div[@id='main-
81     wrapper']/aside[1]/div[1]/nav[1]/ul[1]/li[3]/ul[1]/li[2]/a[1]").click()
82
83     # wait for grade card window to open
84     wait(browser, 30).until(
85         expected_conditions.number_of_windows_to_be(2)
86     )
87
88     # switch to new window
89     browser.switch_to.window(browser.window_handles[-1])
90
91     params = {
92         'outputFormat': 'pdf',
93         'printBackground': True,
94         'marginTop': 0,
95         'marginBottom': 0,
96         'marginLeft': 0,
97         'marginRight': 0,
98     }
99
100    # execute chrome dev tools print command
101    content = browser.execute_cdp_cmd('Page.printToPDF', params)
102
103    # convert data to binary
104    pdf_data = base64.b64decode(content['data'])
105
106    # save data as pdf
107    with(open(f'{path}{username}.pdf', 'wb')) as pdf_file:
108        pdf_file.write(pdf_data)
109        print("Download completed.")
110    except Exception as err:
111        print(err)
112    finally:
113        browser.quit()
114
115    if __name__ == "__main__":
116        username = "223CS3152"
117        password = "*****"
118        downloadResult(username, password)

```

## 2 Website 1:IRCTC <https://www.irctc.co.in/>

## 2.1 Selenium WebDriver Screenshot IRCTC website.



The screenshot shows the Spyder IDE interface with the following details:

- Left Panel (Code Editor):** A Python script titled `web1_irctc.py` is open. The code uses Selenium WebDriver to interact with the IRCTC website (`https://www.irctc.co.in/nget/train-search`). It defines a function `getBrowser` to initialize the browser and a function `searchTrain` to perform a movie search. The script then attempts to find an element by XPath and send keys to it. Finally, it prints an error message and closes the browser.
- Right Panel (IPython Console):** The IPython console shows the execution of the script. It runs the file `web1_irctc.py` in the directory `/Users/tapasmanna/coding/College projects/Software-Testing/tool3-WebDriver/CodeFiles`. The output shows the browser being opened and the search element being found, but the script fails to execute the search due to an exception. The browser is then closed.
- Bottom Status Bar:** The status bar shows the environment as `conda: base (Python 3.11.5)`, completion status as `Completions: conda(base)`, and the current file as `IPython Console`.

Figure 2: IRCTC Website

## 2.2 IRCTC : Selenium WebDriver python code

```
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.common.keys import Keys
4 from selenium.webdriver.support.ui import WebDriverWait
5 from selenium.webdriver.support import expected_conditions as EC
6
7 from time import sleep
8
9 # URL of the site
10 url = "https://www.irctc.co.in/irctc/train-search"
11
12 def getBrowser():
13     options = webdriver.ChromeOptions()
14     options.use_chromium = True
15
16     # options.add_argument('--headless') # Run Edge in headless mode
17     return webdriver.Chrome(options=options)
18
19
20
21
22 # Function to perform movie search
23 def searchTrain(station_from,station_to):
24
25
26
```

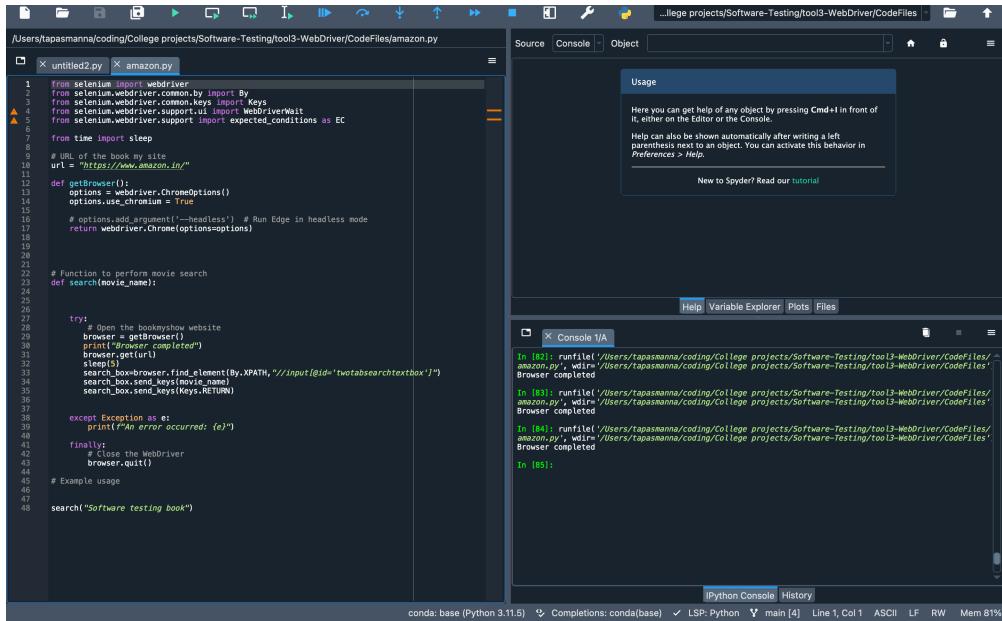
```

27     try:
28         # Open the website
29         browser = getBrowser()
30         print("Browser completed")
31         browser.get(url)
32
33         des_from=browser.find_element(By.XPATH,"//body/app-root[1]/app-home[1]/div[3]/div[1]/app-main-page[1]/div[1]/div[1]/div[1]/div[1]/div[1]/app-jp-input[1]/div[1]/form[1]/div[2]/div[1]/div[1]/p-autocomplete[1]/span[1]/input[1]")
34         des_from.send_keys(station_from)
35         des_from.send_keys(Keys.RETURN)
36
37         des_to=browser.find_element(By.XPATH,"/html/body/app-root/app-home/div[3]/div/app-main-page/div/div/div[1]/div[1]/div[1]/app-jp-input/div/form/div[2]/div[1]/div[2]/p-autocomplete/span/input")
38         des_to.send_keys(station_to)
39
40         des_to.send_keys(Keys.RETURN)
41         sleep(5)
42
43     except Exception as e:
44         print(f"An error occurred: {e}")
45
46     finally:
47         # Close the WebDriver
48         browser.quit()
49
50 # Example usage
51
52 station_from="ROURKELA - ROU (ROURKELA)"
53 station_to="SURAT - ST"
54 searchTrain(station_from,station_to);

```

### 3 Website 2: <https://www.amazon.in/>

#### 3.1 Selenium WebDriver Screenshot amazon website.



The screenshot shows the Spyder IDE interface. On the left, the code editor displays a Python script named 'amazon.py' with the following content:

```
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.common.keys import Keys
4 from selenium.webdriver.support.ui import WebDriverWait
5 from selenium.webdriver.support import expected_conditions as EC
6
7 from time import sleep
8
9 # URL of the book my site
10 url = "https://www.amazon.in/"
11
12 def getBrowser():
13     options = webdriver.ChromeOptions()
14     options.use_chromium = True
15
16     # options.add_argument('--headless') # Run Edge in headless mode
17     return webdriver.Chrome(options=options)
18
19
20
21 # Function to perform movie search
22 def search(movie_name):
23
24     try:
25         # Open the bookmyshow website
26         browser = getBrowser()
27         browser.get(url)
28         browser.implicitly_wait(10)
29         browser.get(url)
30         sleep(5)
31         search_box = browser.find_element(By.XPATH, "//input[@id='twotabsearchtextbox']")
32         search_box.send_keys(movie_name)
33         search_box.send_keys(Keys.RETURN)
34
35     except Exception as e:
36         print(f"An error occurred: {e}")
37
38     finally:
39         # Close the WebDriver
40         browser.quit()
41
42     # Example usage
43
44
45     # search("Software testing book")
46
47
48 search("Software testing book")
```

On the right, the IPython Console shows the execution of the script. It displays three run logs:

- In [82]: runfile('/Users/tapasmania/coding/College projects/Software-Testing/tool3-WebDriver/CodeFiles/amazon.py', wdir='/Users/tapasmania/coding/College projects/Software-Testing/tool3-WebDriver/CodeFiles')
- In [83]: runfile('/Users/tapasmania/coding/College projects/Software-Testing/tool3-WebDriver/CodeFiles/amazon.py', wdir='/Users/tapasmania/coding/College projects/Software-Testing/tool3-WebDriver/CodeFiles')
- In [84]: runfile('/Users/tapasmania/coding/College projects/Software-Testing/tool3-WebDriver/CodeFiles/amazon.py', wdir='/Users/tapasmania/coding/College projects/Software-Testing/tool3-WebDriver/CodeFiles')

The console output indicates that the browser completed its tasks.

Figure 3: Amazon Website

#### 3.2 Amazon : Selenium WebDriver python code

```
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.common.keys import Keys
4 from selenium.webdriver.support.ui import WebDriverWait
5 from selenium.webdriver.support import expected_conditions as EC
6
7 from time import sleep
8
9 # URL of the site
10 url = "https://www.amazon.in/"
11
12 def getBrowser():
13     options = webdriver.ChromeOptions()
14     options.use_chromium = True
15
16     # options.add_argument('--headless') # Run Edge in headless mode
17     return webdriver.Chrome(options=options)
18
19
20
21 # Function to perform movie search
22 def search(query):
23
24
25
26
```

```

27     try:
28         # Open the website
29         browser = getBrowser()
30         print("Browser completed")
31         browser.get(url)
32         sleep(5)
33         search_box=browser.find_element(By.XPATH, "//input[@id='twotabsearchtextbox']")
34         search_box.send_keys(query)
35         search_box.send_keys(Keys.RETURN)
36
37     except Exception as e:
38         print(f"An error occurred: {e}")
39
40     finally:
41         # Close the WebDriver
42         browser.quit()
43
44
45 # Example usage
46
47
48 search("Software testing book")

```

## 4 Website 3:MyGov <https://www.mygov.in/>

### 4.1 Selenium WebDriver Screenshot MyGov website.

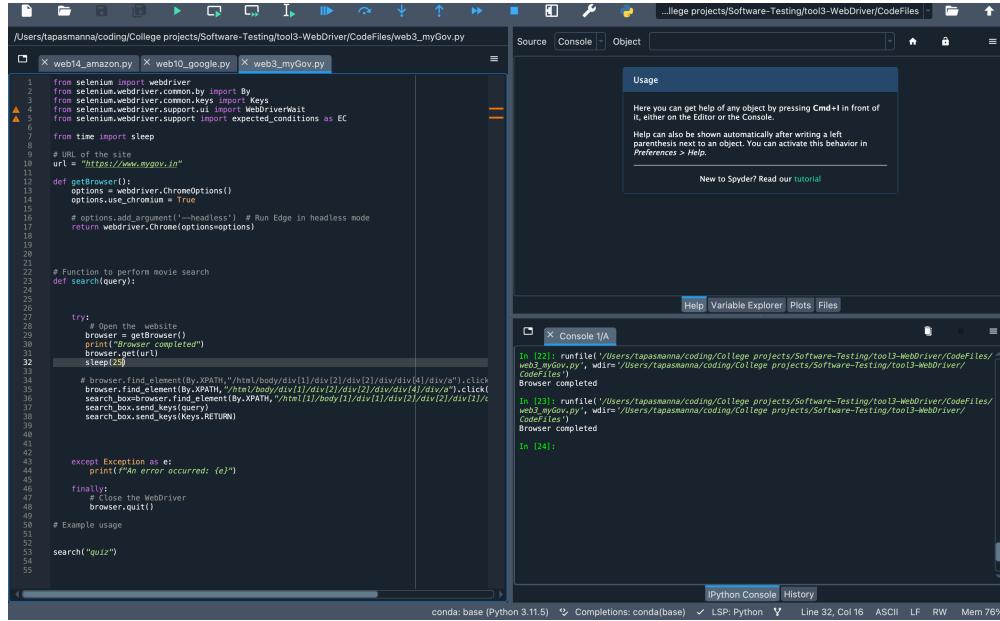


Figure 4: MyGov Website

### 4.2 MyGov : Selenium WebDriver python code

```
1 from selenium import webdriver
```

```

2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.common.keys import Keys
4 from selenium.webdriver.support.ui import WebDriverWait
5 from selenium.webdriver.support import expected_conditions as EC
6
7 from time import sleep
8
9 # URL of the site
10 url = "https://www.mygov.in"
11
12 def getBrowser():
13     options = webdriver.ChromeOptions()
14     options.use_chromium = True
15
16     # options.add_argument('--headless') # Run Edge in headless mode
17     return webdriver.Chrome(options=options)
18
19
20
21
22 # Function to perform movie search
23 def search(query):
24
25
26
27     try:
28         # Open the website
29         browser = getBrowser()
30         print("Browser completed")
31         browser.get(url)
32         sleep(25)
33
34         # browser.find_element(By.XPATH ,"/html/body/div[1]/div[2]/div[2]/div/div[4]/div/a").click()
35         browser.find_element(By.XPATH ,"/html/body/div[1]/div[2]/div[2]/div/div[4]/div/a").click()
36         search_box=browser.find_element(By.XPATH ,"/html[1]/body[1]/div[1]/div[2]/div[2]/div[1]/div[4]/div[1]/div[1]/div[1]/input[1]")
37         search_box.send_keys(query)
38         search_box.send_keys(Keys.RETURN)
39
40
41
42
43     except Exception as e:
44         print(f"An error occurred: {e}")
45
46
47     finally:
48         # Close the WebDriver
49         browser.quit()
50
51
52
53 # Example usage
54
55
56 search("quiz")

```

## 5 Website 4:<https://vedabase.io/>

## 5.1 Selenium WebDriver Screenshot vedabase website.

The screenshot shows the Spyder IDE interface with two main panes. The left pane displays a Python script named `web2_bookMyShow.py` in a code editor. The script uses Selenium WebDriver to interact with the `bookMyShow` website. It includes a function `readBG()` that performs a search and extracts movie reviews. The right pane shows a Jupyter Notebook cell with the following content:

```
...lege projects/Software-Testing/tool3-WebDriver/CodeFiles/vedaBase.py

Source Console Object

Usage

Here you can get help of any object by pressing Cmd+I in front of it, either on the Editor or the Console.

Help can also be automatically shown after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our tutorial

Help Variable Explorer Plots Files

Console 1/A
In [9]: runfile('/Users/tapasmanna/coding/College projects/Software-Testing/tool3-WebDriver/CodeFiles/vedaBase.py', wdir='/Users/tapasmanna/coding/College projects/Software-Testing/tool3-WebDriver/CodeFiles/vedaBase.py')
Browser completed
In [10]: runfile('/Users/tapasmanna/coding/College projects/Software-Testing/tool3-WebDriver/CodeFiles/vedaBase.py', wdir='/Users/tapasmanna/coding/College projects/Software-Testing/tool3-WebDriver/CodeFiles/vedaBase.py')
Browser completed

In [9]:
```

The Jupyter cell has run twice, showing the output "Browser completed" each time. The status bar at the bottom indicates the environment is "conda: base (Python 3.11.5)" and the file is "IPython Console History".

Figure 5: VedaBase Website

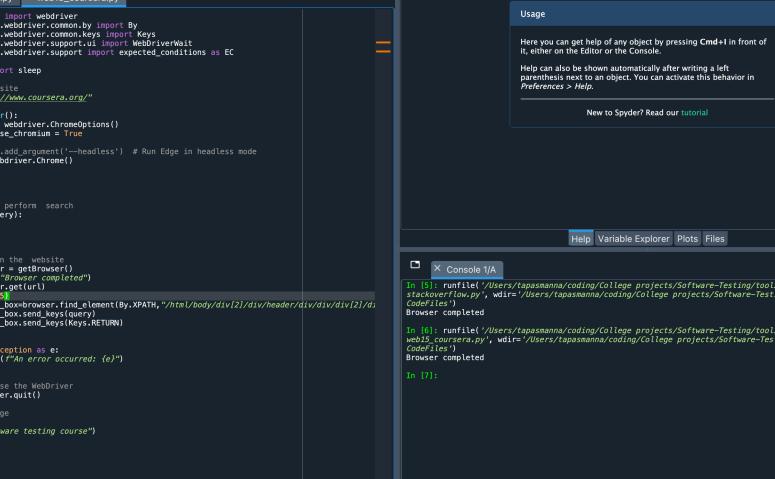
## 5.2 VedaBase : Selenium WebDriver python code

```
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.common.keys import Keys
4 from selenium.webdriver.support.ui import WebDriverWait
5 from selenium.webdriver.support import expected_conditions as EC
6
7 from time import sleep
8
9 # URL of the site
10 url = "https://vedabase.io/en/"
11
12 def getBrowser():
13     options = webdriver.ChromeOptions()
14     options.use_chromium = True
15
16     # options.add_argument('--headless') # Run chrome in headless mode
17     return webdriver.Chrome(options=options)
18
19
20
21 def readBG():
22
23
24
25     try:
26         # Open the website
```

```
27     browser = getBrowser()
28     print("Browser completed")
29     browser.get(url)
30     browser.find_element(By.XPATH, "/html[1]/body[1]/div[1]/div[3]/div[1]/div[1]/a[1]").click()
31     browser.find_element(By.XPATH, "//*[@id='bb179']/a").click()
32
33     sleep(5)
34
35
36
37     except Exception as e:
38         print(f"An error occurred: {e}")
39
40     finally:
41         # Close the WebDriver
42         browser.quit()
43
44 # Example usage
45
46
47 readBG()
```

## 6 Website 5:<https://www.coursera.org/>

## 6.1 Selenium WebDriver Screenshot coursera website.



The screenshot shows the Spyder IDE interface with the following details:

- Editor:** The left pane displays a Python script named `web15_courses.py` containing Selenium code to search for "Software testing course" on `www.coursera.org`.
- Source:** The top right pane shows the `Object` tab of the Help browser, with the "Usage" section open, providing help for the `expected_conditions` module.
- Console:** The bottom right pane shows the `Console 1/A` tab with the following history:
  - In [5]: `runfile('/Users/tapasmania/coding/College projects/Software-Testing/tool3-WebDriver/CodeFiles/stackoverflow.py', wdir='/Users/tapasmania/coding/College projects/Software-Testing/tool3-WebDriver/CodeFiles/')`
  - Browser completed
  - In [6]: `runfile('/Users/tapasmania/coding/College projects/Software-Testing/tool3-WebDriver/CodeFiles/stackoverflow.py', wdir='/Users/tapasmania/coding/College projects/Software-Testing/tool3-WebDriver/CodeFiles/')`
  - Browser completed
  - In [7]:
- Help:** A tooltip at the bottom right indicates "New to Spyder? Read our tutorial".

Figure 6: Coursera Website

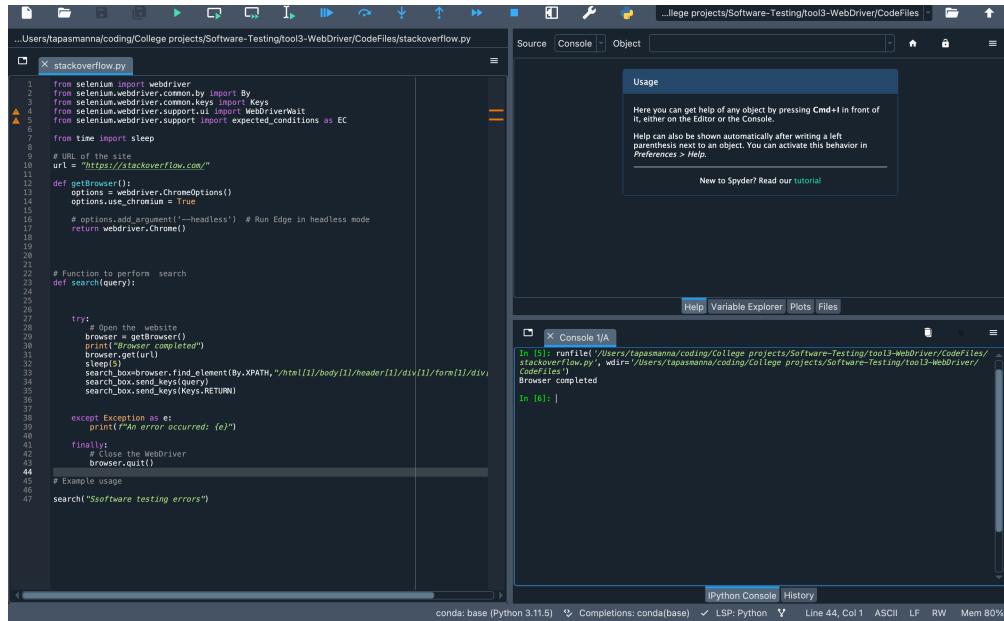
## 6.2 Coursera : Selenium WebDriver python code

```
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.common.keys import Keys
4 from selenium.webdriver.support.ui import WebDriverWait
5 from selenium.webdriver.support import expected_conditions as EC
6
7 from time import sleep
8
9 # URL of the site
10 url = "https://www.coursera.org/"
11
12 def getBrowser():
13     options = webdriver.ChromeOptions()
14     options.use_chromium = True
15
16     # options.add_argument('--headless') # Run Edge in headless mode
17     return webdriver.Chrome()
18
19
20
21
22 # Function to perform search
23 def search(query):
24
25
26
```

```
27 try:
28     # Open the website
29     browser = getBrowser()
30     print("Browser completed")
31     browser.get(url)
32     sleep(5)
33     search_box=browser.find_element(By.XPATH,"/html/body/div[2]/div/header/div/
34 div/div[2]/div[1]/div[3]/div/form/div/div[1]/input")
35     search_box.send_keys(query)
36     search_box.send_keys(Keys.RETURN)
37
38 except Exception as e:
39     print(f"An error occurred: {e}")
40
41 finally:
42     # Close the WebDriver
43     browser.quit()
44
45 # Example usage
46
47 search("Software testing course")
```

## 7 Website 6: <https://stackoverflow.com/>

### 7.1 Selenium WebDriver Screenshot stackOverFlow website.



The screenshot shows the Spyder Python IDE interface. On the left, the code editor displays a file named 'stackoverflow.py' containing Python code for a Selenium WebDriver. The code defines a function 'getBrowser()' to initialize a Chrome browser with the 'use\_chromium' option set to True. It then defines a 'search(query)' function that opens the browser, finds an element by XPATH, sends a query to the search box, and performs a search. Finally, it prints an error message if an exception occurs and closes the browser. The code also includes a comment about running Edge in headless mode. On the right, the IPython Console shows the command 'runfile()' being run, and the output shows the browser completed. The status bar at the bottom indicates the code is run in a conda:base environment with Python 3.11.5.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

from time import sleep

# URL of the site
url = "https://stackoverflow.com/"

def getBrowser():
    options = webdriver.ChromeOptions()
    options.use_chromium = True

    # options.add_argument('--headless') # Run Edge in headless mode
    return webdriver.Chrome()

# Function to perform search
def search(query):

    try:
        # Open the website
        browser = getBrowser()
        browser.get(url)
        browser.get(query)
        sleep(5)
        search_element = browser.find_element(By.XPATH, "/html[1]/body[1]/header[1]/div[1]/form[1]/div[1]/input[1]")
        search_element.send_keys(query)
        search_element.send_keys(Keys.RETURN)

    except Exception as e:
        print(f"An error occurred: {e}")

    finally:
        # Close the WebDriver
        browser.quit()

# Example usage
search("Software testing errors")
```

Figure 7: StackOverFlow Website

### 7.2 stackOverFlow : Selenium WebDriver python code

```
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.common.keys import Keys
4 from selenium.webdriver.support.ui import WebDriverWait
5 from selenium.webdriver.support import expected_conditions as EC

6
7 from time import sleep
8
9 # URL of the site
10 url = "https://stackoverflow.com/"

11
12 def getBrowser():
13     options = webdriver.ChromeOptions()
14     options.use_chromium = True

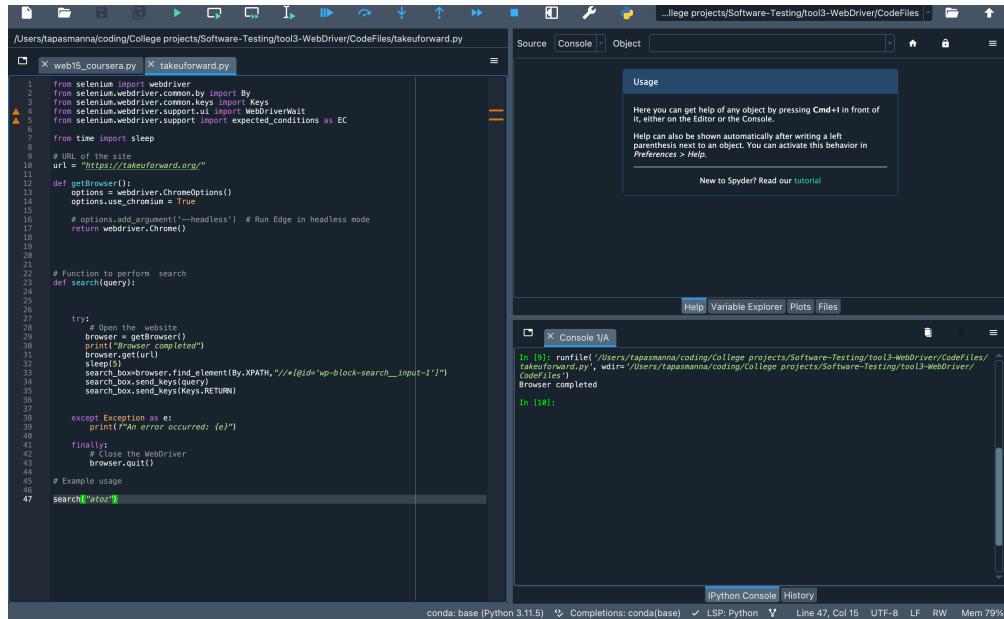
15
16     # options.add_argument('--headless') # Run Edge in headless mode
17     return webdriver.Chrome()

18
19
20
21
22 # Function to perform search
23 def search(query):
24
25
26
```

```
27 try:
28     # Open the website
29     browser = getBrowser()
30     print("Browser completed")
31     browser.get(url)
32     sleep(5)
33     search_box=browser.find_element(By.XPATH,"/html[1]/body[1]/header[1]/div
34 [1]/form[1]/div[1]/input[1]")
35     search_box.send_keys(query)
36     search_box.send_keys(Keys.RETURN)
37
38 except Exception as e:
39     print(f"An error occurred: {e}")
40
41 finally:
42     # Close the WebDriver
43     browser.quit()
44
45 # Example usage
46
47 search("Software testing errors")
```

## 8 Website 7: <https://takeuforward.org/>

### 8.1 Selenium WebDriver Screenshot takeuforward website.



The screenshot shows the Spyder IDE interface with the following details:

- Code Editor:** The left pane displays a Python script named `takeuforward.py` containing code for a Selenium WebDriver. The code includes imports for Selenium, WebDriver, and time, defines a `getBrowser` function to initialize a Chrome browser, and a `search` function to perform a search on the website. A search for "sizox" is shown at the bottom of the script.
- Console:** The right pane shows the Python console output. It includes a help dialog for the `Object` class, a message about help being shown automatically, and the command `In [10]: runfile('/Users/tapasmania/coding/College projects/Software - Testing/tool3-WebDriver/CodeFiles/takeuforward.py', wdir='/Users/tapasmania/coding/College projects/Software - Testing/tool3-WebDriver/CodeFiles')` followed by "Browser completed".
- Bottom Status Bar:** The status bar shows "conda: base (Python 3.11.5)" and "LSP: Python".

Figure 8: TakeUforward Website

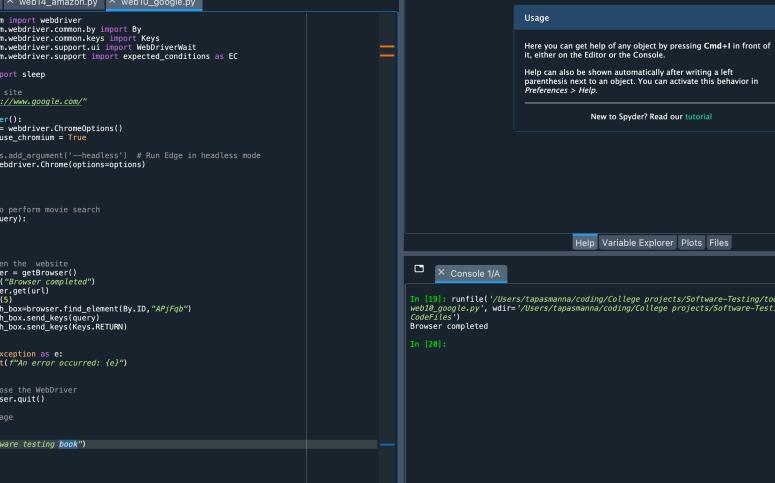
### 8.2 Takeuforward : Selenium WebDriver python code

```
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.common.keys import Keys
4 from selenium.webdriver.support.ui import WebDriverWait
5 from selenium.webdriver.support import expected_conditions as EC
6
7 from time import sleep
8
9 # URL of the site
10 url = "https://takeuforward.org/"
11
12 def getBrowser():
13     options = webdriver.ChromeOptions()
14     options.use_chromium = True
15
16     # options.add_argument('--headless') # Run Edge in headless mode
17     return webdriver.Chrome()
18
19
20
21 # Function to perform search
22 def search(query):
23
24
25
26
```

```
27 try:
28     # Open the website
29     browser = getBrowser()
30     print("Browser completed")
31     browser.get(url)
32     sleep(5)
33     search_box=browser.find_element(By.XPATH,"//*[@id='wp-block-search__input-1']")
34     search_box.send_keys(query)
35     search_box.send_keys(Keys.RETURN)
36
37
38 except Exception as e:
39     print(f"An error occurred: {e}")
40
41 finally:
42     # Close the WebDriver
43     browser.quit()
44
45 # Example usage
46
47 search("at0z")
```

## 9 Website 8:<https://google.com/>

## 9.1 Selenium WebDriver Screenshot google classroom website.



The screenshot shows the Spyder IDE interface with the following details:

- Code Editor:** The left pane displays a Python script named `untitled5.py` containing code for web testing using Selenium and WebDriver. The code includes imports for Selenium, WebDriver, and Keys, and defines functions for searching on Google and Amazon.
- IPython Console:** The right pane shows the output of the script execution. It includes a "Usage" section with documentation for the `object` command, a "Help" section, and a "New to Spyder? Read our tutorial" link. The console output shows the command `!runfile` being run, followed by the message "Browser completed".
- Toolbars and Menus:** Standard Spyder toolbars and menus are visible at the top, including File, Edit, View, Insert, Cell, Run, Help, and Spyder.
- Status Bar:** The bottom right corner shows the status bar with information like "code: base (Python 3.11.5)", "Completions: conda(base)", "LSP: Python", "Line 48, Col 30", "UTF-8", "LF", "RW", and "Mem 81%".

Figure 9: google classroom Website

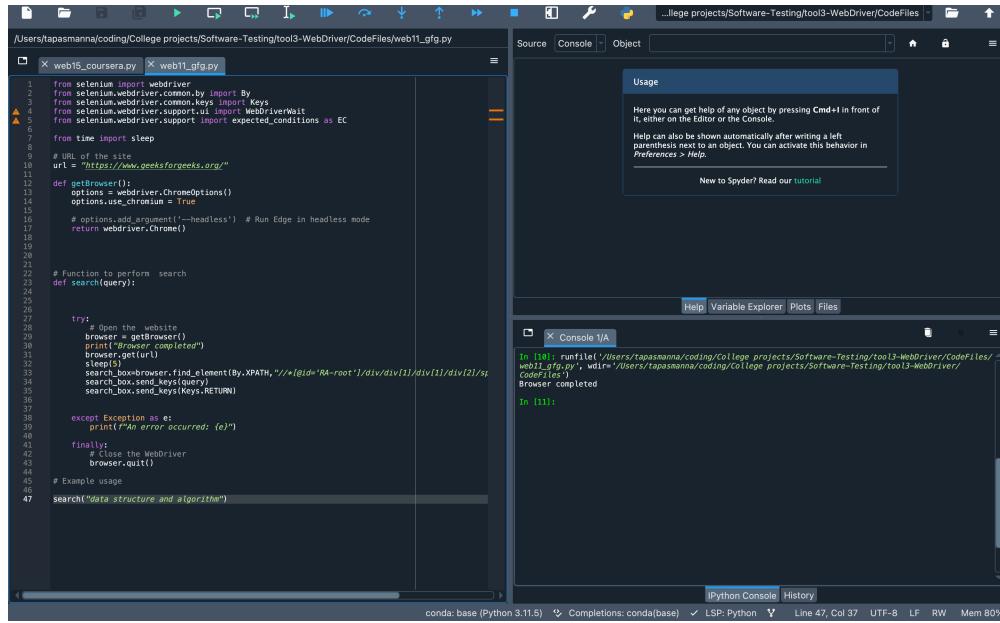
## 9.2 Geeks for Geeks Classroom : Selenium WebDriver python code

```
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.common.keys import Keys
4 from selenium.webdriver.support.ui import WebDriverWait
5 from selenium.webdriver.support import expected_conditions as EC
6
7 from time import sleep
8
9 # URL of the site
10 url = "https://www.google.com/"
11
12 def getBrowser():
13     options = webdriver.ChromeOptions()
14     options.use_chromium = True
15
16     # options.add_argument('--headless') # Run Edge in headless mode
17     return webdriver.Chrome(options=options)
18
19
20
21
22 # Function to perform movie search
23 def search(query):
24
25
26
```

```
27 try:
28     # Open the website
29     browser = getBrowser()
30     print("Browser completed")
31     browser.get(url)
32     sleep(5)
33     search_box=browser.find_element(By.ID,"APjFqb")
34     search_box.send_keys(query)
35     search_box.send_keys(Keys.RETURN)
36
37
38 except Exception as e:
39     print(f"An error occurred: {e}")
40
41 finally:
42     # Close the WebDriver
43     browser.quit()
44
45 # Example usage
46
47
48 search("Software testing book")
```

## 10 Website 9:<https://www.geeksforgeeks.org/>

### 10.1 Selenium WebDriver Screenshot gfg website.



The screenshot shows the Spyder IDE interface. On the left, the code editor displays a Python script named 'web11\_gfg.py'. The code uses the Selenium WebDriver library to navigate to the GeeksforGeeks homepage and perform a search. The right side of the interface includes a 'Usage' help panel, a 'Console' tab showing the execution of the script, and an 'IPython Console' tab at the bottom.

```
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.common.keys import Keys
4 from selenium.webdriver.support import WebDriverWait
5 from selenium.webdriver.support import expected_conditions as EC
6
7 from time import sleep
8
9 # URL of the site
10 url = "https://www.geeksforgeeks.org/"
11
12 def getBrowser():
13     options = webdriver.ChromeOptions()
14     options.use_chromium = True
15
16     # options.add_argument('--headless') # Run Edge in headless mode
17     return webdriver.Chrome()
18
19
20
21
22 # Function to perform search
23 def search(query):
24
25     try:
26         # Open the website
27         browser = getBrowser()
28         print("Browser completed")
29         browser.get(url)
30         sleep(2)
31         search_box = browser.find_element(By.XPATH, "//input[@id='q']")
32         search_box.send_keys(query)
33         search_box.send_keys(Keys.RETURN)
34
35     except Exception as e:
36         print(f"An error occurred: {e}")
37
38     finally:
39         # Close the WebDriver
40         browser.quit()
41
42
43 # Example usage
44
45 search("data structure and algorithm")
46
47
```

Figure 10: Geeks for geeks Website

### 10.2 Gfg : Selenium WebDriver python code

```
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.common.keys import Keys
4 from selenium.webdriver.support import WebDriverWait
5 from selenium.webdriver.support import expected_conditions as EC
6
7 from time import sleep
8
9 # URL of the site
10 url = "https://www.geeksforgeeks.org/"
11
12 def getBrowser():
13     options = webdriver.ChromeOptions()
14     options.use_chromium = True
15
16     # options.add_argument('--headless') # Run Edge in headless mode
17     return webdriver.Chrome()
18
19
20
21
22 # Function to perform search
23 def search(query):
24
25
```

```
27 try:
28     # Open the website
29     browser = getBrowser()
30     print("Browser completed")
31     browser.get(url)
32     sleep(5)
33     search_box=browser.find_element(By.XPATH,"//*[@id='RA-root']//div/div[1]/div[1]/div[2]/span/span/span[1]/input")
34     search_box.send_keys(query)
35     search_box.send_keys(Keys.RETURN)
36
37
38 except Exception as e:
39     print(f"An error occurred: {e}")
40
41 finally:
42     # Close the WebDriver
43     browser.quit()
44
45 # Example usage
46
47 search("data structure and algorithm")
```

## 11 Website 13:<https://www.linkedin.com/>

### 11.1 Selenium WebDriver Screenshot linkdin website.

Figure 11: Linkdin Website

### 11.2 Takeuforward : Selenium WebDriver python code

12 Website 10:<https://www.flipkart.com/>

## 12.1 Selenium WebDriver Screenshot Flipkart website.

The screenshot shows the Spyder IDE interface with the following details:

- Code Editor:** The left pane displays a Python script named `web4_flipkart.py`. The code uses Selenium WebDriver to interact with the `flipkart.com` website, specifically targeting the search bar and results page. It includes imports for Selenium, time, and expected\_conditions, and defines a function `search` that performs a search and prints the results.
- IPython Console:** The right pane shows the IPython console with the following output:

```
In [100]: runfile('/Users/tapasmania/coding/College projects/Software - Testing/tool3-WebDriver/CodeFiles/chatbot.py', wdir='/Users/tapasmania/coding/College projects/Software - Testing/tool3-WebDriver/CodeFiles'
Browser completed
In [101]:
```
- Help and Documentation:** A floating window titled "Usage" provides information on how to use the `Cmd+I` keyboard shortcut to get help for objects in the Editor.
- Toolbars and Menus:** Standard Spyder toolbars and menus like Help, Variable Explorer, Plots, and Files are visible.

Figure 12: Flipkart Website

## 12.2 Flipkart : Selenium WebDriver python code

```
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.common.keys import Keys
4 from selenium.webdriver.support.ui import WebDriverWait
5 from selenium.webdriver.support import expected_conditions as EC
6
7 from time import sleep
8
9 # URL of the site
10 url = "https://www.flipkart.com/"
11
12 def getBrowser():
13     options = webdriver.ChromeOptions()
14     options.use_chromium = True
15
16     # options.add_argument('--headless') # Run Edge in headless mode
17     return webdriver.Chrome()
18
19
20
21
22 # Function to perform search
23 def search(query):
24
25
26
```

```
27 try:
28     # Open the website
29     browser = getBrowser()
30     print("Browser completed")
31     browser.get(url)
32     sleep(5)
33     search_box=browser.find_element(By.XPATH,"/html[1]/body[1]/div[1]/div[1]/
34     div[1]/div[1]/div[1]/div[1]/div[1]/div[1]/div[1]/div[1]/div[1]/div[1]/div[1]/
35     header[1]/div[1]/div[2]/form[1]/div[1]/div[1]/input[1]")
36     search_box.send_keys(query)
37     search_box.send_keys(Keys.RETURN)
38
39 except Exception as e:
40     print(f"An error occurred: {e}")
41
42 finally:
43     # Close the WebDriver
44     browser.quit()
45
46 # Example usage
47 search("Software testing book")
```