**Software Engineering Tools Lab**

**Assignment No-9**

(Module 5- Software testing using tools) **Due date-16/05/2022**

**Name : Onkar Santosh Gavali**

**PRN: 2019BTECS00037**

# Q 1. What is Selenium? What are the features of Selenium?

1. **Selenium** is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms.
2. 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.
3. Selenium Software is not just a 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 Features

* Selenium is an open source and portable Web testing Framework.
* Selenium IDE provides a playback and record feature for authoring tests without the need to learn a test scripting language.
* It can be considered as the leading cloud-based testing platform which helps testers to record their actions and export them as a reusable script with a simple-to-understand and easy-to-use interface.
* Selenium supports various operating systems, browsers and programming languages. Following is the list:
* Programming Languages: C#, Java, Python, PHP, Ruby, Perl, and JavaScript
* Operating Systems: Android, iOS, Windows, Linux, Mac, Solaris.
* Browsers: Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera, Safari, etc.
* It also supports parallel test execution which reduces time and increases the efficiency of tests.
* Selenium can be integrated with frameworks like Ant and Maven for source code compilation.
* Selenium can also be integrated with testing frameworks like TestNG for application testing and generating reports.
* Selenium requires fewer resources as compared to other automation test tools.
* WebDriver API has been indulged in selenium whichis one of the most important modifications done to selenium.
* Selenium web driver does not require server installation, test scripts interact directly with the browser.
* Selenium commands are categorized in terms of different classes which make it easier to understand and implement.
* Selenium Remote Control (RC) in conjunction with WebDriver API is known as Selenium 2.0. This version was built to support the vibrant web pages and Ajax.

Q 2. Install Selenium tool on your machine (Provide screen shots for successful installation)



### Q 3. Using Selenium tool create test cases as given below for a web application GoDaddy.com

### Automate Browser Actions on GoDaddy.com with Selenium

#### Test Case - Open Godaddy.com and maximize browser window.

Steps to Automate:

1. Launch browser of your choice say, Firefox, chrome etc.
2. Open this URL - <https://www.godaddy.com/>
3. Maximize or set size of browser window.
4. Close browser.

Code –

from selenium import webdriver

from selenium import  webdriver

from selenium.webdriver.common.by import By

from selenium.common.exceptions import NoSuchElementException

from webdriver\_manager.chrome import ChromeDriverManager

import time

from selenium.webdriver.common.keys import Keys

# Creating an instance webdriver

option = webdriver.ChromeOptions()

browser = webdriver.Chrome(ChromeDriverManager().install(),options=option)

browser.get(' https://www.godaddy.com/')

time.sleep(2)

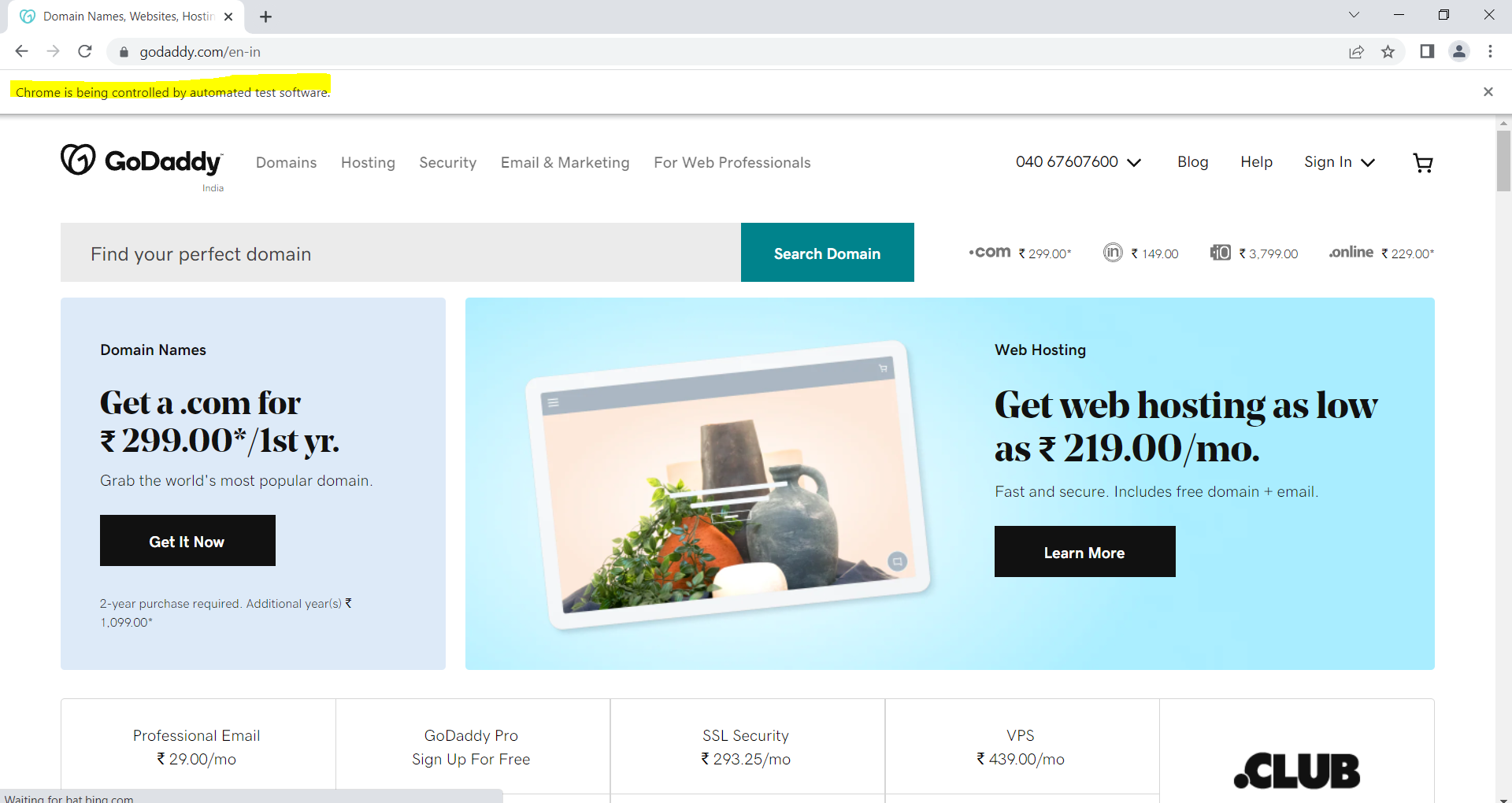
browser.maximize\_window()

time.sleep(10)

# closing the browser

browser.close()

Output



#### 2. Test Case - Open Godaddy.com and Print it's Page title.

#### Steps to Automate:

#### Launch browser of your choice say., Firefox, chrome etc.

#### Open this URL - <https://www.godaddy.com/>

#### Maximize or set size of browser window.

#### Get Title of page and print it.

#### Get URL of current page and print it.

#### Close browser.

Code-

from lib2to3.pgen2 import driver

from selenium import webdriver

from selenium import  webdriver

from selenium.webdriver.common.by import By

from selenium.common.exceptions import NoSuchElementException

from webdriver\_manager.chrome import ChromeDriverManager

import time

from selenium.webdriver.common.keys import Keys

# Creating an instance webdriver

option = webdriver.ChromeOptions()

browser = webdriver.Chrome(ChromeDriverManager().install(),options=option)

browser.get(' https://www.godaddy.com/')

time.sleep(2)

browser.maximize\_window()

time.sleep(10)

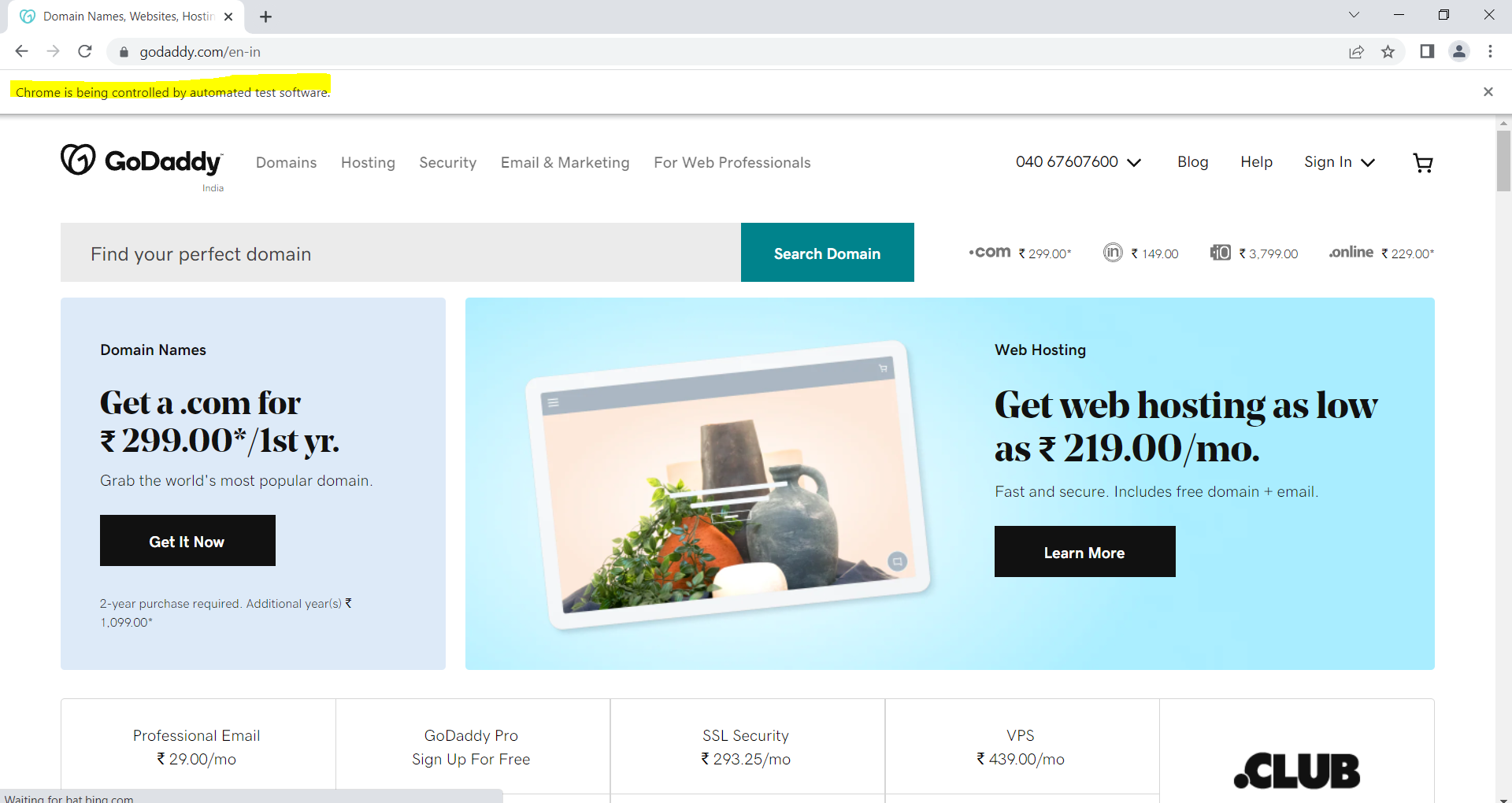
print("Title : ",browser.title)

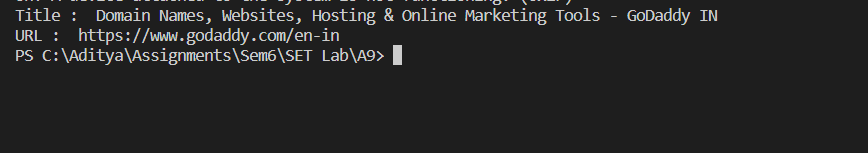
print("URL : ",browser.current\_url)

# closing the browser

browser.close()

Output –





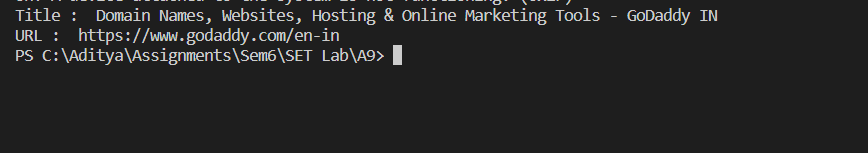
#### 3. Test Case - Open Godaddy.com and Validate Page Title

Steps to Automate:

1. Launch browser of your choice say., Firefox, chrome etc.
2. Open this URL - <https://www.godaddy.com/>
3. Maximize or set size of browser window.
4. Get Title of page and validate it with expected value.
5. Get URL of current page and validate it with expected value.
6. Get Page source of web page.
7. And Validate that page title is present in page source.
8. Close browser
9. from lib2to3.pgen2 import driver
10. from selenium import webdriver
11. from selenium import  webdriver
12. from selenium.webdriver.common.by import By
13. from selenium.common.exceptions import NoSuchElementException
14. from webdriver\_manager.chrome import ChromeDriverManager
15. import time
16. from selenium.webdriver.common.keys import Keys
17. # Creating an instance webdriver
18. option = webdriver.ChromeOptions()
19. browser = webdriver.Chrome(ChromeDriverManager().install(),options=option)
20. browser.get(' https://www.godaddy.com/')
21. time.sleep(2)
22. browser.maximize\_window()
23. time.sleep(10)
24. print("Title : ",browser.title)
25. print("URL : ",browser.current\_url)

# closing the browser

1. browser.close()



### Automate GoDaddy.com website menu links with Selenium

#### 1. Test Case - Automate the first menu link of godaddy.com with Selenium.

Steps to Automate:

1. Launch browser of your choice like, firefox, chrome etc., and using selenium webdriver.
2. Open website url - <https://godaddy.com/>
3. Maximize browser window.
4. Set timeout using implicit wait command of Selenium Webdriver.
5. Click on the first menu link, which is 'Domains'. It will open up a sub- menu, click on the 'Domain Name Search' link from the sub-menu.
6. Now, we'll get the value of the Page title manually before automating, just to know what should be the expected out put of our script. Following are the steps to automate it:

* Right click on Domain Name Search page and click on 'Inspect' option.
* It'll open up the html viewer of that page.
* Under the 'Elements' tab, search for "<head>" tag and if you find it. Then search for "<title>" tag.
* Copy the text written inside pair of "<title> and </title>" tags.
* In the following pic, value of the title is "Domain Name Search | Check Domain Availability - GoDaddy IN". This is should be our expected value.

1. Get the value of title of 'Domain Name Search' page using Selenium Webdriver's command in your script and print it.
2. We already know the expected value, we'll match value fetched in step 7 with our expected value, if it matches print pass either fail.

Code –

from lib2to3.pgen2 import driver

from xml.dom.minidom import Element

# from xml.etree import cElementTree

from selenium import webdriver

import time

# Here Chrome will be used

from selenium import webdriver

from selenium import  webdriver

from selenium.webdriver.common.by import By

from selenium.common.exceptions import NoSuchElementException

from webdriver\_manager.chrome import ChromeDriverManager

import time

from selenium.webdriver.common.keys import Keys

# Creating an instance webdriver

option = webdriver.ChromeOptions()

driver = webdriver.Chrome(ChromeDriverManager().install(),options=option)

# driver = webdriver.Chrome()

# URL of website

url = "https://www.godaddy.com/"

# Opening the website

# driver.set\_page\_load\_timeout(5)

driver.get(url)

# driver.implicitly\_wait(0.8);

#Q 1.1 Getting current URL source code

# get\_title = driver.title

# print(get\_title)

# driver.implicitly\_wait(2);

#Q 1.2. Validating Domain Name :

# if(get\_title == "Domain Names, Websites, Hosting & Online Marketing Tools - GoDaddy IN"):

#   print("Verified Domain Name")

# else :print("Not Verified")

#hamuburger

element  = driver.find\_element\_by\_xpath("/html/body/header/div/div[1]/nav/button");

time.sleep(2);

element.click();

element = driver.find\_element\_by\_xpath("/html/body/header/div/div[1]/nav/div[2]/div/ul/li[1]/span");

time.sleep(2);

element.click();

# /html/body/header/div/div[1]/nav/div[2]/div/ul/li[1]/span

element = driver.find\_element\_by\_xpath("/html/body/header/div/div[1]/nav/div[2]/div/ul/li[1]/div/div/div[1]/ul/li[2]/a");

time.sleep(2);

element.click();

get\_title = driver.title

print(get\_title)

# if(get\_title == "Domain Name Search - Buy and Register Available Domains - GoDaddy IN"):

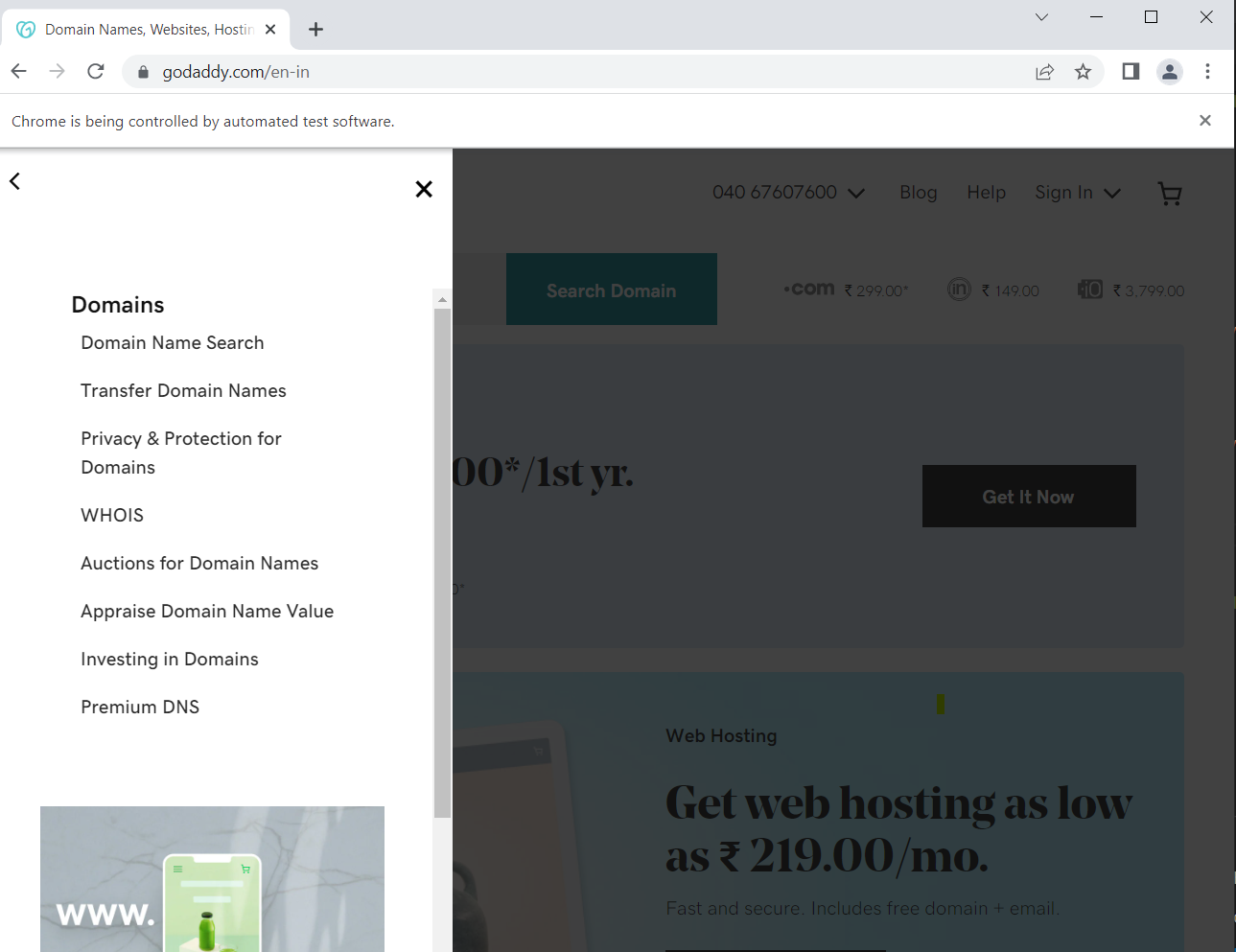
#     print("Validated title: ",get\_title)

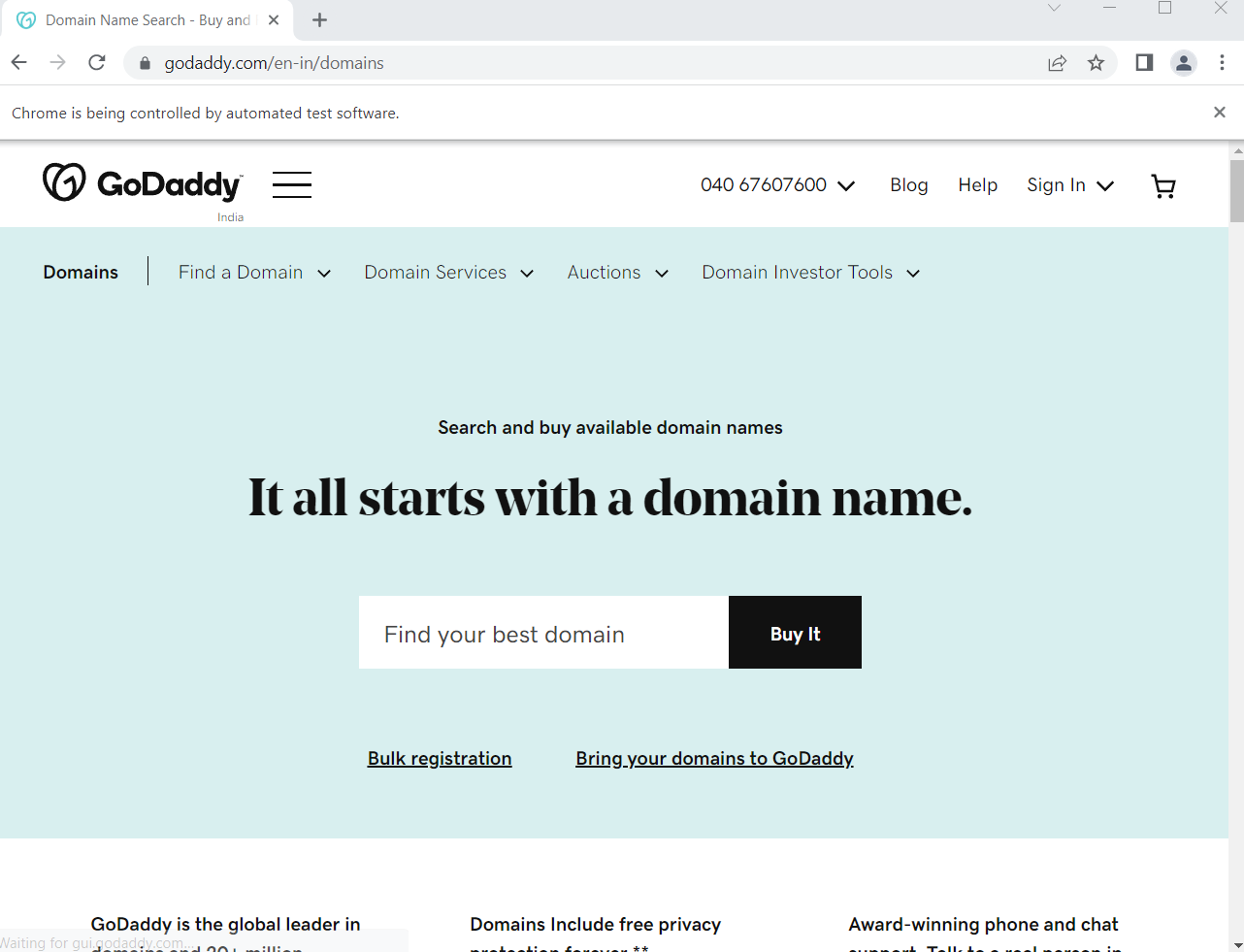
# else:

#     print("Not validated")

# driver.quit()

Output –





#### 2. Test Case - Automate all the menu links of godaddy.com with Selenium.

Steps to Automate:

* 1. Launch browser of your choice like, firefox, chrome etc.,using selenium webdriver.
  2. Open website url - https://godaddy.com/
  3. Maximize browser window.
  4. Set timeout using implicit wait command of Selenium Webdriver.
  5. Click on the first link from the menu, it will open up the sub-menu with multiple sub-menu links.
  6. Click on the first sub-menu link.
  7. Get the page title and validate it.
  8. Go back to the home page by clicking on the GoDaddy logo.
  9. Repeat the steps 5 to 8 and cover all the menu and sub-menu items present one by one.

Code –

from selenium import webdriver

from selenium import  webdriver

from selenium.webdriver.common.by import By

from selenium.common.exceptions import NoSuchElementException

from webdriver\_manager.chrome import ChromeDriverManager

import time

from selenium.webdriver.common.keys import Keys

# Creating an instance webdriver

option = webdriver.ChromeOptions()

driver = webdriver.Chrome(ChromeDriverManager().install(),options=option)

driver.get('https://www.godaddy.com/en-in')

#first

print("\nFirst Menu Link\n");

element = driver.find\_element(By.ID,"id-3a34578f-40f3-4d2f-aaa4-0a1320f3fe6a");

driver.execute\_script("arguments[0].click();", element)

time.sleep(7);

print("Title of the WebPage= "+driver.title)

print("Link of the Webpage= "+driver.current\_url)

element = driver.find\_element(By.ID,"id-b1c657aa-b81f-4909-86bb-e349818d2dfb");

driver.execute\_script("arguments[0].click();", element)

time.sleep(7);

print("Title of the WebPage= "+driver.title)

print("Link of the Webpage= "+driver.current\_url)

#second

print("\nSecond Menu Link\n")

element = driver.find\_element(By.ID,"id-7c8834e5-3239-457a-93d1-9e7432567e4a");

driver.execute\_script("arguments[0].click();", element)

time.sleep(7);

print("Title of the WebPage= "+driver.title)

print("Link of the Webpage= "+driver.current\_url)

element = driver.find\_element(By.ID,"id-b1c657aa-b81f-4909-86bb-e349818d2dfb");

driver.execute\_script("arguments[0].click();", element)

time.sleep(7);

print("Title of the WebPage= "+driver.title)

print("Link of the Webpage= "+driver.current\_url)

#third

print("\nThird Menu Link\n")

element = driver.find\_element(By.ID,"id-7f7e18ab-b77d-44bb-80f5-09045b8b8d61");

driver.execute\_script("arguments[0].click();", element)

time.sleep(7);

print("Title of the WebPage= "+driver.title)

print("Link of the Webpage= "+driver.current\_url)

element = driver.find\_element(By.ID,"id-b1c657aa-b81f-4909-86bb-e349818d2dfb");

driver.execute\_script("arguments[0].click();", element)

time.sleep(7);

print("Title of the WebPage= "+driver.title)

print("Link of the Webpage= "+driver.current\_url)

#forth

print("\nFourth Menu Link\n")

element = driver.find\_element(By.ID,"id-ffacc702-3466-40a7-b5b6-03b909e11509");

driver.execute\_script("arguments[0].click();", element)

time.sleep(7);

print("Title of the WebPage= "+driver.title)

print("Link of the Webpage= "+driver.current\_url)

element = driver.find\_element(By.ID,"id-b1c657aa-b81f-4909-86bb-e349818d2dfb");

driver.execute\_script("arguments[0].click();", element)

time.sleep(7);

print("Title of the WebPage= "+driver.title)

print("Link of the Webpage= "+driver.current\_url)

driver.close()

Output –

