

Report 2

by Upendar Reddy

Submission date: 30-Apr-2022 11:02PM (UTC+0530)

Submission ID: 1824762749

File name: report_2.docx (1.23M)

Word count: 2585

Character count: 15427

1

CAPSTONE PROJECT REPORT

(Project Term January-May 2022)

Amazon Automation

Submitted by

MohithVenkata Sai Vemula- 11802939

Naveen ChanduParitala- 11805079

Upendar Reddy Baireddy- 11805157

BommaPreetham- 11805205

5

Project Group Number

CSERGC0148

Course Code :CSE445

Under the Guidance of

Ms. Puneet

(Assistant Professor)

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

June, 2019



L OVELY
P ROFESSIONAL
U NIVERSITY

**TOPIC APPROVAL PERFORMA**

School of Computer Science and Engineering (SCSE)

Program : P132::B.Tech. (Computer Science & Engineering)

COURSE CODE : CSE445

REGULAR/BACKLOG : Regular

GROUP NUMBER : CSERGC0148

Supervisor Name : Puneet

UID : 27394

Designation : Assistant Professor

Qualification : _____

Research Experience : _____

SR.NO.	NAME OF STUDENT	Prov. Regd. No.	BATCH	SECTION	CONTACT NUMBER
1	Bomma Preetham	11805205	2018	K18GE	9121211496
2	Baireddy Upendar Reddy	11805157	2018	K18GE	6302915887
3	Paritala Naveen Chandu	11805079	2018	K18CJ	9063826172
4	Vemula Mohith Venkata Sai	11802939	2018	K18CJ	7732001827

SPECIALIZATION AREA : Software Engineering

Supervisor Signature: _____

PROPOSED TOPIC : Amazon Automation

Qualitative Assessment of Proposed Topic by PAC		
Sr.No.	Parameter	Rating (out of 10)
1	Project Novelty: Potential of the project to create new knowledge	6.75
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	7.00
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	7.25
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	7.50
5	Social Applicability: Project work intends to solve a practical problem.	6.75
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	6.50

PAC Committee Members		
PAC Member (HOD/Chairperson) Name: Dr. Dalwinder Singh	UID: 11265	Recommended (Y/N): Yes
PAC Member (Allied) Name: Asha Rani	UID: 11332	Recommended (Y/N): Yes
PAC Member 3 Name: Dr. Parampreet Kaur	UID: 18758	Recommended (Y/N): Yes

Final Topic Approved by PAC: Amazon Automation

Overall Remarks: Approved

PAC CHAIRPERSON Name: 14307::Raj Karan Singh

Approval Date: 11 Mar 2022

4/26/2022 3:45:46 PM

1 **DECLARATION**

We hereby declare that the project work entitled "Amazon Automation" is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in Computer Science from Lovely Professional University, Phagwara, under the guidance of Ms. Puneet during January-May 2022. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Project Group Number: CSERGC0148

Name of Student 1: Mohith Venkata Sai Vemula

Registration Number: 11802939

Name of Student 2: Naveen Chandu Paritala

Registration Number: 11805079

Name of Student 3: Upendar Reddy Baireddy

Registration Number: 11805157

Name of Student 4: Bomma Preetham

Registration Number: 11805205

(Mohith)

Date: 19/04/2022

(Naveen)

Date: 19/04/2022

(Upendar)

Date: 19/04/2022

(Preetham)

Date: 19/04/2022

¹CERTIFICATE

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in Computer Science from Lovely Professional University, Phagwara.

Signature and Name of the Mentor

Puneet

Designation

Assistant Professor

School of Computer Science and Engineering,

Lovely Professional University,

Phagwara, Punjab.

Date :19/04/2022

1 **TABLE OF CONTENTS**

Inner first page.....	(i)
PAC form.....	(ii)
Declaration.....	(ii)
Certificate.....	(iii)
Table of Contents.....	(iv)

1. Introduction

- 1.1 What is Selenium?**
- 1.2 Selenium Components**
- 1.3 What is Selenium WebDriver?**
- 1.4 Drawbacks of Selenium WebDriver**

2. Project Overview

- 2.1 What is Automation?**
- 2.2 Project Description**
 - 2.2.1 Importing the necessary libraries**
 - 2.2.2 Accessing the WebDriver and opening the website**
 - 2.2.3 Maximize Window**
 - 2.2.4 Sign-in or Sign-up page**
 - 2.2.5 Search item and display the item**
 - 2.2.6 Add item to Cart**
 - 2.2.7 Your Account**

3. Source Code

4. Challenges Faced and Areas of

- 4.1 Challenge faced in Sign Up page**
- 4.2 Concern in Sign Up page**
- 4.3 Empty fields in Cost and Rating**

1. INTRODUCTION

2 1.1 What is Selenium?

Selenium refers to a suite of tools that are widely used in the testing community when it comes to cross-browser testing. Selenium cannot automate desktop applications; it can only be used in browsers. It is considered to be one of the most preferred tool suites for automation testing of web applications as it provides support for popular web browsers which makes it very powerful.

It supports a number of browsers

Google Chrome 12+, Internet Explorer 7,8,9,10, Safari 5.1+, Opera 11.5, Firefox 3+) and operating systems (Windows, Mac, Linux/Unix.

Selenium also provides compatibility with different programming languages –

C#, Java, JavaScript, Ruby, Python, PHP.

Testers can choose which language to design test cases in, thus making Selenium highly favorable for its flexibility.

2 1.2 Selenium Components

The Selenium test suite comprises four main components:-

- Selenium IDE
- Selenium RC
- Selenium WebDriver
- Selenium Grid

From the Given four components of Selenium we will use the 3rd component i.e. , Selenium WebDriver in our entire project.

2 1.3 What is Selenium WebDriver?

Selenium WebDriver is a web framework that permits you to execute cross-browser tests. This tool is used for automating web-based application testing to verify that it performs expectedly.

Selenium WebDriver allows you to choose a programming language to create test scripts. It is an advancement over Selenium RC to overcome a few limitations.

1.4 Drawbacks of Selenium WebDriver

2
Selenium WebDriver is not capable of handling window components, but this drawback can be overcome by using tools like Sikuli, Auto IT, etc.

2. Project Overview

The title of the project is Amazon Automation – Guide for Amazon. Automation is the term which we regularly keep hearing in our daily lives.

2.1 What is Automation?

Automation can be defined as a technique of making a process or a system operate automatically (less human interaction). When we automate a process or a task it reduces the effort and consumes less time.

2.2 Project Description

When we open Amazon website or app if you are an old user you sign in using credentials, search item if we want to order any or add item to cart for later purchase. Then if we want to update our account details we go into 'your account' and update necessary information and join prime if you are interested. However if you are a new user you create an account first.

The entire project works on amazon website (www.amazon.in) with the help of python language. We use the specific libraries to run commands.

This project helps you just like a guide by bringing all the information and tasks together at one place, all you need to do is select the option from the list and the task is performed for you. Making the technology easily usable and user friendly is the main objective of this project.

- We start by installing the Selenium Library .

```
pip install -U selenium
```

- Then we download the Chrome Driver that is compatible with the current version of our chrome browser using the link below.

```
https://chromedriver.chromium.org/downloads
```

Now we can start our program in any app we are comfortable with. I use the PyCharmapp in my project.

2.2.1 Importing the necessary libraries

```
from selenium import webdriver
from selenium.webdriver.support.select import Select
import time
```

2.2.2 Accessing the WebDriver and opening the website

4

We create a variable `driver` and assign the downloaded `chromedriver` extension to it. By using `“get”` we can open the website.

```
driver = webdriver.Chrome('chromedriver.exe')
driver.get('https://www.amazon.in/')

```

2.2.3 Maximize Window

After successfully completing step 1 and step 2, when we run it a new automated page is opened to maximize that page we need to use maximize command

```
driver.maximize_window()

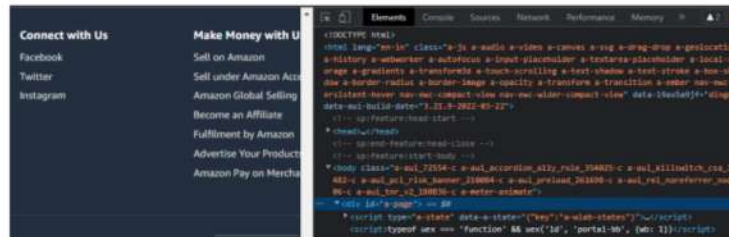
```

4

Now finally we get a chrome page with heading `chrome is being controlled by automated test software`.

Now to automate the website we need to inspect the website that we are using, In our case we need to right click on any spot in amazon website and the click on the inspect option. This gives you the code used for amazon website.

Whenever you inspect particular field in website the code is given to your right side of screen as shown in the picture.



2.2.4 Sign-in or Sign-up page

When we use any shopping website or any other website, we can access the features to its full extent only when we have an account. So let's start by automating the sign in/ sign up page.

In order to get the code for some field in website we need to right click on that particular field and click on inspect, then the tag name, class name or id of the field is displayed. We can access this in our code by using the variable we created earlier. The `“driver”` variable is used with `“find_element”` in the following way.

```
driver.find_element_

```

There are various ways to access the fields in a website, such as

- Id
- Class

- X-path
- Name
- Link_Text

We can use both `find_elements` and `find_element` but its use depends on various situations. When we use a for loop in the condition we give `"find_elements"` and inside the loop we give `"find_element"`

The Picture below shows all the available elements(ways to access):

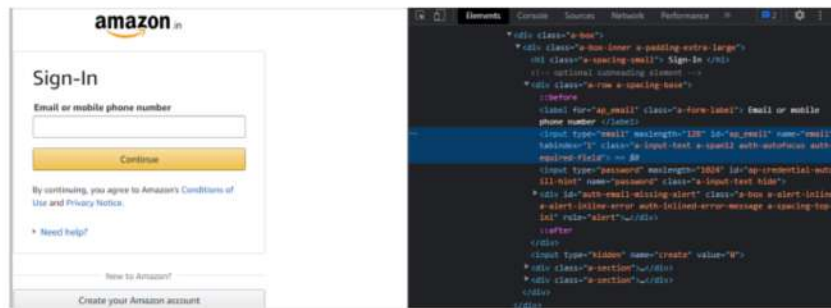
```
m find_element_by_css_selector(self, css_select... WebDriver
m find_element_by_xpath(self, xpath)           WebDriver
m find_element_by_id(self, id_)                 WebDriver
m find_elements_by_css_selector(self, css_selec... WebDriver
m find_element_by_class_name(self, name)         WebDriver
m find_element_by_link_text(self, link_text)     WebDriver
m find_element_by_name(self, name)               WebDriver
m find_element_by_partial_link_text(self, link... WebDriver
m find_element_by_tag_name(self, name)           WebDriver
m find_elements_by_xpath(self, xpath)            WebDriver
m find_elements_by_tag_name(self, name)          WebDriver
m find_elements_by_id(self, id_)                 WebDriver
m find_elements_by_class_name(self, name)        WebDriver
m find_elements_by_link_text(self, text)         WebDriver
m find_elements_by_name(self, name)              WebDriver
m find_elements_by_partial_link_text(self, link... WebDriver
```

Sign-In Page of Amazon:

If you are having an account in amazon you can run the commands for sign in page and in case if you forget the password we can run the commands for forget password.

When we inspect the code for the input column in sign in page the class or id the elements that are available for the field is given. When we hover the mouse onto the highlighted code, the class of the input box is displayed above the box with a `."` and id is displayed with `"#"`.





After accessing the input field we enter details in the field using `“.send_keys”` and then to click on continue button we need to use `“.click()”` after accessing the button with the class or id.

Code for Sign-In page:

```
login_no = driver.find_element_by_id('ap_email')
login_no.send_keys(1234567890)

continue_button = driver.find_element_by_id('continue')
continue_button.click()

login_pass = driver.find_element_by_id('ap_password')
login_pass.send_keys('Your Password')

sign_in_submit = driver.find_element_by_id('signInSubmit')
sign_in_submit.click()
```

Code for Forget Password:

```
# Click on forget password
driver.find_element_by_id('auth-fpp-link-bottom').click()
driver.find_element_by_id('continue').click()

# Enter otp
otp = driver.find_element_by_id('cvf-input-code')
3.p.send_keys(input("Enter the otp received to phone: "))
driver.find_element_by_css_selector('.a-button-input').click()

# Enter new password
pass_new = driver.find_element_by_id('ap_fpp_password')
pass_new.send_keys(input("Enter new password: "))
# Retype the same password
pass_new_2 = driver.find_element_by_id('ap_fpp_password_check')
pass_new_2.send_keys(input("Enter new password again: "))
driver.find_element_by_id('continue').click()
```

The same way by accessing every required field we can automate the Sign Up page.

Code for Sign-Up page:

```
driver.find_element_by_css_selector('#createAccountSubmit').click()
time.sleep(1)

name = driver.find_element_by_id('ap_customer_name')
```

We can either enter details or take input from user using input command

```
name_ = input("Enter Name that is unique: ")
```

When you try to automate the sign-up page the issue we face is that there are 2 different types of sign up page and they open at random.

So we need to automate both the pages using "if - else" condition.

Code for 1st type of sign up page

```
for label in driver.find_elements_by_css_selector('.a-form-label'):
    if (label.text) == 'Mobile number or email':

        email = driver.find_element_by_id('ap_email')
        # email_ = input('Enter email: ')
        email.send_keys('*****@gmail.com')
        time.sleep(1)
        pass_ = driver.find_element_by_id('ap_password')
        # pass_ = input('Enter password 8 characters long with a number and Capital: ')
        pass_.send_keys('*****')
        time.sleep(1)
        pass_1 = driver.find_element_by_id('ap_password_check')
        # pass_1 = input('Re enter the same password: ')
        pass_1.send_keys('*****')
        time.sleep(1)
```

In this case the otp is sent to your mail.

We can enter otp using code

```
otp1 = (otp_type.find_element_by_id('cvf-input-code'))
otp2 = int(input("Enter the otp received to your mail:"))
otp1.send_keys(otp2)
time.sleep(2)
```

Then we can submit otp using

```
otp_type.find_element_by_id('auth-verify-button').click()
```

Code for 2nd type of sign up page

```
elif (label.text) == 'Mobile number':

    phone_no = driver.find_element_by_id('ap_phone_number')
    # phone_no = int(input('Enter mobile number never used before: '))
```

```

phone_no.send_keys(1234567890)
time.sleep(1)

email = driver.find_element_by_id('ap_email')
# email = input('Enter email: ')
email.send_keys('*****@gmail.com')
time.sleep(1)

pass_ = driver.find_element_by_id('ap_password')
# pass_ = input('Enter password 8 characters long with a number and Capital: ')
pass_.send_keys('*****')
time.sleep(1)

```

In this case the otp is sent to your mobile.

```

otp1 = (otp_type.find_element_by_id('auth-pv-enter-code'))
# otp2 = int(input("Enter the otp received to your mobile:"))
otp1.send_keys(123456)
time.sleep(2)
otp_type.find_element_by_id('auth-verify-button').click()

```

2.2.5 Search item and display the item link in excel

In this step we search for any item in the search box using the code and then we display the name, cost, rating and link of the item searched in excel sheet.

Code for extracting item name:

```

for item in driver.find_elements_by_css_selector('s-result-item.s-asin'):
    item_name = item.find_element_by_css_selector('a-color-base a-text-normal')
    print(item_name.text)

```

Code for extracting item cost:

```

for item in driver.find_elements_by_css_selector('s-result-item.s-asin'):
    try:
        item_cost = item.find_element_by_css_selector('a-price-whole')
        print(f'{item_cost.text}\n')
    except:
        print("Cost is not given\n")

```

After getting the details we can create a csv file and add details to file using the command.

```

with open("Amazon_details.csv", 'w', encoding='utf-8') as f:

```

Code for extracting item link:

```

for item in driver.find_elements_by_css_selector('s-result-item.s-asin'):
    item_link = item.find_element_by_tag_name("a").get_attribute('href')
    print(item_link)

```


2.2.6 Add item to Cart

To add an item to the cart, we need to open the item in a new page and then click on add to cart

lst has all the links of items shown in first page after search.

```
lst = list(map(str, lst.split()))
for i in range(3):
    driver.get('3' + lst[i])
    7 id_cart = driver.find_element_by_id('add-to-cart-button')
    time.sleep(2)
    item_name_ = driver.find_element_by_id('productTitle')

    # name
    print(item_name_.text)

    # link
    print(lst[i])

    # rating: 3 6
    for j in driver.find_elements_by_css_selector('.a-section.a-spacing-none.a-spacing-top-mini'
                                                  '.cr-widget-ACR'):
        try:
            3
            rating = j.find_element_by_css_selector('a-size-medium.a-color-base')
            print(rating.text)
            time.sleep(2)

        except:
            print("Rating is not given for this item")

    add_cart.click()
    print("Item successfully added to cart")
    print(" * * 200")
    time.sleep(3)

    print("\nDetails successfully obtained...\n")
    driver.find_element_by_xpath('//*[@id="attach-close_sideSheet-link"]').click()
```

2.2.7 Your Account

In amazon to access your account we need to visit 'your account' option. When we click on your account six different choices are displayed.

They are:

1. Your Orders
2. Login and Security
3. Prime Account
4. Address
5. Payment Options
6. Amazon Pay Balance

1. Your Orders:

```
print('1. Orders\n2. Buy again\n3. Not yet shipped\n4. Cancelled Orders\n5. Go back')
your_ord_inp = int(input('Enter any option from list: '))
```

We will be asking the user to select necessary options and then take them to the particular page. This process is same for all the pages.

We use 'Select' to work on dropdown button. There are 3 ways to access the dropdown values.

```
drop_d_past = Select(driver.find_element_by_css_selector
('a-native-dropdown'))
```

1. Select by Index:

```
drop_d_past.select_by_index(i)
```

2. Select by Value:

```
drop_d_past.select_by_value('Value')
```

3. Select by Visible_text:

```
drop_d_past.select_by_visible_text('Visible Text')
```

2. Login and Security:

After entering this page we get several options regarding the content such as

```
print("Login and security:\n\n")
"1. Edit Name\n"
"2. Edit Mail id\n"
"3. Edit Mobile Number\n"
"4. Password\n"
"5. 2 Step Authentication\n"
"6. Secure Account\n"
"7. Exit Page")
```

After entering any number from the above list we will be taken to that page and then we can update the information or else if you want to leave you can enter "7" it takes back to homepage

3. Prime Account:

In this page you can add a prime account based on packages and access the terms and conditions.

4. Address:

Here we can add new address or update the existing address or remove it.

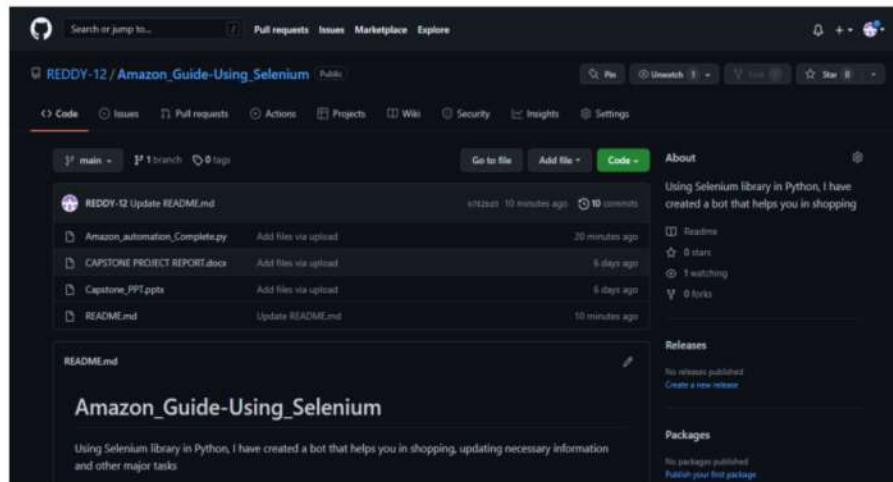
The same way Payment Options and Amazon Pay balance are automated.

3. Source Code:

The code is mentioned in the below link. You can find the file there by the name "Amazon_automation_Complete.py"

https://github.com/REDDY-12/Amazon_Guide-Using_Selenium

Use the below picture as the reference to find the file faster



4. Challenges Faced and Areas of concern

4.1 Challenge faced in Sign Up page

When we started automating the sign up page, the first challenge we faced is that there were 2 different types of login forms which open randomly every time we open the page. The solution to solve this problem is if-else loop. We created loop to enter data based on the sub - heading of the form.

4.2 Concern in Sign Up page

Every website uses captcha to stop the robot from operating the website. In our case after submitting the sign up form there is a puzzle popping up to restrict the bot. As the main purpose of the puzzle is to restrict the bot, we were unable to automate the captcha. So the user needs to enter the otp directly in the automated page created. After creating an account user can come back and run the bot again and go for sign in.

4.3 Empty fields in Cost and Rating

When we see the details of search items in amazon not all items have cost and rating displayed. In this case when we fetch the cost or rating we get an error when the field is empty.

So to overcome this issue we used `try - except` block. This block helps us to deal with the exception and run the program without any interruption.

4.4 Infinite loop after Sign Out

After signing in successfully and after performing all the necessary actions, if you click on sign out you will be logged out of the account but the program prints one line infinitely and we were unable to rectify the issue but if you do the same thing after entering through forget password the code exits successfully.

5. Conclusion

- When a person is new to any website he/she doesn't know where to start with. We usually need assistance from someone.
- Using automation we can make the interaction with the website much easier to the users without any external assistance. It is possible to automate entire website with ease using selenium.
- The sole purpose of "Automation" is to reduce the human effort.

- Though we can automate almost everything, there are some tabs that can never be automated, the reason being the purpose of those tags is to keep bots from using the site.
- Overall using automation can make us more closer to the technical world.

Report 2

ORIGINALITY REPORT

19%

SIMILARITY INDEX

16%

INTERNET SOURCES

4%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

www.coursehero.com

Internet Source

8%

2

www.browserstack.com

Internet Source

7%

3

Sujay Raghavendra. "Python Testing with Selenium", Springer Science and Business Media LLC, 2021

Publication

2%

4

Art Yudin. "Chapter 4 Gathering Data with Python", Springer Science and Business Media LLC, 2021

Publication

1%

5

www.scribd.com

Internet Source

1%

6

top800.radiohamburg.de

Internet Source

<1%

7

Mikhail Demianenko, Carlo Iapige De Gaetani. "A Procedure for Automating Energy Analyses in the BIM Context Exploiting Artificial Neural

<1%

Networks and Transfer Learning Technique", Energies, 2021

Publication

8

Koushal Kumar, Gour Sundar Mitra Thakur.
"Advanced Applications of Neural Networks
and Artificial Intelligence: A Review",
International Journal of Information
Technology and Computer Science, 2012

Publication

<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography On