American International University-Bangladesh (AIUB)
Department of Computer Science
Faculty of Science &Technology (FST)
Summer 22-23

Section: A
Software Quality and Testing

Selenium Software Testing

A Report submitted
By

SN	Student Name	Student ID
1	Ananya Chowdhury	18-39028-3
2	Tanzima Zahir	18-38958-3
3	MD. Ashraf Hossain Rifat	18-38928-3
4	Israt Jahan Pritha	18-39023-3

Under the supervision of

Abhijit Bhowmik
Department of Computer Science
Faculty of Science and Information Technology
American International University-Bangladesh

Software: Selenium

Installing Selenium: Before installing Selenium, we have to install Python first. As we have installed Python in our pc so we just skip that. Now for installing selenium we open Command Prompt and use this command - “pip install selenium”(shown in figure 1). After successfully installed selenium there would be appear a message like “Requirement already satisfied”(figure 2).

Figure 1: Selenium installation in CMD

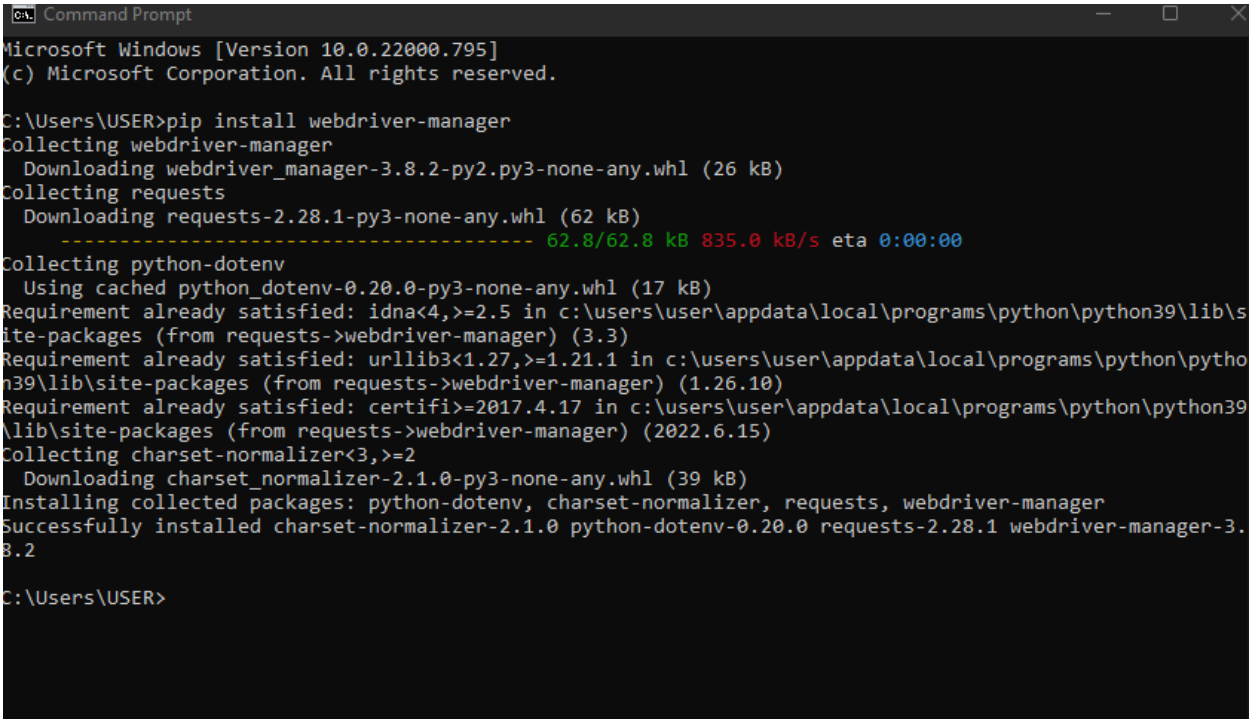

Figure 2: Webdriver manager installation

Now we have to download Google Chrome driver as per our Google Chrome version. We have download Chromedriver_win32. After that we paste the Chromedriver to selenium folder and open our Laravel project in Selenium folder by Visual Studio Code.

Test Case:

Test Case for Login Module :

Project Name: E Commerce Site	**Test Designed by:** Aurthy
Test Case ID: Login_01	**Test Designed date:** 17-07-2022
Test Priority (Low, Medium, High): High	**Test Executed by:** Ananya
Module Name: Login Session	**Test Execution date:** 18-07-2022
Test Title: verify login with valid username and password	
Description: Test website login page	
Precondition (If any): User must have valid username and password	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the website 2. Click "Login" button 3. Enter username 4. Enter password 5. Click Login	Username: Ayon07 Password: 123456	User should login into the application	As expected	Pass

Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database.

Test Case for User create Module :

Project Name: E Commerce Site	**Test Designed by:** Aurthy
Test Case ID: create_user_001	**Test Designed date:** 17-07-2022
Test Priority (Low, Medium, High): High	**Test Executed by:** Ananya
Module Name: Creating user	**Test Execution date:** 18-07-2022
Test Title: Create user for the website	
Description: Create User for the website for business purpose.	

Precondition (If any):

1. Need to be logged in on the website

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the website 2. Click "Login" button 3. Enter username 4. Enter password 5. Click Login 6. Click Create User 7. Enter User Name 8. Click submit	User Name: Ayon	User created successfully	As expected	Pass

Post Condition: Admin redirect to the dashboard page with a success message.

Testing Process:

- **Login Module:**

After installing all the necessary files we log in to the server to open our project. After getting the IP-link we connected to the software that we asked the automated software to search for "username " and write "Ayon07". If the user name input is done correctly, we will get a message as user name input completed. The same goes to the password. In password field we will insert "123456" then press the button "Login". If password works properly then we will get the answer as succussed or failed.

login.py > ...

```python
from selenium.webdriver.common.by import By
from selenium import webdriver
import time
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))
# driver = webdriver.Chrome(executable_path="chromedriver.exe")
driver.get("http://127.0.0.1:8000/login")
time.sleep(2)
#name = u_username
name = driver.find_element(By.ID, "u_username")
name.send_keys("Ayon07")
time.sleep(2)
print("Username input completed")
password = driver.find_element(By.ID, "u_password")
password.send_keys("123456")
print("Password input completed")
time.sleep(2)
#password = u_password
# id = login

button = driver.find_element(By.ID, "login")
button.click()
print("Login button clicked")

if driver.title == "User Dash":
    print("Successfully  Logged In")
else:
    print("Login Failed")
time.sleep(2)
driver.quit()
```

Laravel project hyper-link

Figure 3: Testing Login Module

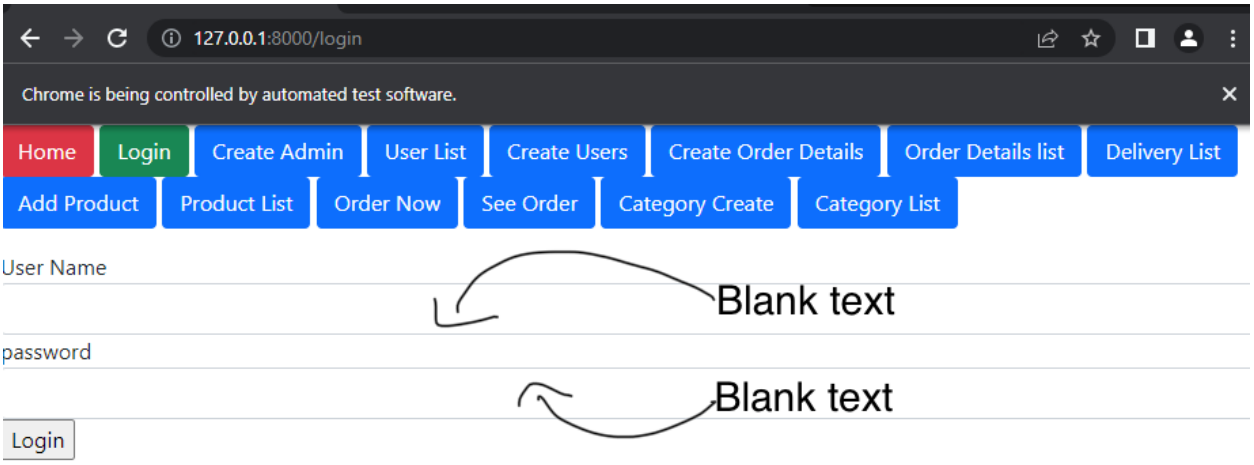

Figure 4: Input field for Login

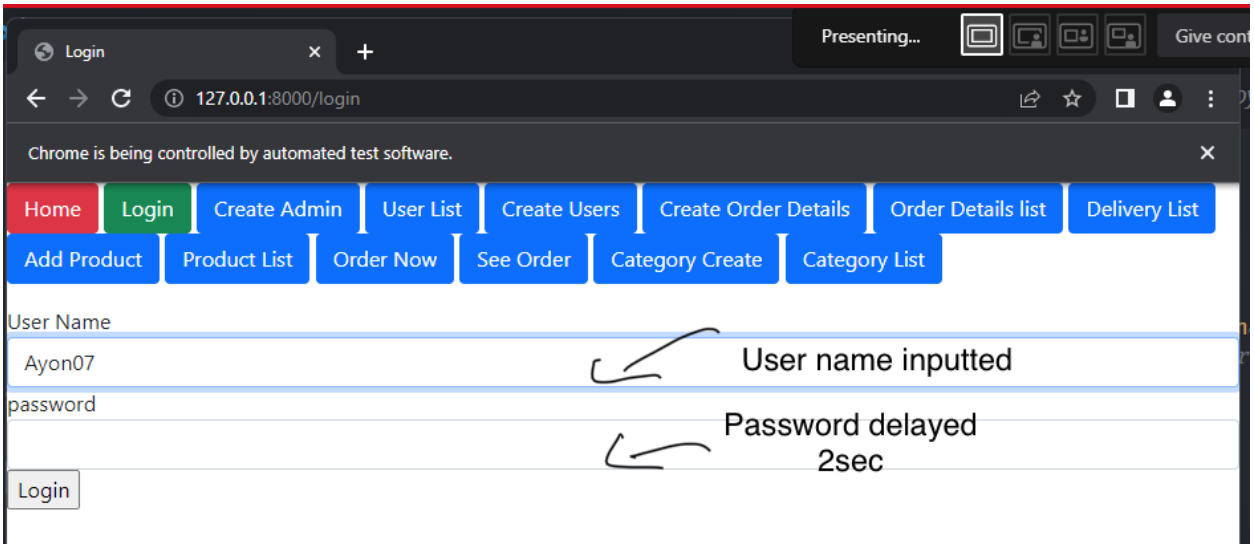

Figure 5: Insert user name

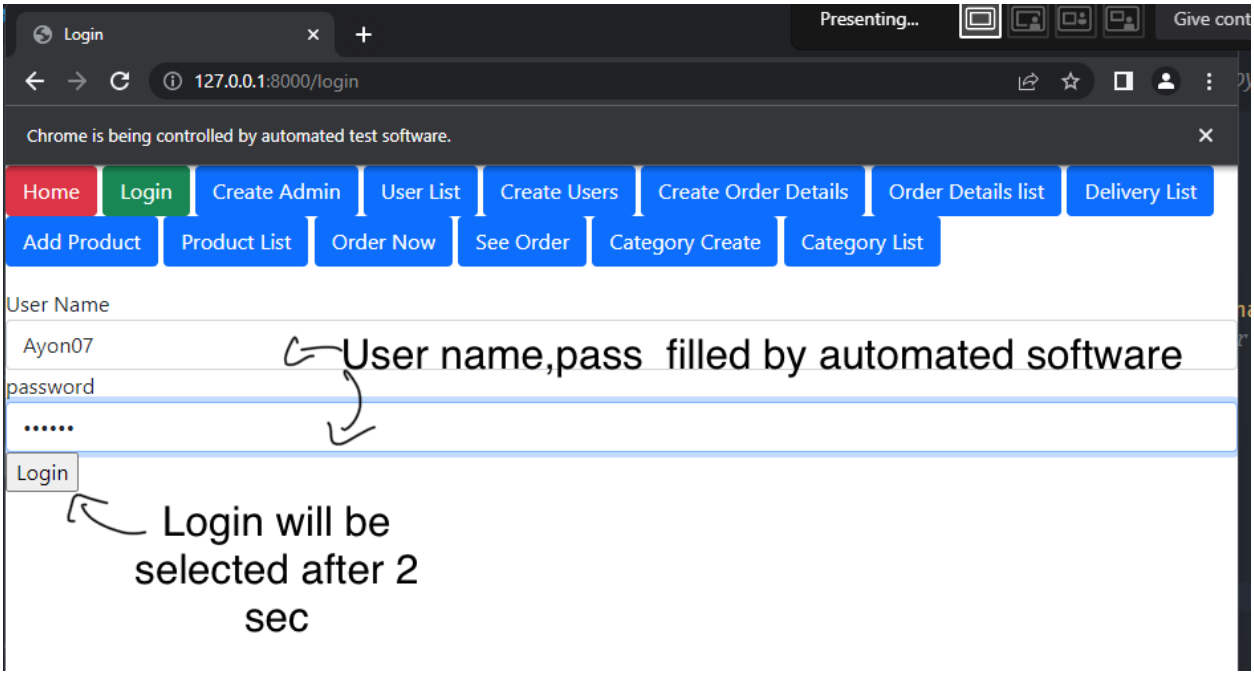

Figure 6: Insert Password

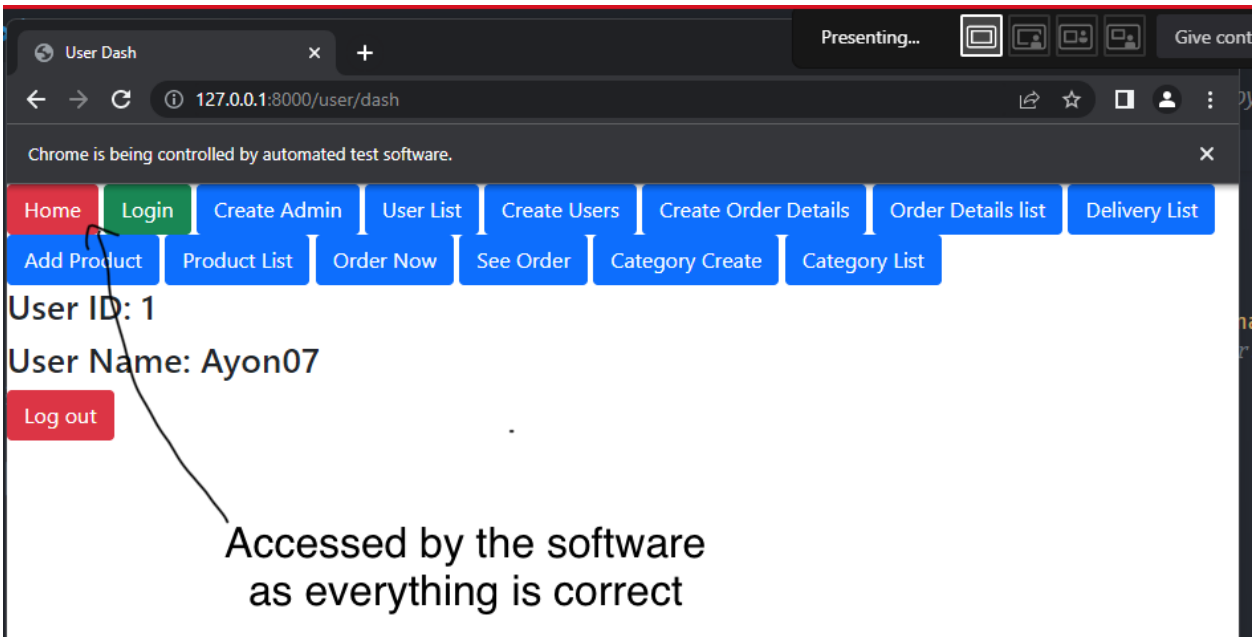

Figure 7: Successfully Login

```
PS D:\12thsem\sqt\selenium> py login.py


DevTools listening on ws://127.0.0.1:64935/devtools/browser/0b987e6f-3dad-496b-9ece-681878cd312b
[23800:23868:0718/204432.904:ERROR:device_event_log_impl.cc(214)] [20:44:32.903] Bluetooth: bluetooth_adapter_w
inrt.cc:1074 Getting Default Adapter failed.
Username input completed
Password input completed
Login button clicked
Successfully  Logged In
```

Accessed

Figure 8: Result of Login

In Figure 3, we have shown the code of login page testing. And after that when we execute the code at first we can see the login page of our ecommerce site and all the input fields are blank there in Figure 4. Also in Figure 5, we can see the username is automatically inserted. After that it will insert the Password which is in Figure 6. Then the login button will be clicked and it will enter the user dash board and we can see the Name of the user in Figure 7. And in Figure 8, we can see the output of the result either the input was given perfectly or not and the login was successful or not.

Result: As we inserted correct username and password provided by the teams and as it is logged in successfully. So we can say that the code was correctly implemented.

- **Error checking on Login Module:**

As correct password and username leads us to get the access of login to the software it's time to check whether the code can distinguish correct input from the wrong input and whether it grant us the assess to log in.

```
from selenium.webdriver.common.by import By
from selenium import webdriver
import time
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))
# driver = webdriver.Chrome(executable_path="chromedriver.exe")
driver.get("http://127.0.0.1:8000/login")
time.sleep(2)
#name = u_username
name = driver.find_element(By.ID, "u_username")
name.send_keys("Ayon07")                                   Correct
time.sleep(2)
print("Username input completed")
password = driver.find_element(By.ID, "u_password")
password.send_keys("1234567")                              Wrong input provided
print("Password input completed")
time.sleep(2)
#password = u_password
# id = login

button = driver.find_element(By.ID, "login")
button.click()
print("Login button clicked")

if driver.title == "User Dash":
    print("Successfully  Logged In")
else:                                                      Message
    print("Login Failed")
time.sleep(2)
driver.quit()
```

Figure 9: Error checking

```
PS D:\12thsem\sqt\selenium> py login.py


DevTools listening on ws://127.0.0.1:63657/devtools/browser/4f46e8eb-cdc6-4e50-b300-ff494ad1847a
[14592:25216:0718/203508.636:ERROR:device_event_log_impl.cc(214)] [20:35:08.635] Bluetooth: bluetooth_adapter_winrt.cc:1074 Getti
ng Default Adapter failed.
Username input completed
Password input completed
Login button clicked
Login Failed
```

Denied access

Figure 10: Login failed as wrong password inserted

Result: We insert a wrong password over here for checking if our code is working properly or not. As we have given the wrong password in password input field in Figure 9, we can see that when all the insertion is done the Login was failed and in the output which is in Figure 10 we can see it printed Login Failed.

- **Adding User Module:**

In this test case we will creating an user automatically using Selenium. For this we will be providing username, password, phone number, Email and if every given data matched with the validation state the program will let us to create a new user.

createuser.py > ...

```python
from selenium.webdriver.common.by import By
from selenium import webdriver
import time
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))
# driver = webdriver.Chrome(executable_path="chromedriver.exe")
driver.get("http://127.0.0.1:8000/admin/create")
time.sleep(5)

name = driver.find_element(By.ID, "u_name")
name.send_keys("Ananyo Chowdhury")

phone = driver.find_element(By.ID, "u_phone")
phone.send_keys("01649889794")

address = driver.find_element(By.ID, "u_address")
address.send_keys("Gulshan, Dhaka")

username = driver.find_element(By.ID, "u_username")
username.send_keys("Ananyo10")


email = driver.find_element(By.ID, "u_email")
email.send_keys("Anany04@gmail.com")


password = driver.find_element(By.ID, "u_password")
password.send_keys("123456")

usertype = driver.find_element(By.ID, "Admin")
usertype.click()
```

Admin creation link

Proving data to the automated software

```python
usertype = driver.find_element(By.ID, "Admin")
usertype.click()

img = driver.find_element(By.ID, "u_profileimg")
img.send_keys("default.jpg")

button = driver.find_element(By.ID, "signup")
button.click()
print("Signup clicked")


if driver.title == "User List":
    print("Successfully  User Added")
else:
    print("User Adding Failed")

time.sleep(5)
driver.quit()
```

Message

Fig 11: Creating User testing Code

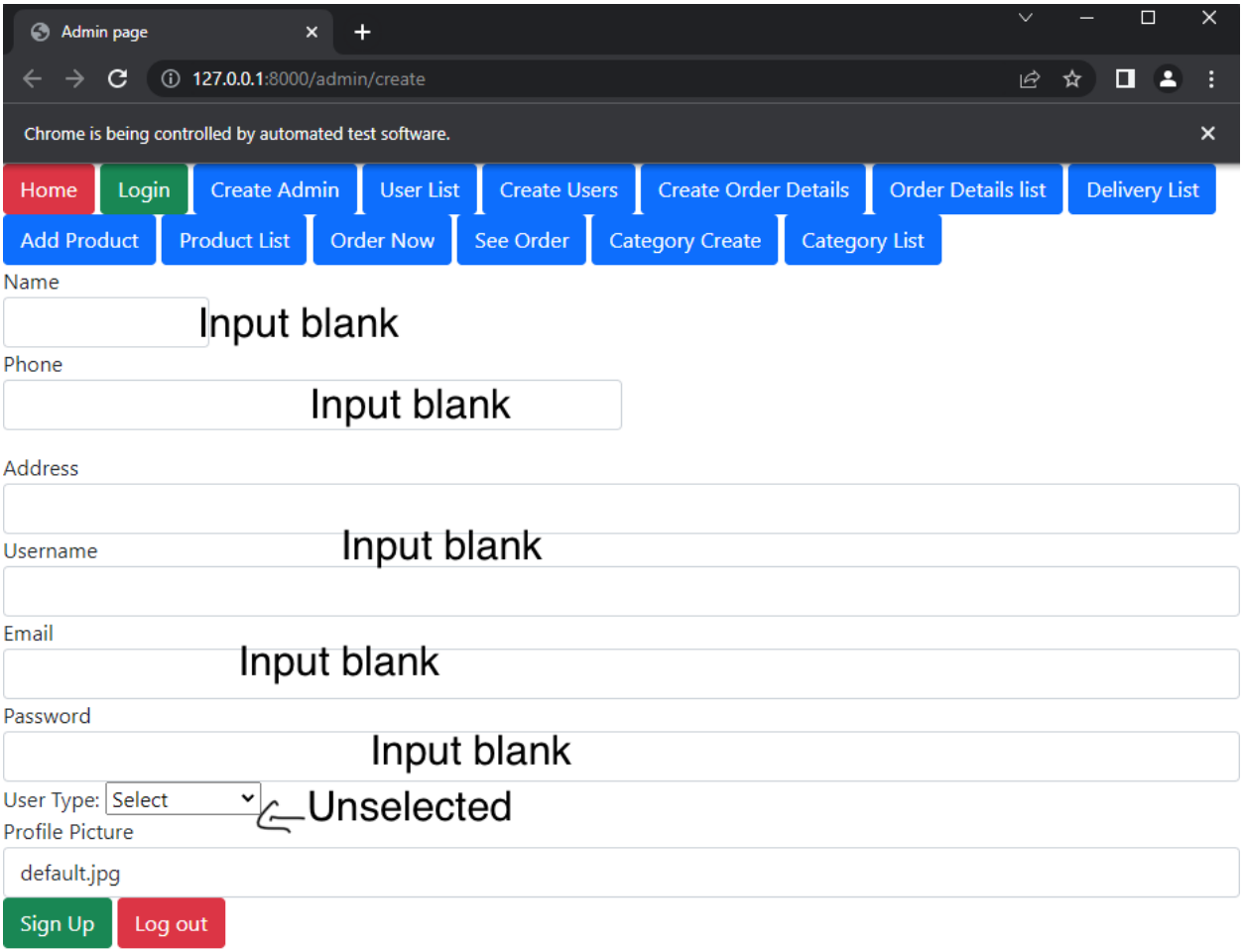

Figure 12: Create User input field

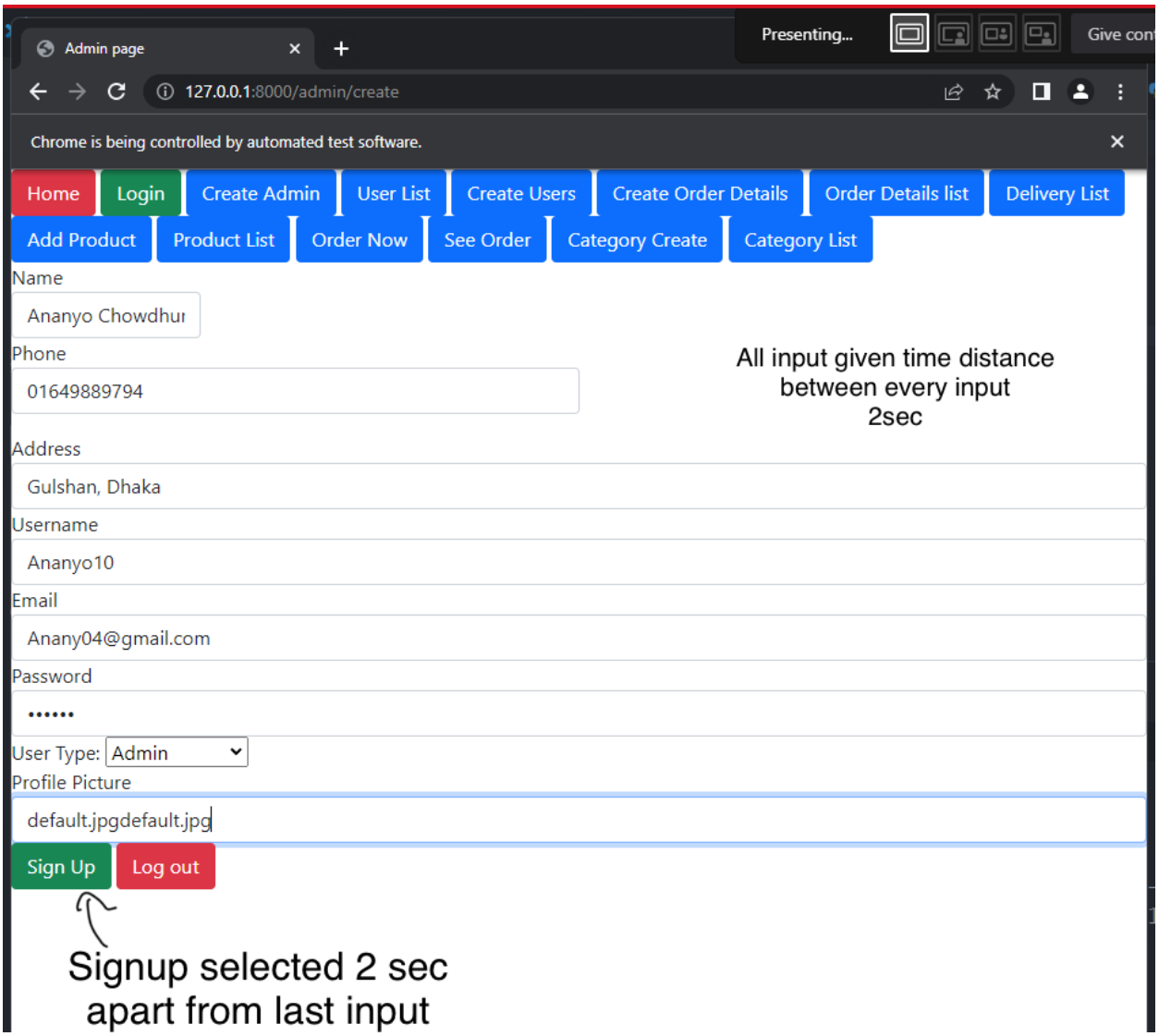

Figure 13: All the input field filled automatically

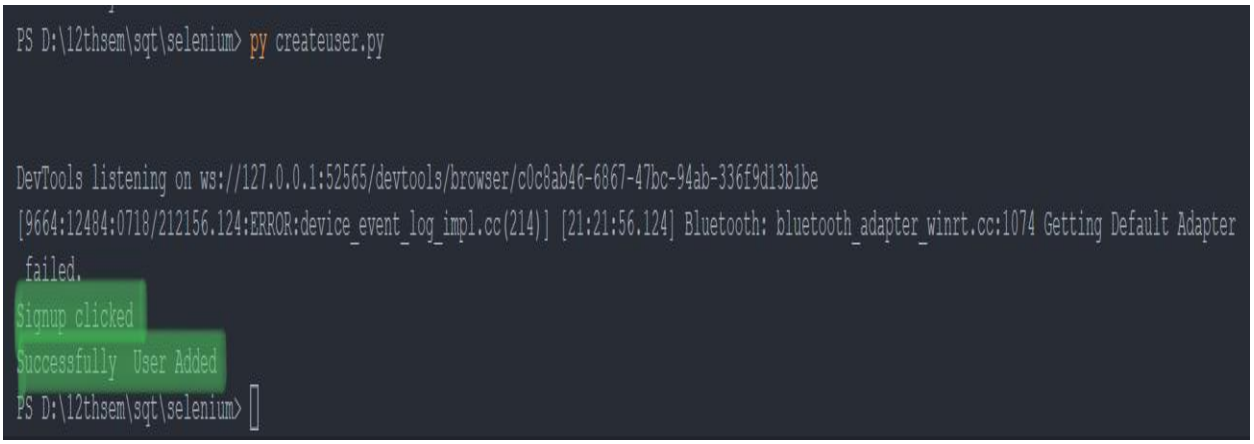

Figure 14: output of the create-user test

Here, On Figure 11, we can see the testing code of creating new user. Then when we execute the code, the chrome file opens the creating user link in Figure 12 where the input field is blank and after that in the Figure 13, the input field was filled automatically. And that Figure 14, output of the test file and user was added successfully.

Result: As we inserted correct and validate inputs we can see that the user was added successfully and it printed "Successfully User Added".

- **Counting Module:**

Here we count all the Anchor tags from the Dashboard and show those sequentially by this code. We have in total 11 Anchor tags in Dashboard.

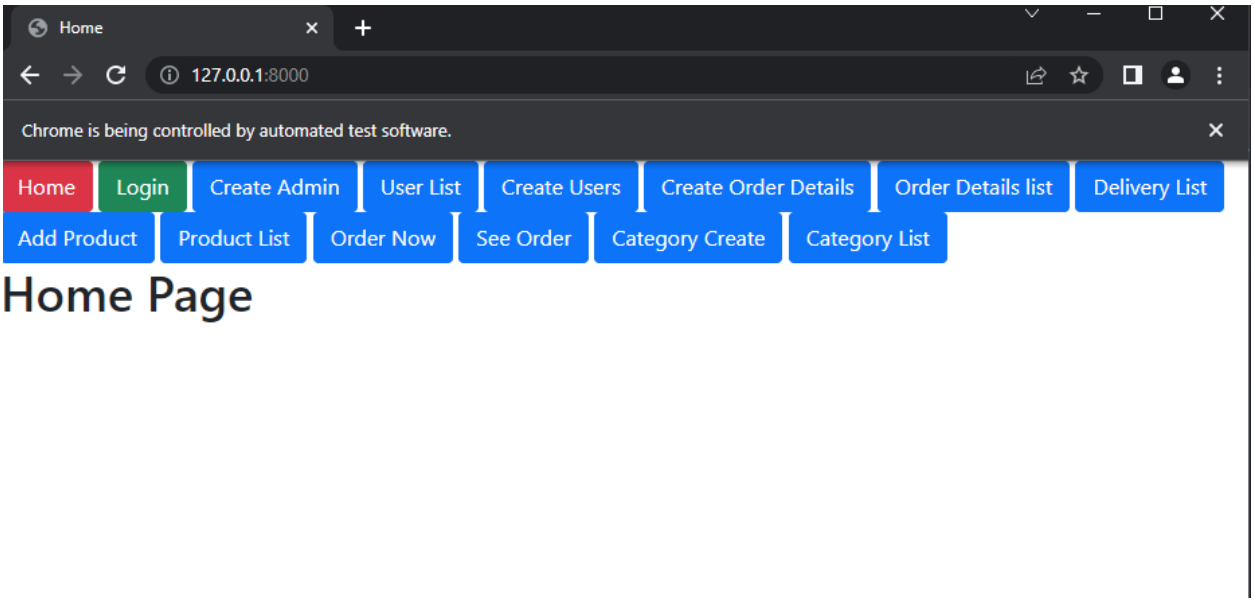

Figure15 :All anchor tags from Dashboard

ok.py | links.py | implicit.py | handle.py | login.py

links.py > ...

```python
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))


# driver = webdriver.Chrome(executable_path="chromedriver.exe")

driver.get("http://127.0.0.1:8000")

links = driver.find_elements(By.TAG_NAME, "a")
print("Length of anchor tag: ", len(links))


# link TEXT
for link in links:
    print(link.text)

time.sleep(4)

driver.find_element(By.LINK_TEXT, "Login").click()
time.sleep(4)

driver.quit()
```

Figure16 : Counting Module

```
Length of anchor tag:  14
Home
Login
Create Admin
User List
Create Users
Create Order Details
Order Details list
Delivery List
Add Product
Product List
Order Now
See Order
Category Create
Category List
```

Number of anchor tag

Name of all anchor tag

Figure17 : Length and list of all anchor tag name

Result: We supposed to get the length of anchor tags along with the names of all anchor tag and we got the actual result.

- **Multitasking handle Module**:

In this test case we are selecting the 5th element on the top navigation bar and moved towards that page and click that button , we can see its on create users page. In this program we are trying to open home from the top navigation . we are selecting this from the title drive. And stay on this case for 5 sec. To opened another page from the navigation bar that is create user . we can see the tin the page that it opened another page on the chrome bar. In this case we are checking we there is any index title name as home if we found home in the bar we will close that page and keep other pages turned open.

handle.py > ...

```python
from selenium import webdriver
from selenium.webdriver.common.by import By
import time
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))

driver.get("http://127.0.0.1:8000")


time.sleep(4)

var = driver.find_element(by=By.XPATH, value="/html/body/a[5]")
print(driver.current_window_handle)

time.sleep(4)
# current_window_handle == parent

handles = driver.window_handles
# window_handles -- child handle

for handle in handles:
    driver.switch_to.window(handle)
    # print(driver.title)

    if driver.title == "Home":
        driver.close()
        print("Home Page closed")

time.sleep(5)

driver.quit()
```

Location of an anchor tag

Fig 18:Multipage Handler

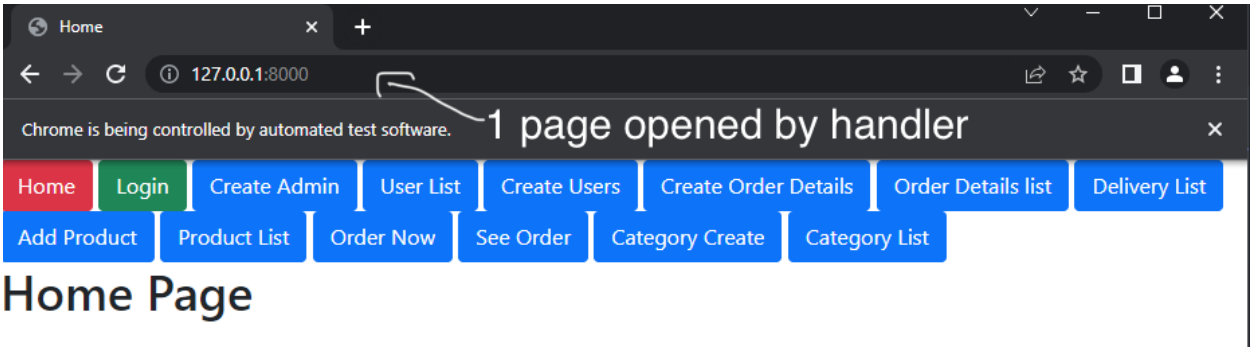

Fig 19: Multipage Handler

In the fig:19 by the help of handle function one page is opened in the chrome named as home anchor tag.as there was is a delay on 4 sec . For 4 second page will remain like this as like code.

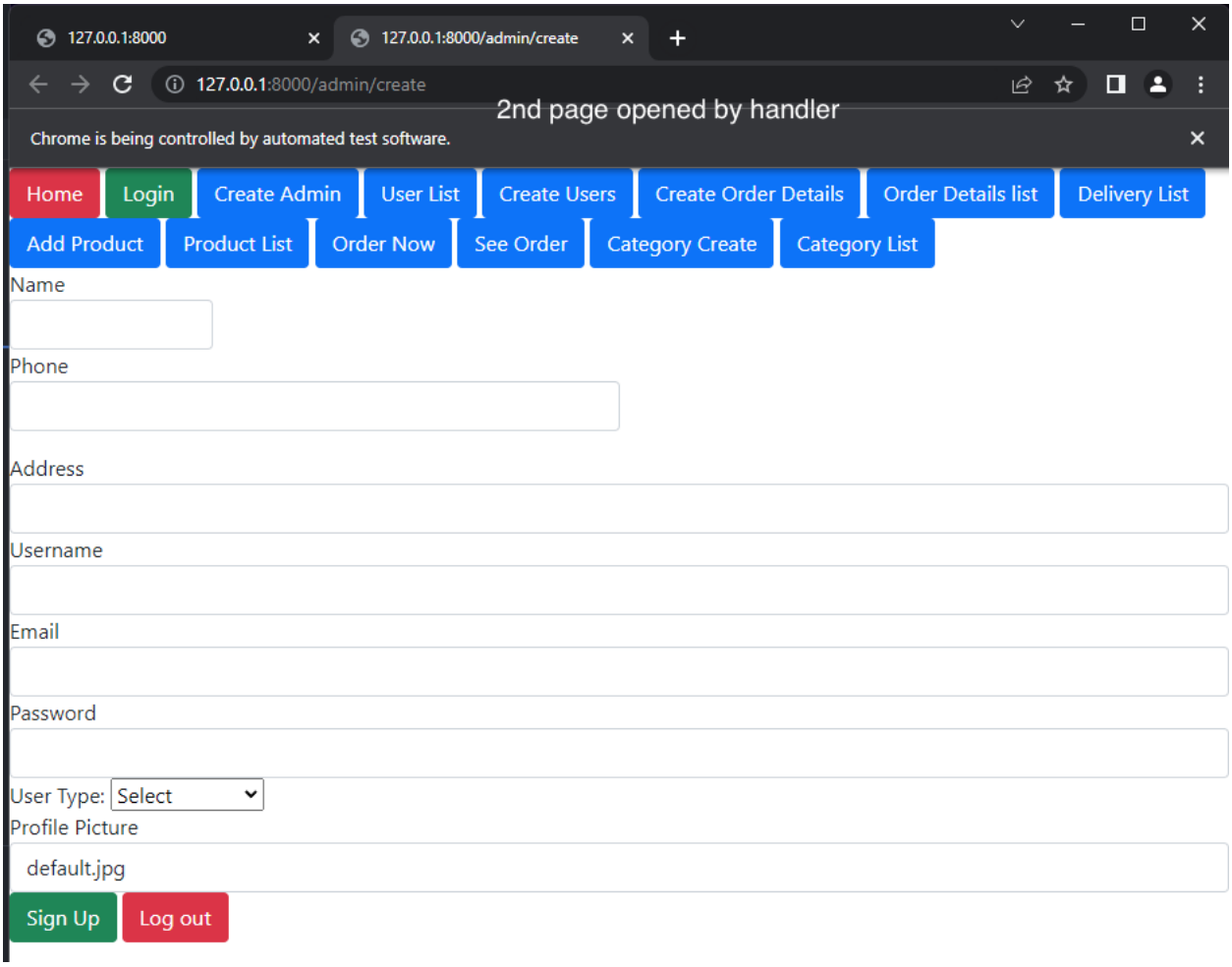

Fig 20:Multipage Handler

In fig:20 , 2nd page is opened by the help of handle function which is done by automated software , this page will remain open for 4sec after that next function will be called and be executed.

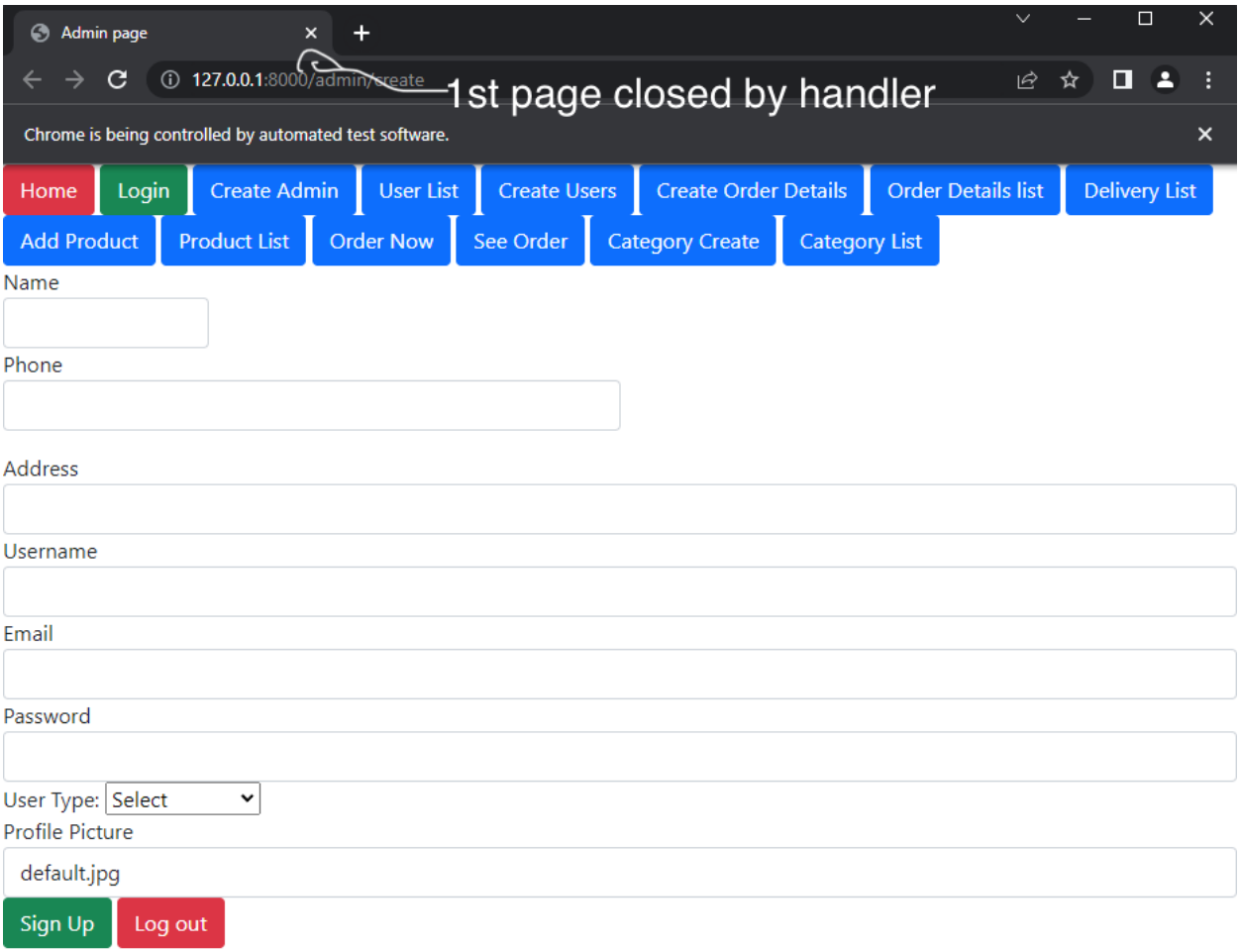

Fig 21:Multipage Handler

In Fig:21 1st page is closed as in the code it was searching by the name home and as it finds that index page it will turn that page off which happens in this fig 23.

Fig 22: Multipage Handler

In the fig:22 we can see the output generated by the automated software page it was designated by is closed.