

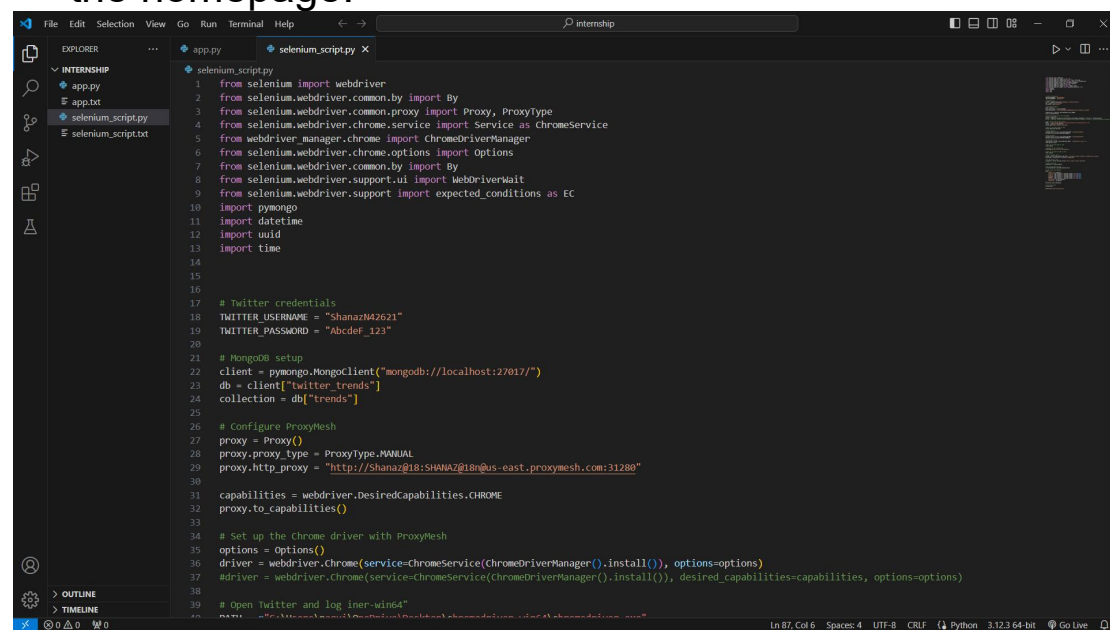
Assignment for Tech Internship via Internshala

Given Task

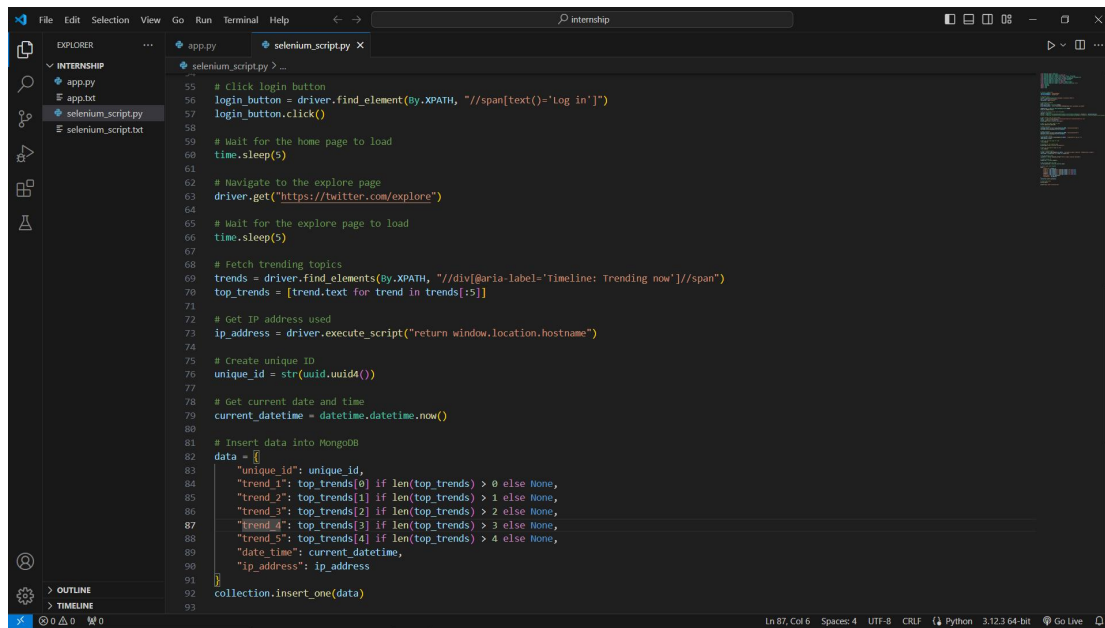
Web scraping with Selenium and ProxyMesh, storing the data in MongoDB, showing a list on a webpage. This task will test your ability to work with web automation tools, proxies, and data extraction techniques.

Task Done by Assigned steps:

1. Write a Selenium script that can read the Twitter home page (on your local computer) and fetch the top 5 trending topics under “What’s Happening” section from the homepage.

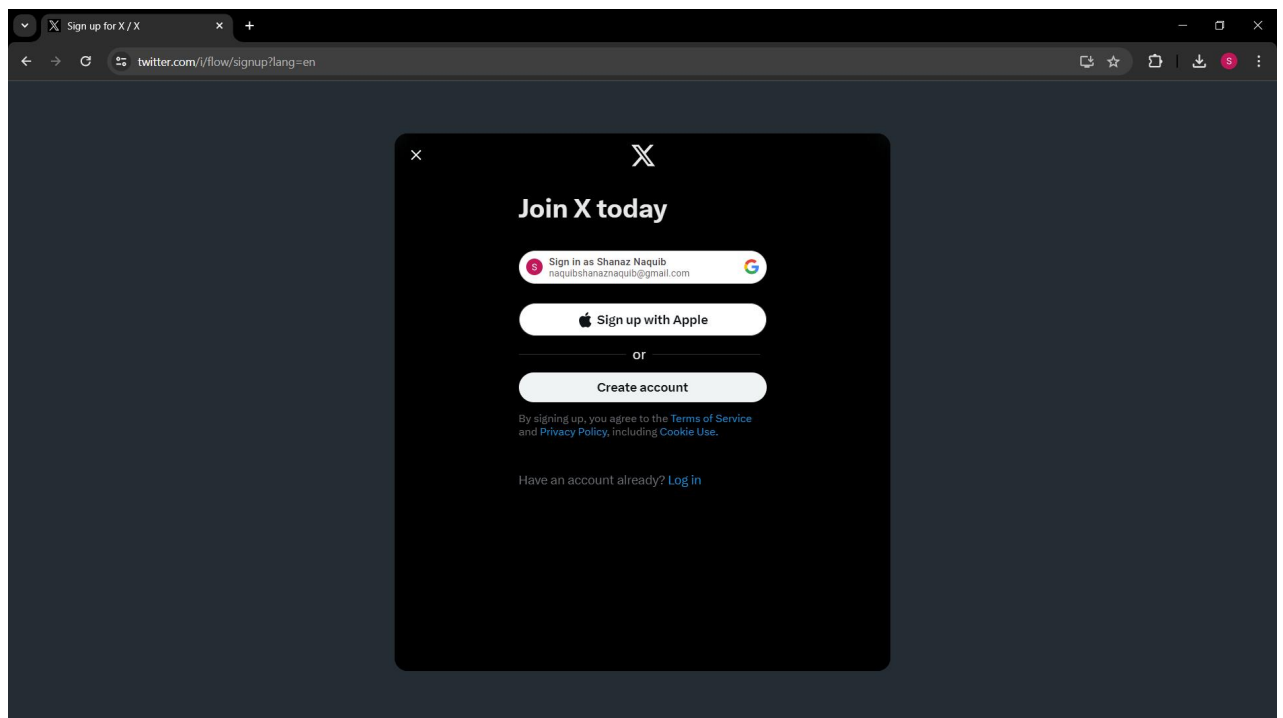


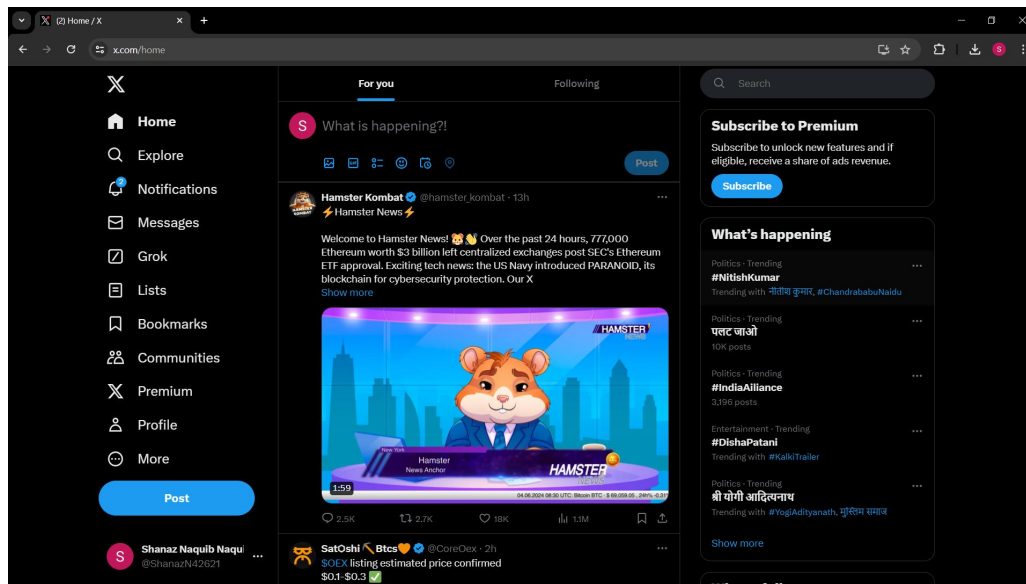
```
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.common.proxy import Proxy, ProxyType
4 from selenium.webdriver.chrome.service import Service as ChromeService
5 from webdriver_manager.chrome import ChromeDriverManager
6 from selenium.webdriver.chrome.options import Options
7 from selenium.webdriver.common.by import By
8 from selenium.webdriver.support.ui import WebDriverWait
9 from selenium.webdriver.support import expected_conditions as EC
10 import pymongo
11 import datetime
12 import uuid
13 import time
14
15
16
17 # Twitter credentials
18 TWITTER_USERNAME = "Shana742621"
19 TWITTER_PASSWORD = "Abcdcf_123"
20
21 # MongoDB setup
22 client = pymongo.MongoClient("mongodb://localhost:27017/")
23 db = client["twitter_trends"]
24 collection = db["trends"]
25
26 # Configure ProxyMesh
27 proxy = Proxy()
28 proxy.proxy_type = ProxyType.MANUAL
29 proxy.http_proxy = "http://Shana742621@us-east.proxyMesh.com:31280"
30
31 capabilities = webdriver.DesiredCapabilities.CHROME
32 proxy_to_capabilities()
33
34 # Set up the Chrome driver with ProxyMesh
35 options = Options()
36 driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()), options=options)
37 #driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()), desired_capabilities=capabilities, options=options)
38
39 # Open Twitter and log in
40 driver.get("https://twitter.com/login")
41 driver.find_element(By.NAME, "username").send_keys(TWITTER_USERNAME)
42 driver.find_element(By.NAME, "password").send_keys(TWITTER_PASSWORD)
43 driver.find_element(By.ID, "login-button").click()
44 time.sleep(10)
45 driver.get("https://twitter.com/trending")
46 time.sleep(10)
47 # Get the trending topics
48 trends = driver.find_elements(By.CSS_SELECTOR, "div[data-testid='Trend']")
49 trends_list = []
50 for trend in trends:
51     topic = trend.find_element(By.CSS_SELECTOR, "div[data-testid='Text']").text
52     trends_list.append(topic)
53
54 # Store the trending topics in MongoDB
55 trends_list.insert(0, "Trending Topics")
56 collection.insert_one(trends_list)
```



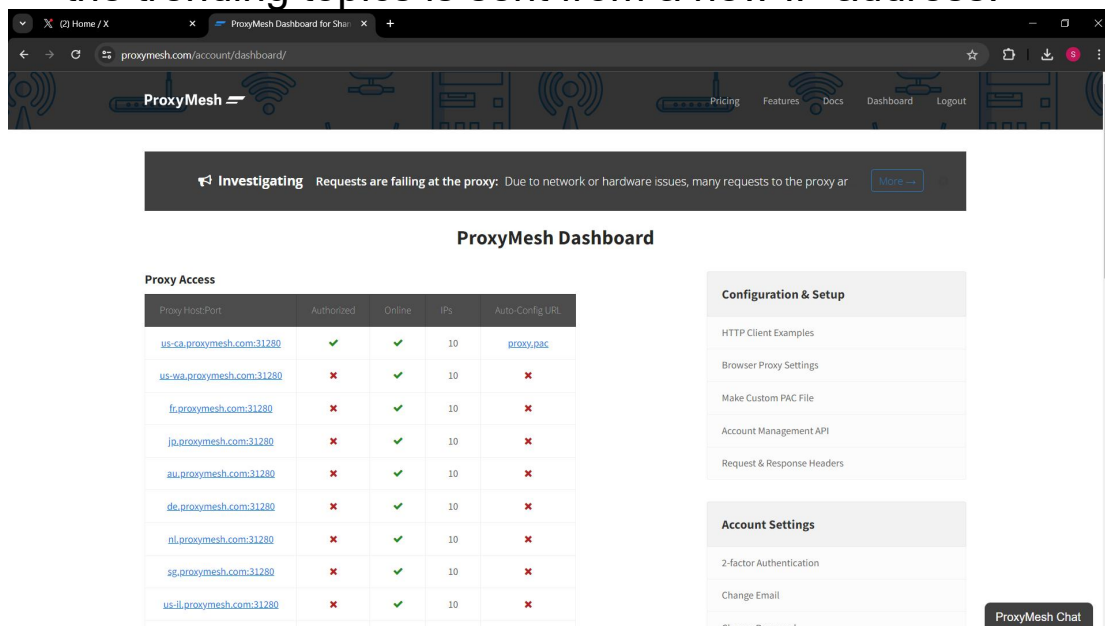
```
55 # Click login button
56 login_button = driver.find_element(By.XPATH, "//span[text()='Log in']")
57 login_button.click()
58
59 # Wait for the home page to load
60 time.sleep(5)
61
62 # Navigate to the explore page
63 driver.get("https://twitter.com/explore")
64
65 # Wait for the explore page to load
66 time.sleep(5)
67
68 # Fetch trending topics
69 trends = driver.find_elements(By.XPATH, "//div[@aria-label='Timeline: Trending now']//span")
70 top_trends = [trend.text for trend in trends[:5]]
71
72 # Get IP address used
73 ip_address = driver.execute_script("return window.location.hostname")
74
75 # Create unique ID
76 unique_id = str(uuid.uuid4())
77
78 # Get current date and time
79 current_datetime = datetime.datetime.now()
80
81 # Insert data into MongoDB
82 data = {
83     "unique_id": unique_id,
84     "trend_1": top_trends[0] if len(top_trends) > 0 else None,
85     "trend_2": top_trends[1] if len(top_trends) > 1 else None,
86     "trend_3": top_trends[2] if len(top_trends) > 2 else None,
87     "trend_4": top_trends[3] if len(top_trends) > 3 else None,
88     "trend_5": top_trends[4] if len(top_trends) > 4 else None,
89     "date_time": current_datetime,
90     "ip_address": ip_address
91 }
92 collection.insert_one(data)
93
```

2. To access Twitter, create/use your own Twitter account, since log in required to see this page.





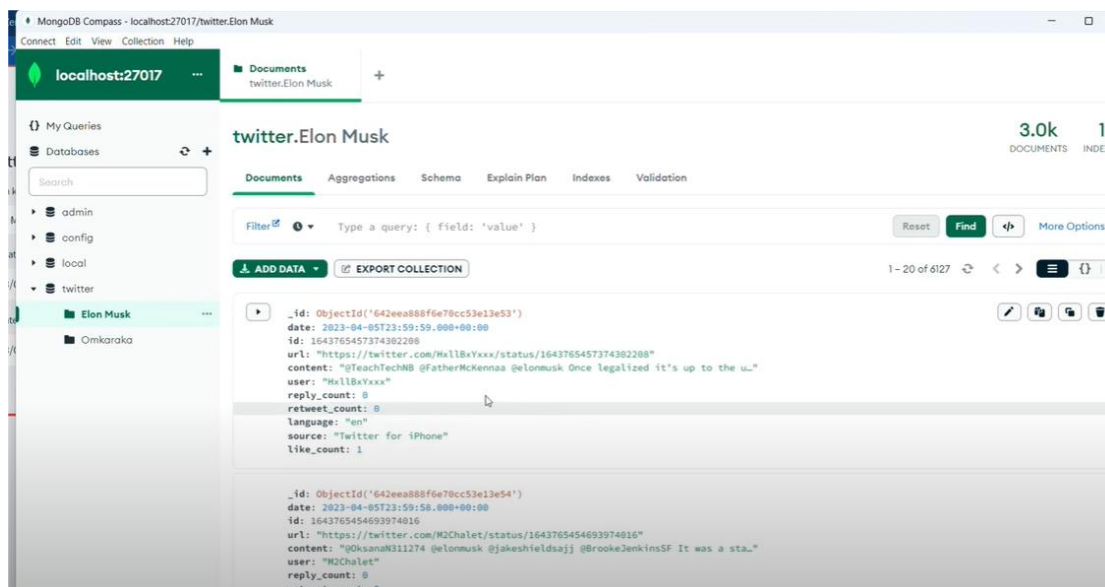
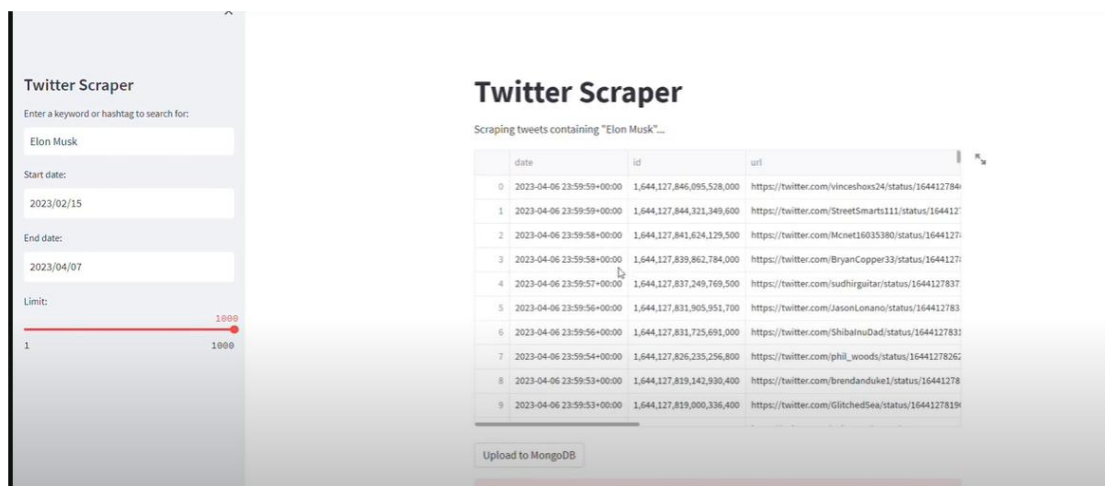
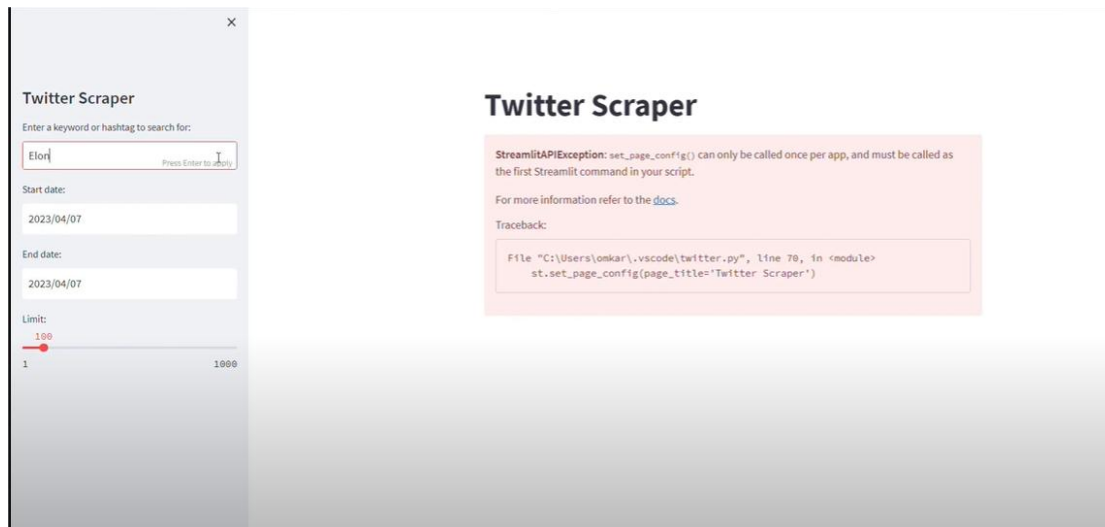
3. Use ProxyMesh such that each new request to scrape the trending topics is sent from a new IP address.



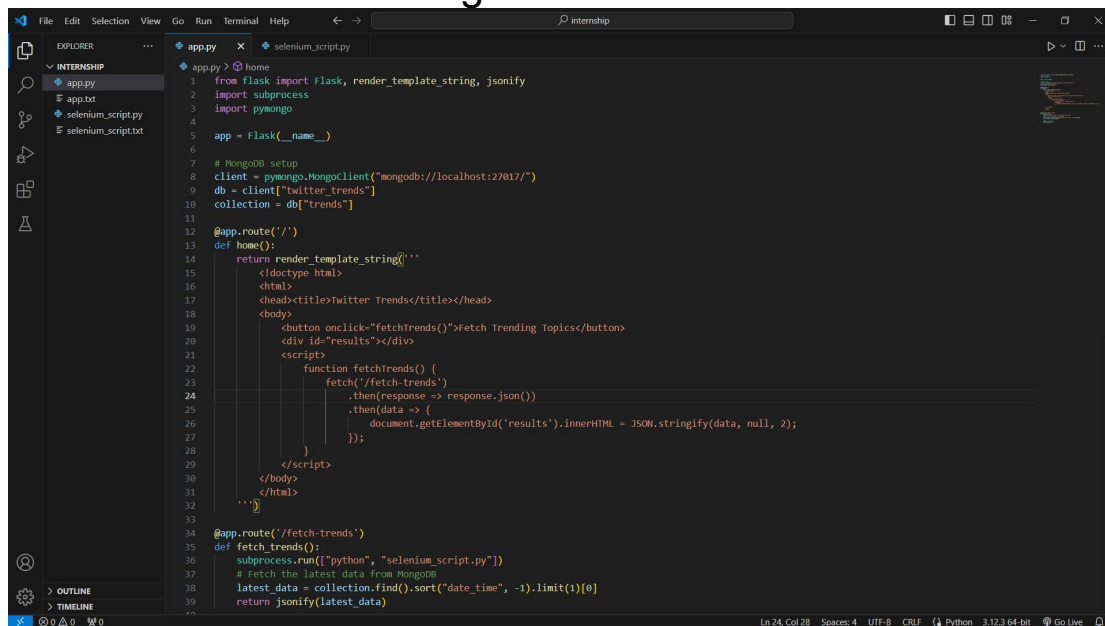
```
12 import uuid
13 import time
14
15
16
17 # Twitter credentials
18 TWITTER_USERNAME = "Shanaaz42621"
19 TWITTER_PASSWORD = "Abcdef_123"
20
21 # MongoDB setup
22 client = pymongo.MongoClient("mongodb://localhost:27017/")
23 db = client["twitter_trends"]
24 collection = db["trends"]
25
26 # Configure ProxyMesh
27 proxy = Proxy()
28 proxy.proxy_type = ProxyType.MANUAL
29 proxy.http_proxy = "http://Shanaaz@18:SHANAZ@18@us-east.proxyMesh.com:31280"
30
31 capabilities = webdriver.DesiredCapabilities.CHROME
32 proxy.to_capabilities()
33
34 # Set up the Chrome driver with ProxyMesh
35 options = Options()
36 driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()), options=options)
37 #driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()), desired_capabilities=capabilities, options=options)
38
39 # Open Twitter and log in
40 PATH = r"C:\Users\naqui\OneDrive\Desktop\chromedriver-win64\chromedriver.exe"
41 driver = webdriver.Chrome(PATH)
42 driver.get("https://twitter.com/login")
43
44 # wait for the login page to load
45 driver.implicitly_wait(10)
46
47 # Enter username
48 username_field = driver.find_element(By.NAME, "session[username]")
49 username_field.send_keys(TWITTER_USERNAME)
50
```

4. Create a unique ID for each time the Selenium script is run and store the results in a MongoDB database (fields required – unique ID, name of trend1, 2, 3, 4, 5, date and time of end of Selenium script, IP address used). We won't need anything else.

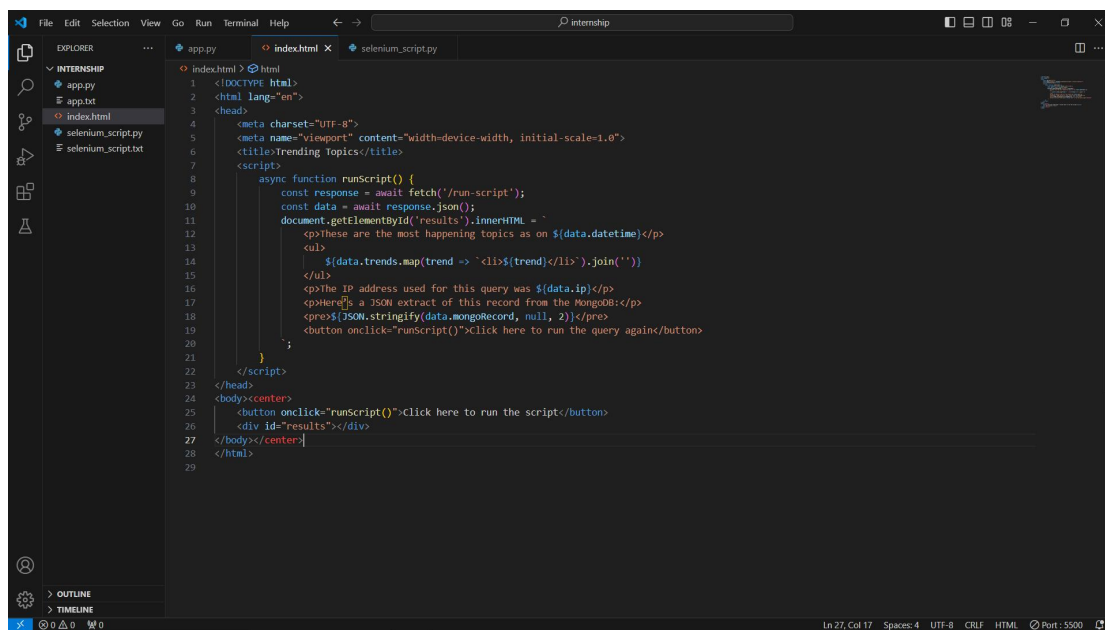
```
1 from flask import Flask, render_template_string, jsonify
2 import subprocess
3 import pymongo
4
5 app = Flask(__name__)
6
7 # MongoDB setup
8 client = pymongo.MongoClient("mongodb://localhost:27017/")
9 db = client["twitter_trends"]
10 collection = db["trends"]
11
12 @app.route('/')
13 def home():
14     return render_template_string('''
15     <doctype html>
16     <html>
17     <head><title>Twitter Trends</title></head>
18     <body>
19     <button onclick="fetchTrends()">Fetch Trending Topics</button>
20     <div id="results"></div>
21     <script>
22     function fetchTrends() {
23         fetch('/fetch-trends')
24         .then(response => response.json())
25         .then(data => {
26             document.getElementById('results').innerHTML = JSON.stringify(data, null, 2);
27         });
28     }
29     </script>
30     </body>
31     </html>
32     ''')
33
34 @app.route('/fetch-trends')
35 def fetch_trends():
36     subprocess.run(["python", "selenium_script.py"])
37     # Fetch the latest data from MongoDB
38     latest_data = collection.find().sort("date_time", -1).limit(1)[0]
39     return jsonify(latest_data)
40
```



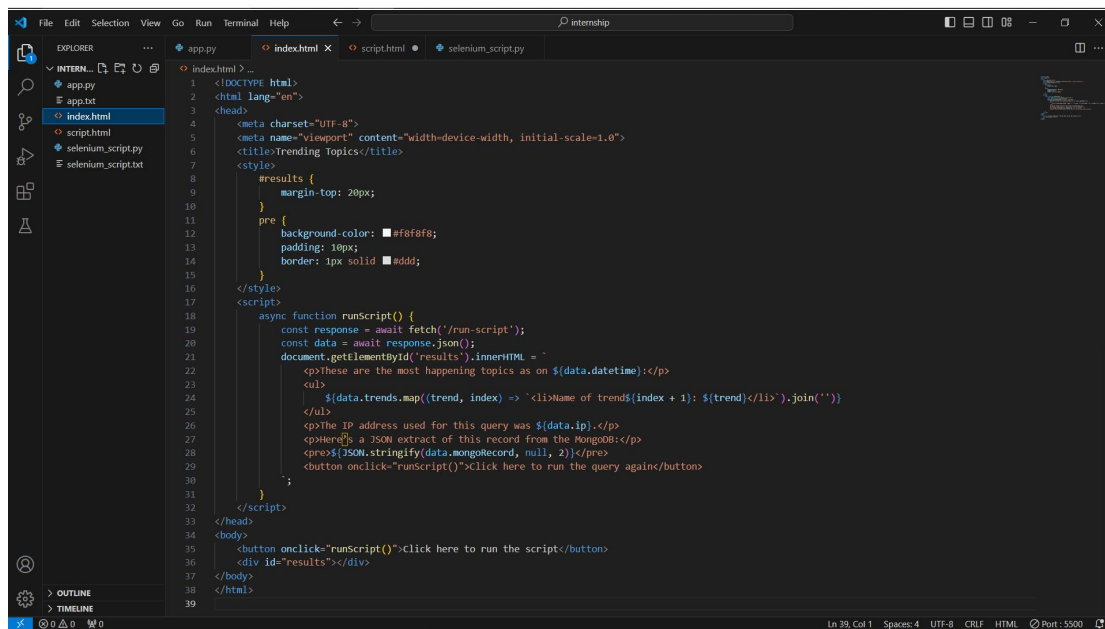
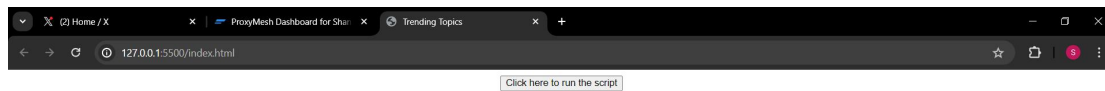
5. Create a simple HTML page which has a button, which when clicked, will the Selenium script, and then show results in the following manner:



```
1 from flask import Flask, render_template_string, jsonify
2 import subprocess
3 import pymongo
4
5 app = Flask(__name__)
6
7 # MongoDB setup
8 client = pymongo.MongoClient("mongodb://localhost:27017/")
9 db = client["twitter_trends"]
10 collection = db["trends"]
11
12 @app.route('/')
13 def home():
14     return render_template_string('''
15     <doctype html>
16     <html>
17     <head><title>Twitter Trends</title></head>
18     <body>
19     <button onclick="fetchTrends()">Fetch Trending Topics</button>
20     <div id="results"></div>
21     <script>
22         function fetchTrends() {
23             fetch('/fetch-trends')
24                 .then(response => response.json())
25                 .then(data => {
26                     document.getElementById('results').innerHTML = JSON.stringify(data, null, 2);
27                 });
28         }
29     </script>
30     </body>
31     </html>
32     ''')
33
34 @app.route('/fetch-trends')
35 def fetch_trends():
36     subprocess.run(["python", "selenium_script.py"])
37     # Fetch the latest data from MongoDB
38     latest_data = collection.find().sort("date_time", -1).limit(1)[0]
39     return jsonify(latest_data)
```



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Trending Topics</title>
7     <script>
8         async function runScript() {
9             const response = await fetch('/run-script');
10             const data = await response.json();
11             document.getElementById('results').innerHTML = `
12             <p>These are the most happening topics as on ${data.datetime}</p>
13             <ul>
14                 ${data.trends.map(trend => `<li>${trend}</li>`).join('')}
15             </ul>
16             <p>The IP address used for this query was ${data.ip}</p>
17             <p>Here's a JSON extract of this record from the MongoDB:</p>
18             <pre>${JSON.stringify(data.mongoRecord, null, 2)}</pre>
19             <button onclick="runScript()">Click here to run the query again</button>
20         }
21     </script>
22 </head>
23 <body><center>
24     <button onclick="runScript()">Click here to run the script</button>
25     <div id="results"></div>
26 </body></center>
27 </html>
```



THANK YOU FOR GIVING THE OPPORTUNITY

SHANAZ NAQUIB