Task-3 Title: Selenium Setup and Interaction with Dynamic Websites(Data Collection)
Date: 11/11/24-12/11/24

---

1. Task Overview

Objective:
To set up the Selenium library, configure browser drivers, and use Selenium to interact with a dynamic website to extract data.

Key Deliverables:

1. A test script demonstrating Selenium's functionality for web interaction.
2. Documentation of the setup process, including screenshots of command-line outputs.

---

2. Tasks Completed

Environment Setup:

1. **Install Selenium Library:**

```
Pip install selenium
```

2. **Download Browser Drivers:**
   - For Chrome: Downloaded **ChromeDriver** from chromedriver.chromium.org.
   - Verified compatibility with the installed version of Google Chrome.

```
chromedriver --version
```

Testing Selenium:

Script:

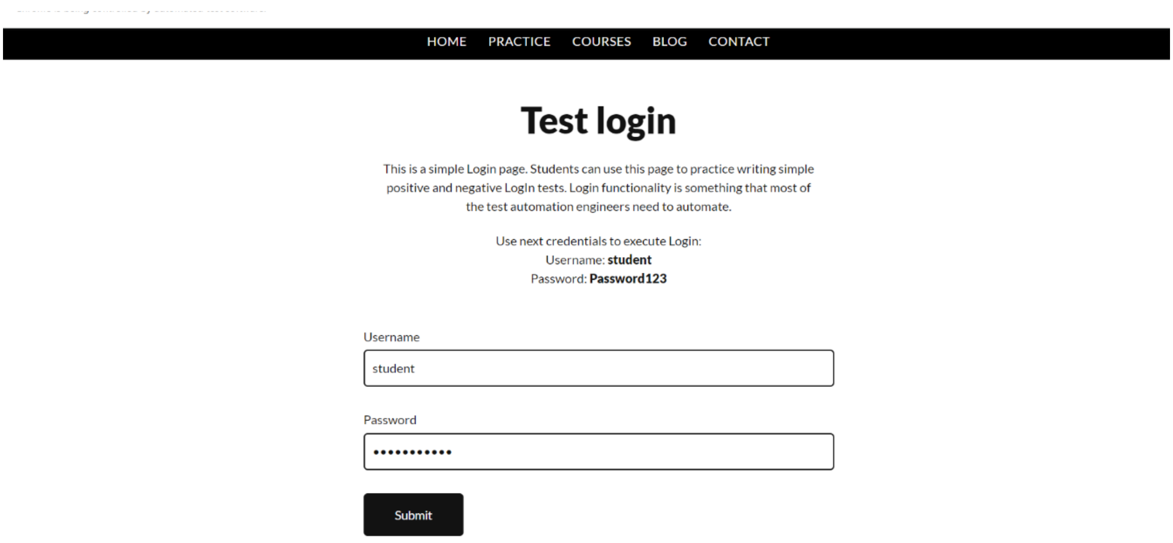
1. Interacted with a dynamic website (e.g., Google Search)

```
from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.chrome.service import Service

from webdriver_manager.chrome import ChromeDriverManager

import time

# Step 1: Set up the WebDriver with Service class

service = Service(ChromeDriverManager().install())  # Automatically manages chromedriver version

driver = webdriver.Chrome(service=service)
```

```
try:
    # Step 2: Open the login page
    driver.get("https://practicetestautomation.com/practice-test-login/")
    time.sleep(2)  # Optional: Wait for the page to load fully
    # Step 3: Locate the username and password fields
    username_field = driver.find_element(By.ID, "username")
    password_field = driver.find_element(By.ID, "password")
    # Step 4: Enter sample credentials (valid test credentials)
    username_field.send_keys("student")
    password_field.send_keys("Password123")
    # Step 5: Click the login button
    login_button = driver.find_element(By.ID, "submit")
    login_button.click()
    time.sleep(15)  # Optional: Wait for the login process
    # Step 6: Verify login success by checking for a specific element
    success_message = driver.find_element(By.XPATH, "//h1[contains(text(), 'Logged In Successfully')]")
    if success_message:
        print("Login successful!")
    else:
        print("Login failed.")
finally:
    # Close the browser after the test
    driver.quit()
```

3. Execution Command:

```
python login.py
```



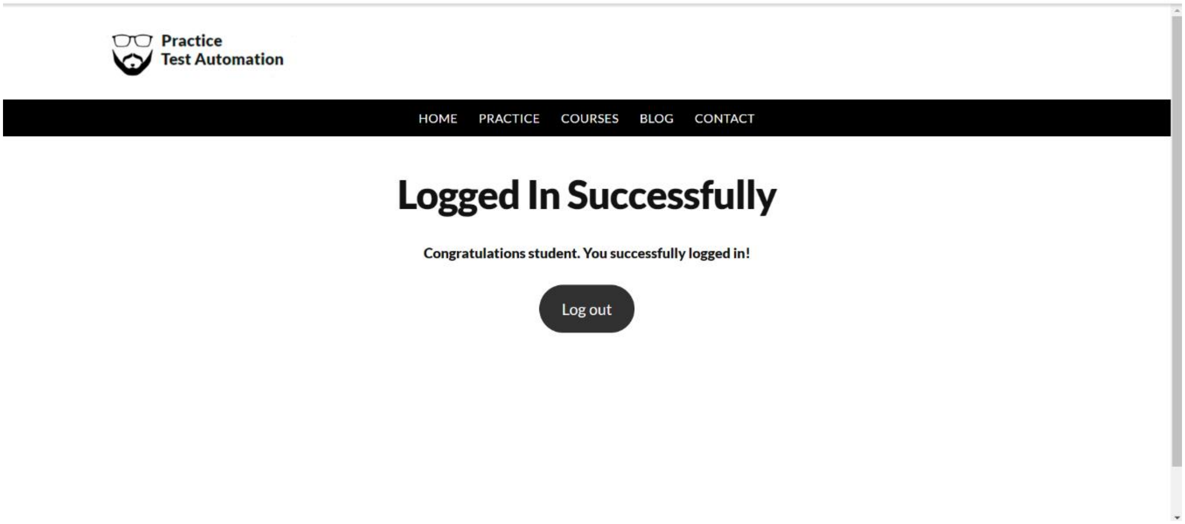
Test login

This is a simple Login page. Students can use this page to practice writing simple positive and negative LogIn tests. Login functionality is something that most of the test automation engineers need to automate.

Use next credentials to execute Login:
Username: **student**
Password: **Password123**

Username: student

Password: •••••••••••



Logged In Successfully

Congratulations student. You successfully logged in!

4. Key Findings

1. Dynamic Website Interaction:
   - Selenium successfully interacted with Google Search and extracted the first result's title.
2. Insights:
   - Ideal for dynamic websites with complex JavaScript-based content.
   - Browser drivers are essential for ensuring compatibility and smooth functionality.

Task4: Data collection Apache Kafka(Data Collection)
Date:13/11/24-15/11/24

1. Task Overview
Objective:
To collect data from multiple sources (e.g., APIs, sensors, log files, or web scraping) in diverse formats (JSON, CSV, XML, AVRO, or Parquet) using appropriate tools, including Apache Kafka and APIs.
Key Deliverables:

- Collected data samples in multiple formats.
- Scripts demonstrating data collection methods.
- Documentation of the tools and processes used

2. Tasks Completed
Data Sources and Tools Used:

Source Identification:

- APIs: Accessed structured datasets via REST API (e.g., Weather Data API).
- Log Files: Collected sample server logs.
- Web Scraping: Used tools like BeautifulSoup for scraping structured data.
- Real-time Data: Set up Apache Kafka for ingesting real-time sensor data.

Implementation Details:
1. Download Kafka:

- Visit Kafka's official downloads page to get the latest stable version.

2. Start Zookeeper: Kafka includes a simple script to start Zookeeper. Run it in a separate terminal:
.\bin\windows\zookeeper-server-start.bat C:\kafka\config\zookeeper.properties

3.Start Kafka Broker: Open another terminal to start the Kafka server:

.\bin\windows\kafka-server-start.bat .\config\server.properties

4. Verify the Installation:

Create a Topic: Kafka topics are where messages are sent and stored. Create a test topic:

kafka-topics.bat --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 3 --topic Indian_weather

List Topics:

bin/kafka-topics.sh --list --bootstrap-server localhost:9092

Read Messages from the Topic: Open a new terminal and run:

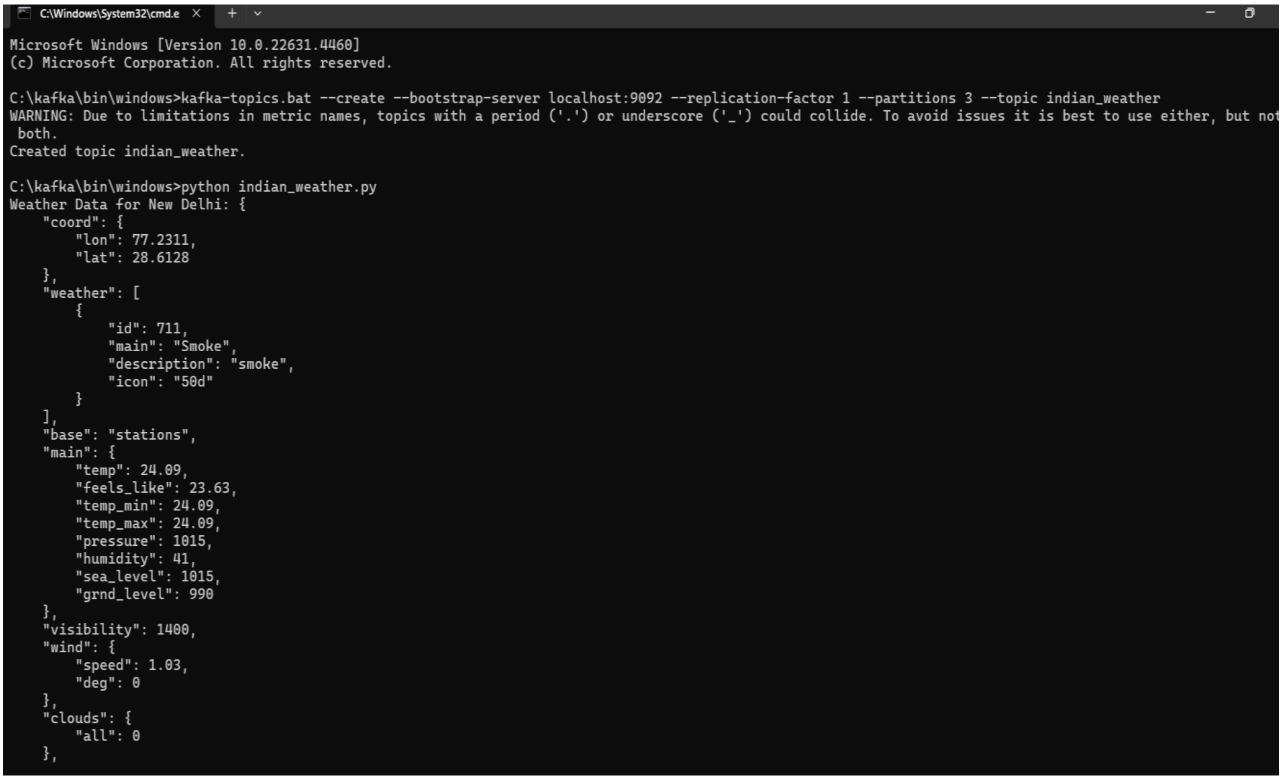
bin/kafka-console-consumer.sh --topic Indian_weather -topic --from-beginning --bootstrap-server localhost:9092

Set Up the Python Script:

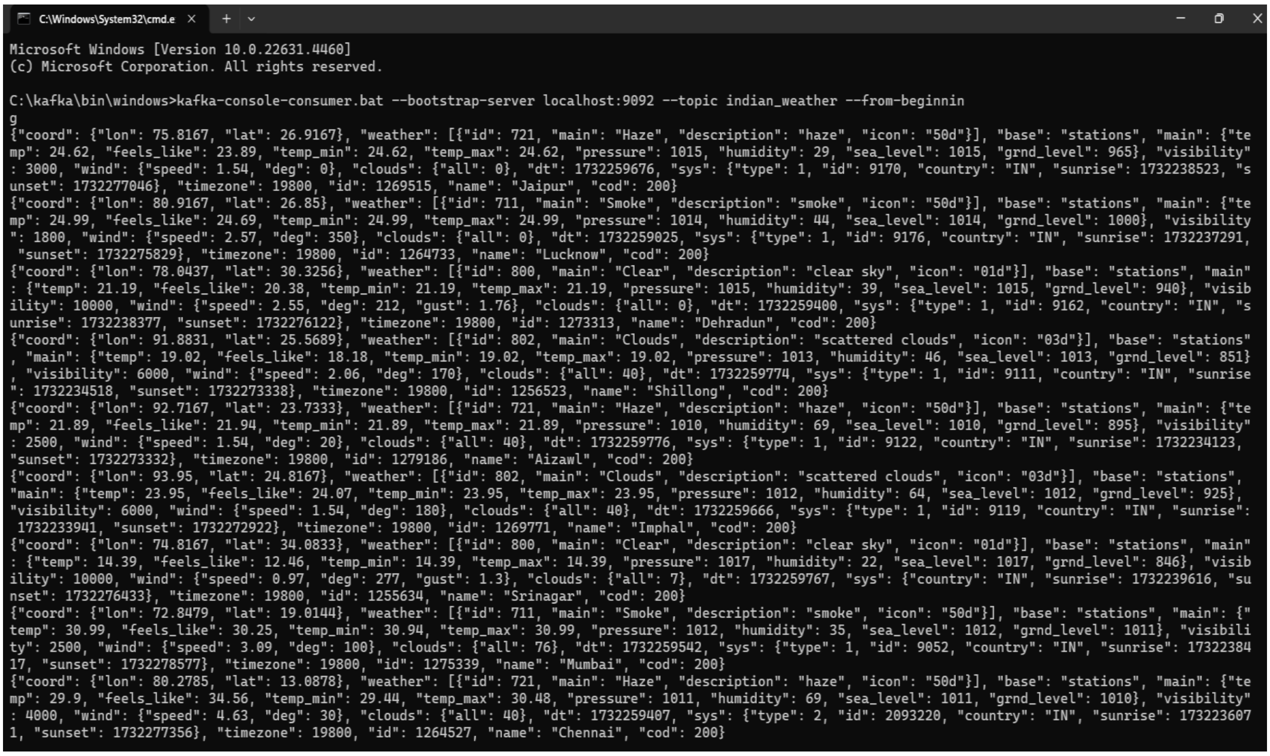
```
from confluent_kafka import Producer
import requests
import json
# Kafka Configuration
KAFKA_TOPIC = "indian_weather"
KAFKA_SERVER = "localhost:9092"
# Define Kafka Producer
def on_delivery(err, msg):
    if err is not None:
        print('Message delivery failed: {}'.format(err))
    else:
        print('Message delivered to {} [{}]'.format(msg.topic(), msg.partition()))

producer = Producer({'bootstrap.servers': KAFKA_SERVER})
# API Key and Endpoint
API_KEY = "d6ceb7623080049e80bedfa21a3d6795"
BASE_URL = "https://api.openweathermap.org/data/2.5/weather"
# List of Indian State Capitals
indian_capitals = [
    "New Delhi", "Mumbai", "Kolkata", "Chennai", "Bangalore", "Hyderabad",
    "Jaipur", "Bhopal", "Lucknow", "Patna", "Thiruvananthapuram",
    "Chandigarh", "Bhubaneswar", "Ranchi", "Dehradun", "Shillong",
    "Gangtok", "Itanagar", "Kohima", "Aizawl", "Agartala", "Imphal",
    "Panaji", "Shimla", "Srinagar", "Amaravati"
]
# Fetch and Send Weather Data for Each Capital
for city in indian_capitals:
    params = {
        "q": city,
        "appid": API_KEY,
        "units": "metric"  # Use "imperial" for Fahrenheit
    }
    response = requests.get(BASE_URL, params=params)
    # Check if Request is Successful
    if response.status_code == 200:
        data = response.json()
        print(f"Weather Data for {city}:", json.dumps(data, indent=4))  # Print data
        # Send data to Kafka
        producer.produce(KAFKA_TOPIC, json.dumps(data), callback=on_delivery)
    else:
        print(f"Error fetching data for {city}: {response.status_code}")
# Ensure all messages are sent
producer.flush()
print(f"Data sent to Kafka topic: {KAFKA_TOPIC}")
```

Run the Python Script from Command Line:

```
python Indian_weather.py
```



```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\kafka\bin\windows>kafka-topics.bat --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 3 --topic indian_weather
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use either, but not
 both.
Created topic indian_weather.

C:\kafka\bin\windows>python indian_weather.py
Weather Data for New Delhi: {
    "coord": {
        "lon": 77.2311,
        "lat": 28.6128
    },
    "weather": [
        {
            "id": 711,
            "main": "Smoke",
            "description": "smoke",
            "icon": "50d"
        }
    ],
    "base": "stations",
    "main": {
        "temp": 24.09,
        "feels_like": 23.63,
        "temp_min": 24.09,
        "temp_max": 24.09,
        "pressure": 1015,
        "humidity": 41,
        "sea_level": 1015,
        "grnd_level": 990
    },
    "visibility": 1400,
    "wind": {
        "speed": 1.03,
        "deg": 0
    },
    "clouds": {
        "all": 0
    },
```



```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\kafka\bin\windows>kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic indian_weather --from-beginnin
g
{"coord": {"lon": 75.8167, "lat": 26.9167}, "weather": [{"id": 721, "main": "Haze", "description": "haze", "icon": "50d"}], "base": "stations", "main": {"te
mp": 24.62, "feels_like": 23.89, "temp_min": 24.62, "temp_max": 24.62, "pressure": 1015, "humidity": 29, "sea_level": 1015, "grnd_level": 965}, "visibility"
: 3000, "wind": {"speed": 1.54, "deg": 0}, "clouds": {"all": 0}, "dt": 1732259676, "sys": {"type": 1, "id": 9170, "country": "IN", "sunrise": 1732238523, "s
unset": 1732277046}, "timezone": 19800, "id": 1269515, "name": "Jaipur", "cod": 200}
{"coord": {"lon": 80.9167, "lat": 26.85}, "weather": [{"id": 711, "main": "Smoke", "description": "smoke", "icon": "50d"}], "base": "stations", "main": {"te
mp": 24.99, "feels_like": 24.69, "temp_min": 24.99, "temp_max": 24.99, "pressure": 1014, "humidity": 44, "sea_level": 1014, "grnd_level": 1000}, "visibility
": 1800, "wind": {"speed": 2.57, "deg": 350}, "clouds": {"all": 0}, "dt": 1732259025, "sys": {"type": 1, "id": 9176, "country": "IN", "sunrise": 1732237291,
 "sunset": 1732275829}, "timezone": 19800, "id": 1264733, "name": "Lucknow", "cod": 200}
{"coord": {"lon": 78.0437, "lat": 30.3256}, "weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01d"}], "base": "stations", "main"
: {"temp": 21.19, "feels_like": 20.38, "temp_min": 21.19, "temp_max": 21.19, "pressure": 1015, "humidity": 39, "sea_level": 1015, "grnd_level": 940}, "visib
ility": 10000, "wind": {"speed": 2.55, "deg": 212, "gust": 1.76}, "clouds": {"all": 0}, "dt": 1732259400, "sys": {"type": 1, "id": 9162, "country": "IN", "s
unrise": 1732238377, "sunset": 1732276122}, "timezone": 19800, "id": 1273313, "name": "Dehradun", "cod": 200}
{"coord": {"lon": 91.8831, "lat": 25.5689}, "weather": [{"id": 802, "main": "Clouds", "description": "scattered clouds", "icon": "03d"}], "base": "stations"
, "main": {"temp": 19.02, "feels_like": 18.18, "temp_min": 19.02, "temp_max": 19.02, "pressure": 1013, "humidity": 46, "sea_level": 1013, "grnd_level": 851}
, "visibility": 6000, "wind": {"speed": 2.06, "deg": 170}, "clouds": {"all": 40}, "dt": 1732259774, "sys": {"type": 1, "id": 9111, "country": "IN", "sunrise
": 1732234518, "sunset": 1732273338}, "timezone": 19800, "id": 1256523, "name": "Shillong", "cod": 200}
{"coord": {"lon": 92.7167, "lat": 23.7333}, "weather": [{"id": 721, "main": "Haze", "description": "haze", "icon": "50d"}], "base": "stations", "main": {"te
mp": 21.89, "feels_like": 21.94, "temp_min": 21.89, "temp_max": 21.89, "pressure": 1010, "humidity": 69, "sea_level": 1010, "grnd_level": 895}, "visibility"
: 2500, "wind": {"speed": 1.54, "deg": 20}, "clouds": {"all": 40}, "dt": 1732259776, "sys": {"type": 1, "id": 9122, "country": "IN", "sunrise": 1732234123,
"sunset": 1732273332}, "timezone": 19800, "id": 1279186, "name": "Aizawl", "cod": 200}
{"coord": {"lon": 93.95, "lat": 24.8167}, "weather": [{"id": 802, "main": "Clouds", "description": "scattered clouds", "icon": "03d"}], "base": "stations",
"main": {"temp": 23.95, "feels_like": 24.07, "temp_min": 23.95, "temp_max": 23.95, "pressure": 1012, "humidity": 64, "sea_level": 1012, "grnd_level": 925},
"visibility": 6000, "wind": {"speed": 1.54, "deg": 180}, "clouds": {"all": 40}, "dt": 1732259666, "sys": {"type": 1, "id": 9119, "country": "IN", "sunrise":
 1732233941, "sunset": 1732272922}, "timezone": 19800, "id": 1269771, "name": "Imphal", "cod": 200}
{"coord": {"lon": 74.8167, "lat": 34.0833}, "weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01d"}], "base": "stations", "main"
: {"temp": 14.39, "feels_like": 12.46, "temp_min": 14.39, "temp_max": 14.39, "pressure": 1017, "humidity": 22, "sea_level": 1017, "grnd_level": 846}, "visib
ility": 10000, "wind": {"speed": 0.97, "deg": 277, "gust": 1.3}, "clouds": {"all": 7}, "dt": 1732259767, "sys": {"country": "IN", "sunrise": 1732239616, "su
nset": 1732276433}, "timezone": 19800, "id": 1255634, "name": "Srinagar", "cod": 200}
{"coord": {"lon": 72.8479, "lat": 19.0144}, "weather": [{"id": 711, "main": "Smoke", "description": "smoke", "icon": "50d"}], "base": "stations", "main": {"
temp": 30.99, "feels_like": 30.25, "temp_min": 30.94, "temp_max": 30.99, "pressure": 1012, "humidity": 35, "sea_level": 1012, "grnd_level": 1011}, "visibili
ty": 2500, "wind": {"speed": 3.09, "deg": 100}, "clouds": {"all": 76}, "dt": 1732259542, "sys": {"type": 1, "id": 9052, "country": "IN", "sunrise": 17322384
17, "sunset": 1732278577}, "timezone": 19800, "id": 1275339, "name": "Mumbai", "cod": 200}
{"coord": {"lon": 80.2785, "lat": 13.0878}, "weather": [{"id": 721, "main": "Haze", "description": "haze", "icon": "50d"}], "base": "stations", "main": {"te
mp": 29.9, "feels_like": 34.56, "temp_min": 29.44, "temp_max": 30.48, "pressure": 1011, "humidity": 69, "sea_level": 1011, "grnd_level": 1010}, "visibility"
: 4000, "wind": {"speed": 4.63, "deg": 30}, "clouds": {"all": 40}, "dt": 1732259407, "sys": {"type": 2, "id": 2093220, "country": "IN", "sunrise": 173223607
1, "sunset": 1732277356}, "timezone": 19800, "id": 1264527, "name": "Chennai", "cod": 200}
```
]