Table of Contents

Awesome! Let's now perform:

✅ Experiment 6: Selenium Script for Automated Testing

🎯 Goal: Use Selenium to automate opening a browser and performing a search

---

💻 Tools Needed:

Tool	Use	Install
Python	Scripting language	Download
pip	Python package manager	Comes with Python
Selenium	Python library to automate browser	`pip install selenium`
ChromeDriver	Driver to control Chrome	chromedriver.chromium.org
Google Chrome	Browser to test	Already installed on most systems

---

✅ Step-by-Step Setup

🔧 Step 1: Install Python + Selenium

Open CMD or PowerShell:

```bash
python --version
pip --version
pip install selenium
```

✅ If installed, you're good to go.

📁 Step 2: Download ChromeDriver

1. Go to: https://chromedriver.chromium.org/downloads
2. Choose version matching your Chrome version
   - (You can check your version in Chrome → Help → About Chrome)
3. Extract and place `chromedriver.exe` in the same folder as your script, or add to PATH

📝 Step 3: Write the Selenium Script (test_google.py)

Open VS Code / Notepad / any editor, paste:

```python

```

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

# Set up Chrome driver
driver = webdriver.Chrome()

# Open Google
driver.get("https://www.google.com")

# Find the search box
search = driver.find_element(By.NAME, "q")
search.send_keys("DevOps")

# Submit the search
search.submit()

# Wait a few seconds
time.sleep(3)

# Print title of page
print("Page title is:", driver.title)

# Close browser
driver.quit()
```

▶️ Step 4: Run the Script

In the same folder as `test_google.py`, run:

```bash
python test_google.py
```

✅ Chrome will:

- Open Google
- Type "DevOps" in search
- Submit the search
- Print the page title
- Close the browser

✅ You Just Performed:

- Automated browser testing using Selenium
- This is widely used in CI/CD pipelines for UI testing!

❓ Bonus: Can I integrate this with Jenkins?

Yes! You can add this as a build step in Jenkins too:

- Create a Freestyle job
- Place the script in workspace
- Add a build step:

```bash
python test_google.py
```

Would you like to now go to:

🧪 Experiment 7: Dockerize PHP project and push to DockerHub?

Let's containerize something next! 🐳

Perfect! You're referring to your Google Chrome version: `136.0.7103.49`, which is exactly what we need to match the correct ChromeDriver.

✅ Step-by-Step: Download ChromeDriver for Version `136`

🔗 Step 1: Go to ChromeDriver Downloads

👉 Visit:
https://googlechromelabs.github.io/chrome-for-testing/
(This is the new official link for ChromeDriver 115+)

🔍 Step 2: Download Matching Driver

1. Scroll down to Chrome 136 section.
2. Under Version: 136.0.7103.39 (or closest available), click:

   ```python
   chromedriver-win64 zip
   ```

3. Download and extract the file → You'll get `chromedriver.exe`

📁 Step 3: Place ChromeDriver Properly

Now do one of the following:

Option	What to do
☑ Recommended	Place `chromedriver.exe` in the same folder as your Python Selenium script
OR	Add the folder path containing `chromedriver.exe` to your system's `Path` variable

---

▶ Step 4: Test It Works

In the folder where `chromedriver.exe` is, run this Python code:

```python
from selenium import webdriver

driver = webdriver.Chrome()
driver.get("https://www.google.com")
driver.quit()
```

✅ If it opens Chrome and exits, you're good to go!

Ready to go back to writing the full Selenium script to search "DevOps"?
Let me know and I'll paste the final code again for testing!