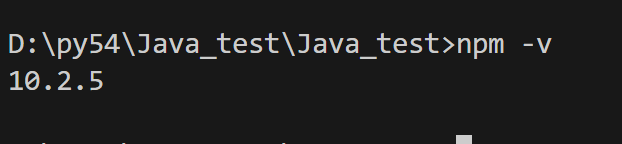
**Installation and Exploration of Selenium for Automated Testing**

**What is Selenium?**

**Selenium is an open-source tool used for automating web browsers. It provides a suite of tools for testing web applications across different browsers and platforms. The Selenium WebDriver API allows you to interact with web pages programmatically by simulating user actions like clicking, typing, navigating, etc.**

**Installation of Selenium WebDriver**

**Step 1: Install Node.js and npm**

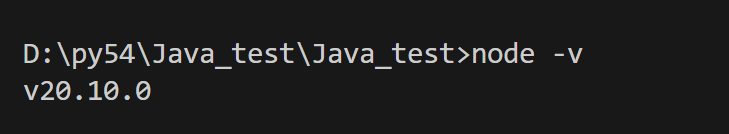
****

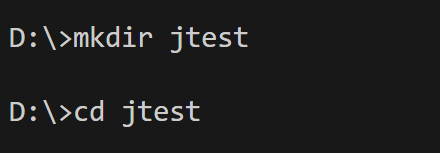
**If you don’t have Node.js installed, download it from** [**Node.js official website**](https://nodejs.org/)**. Follow the instructions for your operating system.**

1. **Verify the installation by running the following commands in the terminal:**

**node -v**

**npm -v**

****

****

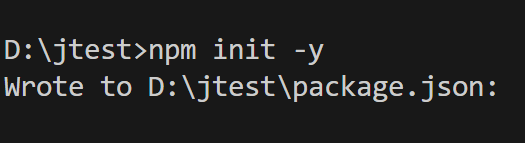
**Step 2: Create a Project Folder**

1. **Open Visual Studio Code.**
2. **Create a new folder named SeleniumLab.**
3. **Open the terminal in VS Code and initialize a Node.js project:**

**bash**

**Copy code**

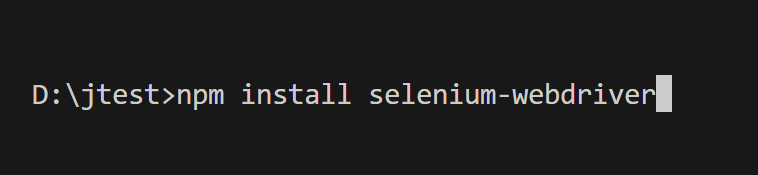
**npm init -y**

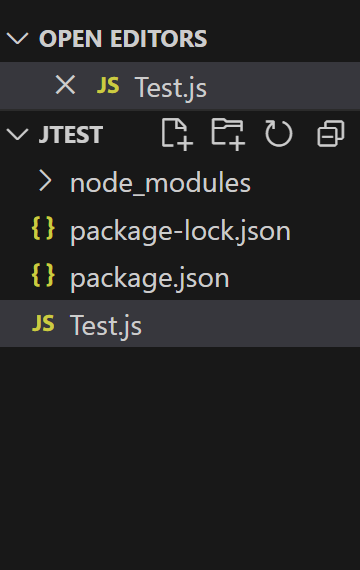
****

**Step 3: Install Selenium WebDriver**

1. **Install the selenium-webdriver package:**

**npm install selenium-webdriver**

****

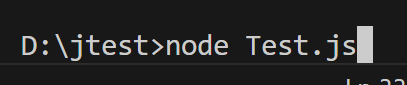
1. **Download the ChromeDriver that matches your Chrome browser version from ChromeDriver Downloads.**
2. **Extract the downloaded ChromeDriver and place it in the project folder.**
3. ****

**Part 3: Writing a Simple Selenium Test Script**

**Step 1: Create a JavaScript Test File**

1. **Inside the SeleniumLab folder, create a new file named test.js.**
2. **Copy and paste the following code into test.js:**

**Test.js**

****

**const { Builder, By, Key, until } = require('selenium-webdriver');**

**async function testGoogleSearch() {**

**// Step 1: Launch the Chrome browser**

**let driver = await new Builder().forBrowser('chrome').build();**

**try {**

**// Step 2: Open the Google homepage**

**await driver.get('https://www.google.com');**

**// Step 3: Locate the search box and enter a search query**

**await driver.findElement(By.name('q')).sendKeys('Selenium WebDriver', Key.RETURN);**

**// Step 4: Wait for the search results to load**

**await driver.wait(until.titleContains('Selenium WebDriver'), 5000);**

**// Step 5: Print the page title in the console**

**let title = await driver.getTitle();**

**console.log('Page Title:', title);**

**// Step 6: Check if the search results are displayed**

**if (title.includes('Selenium WebDriver')) {**

**console.log('Test Passed: Search results loaded successfully.');**

**} else {**

**console.log('Test Failed: Search results not loaded.');**

**}**

**} catch (error) {**

**console.error('Error:', error);**

**} finally {**

**// Step 7: Close the browser**

**await driver.quit();**

**}**

**}**

**testGoogleSearch();**

**Explanation of the Code:**

1. **Import Selenium WebDriver:**
   * **Builder: Used to create a new browser instance.**
   * **By: Used to locate elements on the web page.**
   * **Key: Simulates keyboard actions.**
   * **until: Used for waiting until a condition is met.**
2. **testGoogleSearch Function:**
   * **Step 1: Launches the Chrome browser.**
   * **Step 2: Opens https://www.google.com.**
   * **Step 3: Finds the search box using its name attribute and types a query (Selenium WebDriver).**
   * **Step 4: Waits until the page title includes "Selenium WebDriver".**
   * **Step 5: Prints the page title.**
   * **Step 6: Verifies if the search results are displayed correctly.**
   * **Step 7: Closes the browser.**

**Part 4: Running the Selenium Test Script**

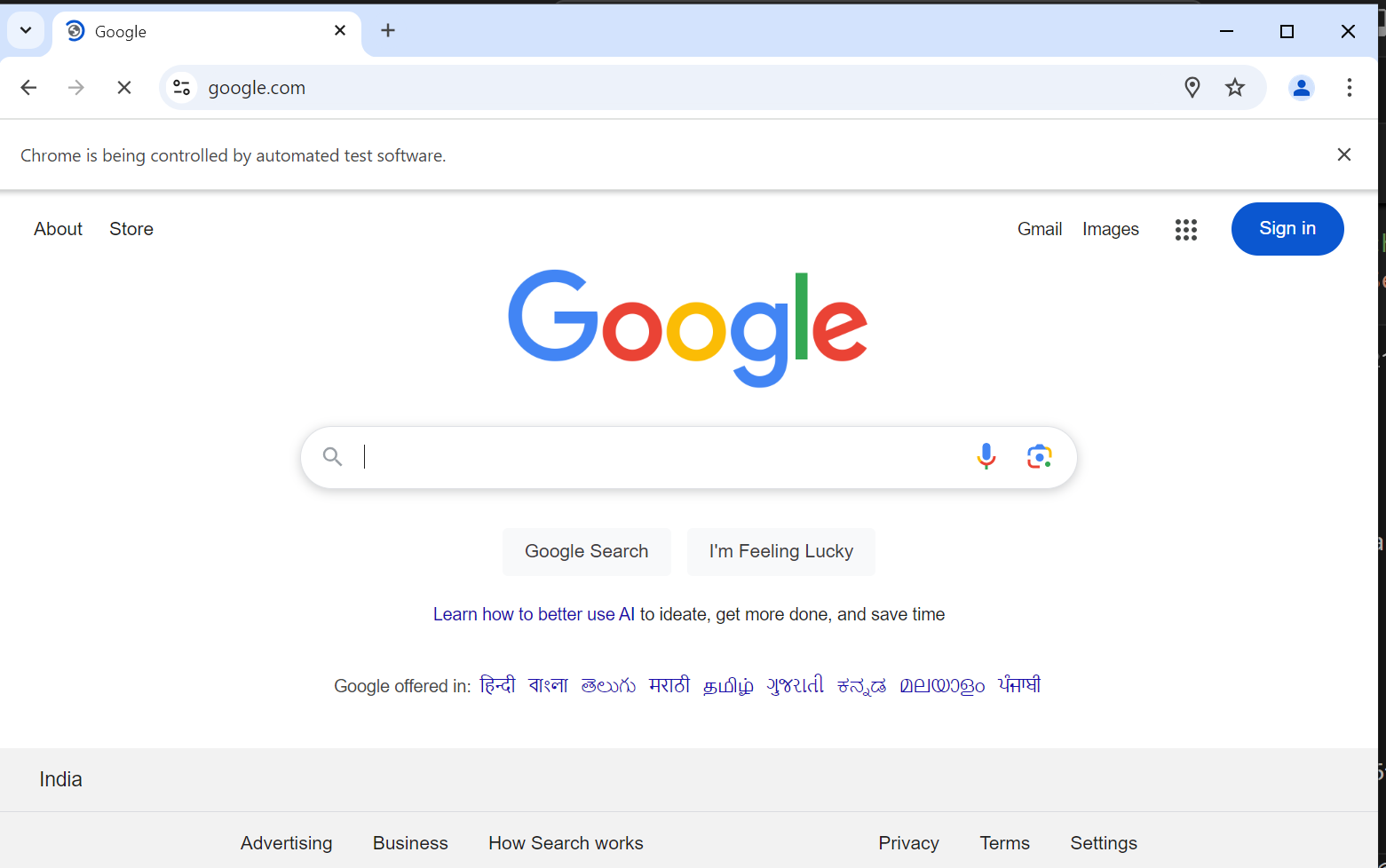
**Step 1: Running the Script**

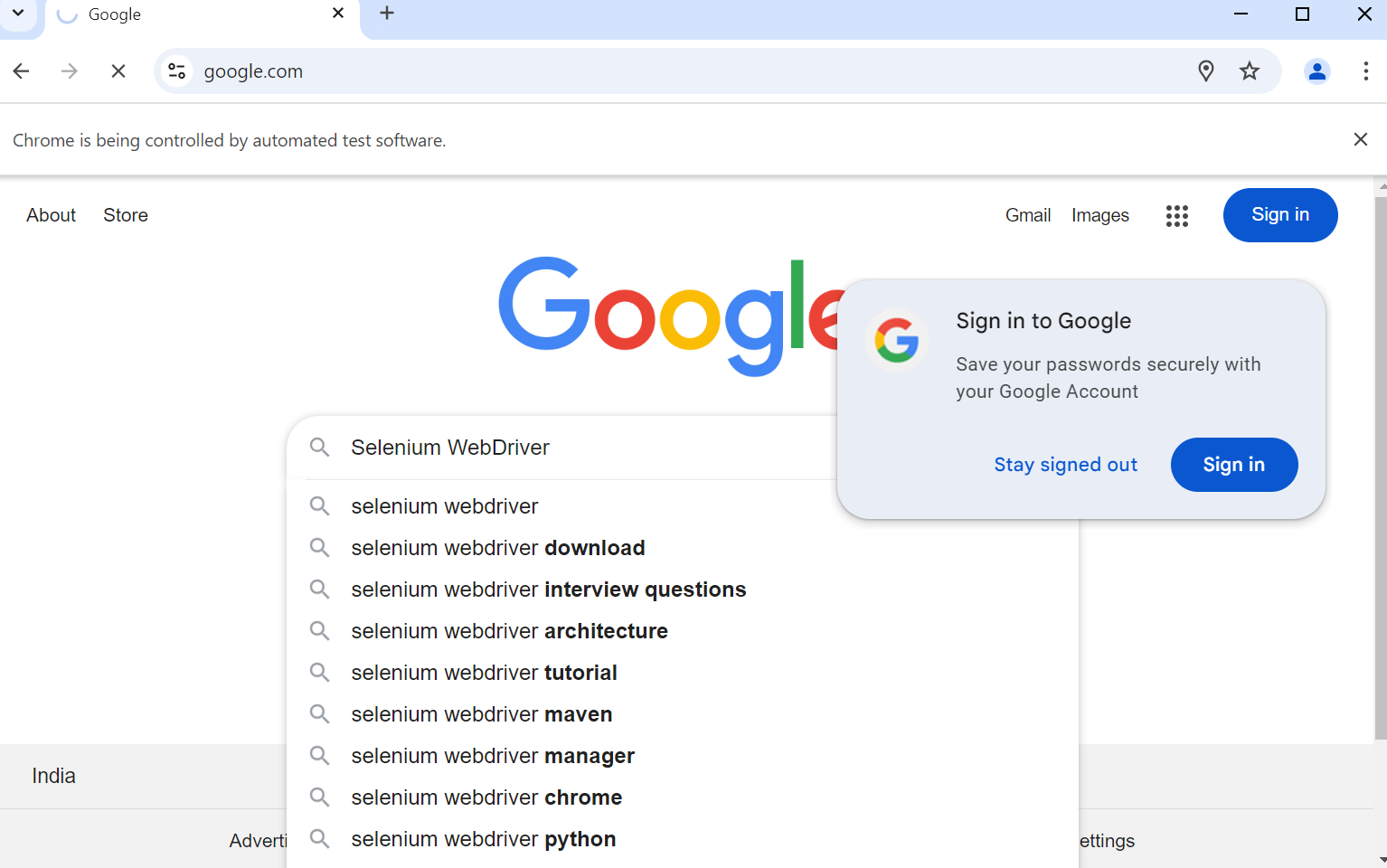
1. **Make sure that ChromeDriver is in the project folder.**
2. **Open the terminal in Visual Studio Code.**
3. **Run the test script:**

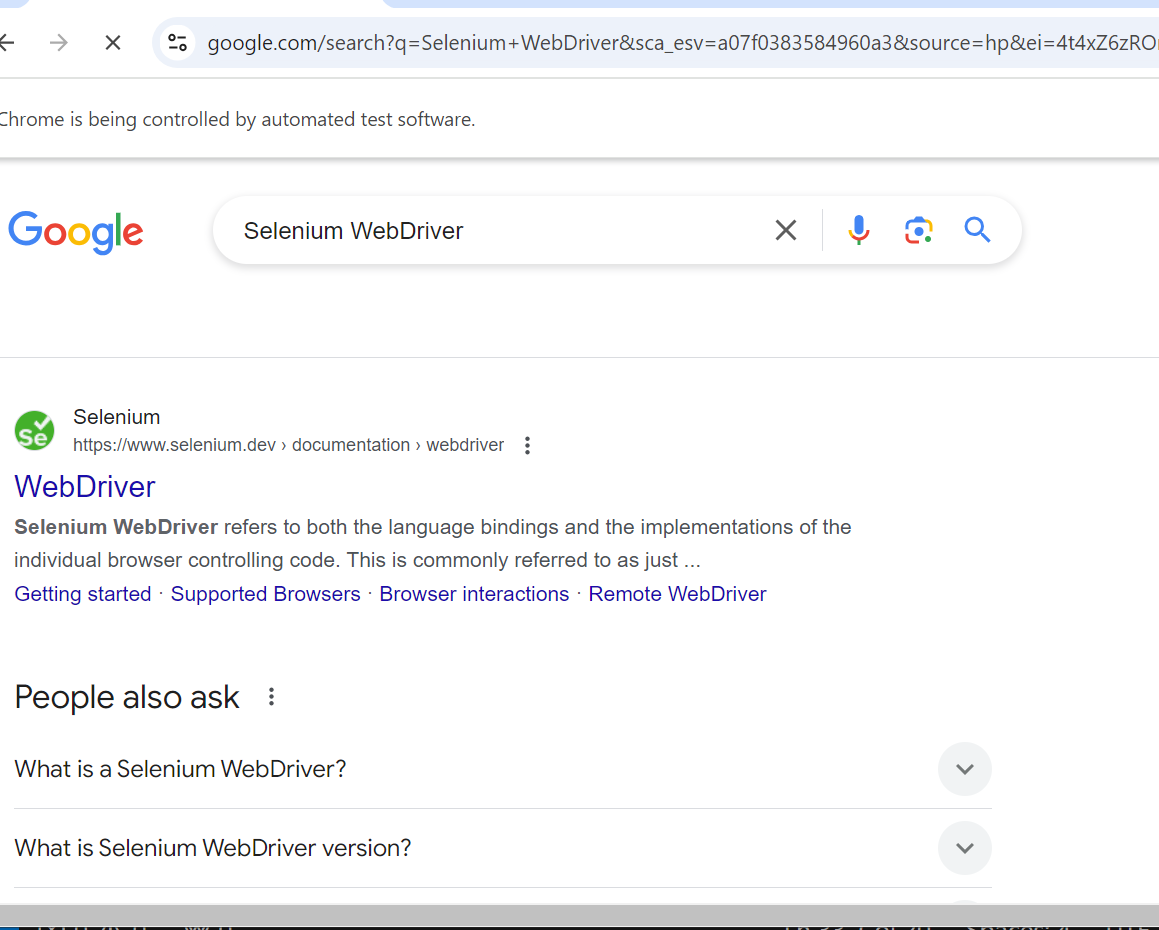
**node Test.js**

**Expected Output:**

* **The Chrome browser should open, navigate to Google, perform a search for "Selenium WebDriver", and display the search results.**
* **The terminal should display:**

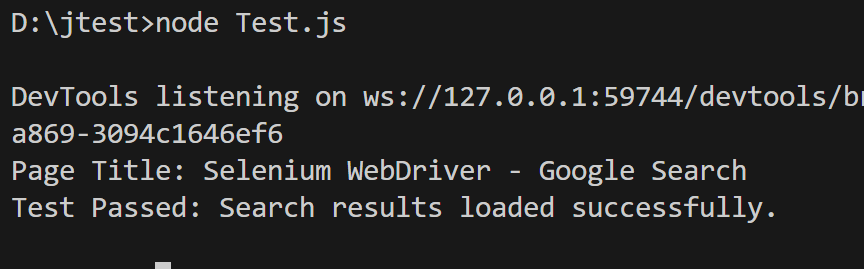






**Page Title: Selenium WebDriver - Google Search**

**Test Passed: Search results loaded successfully.**



**Part 5: Exploring Selenium Commands**

**Commonly Used Selenium Commands**

1. **Navigating to a URL:**

**await driver.get('https://example.com');**

1. **Locating Elements:**
   * **By ID:**

**await driver.findElement(By.id('elementId'));**

* + **By Class Name:**

**await driver.findElement(By.className('className'));**

* + **By Name:**

**await driver.findElement(By.name('elementName'));**

1. **Performing Actions:**
   * **Click an Element:**

**await driver.findElement(By.id('buttonId')).click();**

* + **Enter Text:**

**await driver.findElement(By.name('inputName')).sendKeys('Hello World');**

1. **Waiting for Elements:**

**await driver.wait(until.elementLocated(By.id('elementId')), 5000);**

1. **Closing the Browser:**

**await driver.quit();**

**Part 6: Viva Questions**

1. **What is Selenium, and what is it used for?**
2. **What is the role of ChromeDriver in Selenium automation?**
3. **Explain the purpose of await and async in the Selenium test script.**
4. **How can you locate web elements using Selenium WebDriver?**
5. **What is the difference between functional testing and automated testing?**

**Part 7: Assignment**

1. **Modify the test script to automate the login process for a sample website.**
2. **Write test cases to validate the registration form of a web application using Selenium.**
3. **Explore the use of other web drivers like FirefoxDriver and EdgeDriver.**