Task 1: Page Title Verification

Topic: Page Interaction Commands

Description: Write a simple Selenium script to open a specified URL and verify the title of the page.

Steps:

Launch Eclipse IDE and create a new Java class in the Selenium project.

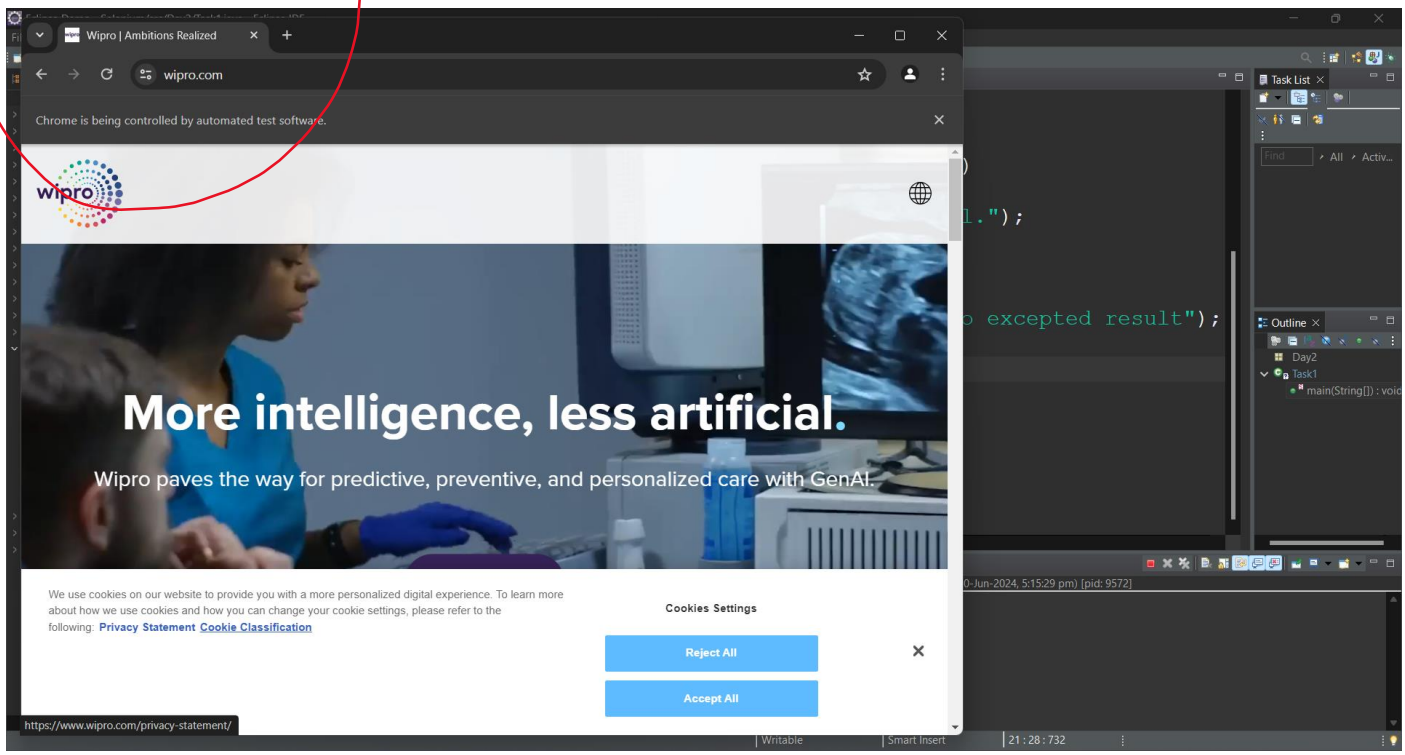
In the main method, instantiate the WebDriver and navigate to 'http://example.com'.

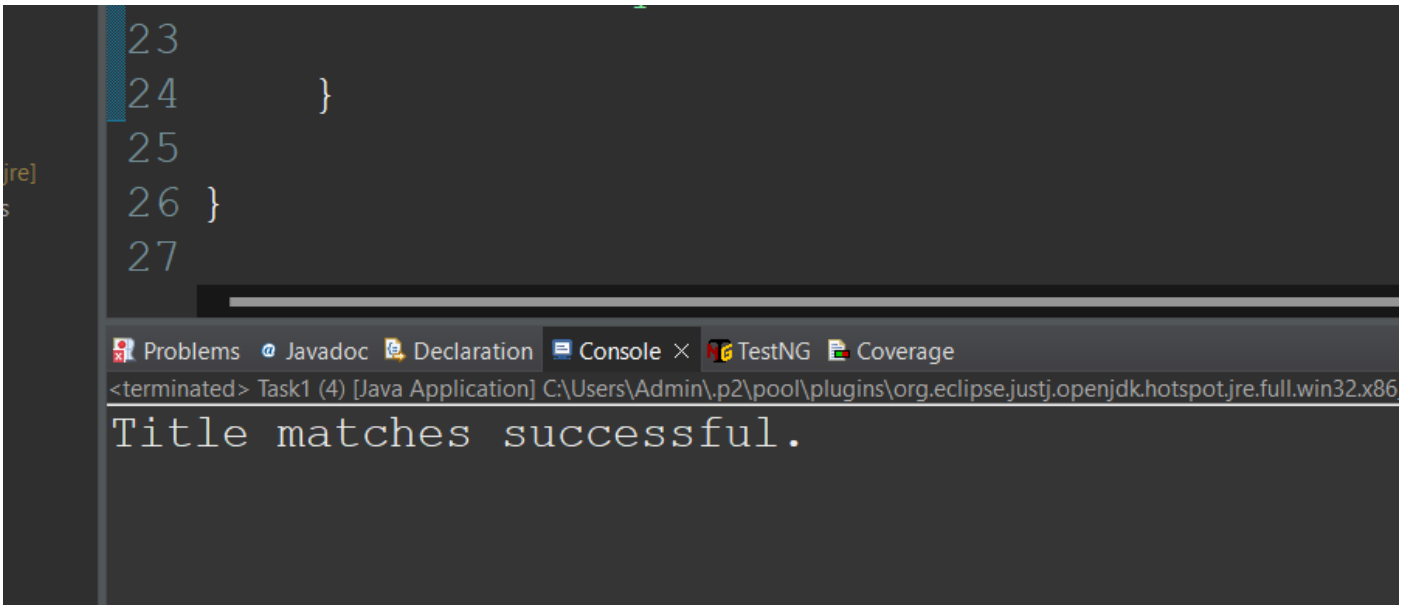
Use driver.getTitle() to retrieve the page title and store it in a variable.

Assert that the page title matches the expected title using a simple if statement that prints out the verification result.

```
package Day2;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Task1 {

        public static void main(String[] args) throws InterruptedException {

        System.setProperty("WebDriver.chrome.driver","C:\\Users\\Admin\\Downloads\\chromedriver-win64\\chromedriver-win64");
    WebDriver driver=new ChromeDriver();
    driver.get("https://www.wipro.com/");
    String title=driver.getTitle();
    if(title.contains("Wipro | Ambitions Realized"))
    {
        System.out.println("Title matches successful.");
    }
    else
    {
        System.out.println("Title Not not matches to excepted result");
    }
    Thread.sleep(4000);
```

```
        driver.quit();

            }
}
```





Close the browser using driver.quit().

Browser closed successfully.

Task 2: Element Interaction and State Verification

Topic: WebElements Commands and Identification

Description: Write a Selenium script to interact with an input field on a form, enter text, and verify the text was entered correctly.

Steps:

Open the same Java class used in Task 1.

Navigate to a web page with an input field (e.g., a search box).

Use driver.findElement() to locate the input field by its name or ID.

Enter text using the sendKeys() method and clear it with clear()

Retrieve the input field value and verify it matches the text entered.

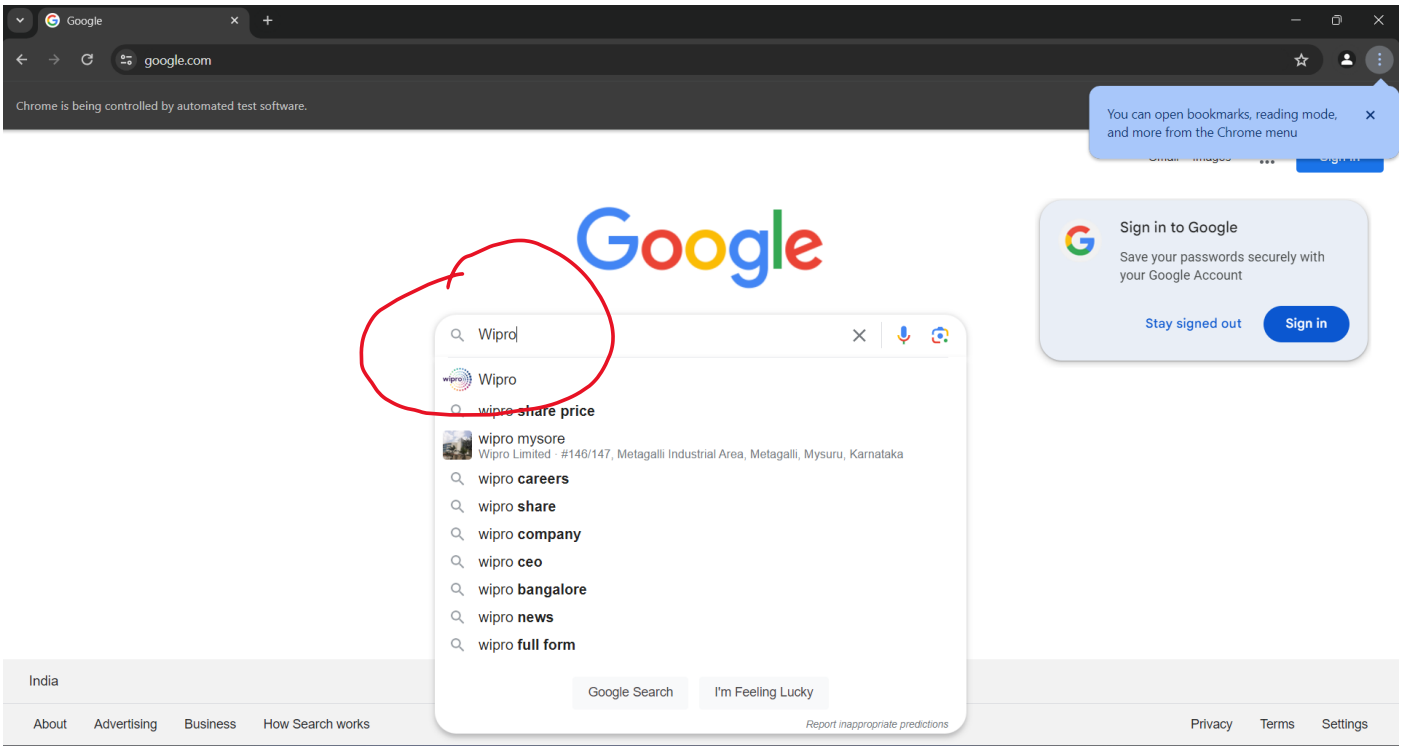
Verify the input field is displayed and enabled by using isDisplayed() and isEnabled().

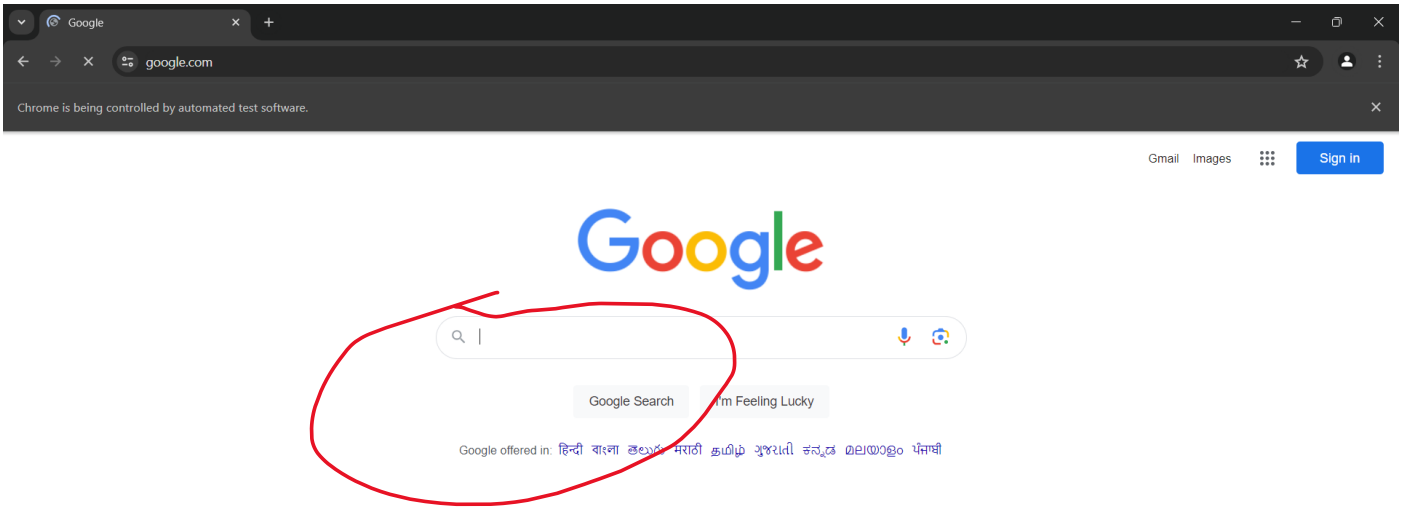
```
package Day2;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class Task2 {

    public static void main(String[] args) throws InterruptedException {

    System.setProperty("WebDriver.chrome.driver","C:\\Users\\Admin\\Downloads\\chromedriver-win64\\chromedriver-win64");
  WebDriver driver=new ChromeDriver();
  driver.get("https://google.com");
  String title=driver.getTitle();

  if(title.contains("Google"))
  {
    System.out.println("Title matches successful.");
  }
```

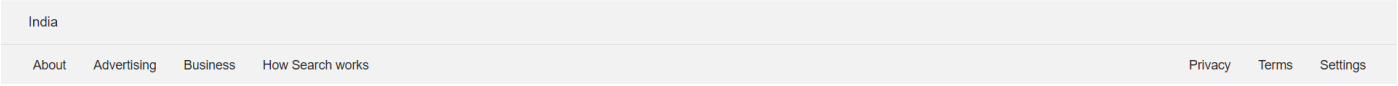
```
else
{
    System.out.println("Title Not not matches to excepted result");
}
WebElement search=driver.findElement(By.id("APjFqb"));
boolean Displayed=search.isDisplayed();
    boolean Enabled=search.isEnabled();

if(Displayed && Enabled)
{
    System.out.println("Search box is enabled.");
}
else
{
    System.out.println("Text box is not enabled.");
}
String input="Wipro";
    search.sendKeys(input);
    String Text = search.getAttribute("value");

    if (Text.equals(input))
    {
   System.out.println("Text verification successful.");
}
    else
    {
   System.out.println("Text verification failed.");
}
    search.click();
    Thread.sleep(4000);
    search.clear();
    Text=search.getAttribute("value");
```

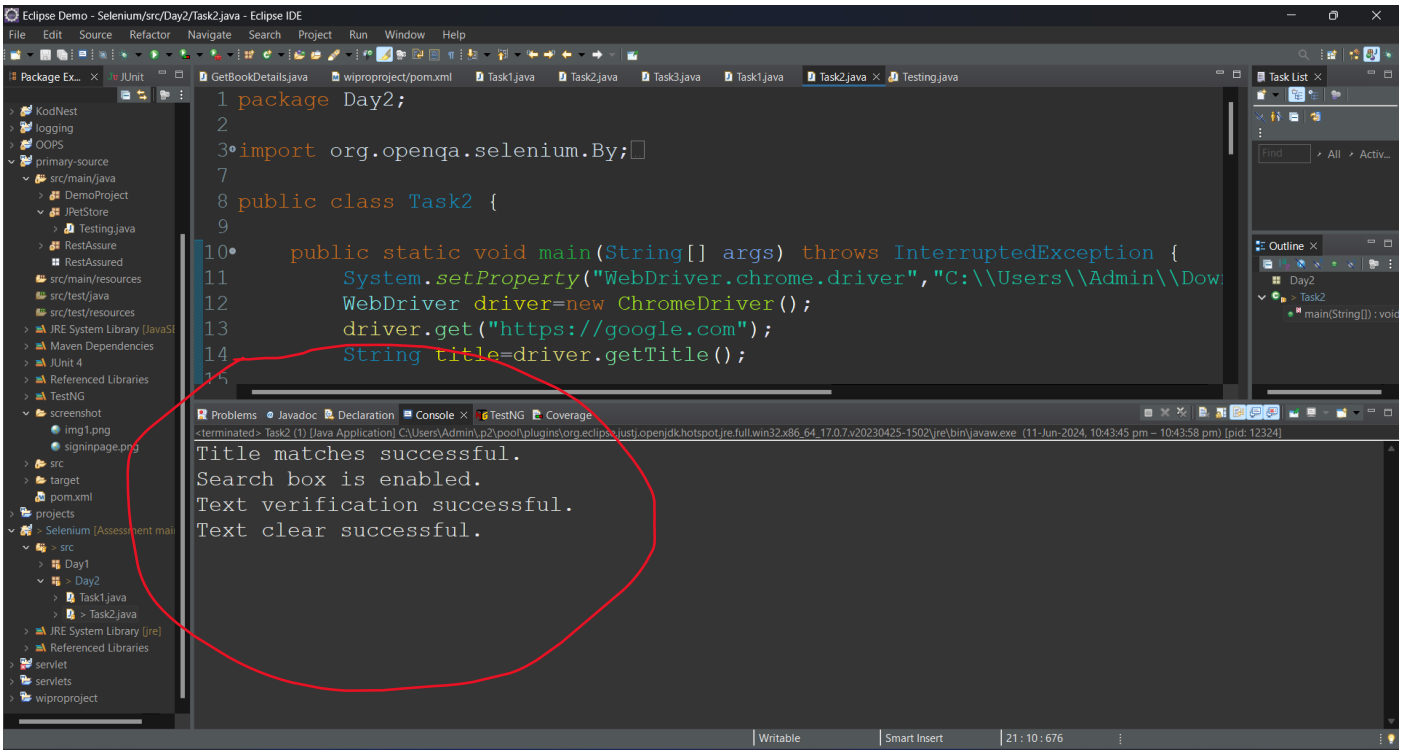
```
    if (Text.isEmpty()) {
        System.out.println("Text clear successful.");
    }
    else
    {
        System.out.println("Text clear failed");
    }
    driver.quit();
        }
}
```

OutPut:




Google
google.com
Chrome is being controlled by automated test software.
Gmail
Images
Sign in
Google
Google Search
I'm Feeling Lucky
Google offered in: हिन्दी বাংলা తెలుగు मराठी தமிழ் ગુજરાતી ಕನ್ನಡ മലയാളം ਪੰਜਾਬੀ


India
About
Advertising
Business
How Search works
Privacy
Terms
Settings


Eclipse Demo - Selenium/src/Day2/Task2.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Ex...
JUnit
GetBookDetails.java
wiproproject/pom.xml
Task1.java
Task2.java
Task3.java
Task1.java
Task2.java
Testing.java
Task List
Outline
Day2
Task2
main(String[]) : void
KodNest
logging
OOPS
primary-source
src/main/java
DemoProject
JPetStore
Testing.java
RestAssure
RestAssured
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE
Maven Dependencies
JUnit 4
Referenced Libraries
TestNG
screenshot
img1.png
signinpage.png
src
target
pom.xml
projects
Selenium [Assessment mai
src
Day1
Day2
Task1.java
Task2.java
JRE System Library [jre]
Referenced Libraries
servlet
servlets
wiproproject
1 package Day2;
2
3 import org.openqa.selenium.By;
7
8 public class Task2 {
9
10 public static void main(String[] args) throws InterruptedException {
11 System.setProperty("WebDriver.chrome.driver","C:\\Users\\Admin\\Dow
12 WebDriver driver=new ChromeDriver();
13 driver.get("https://google.com");
14 String title=driver.getTitle();
Problems
Javadoc
Declaration
Console
TestNG
Coverage
<terminated> Task2 (1) [Java Application] C:\Users\Admin\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (11-Jun-2024, 10:43:45 pm – 10:43:58 pm) [pid: 12324]
Title matches successful.
Search box is enabled.
Text verification successful.
Text clear successful.
Writable
Smart Insert
21 : 10 : 676

Task 3: Locator Strategy Implementation

Topic: Locator Techniques and Effective Element Selection

Description: Craft CSS Selector and XPath locators for an element on a web page and demonstrate the selection in a Selenium script.

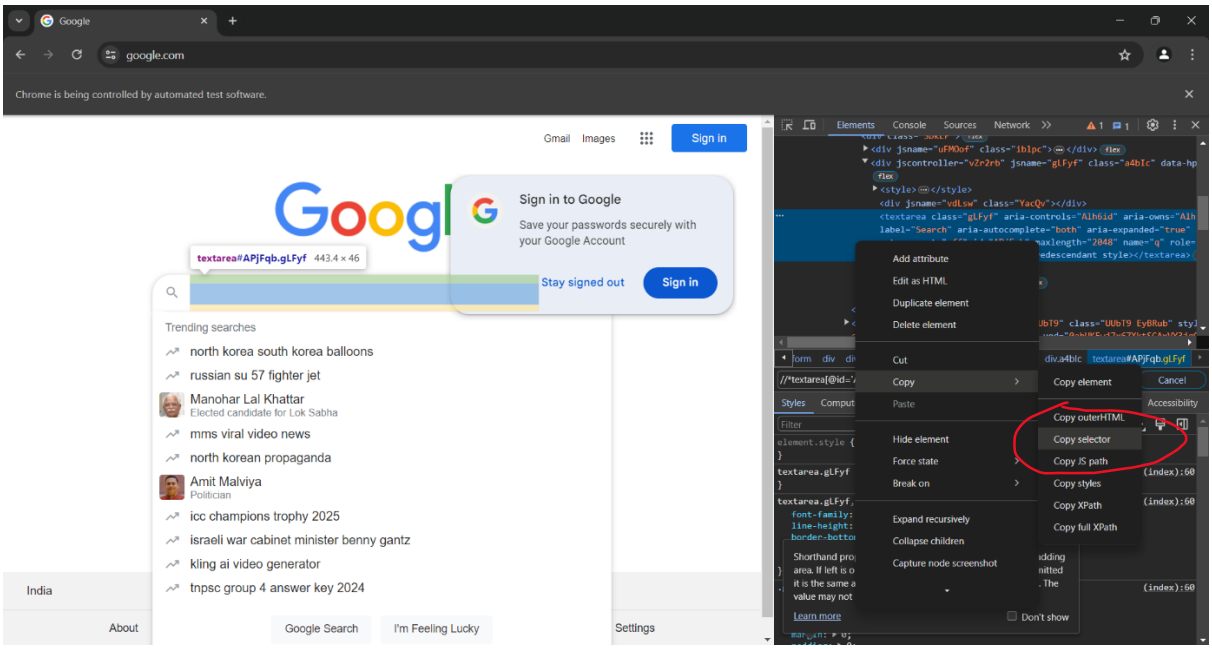
Steps:

Use the Eclipse IDE to open the existing Selenium project.

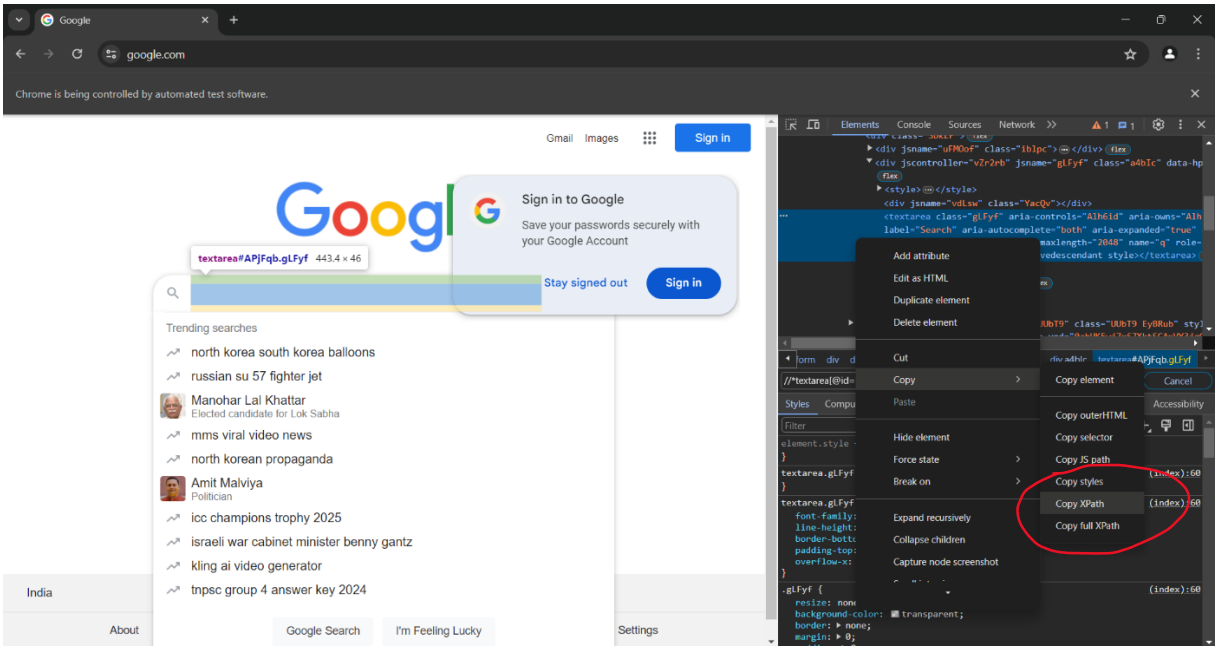
Open a web page that has a uniquely identifiable element (like a button with an ID).

Write two separate locators in the Selenium script: one using CSS Selector and one using XPath.

CSS Selector:



Xpath:



Use driver.findElement() to find the element using both locators separately, printing out a confirmation that the element has been found.

```
package Day2;


import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;


public class Task3 {


        public static void main(String[] args) throws InterruptedException {


        System.setProperty("WebDriver.chrome.driver","C:\\Users\\Admin\\Downloads\\chromedriver-win64\\chromedriver-win64");

    WebDriver driver=new ChromeDriver();

    driver.get("https://google.com");

    WebElement selector=driver.findElement(By.cssSelector("#APjFqb"));

    selector.sendKeys("css selector");

    Thread.sleep(2000);

    System.out.println("Element found using css selector");

    selector.clear();

    Thread.sleep(2000);

    WebElement xpath=driver.findElement(By.xpath("//*[@id=\"APjFqb\"]"));

    xpath.sendKeys("XPath");

    Thread.sleep(2000);

    System.out.println("Element found using xpath.");

    driver.quit();

        }

}
```

Output:

```
19          Thread.sleep(2000);
20          WebElement xpath=driver.findElem
```

Problems Javadoc Declaration Console TestNG Coverage

```
<terminated> Task3 (2) [Java Application] C:\Users\Admin\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.wi
Element found using css selector
Element found using xpath.
```

Discuss the difference between absolute and relative XPath expressions and when to use each.

- XPath is slower in speed and CSS selector is faster than XPath.
- XPath allows bidirectional flow and CSS selector allows only one direction flow.
- Xpath allows identification with the help of visible text appearing on screen with the help of text() function. CSS does not have this feature.
- There are two types of Xpath – absolute and relative. But CSS has no types.