**ASSISTED PROJECT SELENIUM EMP ID – 2587303**

**NAME : R.HIMABINDU**

Demonstrate the features of Selenium.

Features of Selenium

•**Open-Source:**

Selenium is a freeware and a portable tool. It has no upfront direct costs involved. The tool can

be freely downloaded and the support for it is freely available, as it is community-based.

•**Supports languages:**

Selenium supports a range of languages, including Java, Perl, Python, C#, Ruby, Groovy, Java

Script, etc. It has its own script, but it doesn’t limit it to that language. It can work with various

languages and whatever the developers/testers are comfortable with.

•**Supports Operating Systems:**

Selenium operates across and supports multiple Operating Systems, (OS) like Windows, Mac,

Linux, UNIX, etc. With Selenium Suite of solutions, a tailored testing suite can be created over

any platform and then executed on another one. For instance, you can create test cases using

Windows OS and run it with ease on a Linux-based system.

•**Supports multiple browsers:**

Selenium provides support across multiple browsers, namely, Internet Explorer, Chrome, Firefox,

Opera, Safari, etc. This becomes highly resourceful while executing tests and testing it across

various browsers simultaneously.

The browsers supported by the Selenium packages are:

•Selenium IDE can be used with Firefox as a plug-in.

•Selenium RC and Webdriver supports diverse browsers, such as Internet Explorer.

•Supports programming languages and frameworks

Selenium integrates with programming languages and various frameworks. For instance, it can

integrate with ANT or Maven type of framework for source code compilation. Furthermore, it can

integrate with the TestNG testing framework for testing applications and reporting purposes. It

can integrate with Jenkins or Hudson for Continuous Integration (CI) and can even integrate with

other Open-Source tools to support other features.

•**Tests across devices**

Selenium Test Automation can be implemented for mobile web application automation on

Android, IPhone, and Blackberry. This can help in generating necessary results and addresses

issues on a continuous basis.

•**Constant updates**

Selenium support is community-based and active community support enables constant updates

and upgrades. These upgrades are readily available and do not require specific training. This

makes Selenium resourceful and cost-effective as well.

•**Loaded Selenium Suites**

Selenium is not just a singular tool or utility, it is a loaded package of various testing tools and so

is referred to as a Suite. Each tool is designed to cater to different testing needs and

requirements of test environments.

Additionally, Selenium comes with capabilities to support Selenium IDE, Selenium Grid, and

Selenium Remote Control (RC).

•**Ease of implementation**Selenium offers a user-friendly interface that helps create and execute tests easily and

effectively. Its open-source features help users to script their own extensions which makes it

easy to develop customized actions and even manipulate at an advanced level. Tests run directly

across browsers and users can watch while the tests are being executed. Additionally,

Selenium’s reporting capabilities are one of the reasons for being chosen, as it allows testers to

extract results and take follow-up actions.

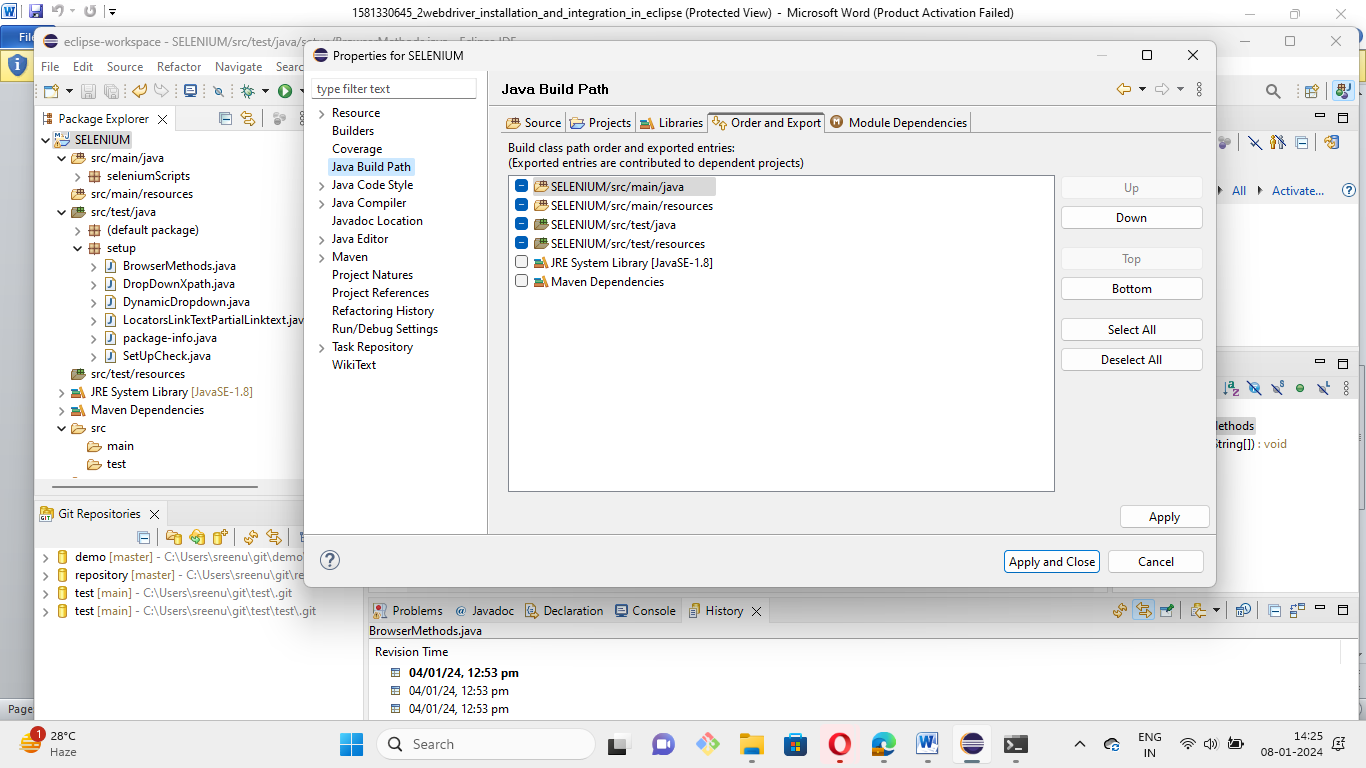
•**Reusability and Add-ons**

Selenium Test Automation Framework uses scripts that can be tested directly across multiple

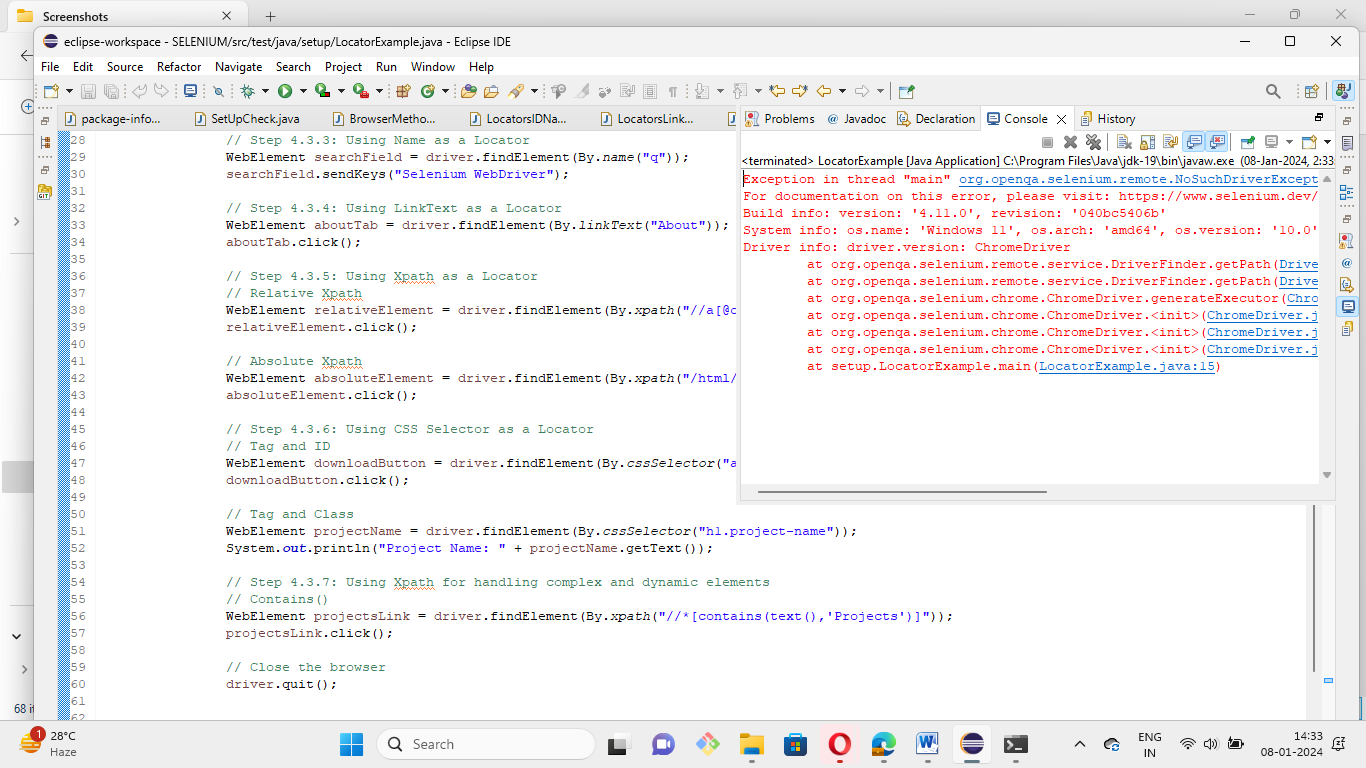
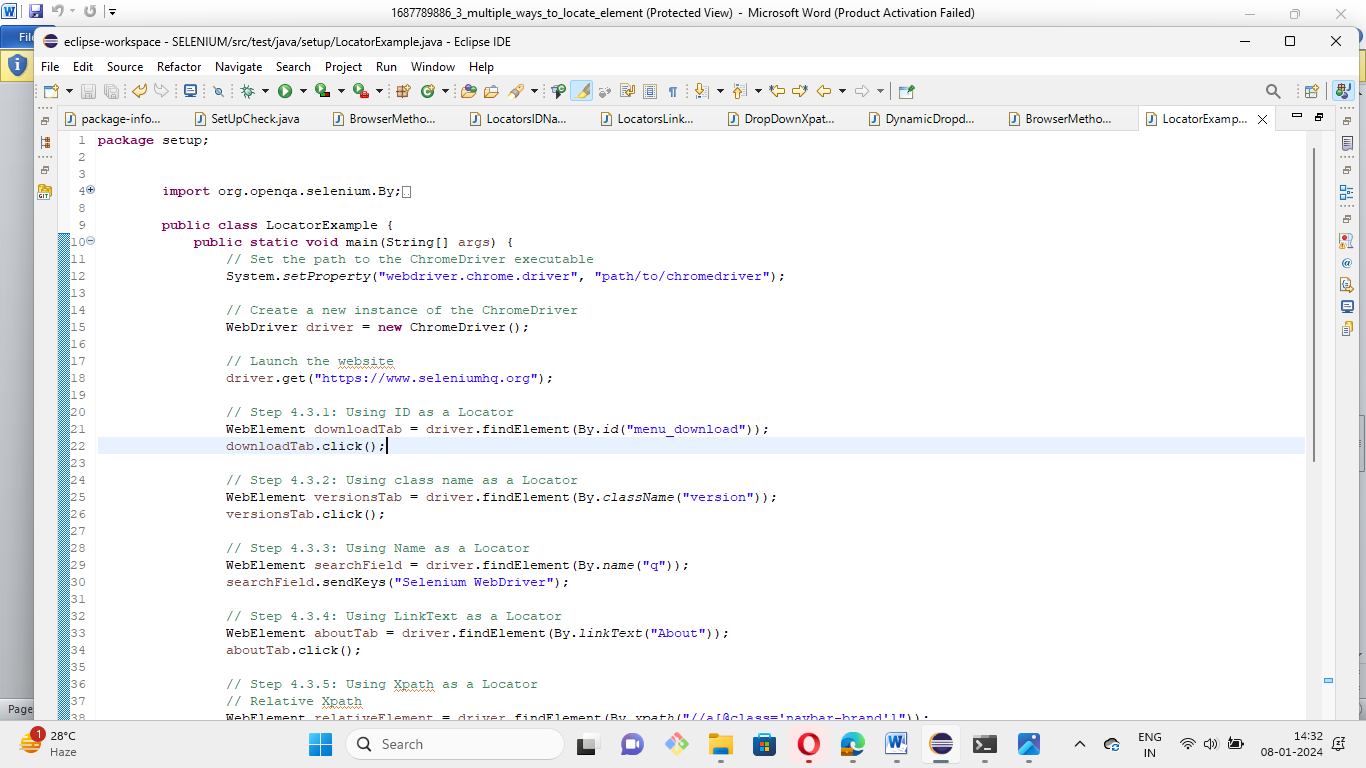
browsers. Concurrently, it is possible to execute multiple tests with Selenium, as it covers almost

all aspects of functional testing by implementing add-on tools that broaden the scope of testing.

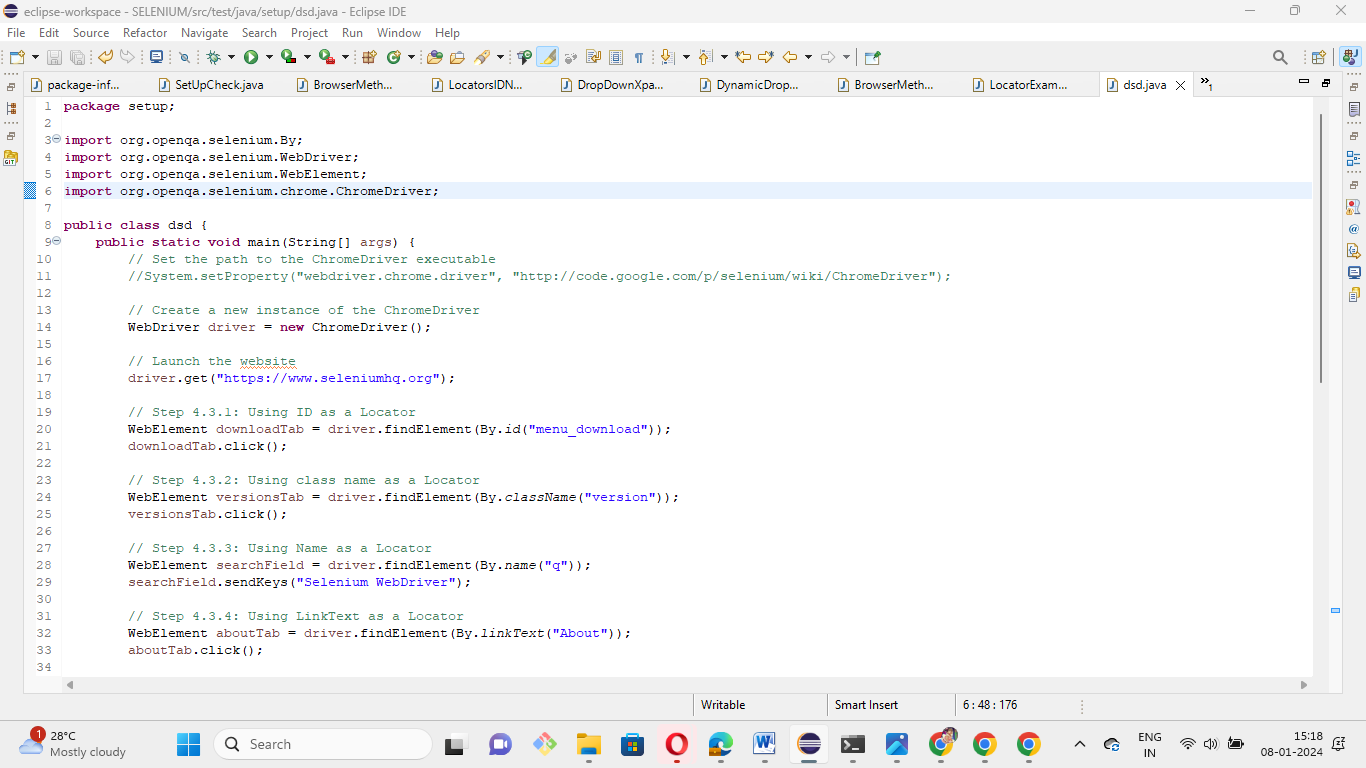
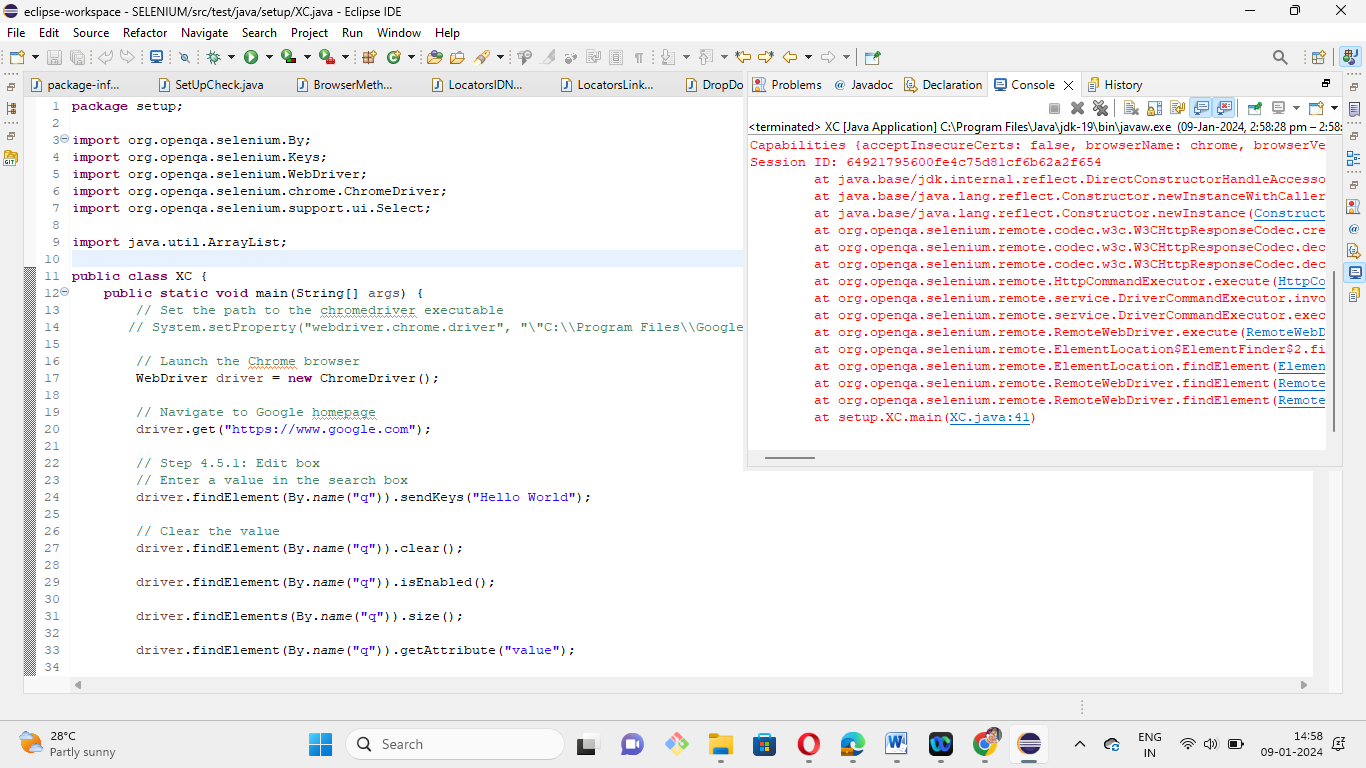
2. Demonstrate how Selenium web driver is installed and integrated in Eclipse.



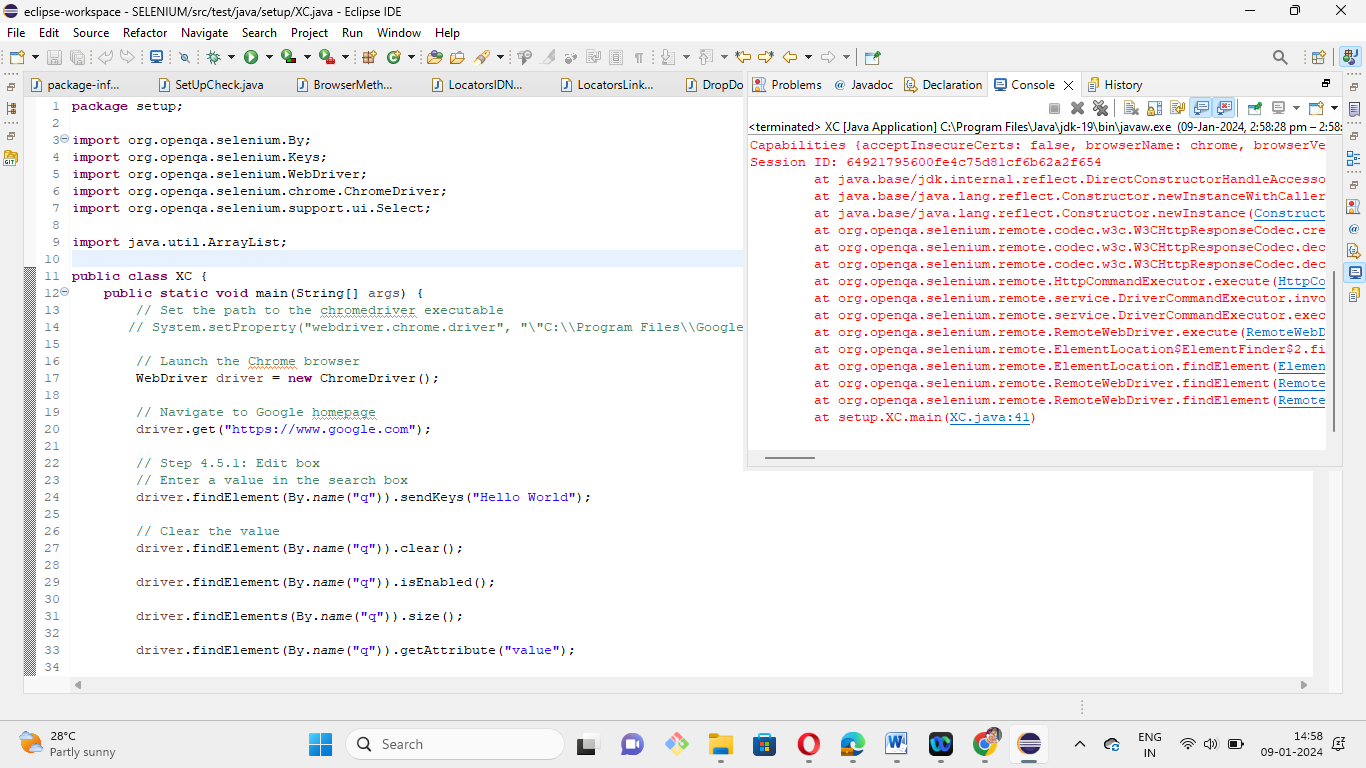
3. Demonstrate how elements are located using Selenium WebDriver.

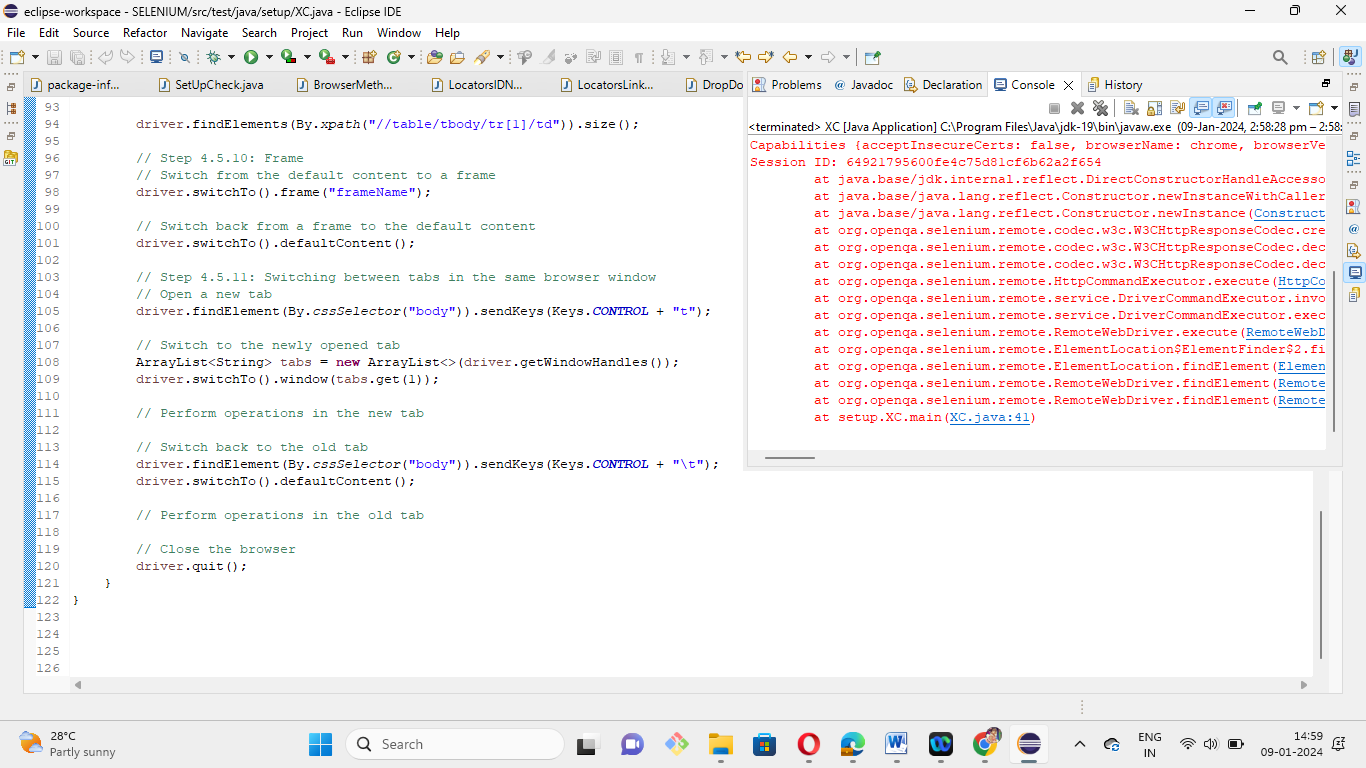
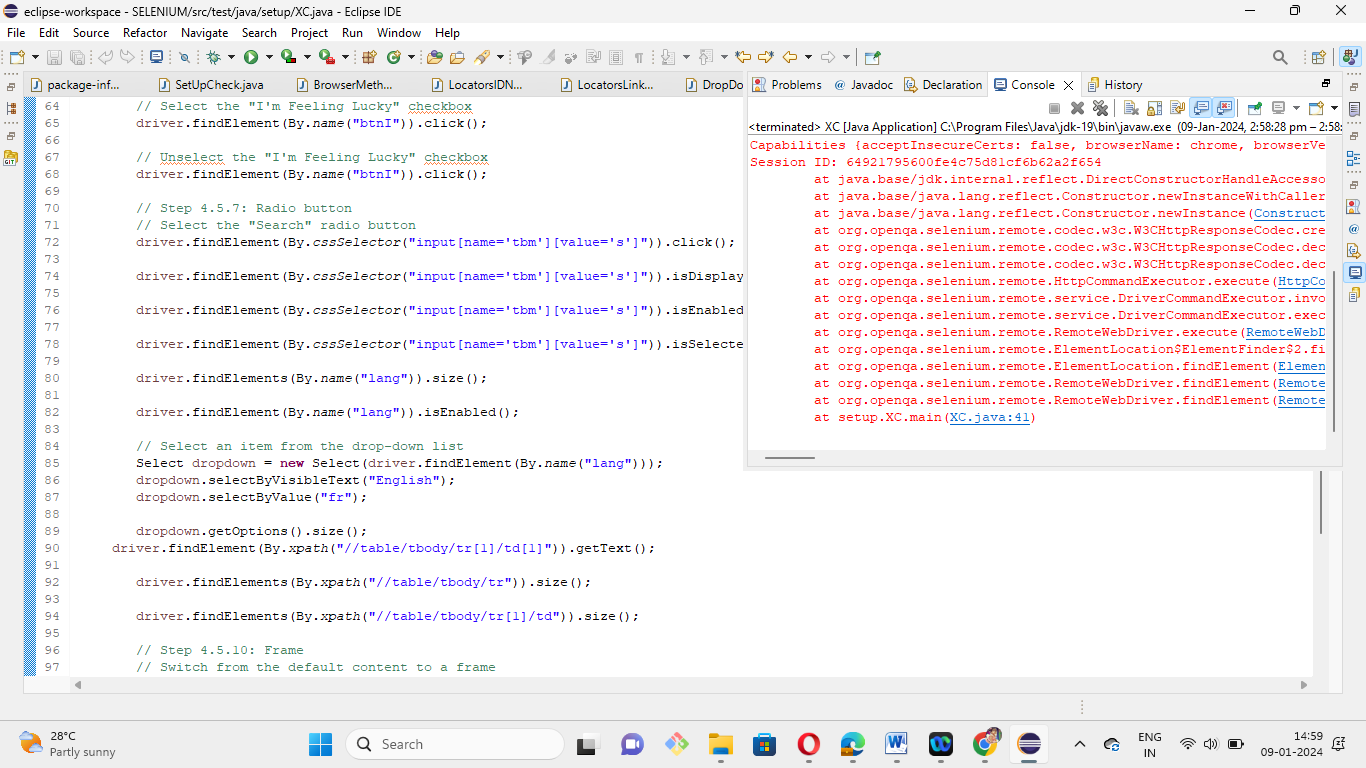
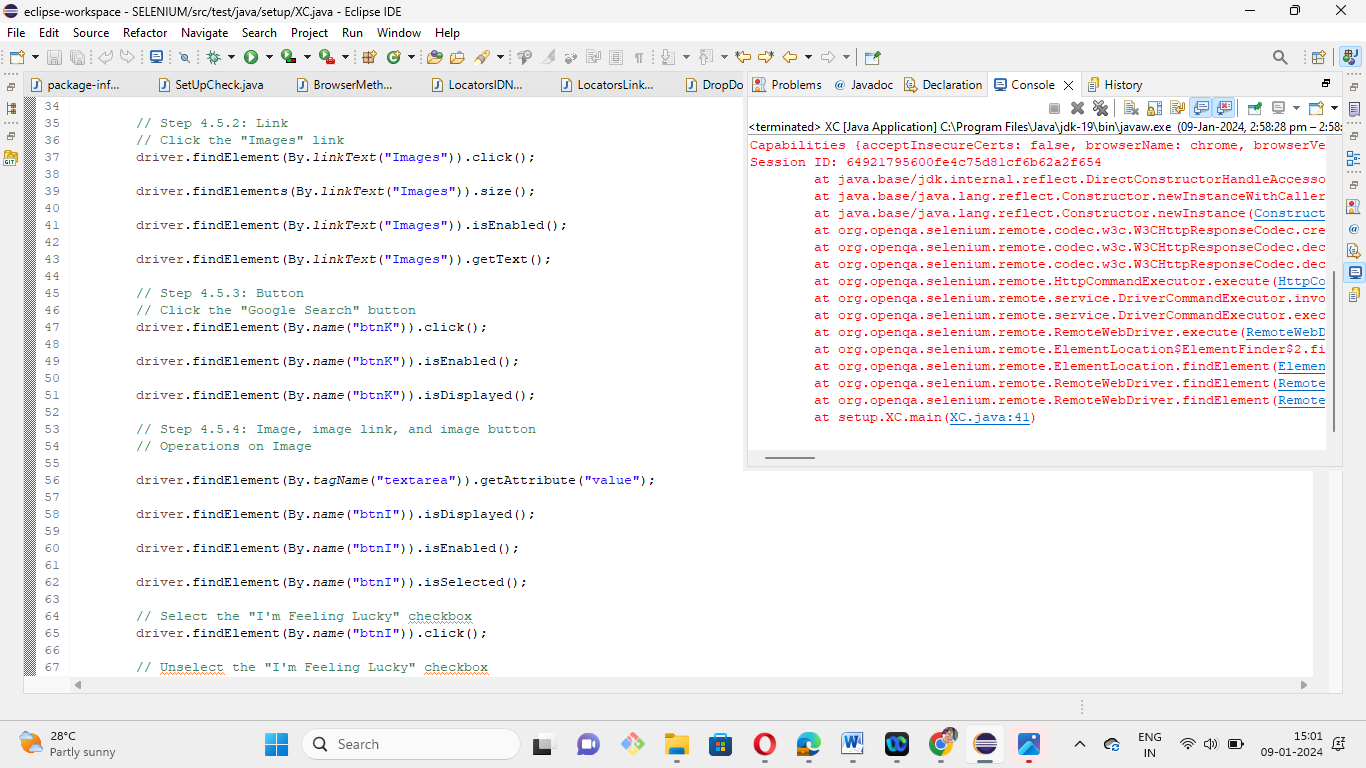
Demonstrate how elements are located using Selenium WebDriver.

4.Demonstrate how elements are located through CSS and XPath.

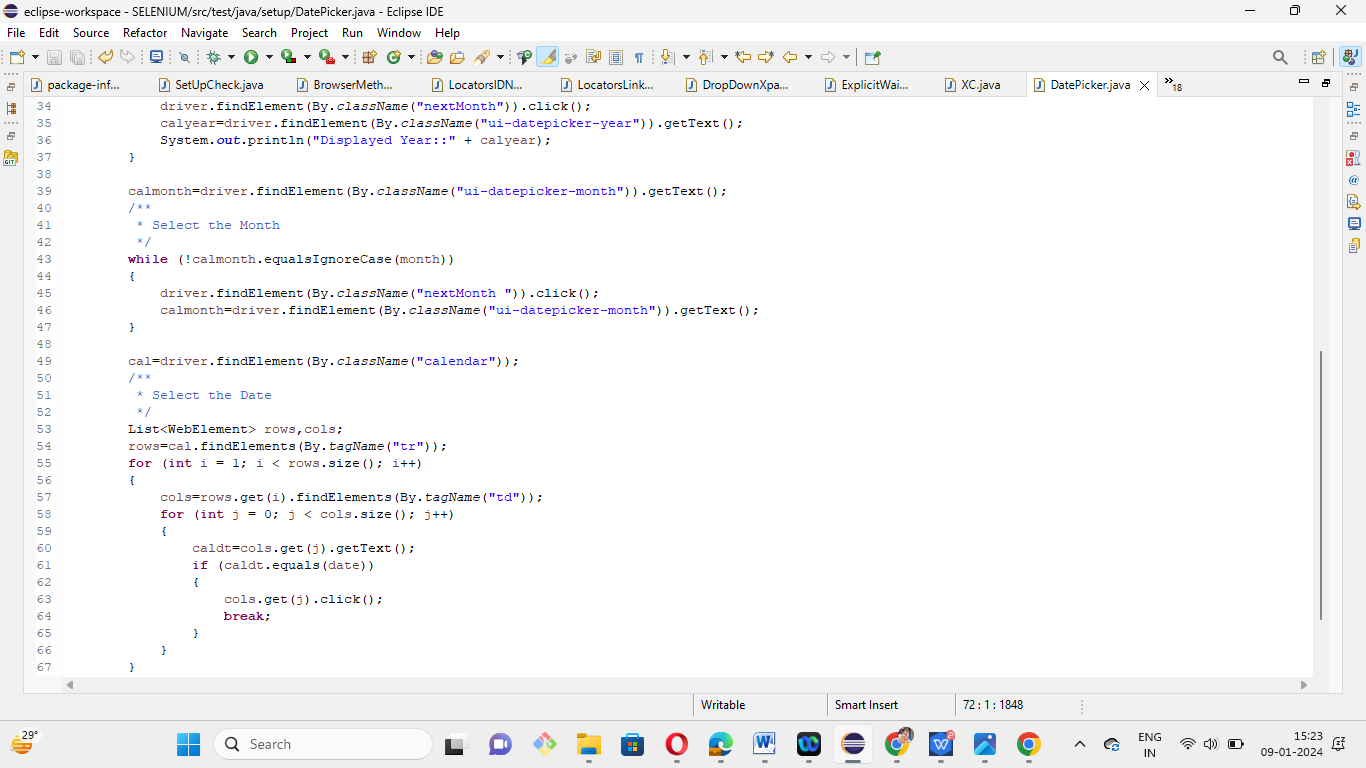
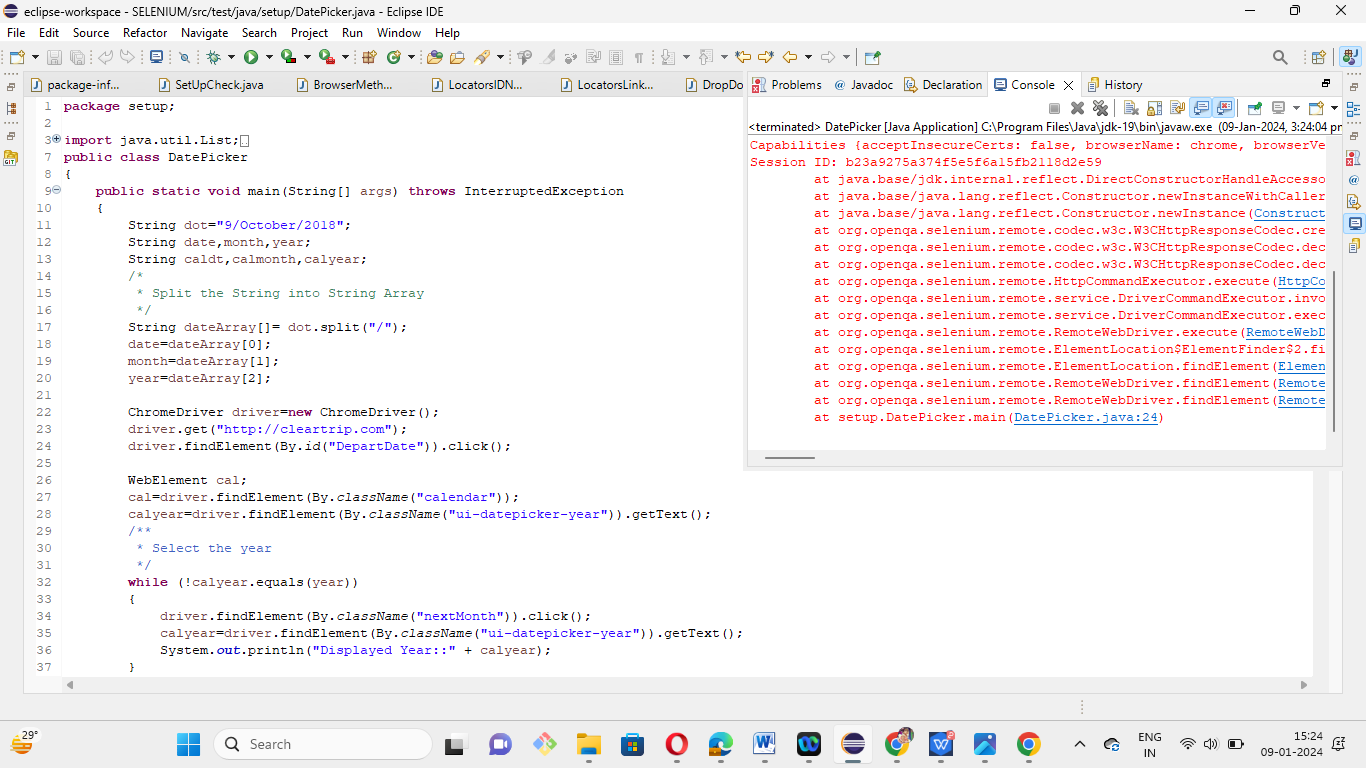


5.Demonstrate how web elements are handled in Selenium.

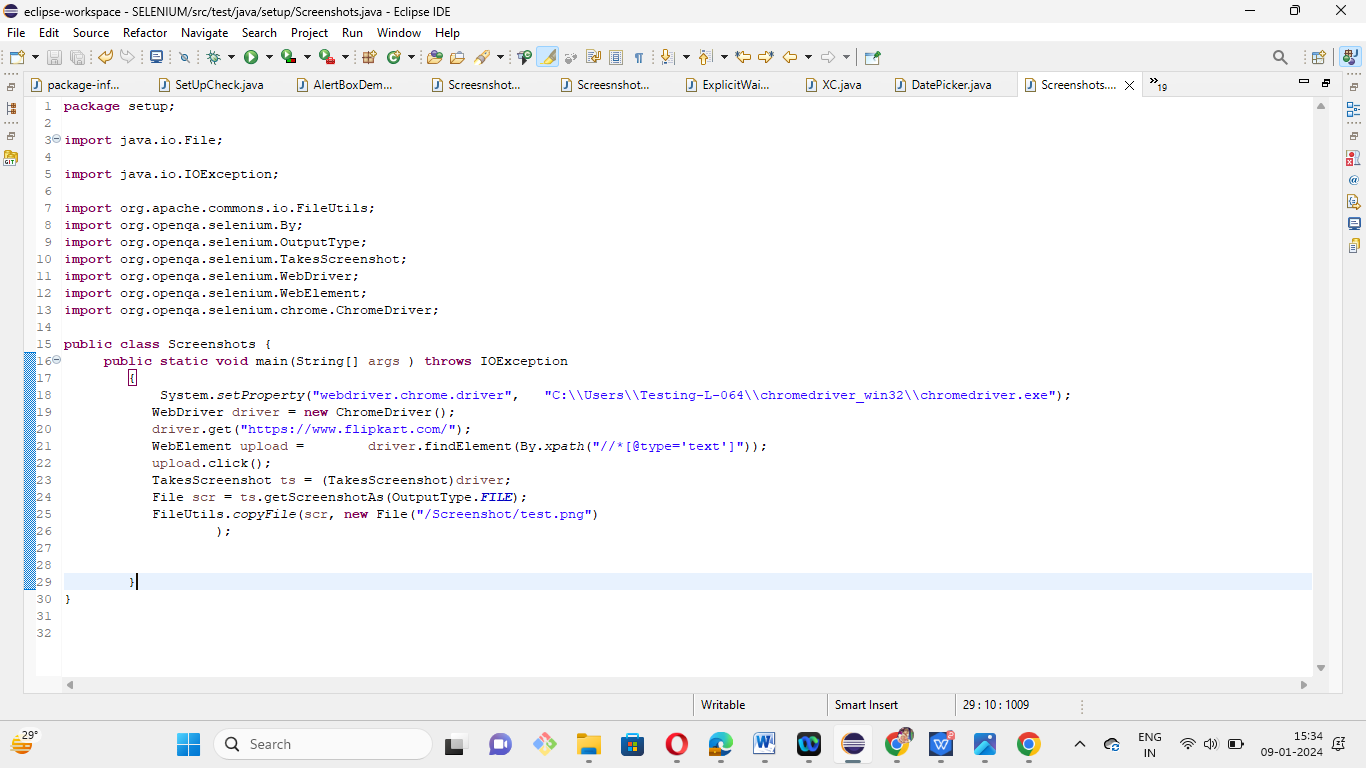




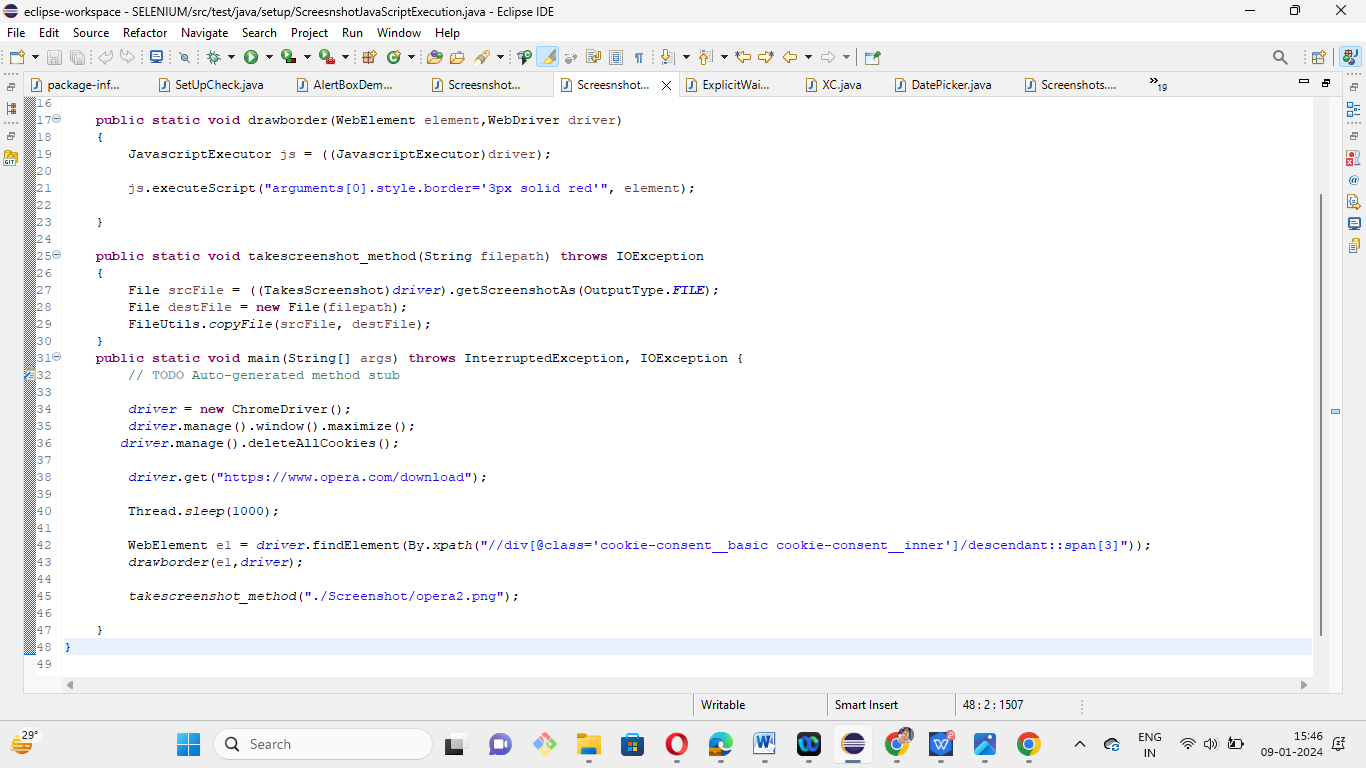
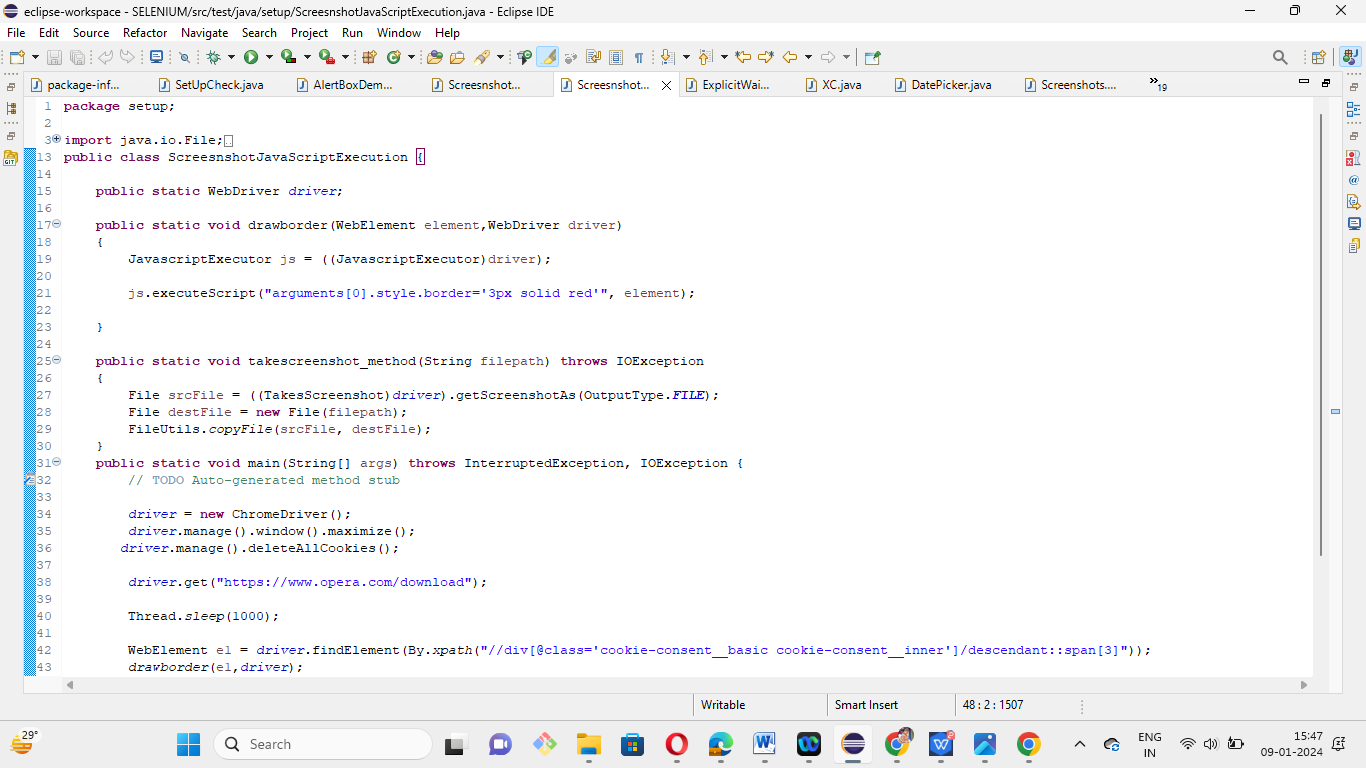
6.Demonstrate how to automate calendars on the web page.



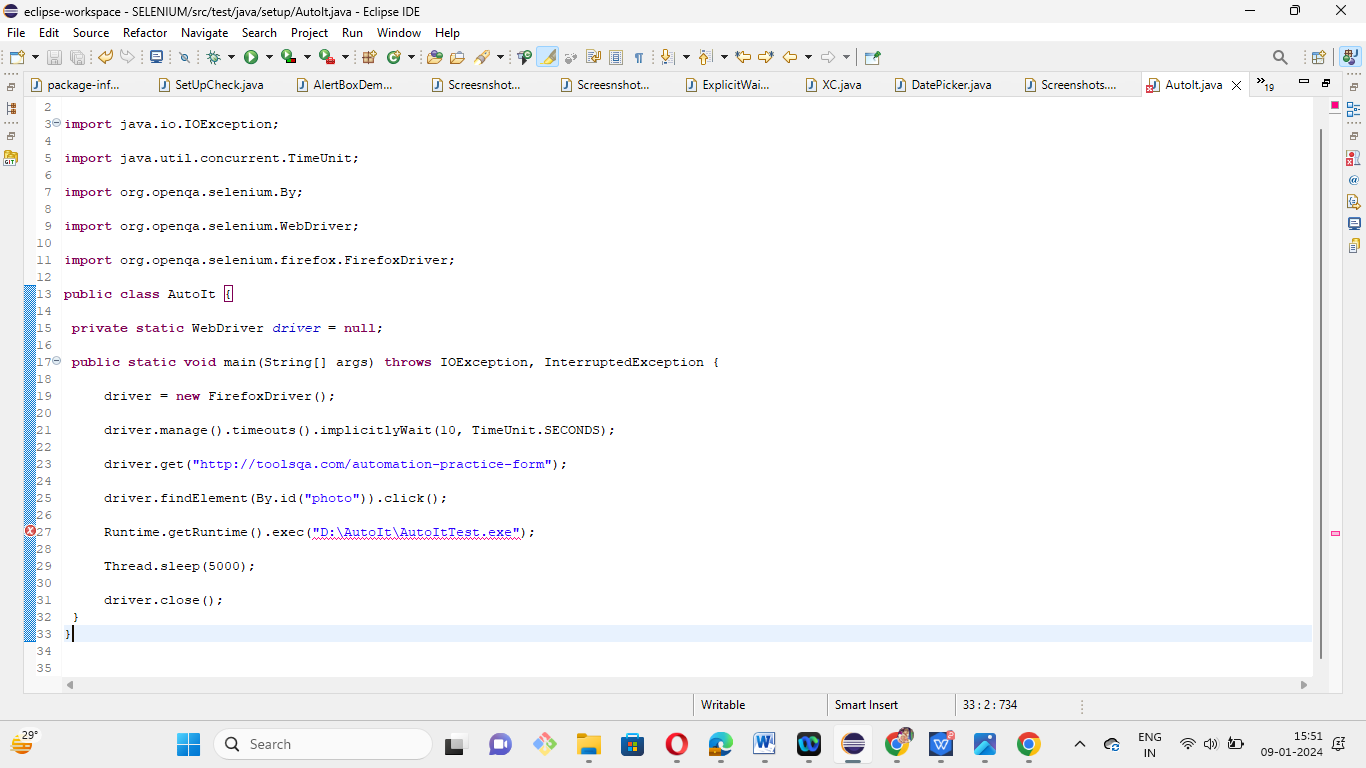
7. Using Selenium WebDriver, write a program to handle alerts.



8. Demonstrate how screenshots are captured and browser profiles are changed in Selenium.



9. Demonstrate installation and configuration of AutoIT.

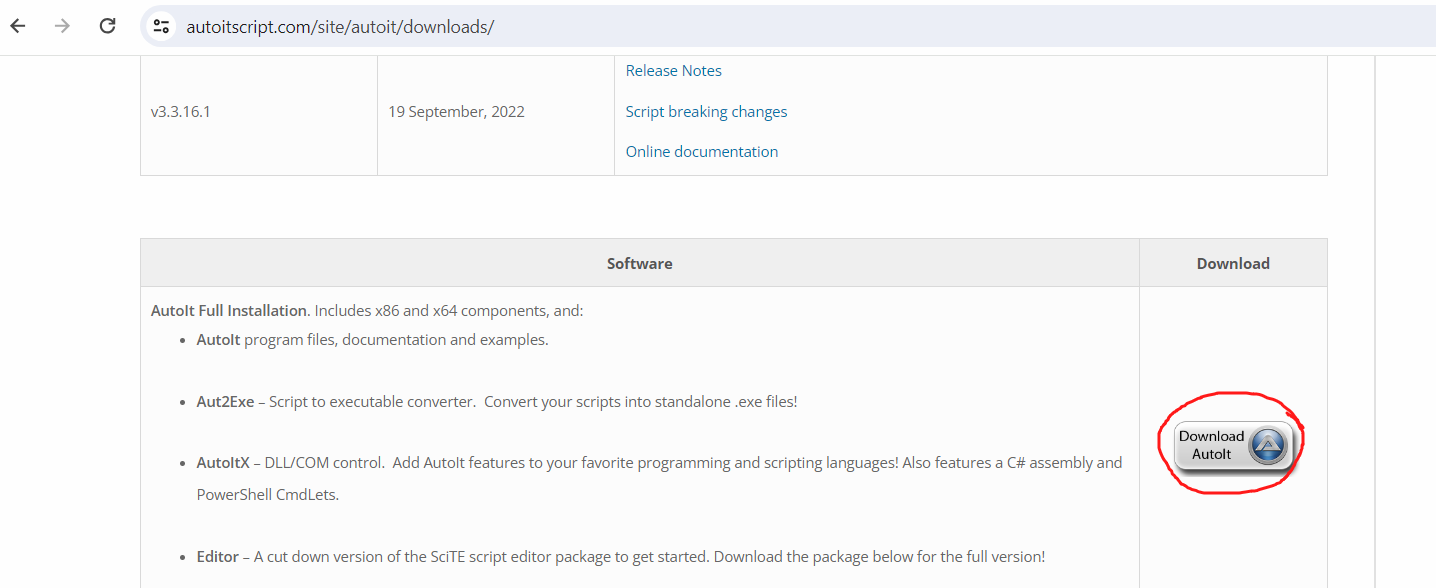


10. Demons trate how file uploads are handled in AutoIT.

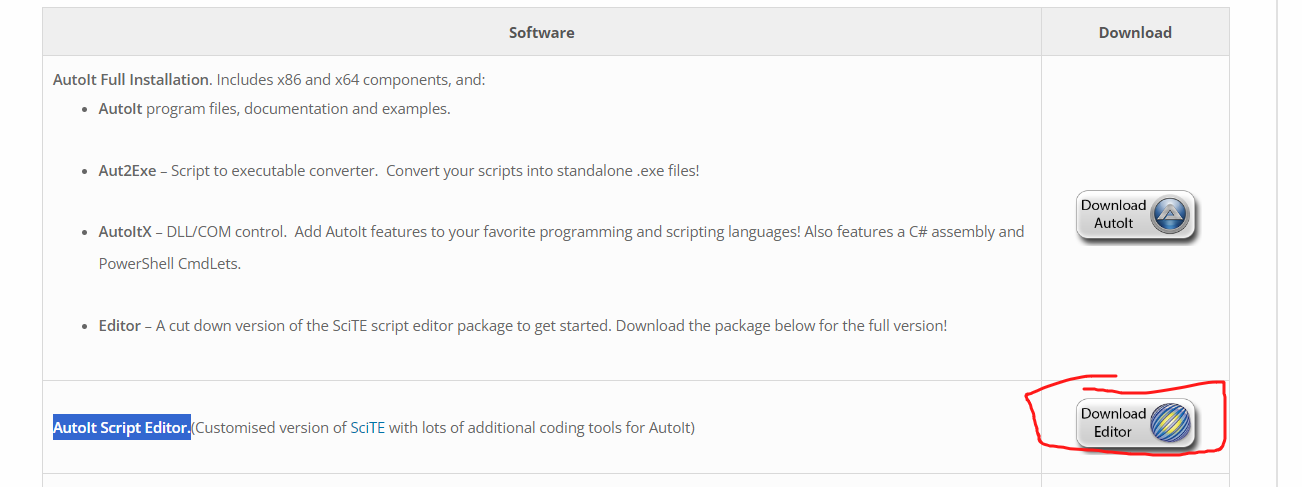
1. Download AutoIT

<https://www.autoitscript.com/site/autoit/downloads/>

Scroll down and click on button as show below



Also download AutoIt Script Editor.



Step 3: Decide the file to be uploaded

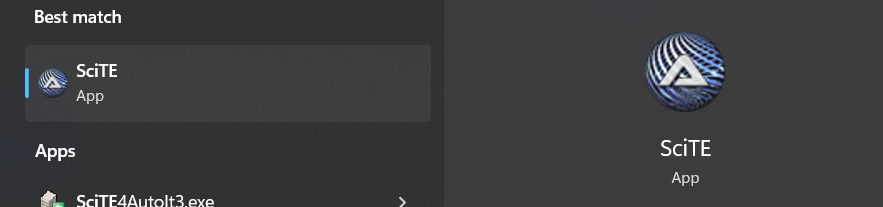
=============

For me it is :  C:\Users\sonal\Documents\ATE-Phase1-SL\JDBCconnectivity.png

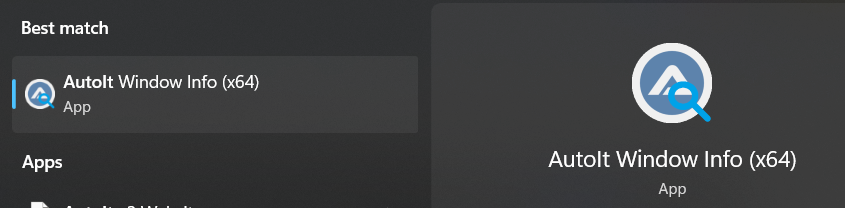
Step 4:

Now we will use AutoIT tool to write script that will first locate the file box and the open button

On the windows machine → open the autoIT editor -> search for app ->Scite



Now open another auto It tool to inspect the file box

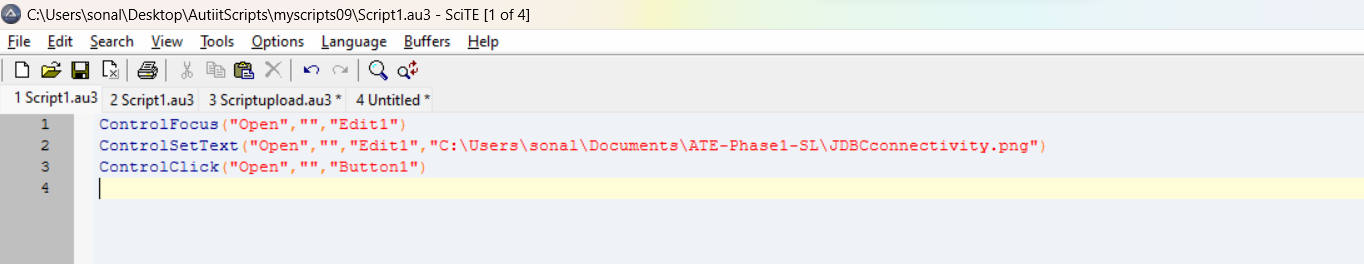


Write the script on the editor

ControlFocus("Open","","Edit1")

ControlSetText("Open","","C:\Users\sonal\Documents\ATE-Phase1-SL\JDBCconnectivity.png")

ControlClick("Open","","Button1")



Save the script