Practical No. 8 Exploring the Features of WebDriver

Date: _______________

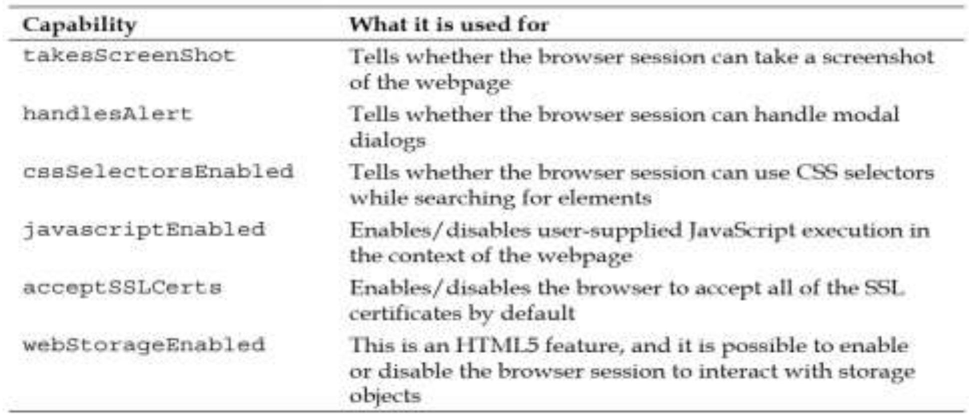
Aim:

To explore the Features of WebDriver like taking screenshot and waits in selenium.

Theory:

You, as a user of WebDriver, have the flexibility to create a session for a browser with your own set of desired capabilities that a browser should or shouldn't have. Using the capabilities feature in WebDriver, you are given a way to specify your choice of how your browser should behave.

Some of the examples of browser capabilities include enabling a browser session to support taking screenshots of the webpage, executing custom JavaScript on the webpage, enabling the browser session to interact with window alerts, and so on.



Capability	What it is used for
`takesScreenShot`	Tells whether the browser session can take a screenshot of the webpage
`handlesAlert`	Tells whether the browser session can handle modal dialogs
`cssSelectorsEnabled`	Tells whether the browser session can use CSS selectors while searching for elements
`javascriptEnabled`	Enables/disables user-supplied JavaScript execution in the context of the webpage
`acceptSSLCerts`	Enables/disables the browser to accept all of the SSL certificates by default
`webStorageEnabled`	This is an HTML5 feature, and it is possible to enable or disable the browser session to interact with storage objects

Taking Screenshot

Taking a screenshot of a webpage is a very useful capability of WebDriver. This is very handy when your test case fails, and you want to see the state of the application when the test case failed. The TakesScreenShot interface in the WebDriver library is implemented by all of the different variants of WebDriver, such as Firefox Driver, Internet Explorer Driver, Chrome Driver, and so on.

The TakesScreenShot capability is enabled in all of the browsers by default. Because this is a read-only capability, a user doesn't have much say on toggling it. Before we see a code example that uses this capability, we should look at an important method of the TakesScreenShot interface—getScreenshotAs().

The API syntax for getScreenshotAs() is as follows:

public <X> X getScreenshotAs(OutputType<X> target)

Here, **OutputType** is another interface of the WebDriver lib. We can ask WebDriver to give your screenshot in three different formats; they are: **BASE64, BYTES (raw data), and FILE**. If you choose the FILE format, it writes the data into a .png file, which will be deleted once the JVM is killed. So, you should always copy that file into a safe location so that it can be used for later reference.

The return type is a specific output that depends on the selected OutputType. For example, selecting **OutputType.BYTES will return a byte array**, and selecting **OutputType.FILE will return a file object.**

Waits

WebDriver provides the test script developers a very handy feature to manage wait time. Wait time is the time your driver will wait for the WebElement to load before it gives up and throws NoSuchElementException.

There are two ways by which you can make WebDriver wait for WebElement. They are **implicit wait time** and **Explicit wait time**.

Implicit wait time

Implicit wait time is used when you want to configure the WebDriver's wait time as a whole for the application under test. Implicit timeouts are common to all the WebElements.

Explicit wait time

Implicit timeout is generic to all the WebElements of a web page. But, if you have one specific WebElement in your application where you want to wait for a very long time, this approach may not work. Setting the implicit wait time to the value of this very long time period will delay your entire test suite execution. So you have to make an exception for only a particular case, like this WebElement. To handle such scenarios, WebDriver has explicit wait time for a WebElement.

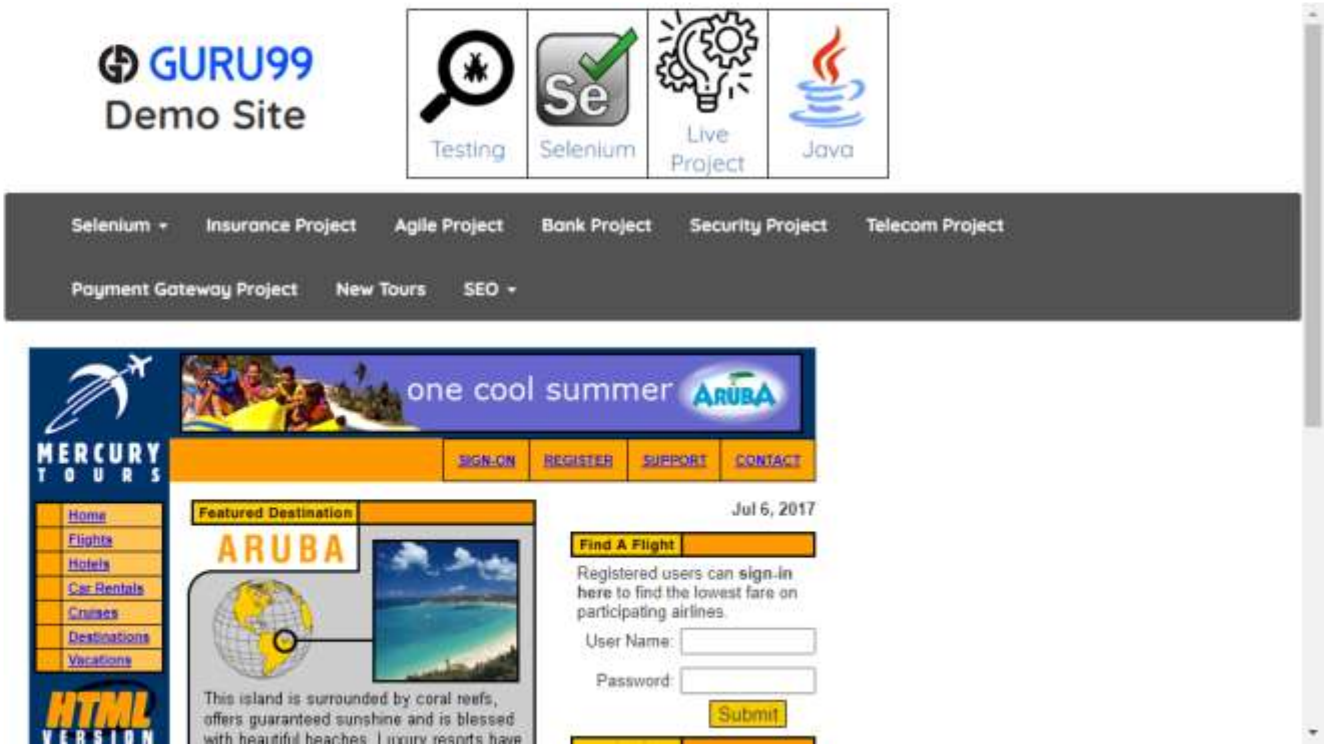
Implementation

1. Write a selenium script to take screenshot of
   http://demo.guru99.com/test/newtours/index.php

Program:

```
package One_Dec;
import java.io.File;
import java.nio.file.Files;
import java.io.IOException;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
public class Screentshot_Demo {
    public static void main(String[] args) throws IOException {
            // TODO Auto-generated method stub
            System.setProperty("Webdriver.chrome.driver", "I:\\SEM-
III\\STQA\\Selenium_Setup\\chromedriver.exe");
            ChromeOptions options = new ChromeOptions();
            options.addArguments("--remote-allow-origins=*");
            WebDriver driver = new ChromeDriver();
            driver.get("http://demo.guru99.com/test/newtours/index.php");
            TakesScreenshot scrShot = ((TakesScreenshot)driver);
            File scrFile = scrShot.getScreenshotAs(OutputType.FILE);
            System.out.println(scrFile.getAbsolutePath());
            File DestFile = new File("I:\\SEM-III\\STQA\\sample\\test.png");
            Files.copy(scrFile.toPath(), DestFile.toPath());}        }
```

Output:



Registered users can sign-in here to find the lowest fare on participating airlines.

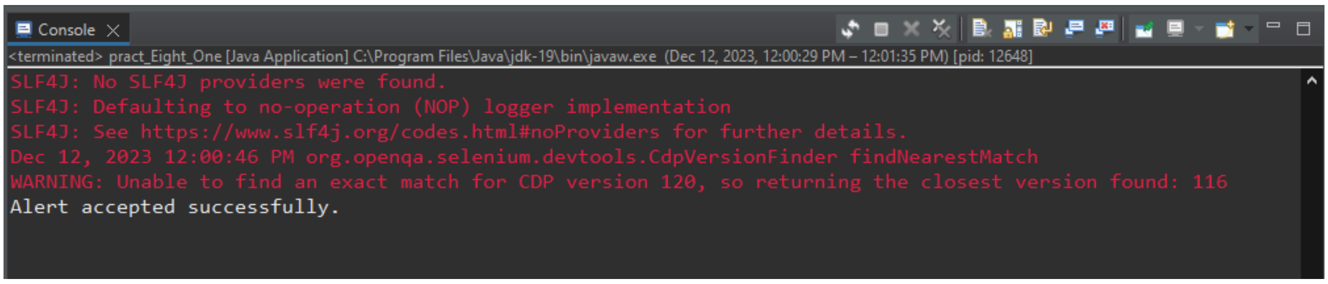
This island is surrounded by coral reefs, offers guaranteed sunshine and is blessed with beautiful beaches. Luxury resorts have

2. Write a selenium script to demonstrate explicit and implicit waits on https://demoqa.com/alerts

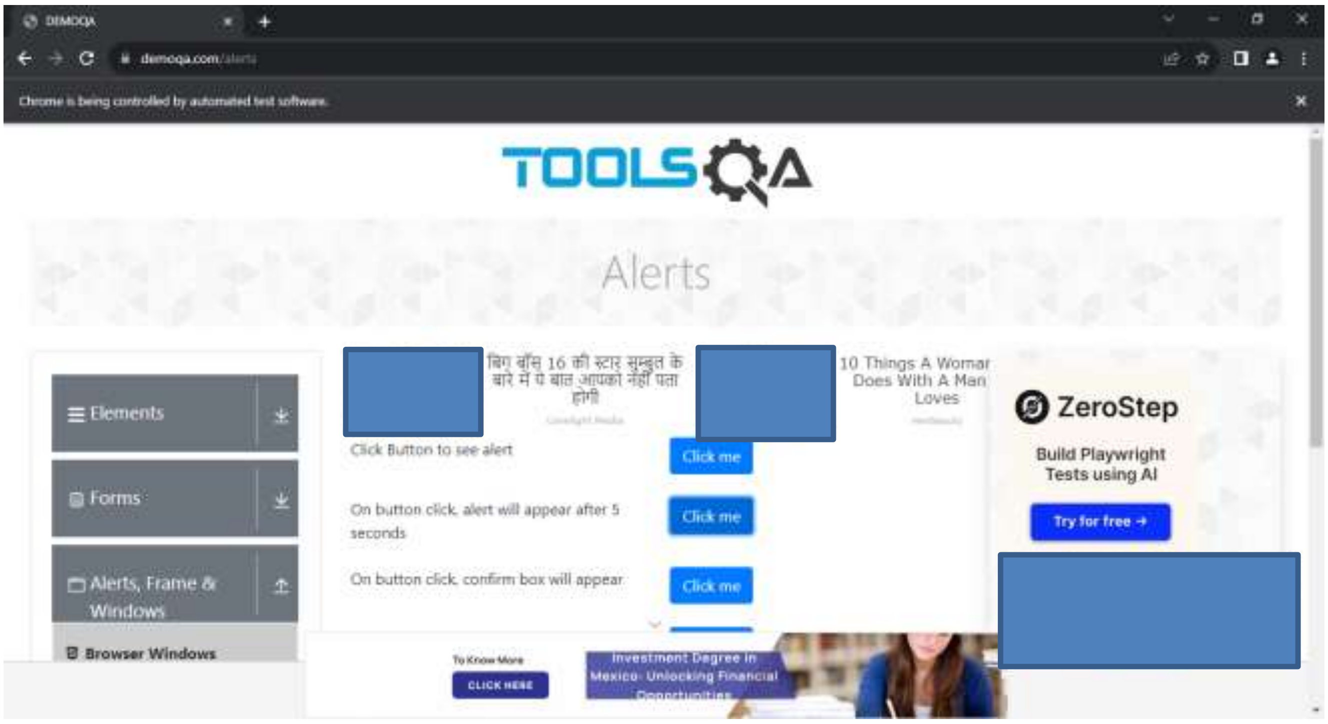
Program:

```
package practTen;

import java.time.Duration;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
public class pract_Eight_One {
        public static void main(String[] args) {
                // TODO Auto-generated method stub
                System.setProperty("Webdriver.chrome.driver", "I:\\SEM-
III\\STQA\\Selenium_Setup\\chromedriver.exe");
                WebDriver driver = new ChromeDriver();
                // Navigate to the demoqa.com/alerts page
                driver.get("https://demoqa.com/alerts");
                // Use implicit wait
                driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
                // Click the button to trigger an alert
                WebElement alertButton = driver.findElement(By.id("timerAlertButton"));
                alertButton.click();
                // Use explicit wait for the alert to be present
                WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(5));
                wait.until(ExpectedConditions.alertIsPresent());
                // Switch to the alert and accept it
                driver.switchTo().alert().accept();
                // Print a message indicating that the alert was accepted
                System.out.println("Alert accepted successfully.");        }       }
```

Output:



```
<terminated> pract_Eight_One [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 12, 2023, 12:00:29 PM – 12:01:35 PM) [pid: 12648]
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Dec 12, 2023 12:00:46 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 120, so returning the closest version found: 116
Alert accepted successfully.
```



[redacted] [redacted] [redacted]

बिग बॉस 16 की स्टार सुम्बुल के बारे में ये बात आपको नहीं पता होगी

10 Things A Woman Does With A Man Loves

Click Button to see alert

On button click, alert will appear after 5 seconds

On button click, confirm box will appear

Conclusion: Learnt to take screenshot of webpage and manage waits in selenium.