10. Evaluating Test Cases

1. Creating a simple Java project

- Open Eclipse
- Go the **File** menu. Choose **New->Java Project**
- Enter the project name as **Test Assertions** . Click on **Next**
- This will create the project files in the Project Explorer

2. Downloading Selenium WebDriver jar, chromdriver.exe, and forefoxdriver.exe

- Go to https://www.seleniumhq.org/download/ to download the **Selenium WebDriver** dependency
- Under the section **Selenium Client & WebDriver Language Bindings,** click on **Download** for **Java client version: 3.141.59**
- On the same page, under **Third Party Drivers, Bindings, and Plugins,** click on **Latest** for **Mozilla Gecko Driver**
- Select the file suitable for your operating system
- Go back to the previous page. Click on **Latest** for **Google Chrome Driver**
- From the current releases, select the appropriate file per your Chrome version

3. Adding the WebDriver dependency in the project

- In the Project Explorer, right click on **Test Assertions**
- Select **Properties** . Select **Java Build Pat** h from the list. Go to **Libraries**
- Click on **Add External JARs** and browse the location where you have downloaded the JAR files
- Select JARs from the **root** folder and the **libs** folder
- Click on **Apply and Close**
- Copy the chromedriver.exe and geckodriver.exe, and paste it your project creating a resource folder

4. Installing TestNG

- Install TestNG in Eclipse.

5. Adding TestNG libraries to the Class Path

- In the Project Explorer, right click on **Test Assertions**
- Select **Properties** . Select **Java Build Pat** h from the list. Go to **Libraries**
- Click on **Add Library.** Select **TestNG** . Click on **Next** . Click on **Finish**
- Click on **Apply and Close**

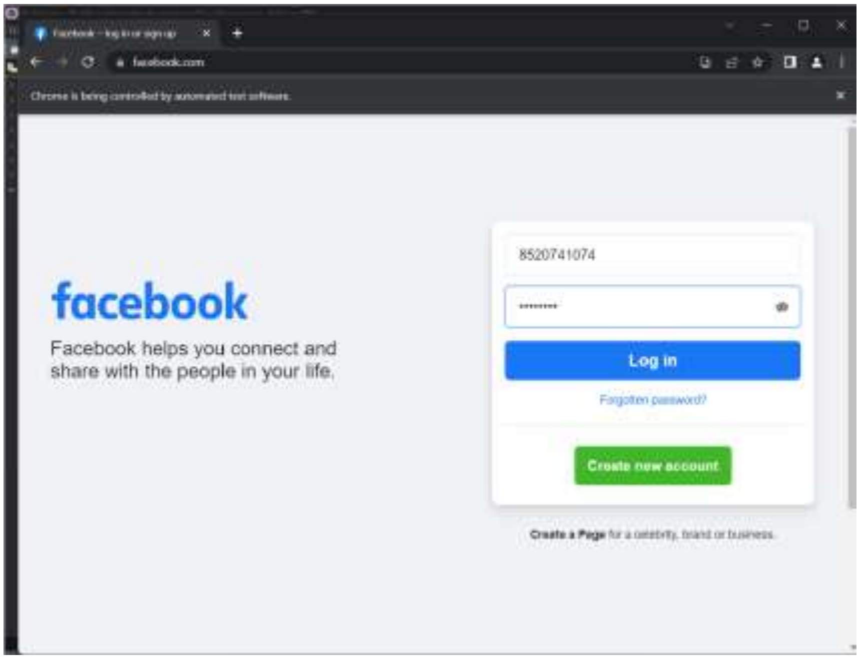
6. Creating a Java class named ParallelTest.java

- In the Project Explorer, expand **Test Assertions->Java Resources**
- Right click on **src** and choose **New->Class**
- In **Class Name,** enter **Assertions** and click on **Finish.** In **Package Name,** enter **com.assert** and click on **Finish** ● Enter the following code: **package** com;

```
 import org.openqa.selenium.By; import
org.openqa.selenium.WebDriver; import
org.openqa.selenium.chrome.ChromeDriver;
 import org.testng.Assert; import
org.testng.annotations.Test; import
org.testng.asserts.SoftAssert;
public class Assertions {

   SoftAssert soft = new SoftAssert();
   WebDriver driver ;
   @Test
   public void Launch() {
      System. setProperty ( "webdriver.chrome.driver" ,
"C:\\Users\\hp\\Downloads\\chromedriver_win32\\chromedriver.exe"
     ); driver = new ChromeDriver(); try {
         Thread. sleep (3000);
      } catch (Exception e )
        { e
        .printStackTrace();
      }
   }

   @Test (dependsOnMethods = { "Launch"
  }) public void Facebook() { driver .get(
  "https://www.facebook.com" );
      soft .assertEquals( "FB Title" , driver
     .getTitle()); try {
            Thread. sleep (2000);
        }  catch  (Exception  e )
           {  e
           .printStackTrace();
        }
    }

    @Test (dependsOnMethods = {  "Facebook"  })
    public  void  Login() {  driver .findElement(By. id ( "email"
       )).sendKeys( "84548451515" );  driver .findElement(By. id (
       "pass" )).sendKeys( "Prudhvi@123" );
        driver .findElement(By. id ( "loginbutton" )).click();
       soft .assertAll();

        try  { Thread. sleep (3000) ;
        }  catch  (Exception  e )
           {  e
           .printStackTrace();
        }

    }
}
```

7. Running the project

- Right click on **Assertions** class. Click on **TestNG->Convert to TestNG**
- Click on **Finish.** It will create a **TestNG.xml** file. Open that file
- Right click. Select **Run As ->TestNG Suite**



Chrome is being controlled by automated test software.

Facebook helps you connect and share with the people in your life.

8520741074

••••••••

Create a Page for a celebrity, brand or business.

Pushing the code to your GitHub repositories : -

- Open your folder where the Project . And then click the right button to open the git bash command prompt.
- Before that, open the github and create a new repository.
- Initialize your repository using the following command: git init
- Add all the files to your git repository using the following command: git add .
- To check the status of the repository use the below command: git status
- Commit the changes using the following command:

git commit . -m “Changes have been committed.”

- To add the files to the repository use the (URL) from the github and use the command;

git remote add origin <url>

- Push the files to the folder you initially created using the following command:

git push origin master.