Installing and Creating Your First Application in ECLIPSE:

Selenium installation is a 3 step process:
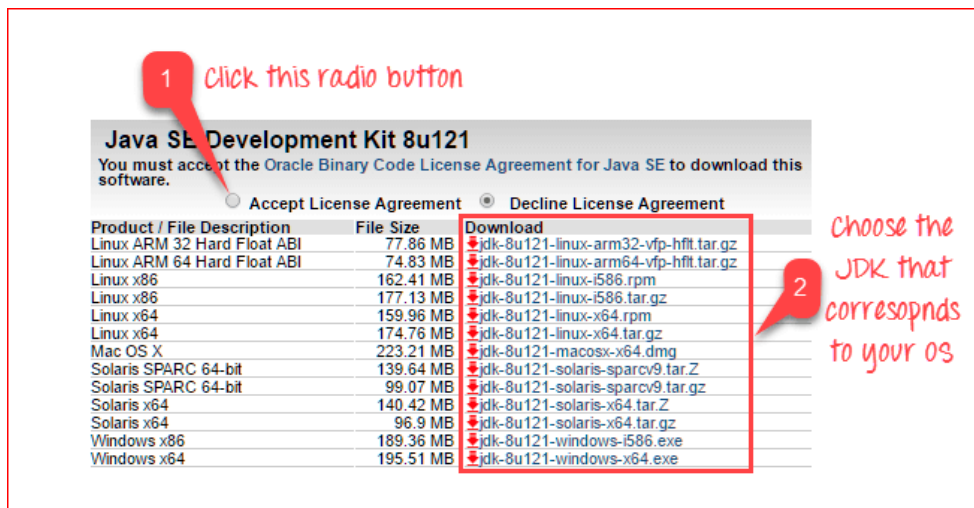
1. Install Java SDK
2. Install Eclipse
3. Install Selenium Webdriver Files

Step 1 – Install Java on your computer

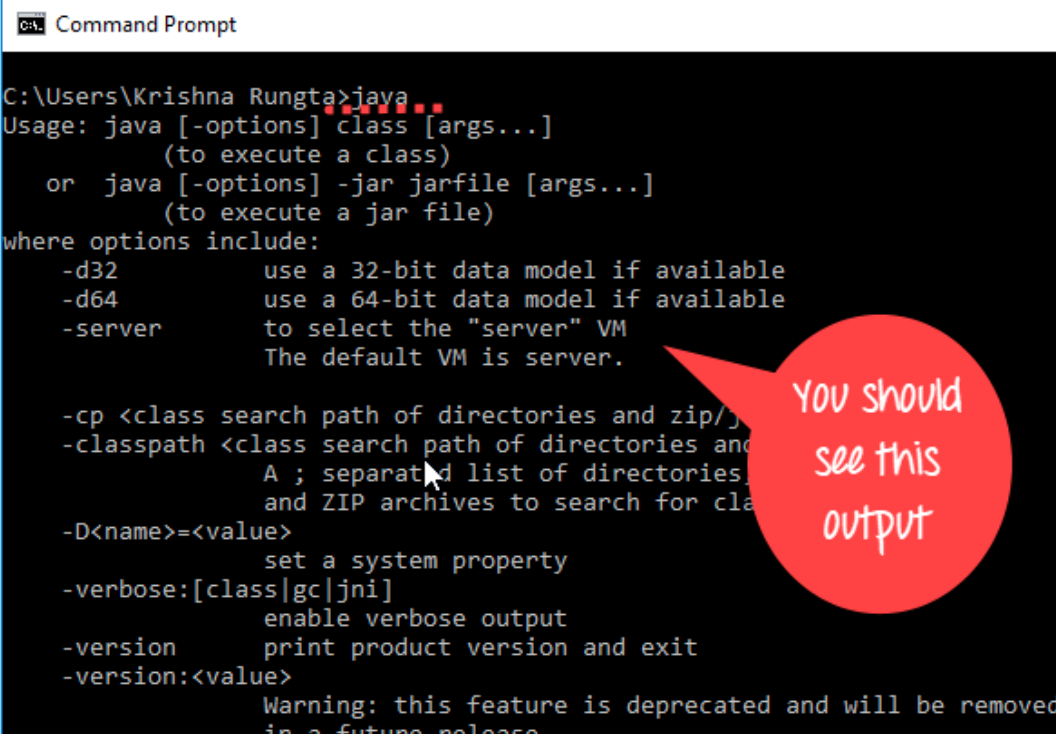Download and install the **Java Software Development Kit (JDK)**

Next –

This JDK version comes bundled with Java Runtime Environment (JRE), so you do not need to download and install the JRE separately.

Once installation is complete, open command prompt and type "java". If you see the following screen you are good to move to the next step
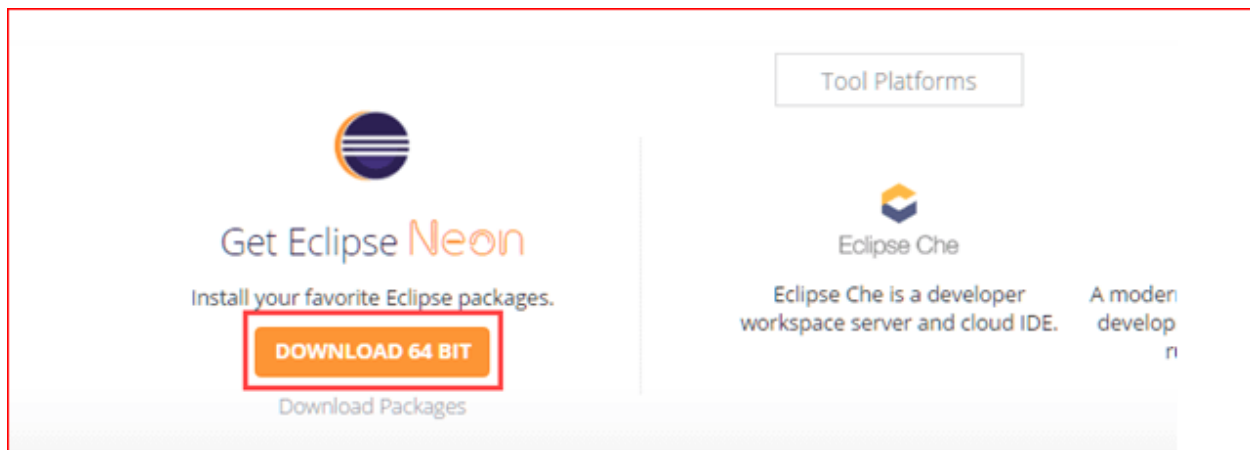

Step 2 – Install Eclipse IDE

Download latest version of **"Eclipse IDE for Java Developers"** here. Be sure to choose correctly between Windows 32 Bit and 64 Bit versions.

You should be able to download an exe file named "eclipse-inst-win64" for Setup.
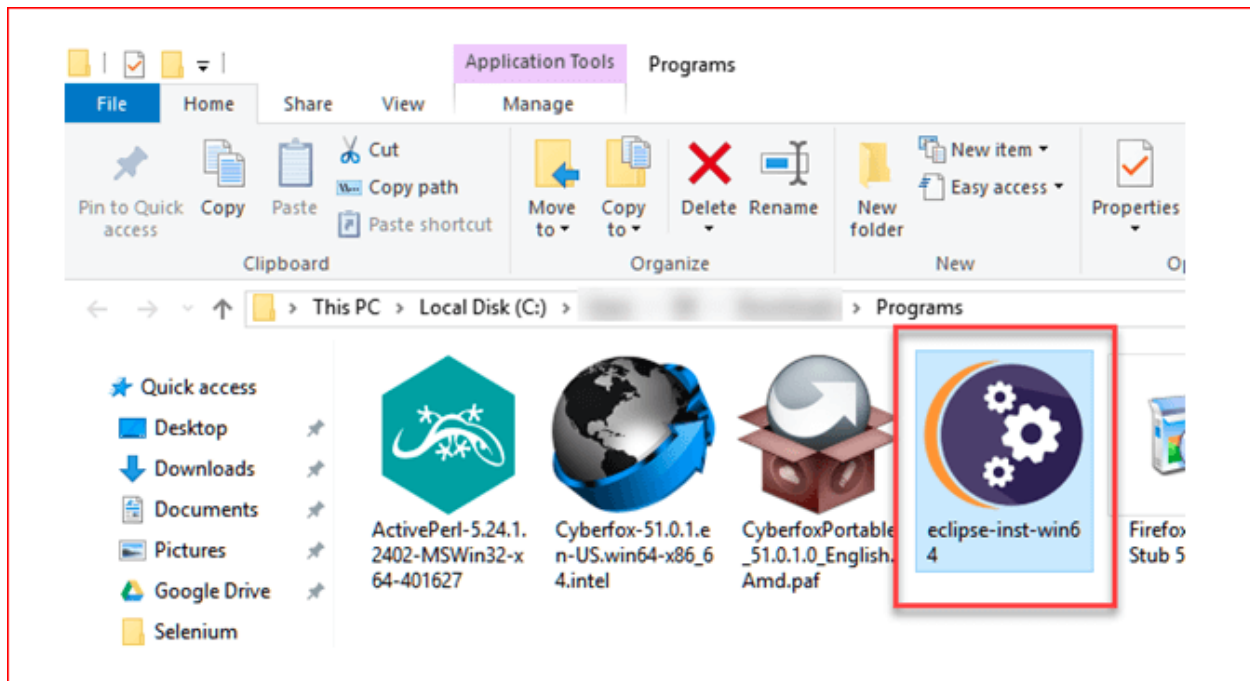
Double-click on file to Install the Eclipse. A new window will open. Click Eclipse IDE for Java Developers.
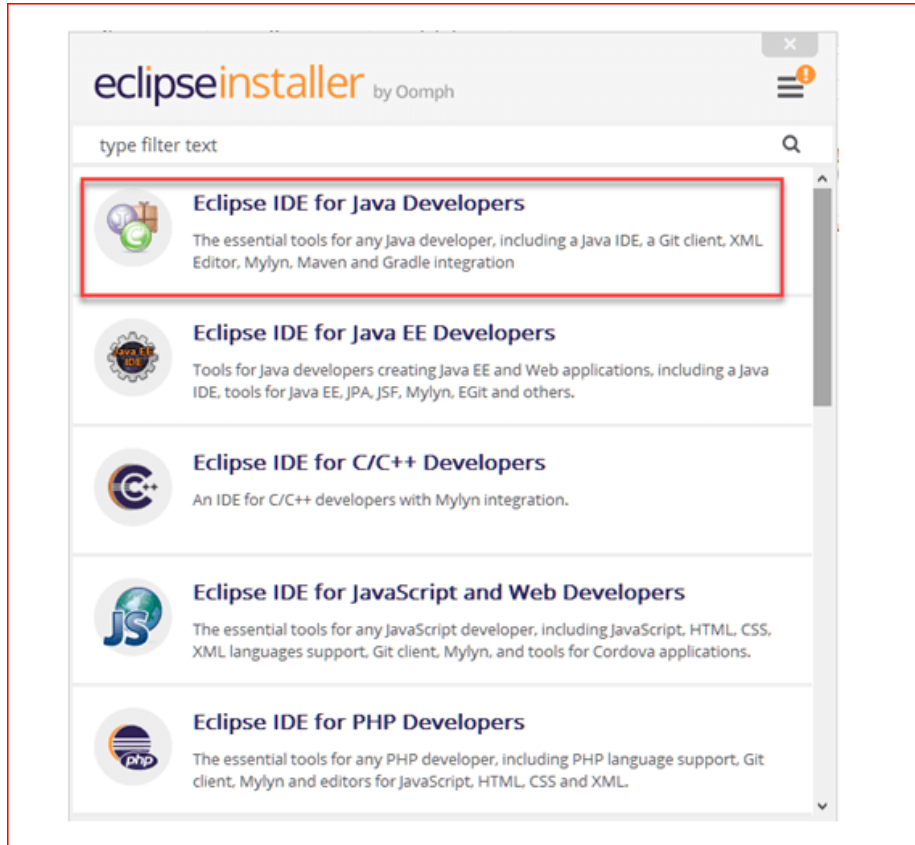
After that, a new window will open which click button marked 1 and change path to "C:\eclipse". Post that Click on Install button marked 2
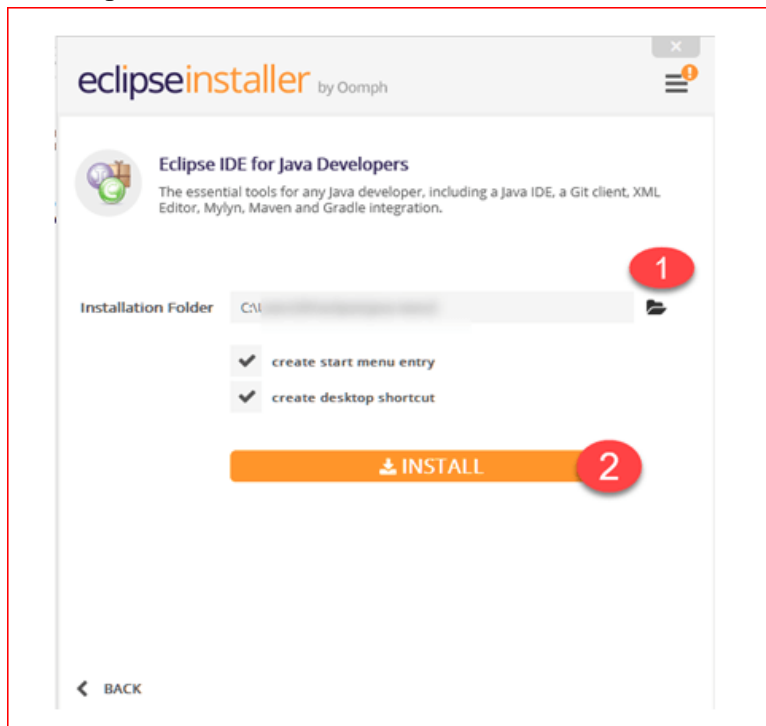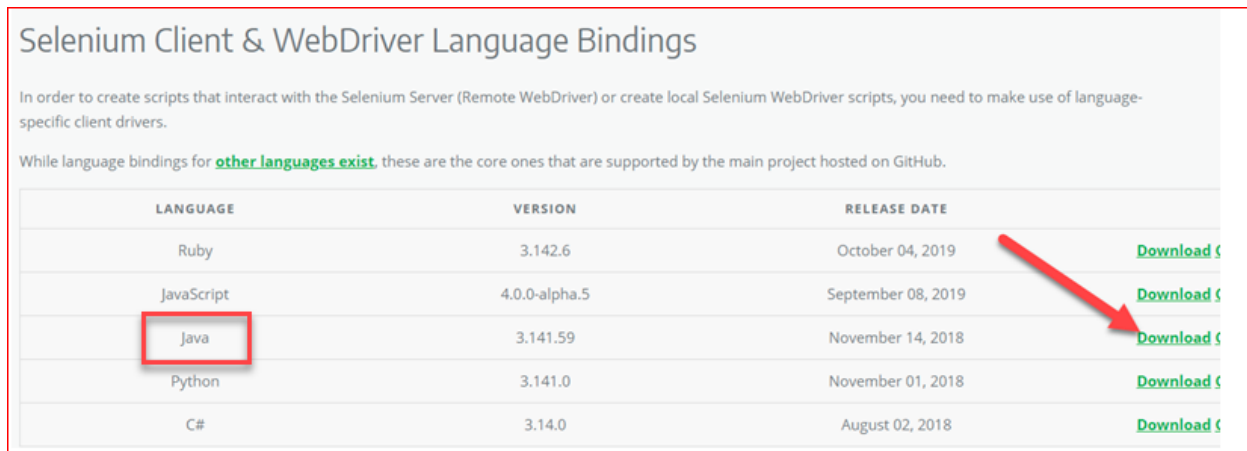
After successful completion of the installation procedure, a window will appear. On that window click on Launch
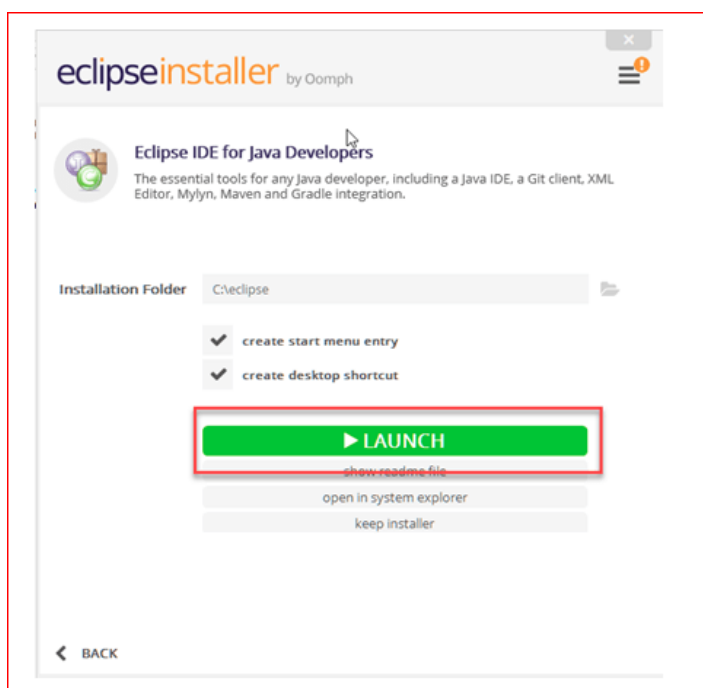
This will start eclipse neon IDE for you.

Step 3 – Download the Selenium Java Client Driver

You can download **Selenium Webdriver for Java Client Driver** here. You will find client drivers for other languages there, but only choose the one for Java.
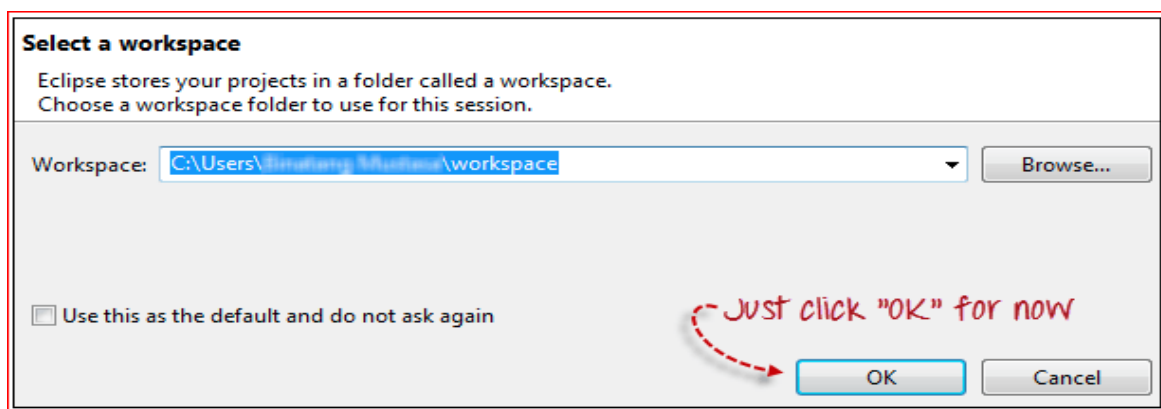
This download comes as a ZIP file named "selenium-3.14.0.zip". For simplicity of Selenium installation on Windows 10, extract the contents of this ZIP file on your C drive so that you would have the directory "C:\selenium-3.14.0\". This directory contains all the JAR files that we would later import on Eclipse for Selenium setup.

Step 4 – Configure Eclipse IDE with WebDriver

1. Launch the "eclipse.exe" file inside the "eclipse" folder that we extracted in step 2. If you followed step 2 correctly, the executable should be located on C:\eclipse\eclipse.exe.
2. When asked to select for a workspace, just accept the default location.

3. Create a new project through File > New > Java Project. Name the project as "newproject".
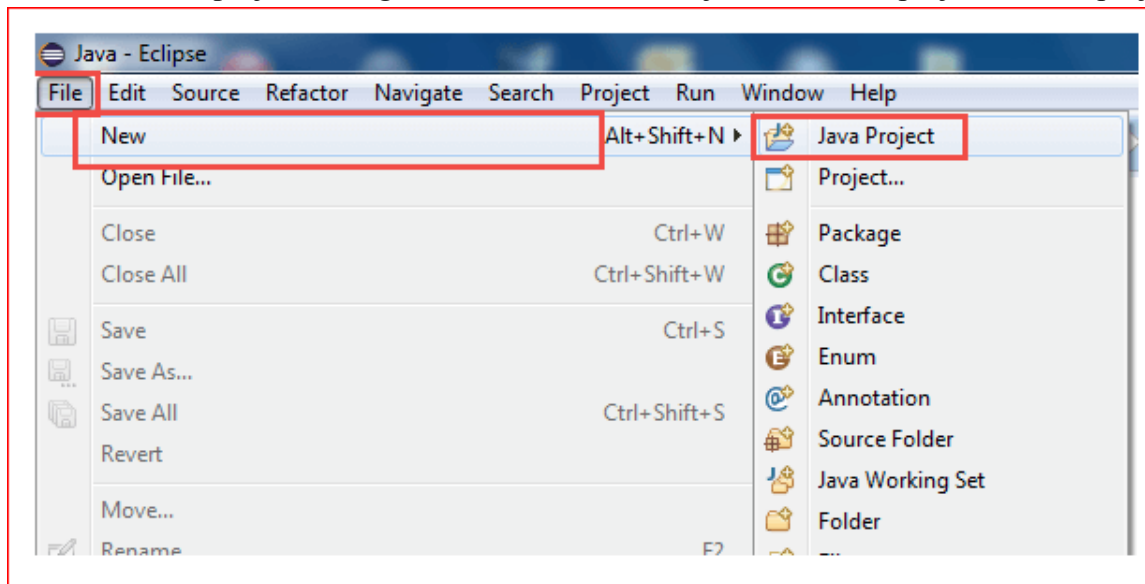
A new pop-up window will open enter details as follow

1. Project Name
2. Location to save project
3. Select an execution JRE
4. Select layout project option
5. Click on Finish button
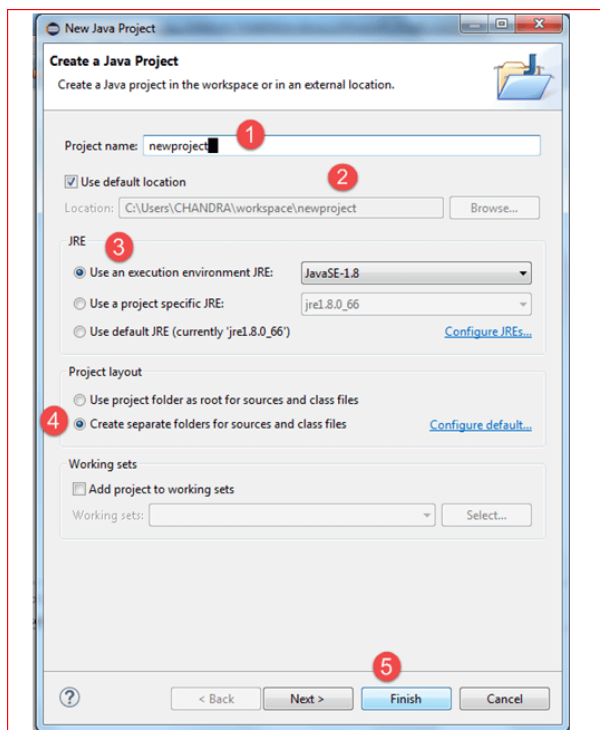
4. In this step,

1. Right-click on the newly created project and
2. Select New > Package, and name that package as "newpackage".
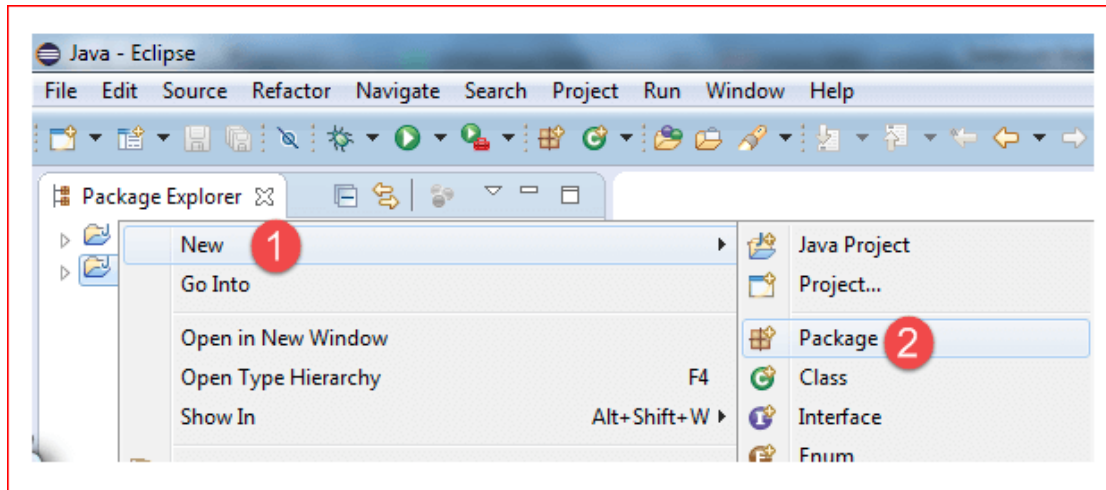
A pop-up window will open to name the package,

1. Enter the name of the package
2. Click on Finish button
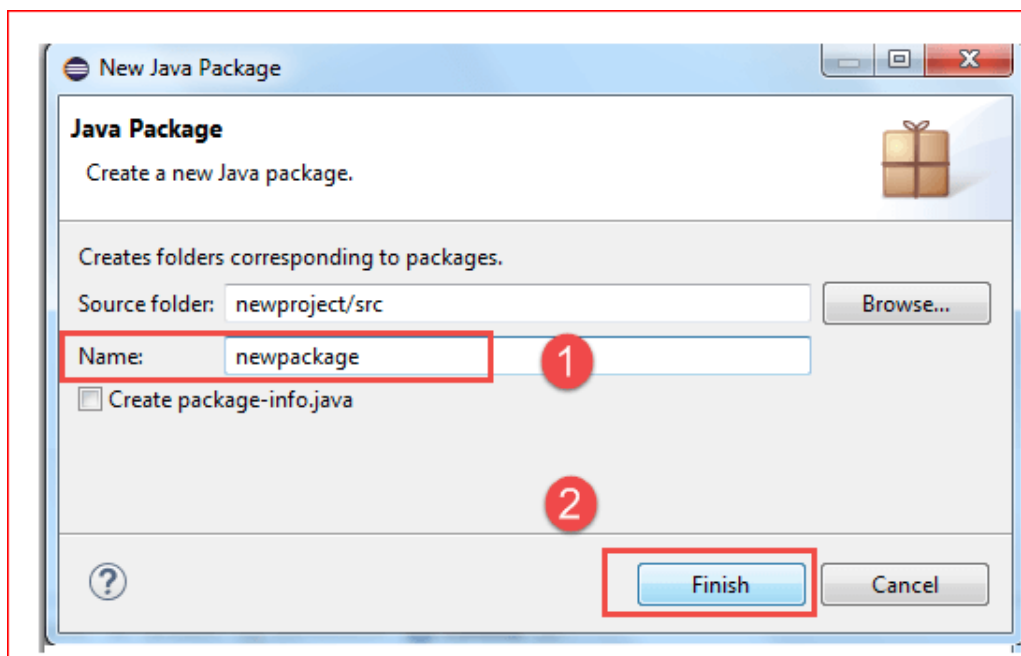
5. Create a new Java class under newpackage by right-clicking on it and then selecting- New > Class, and then name it as "MyClass". Your Eclipse IDE should look like the image below.
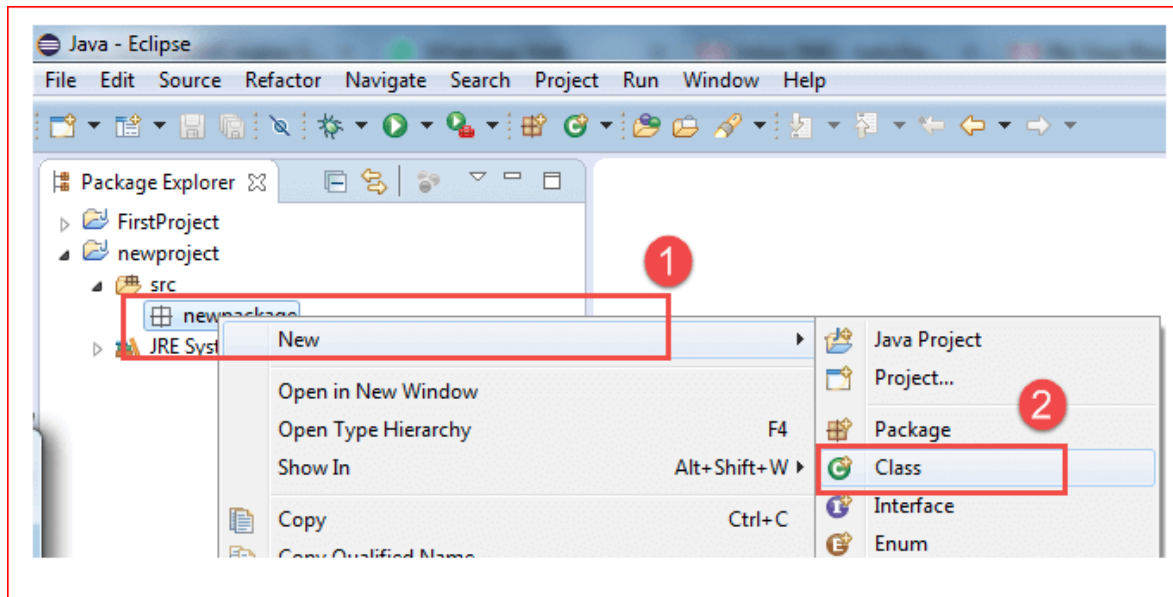
When you click on Class, a pop-up window will open, enter details as

1. Name of the class
2. Click on Finish button
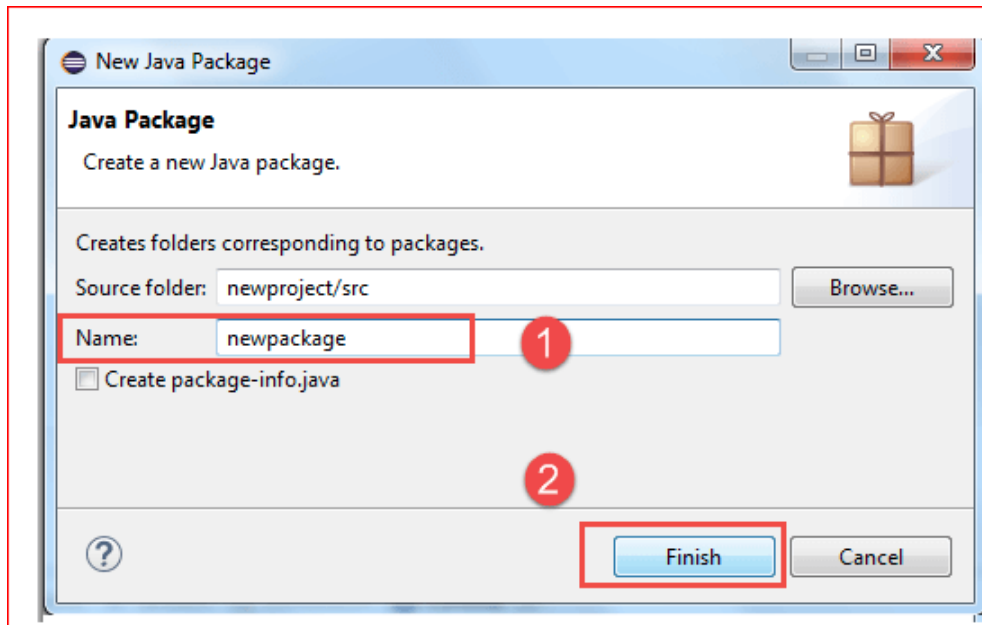
This is how it looks like after creating class.
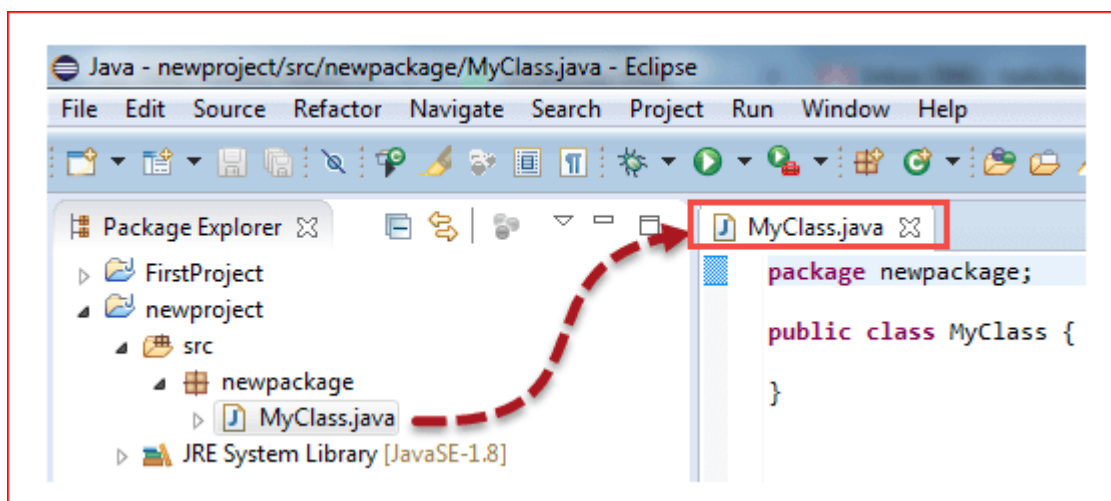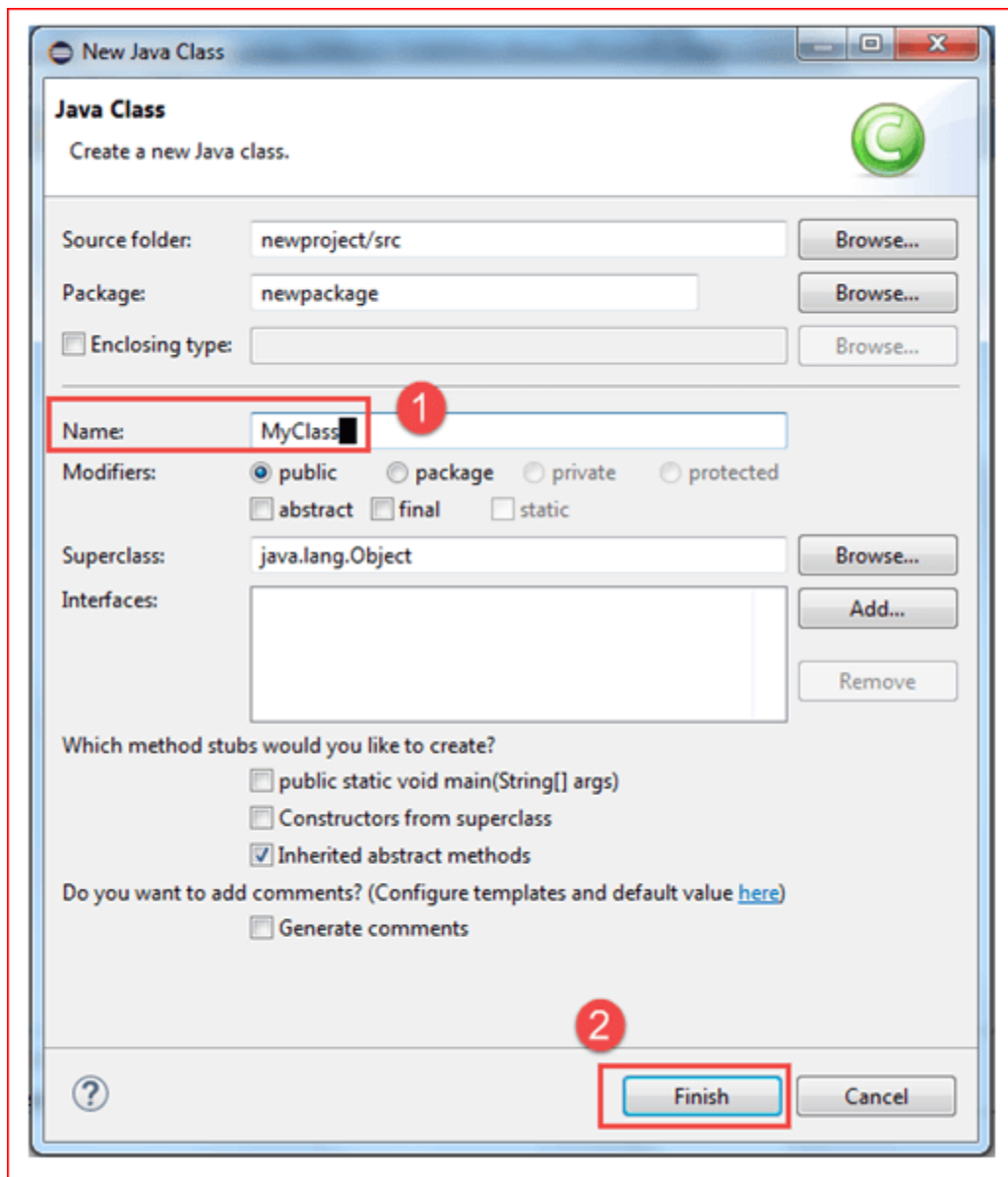
Now selenium WebDriver's into Java Build Path
In this step,

1. Right-click on "newproject" and select **Properties**.
2. On the Properties dialog, click on "Java Build Path".
3. Click on the **Libraries** tab, and then
4. Click on "Add External JARs.."
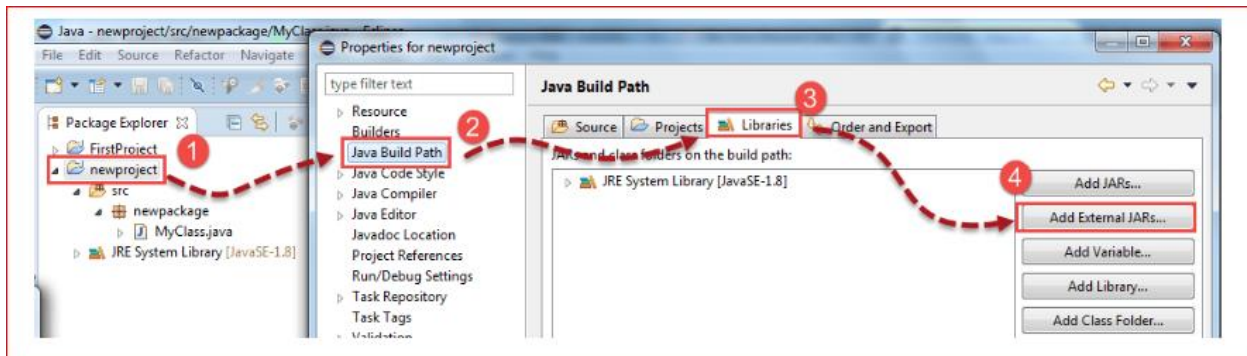
When you click on "Add External JARs.." It will open a pop-up window. Select the JAR files you want to add.
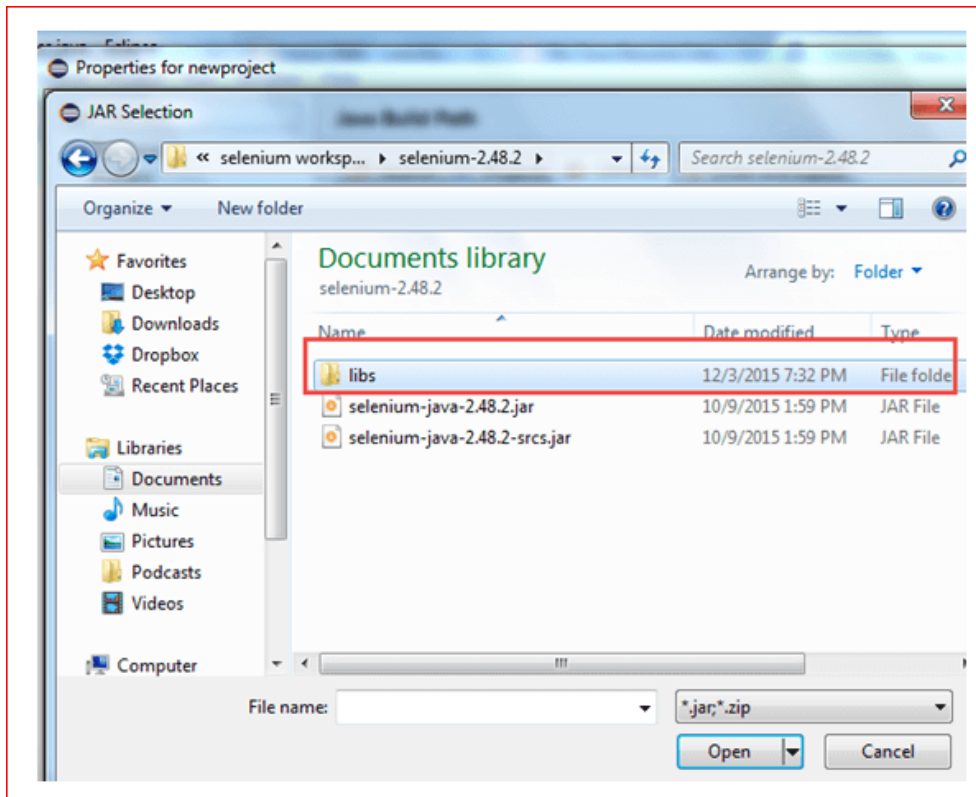
After selecting jar files, click on OK button.
Select all files inside the lib folder.
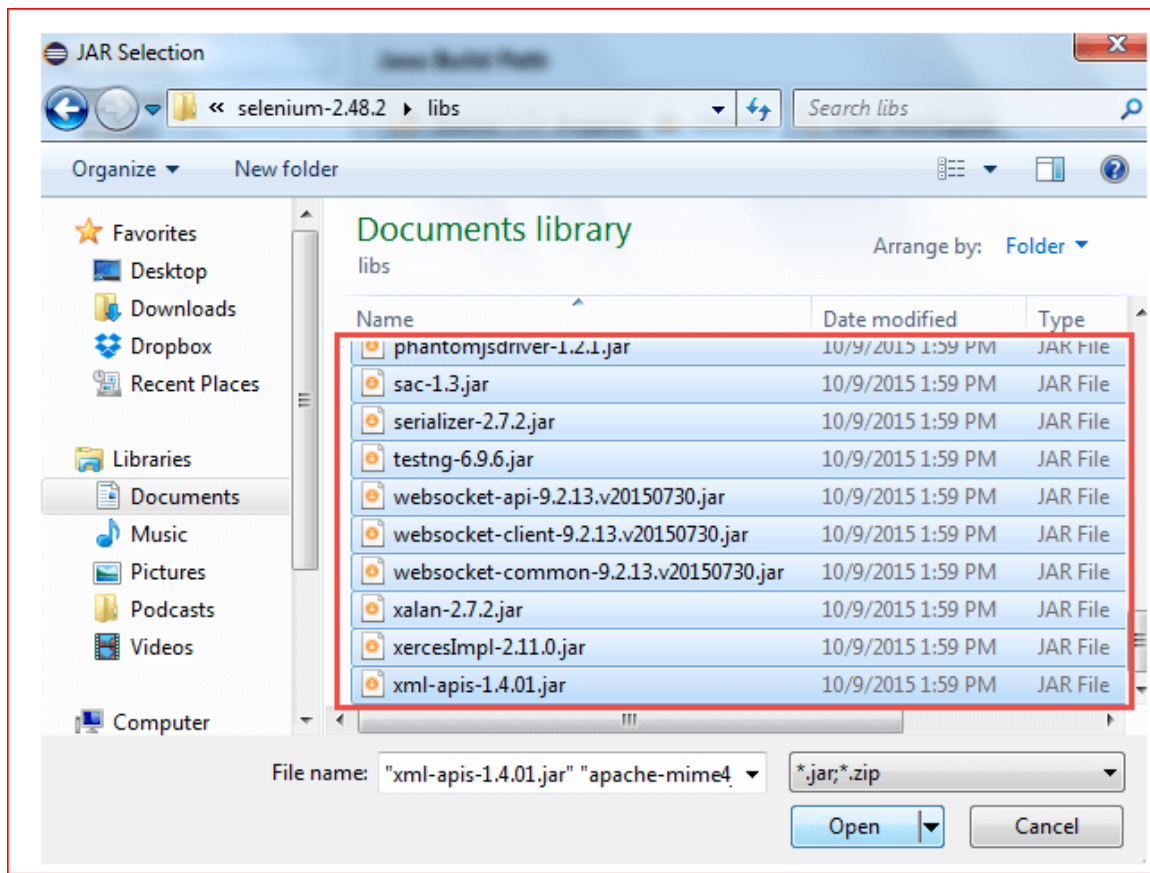
Select files outside lib folder
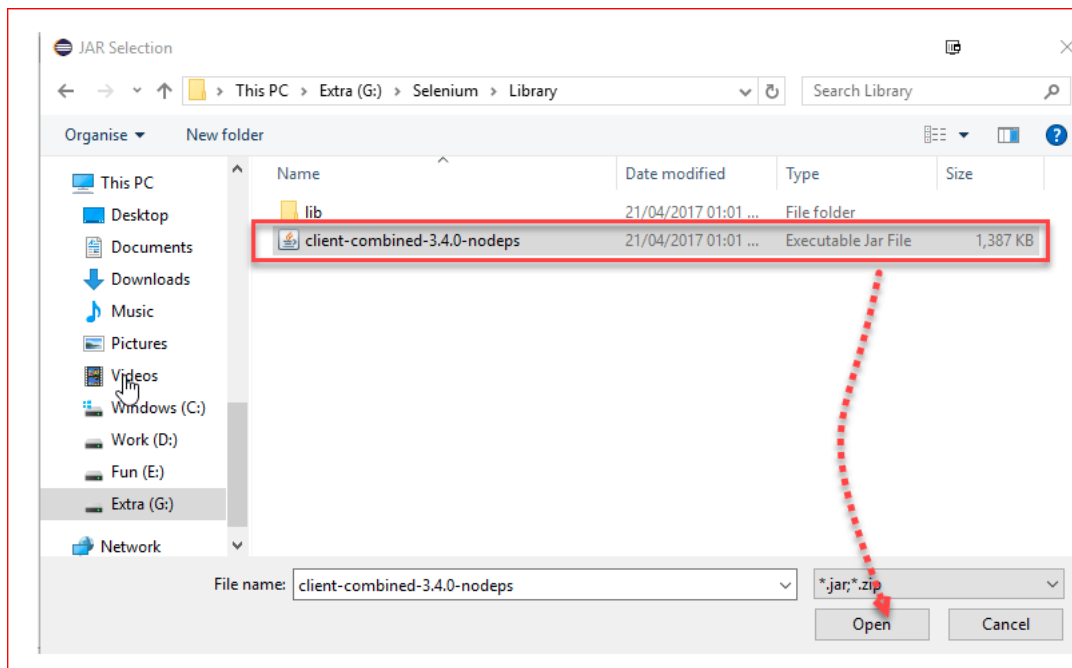
Once done, click "Apply and Close" button
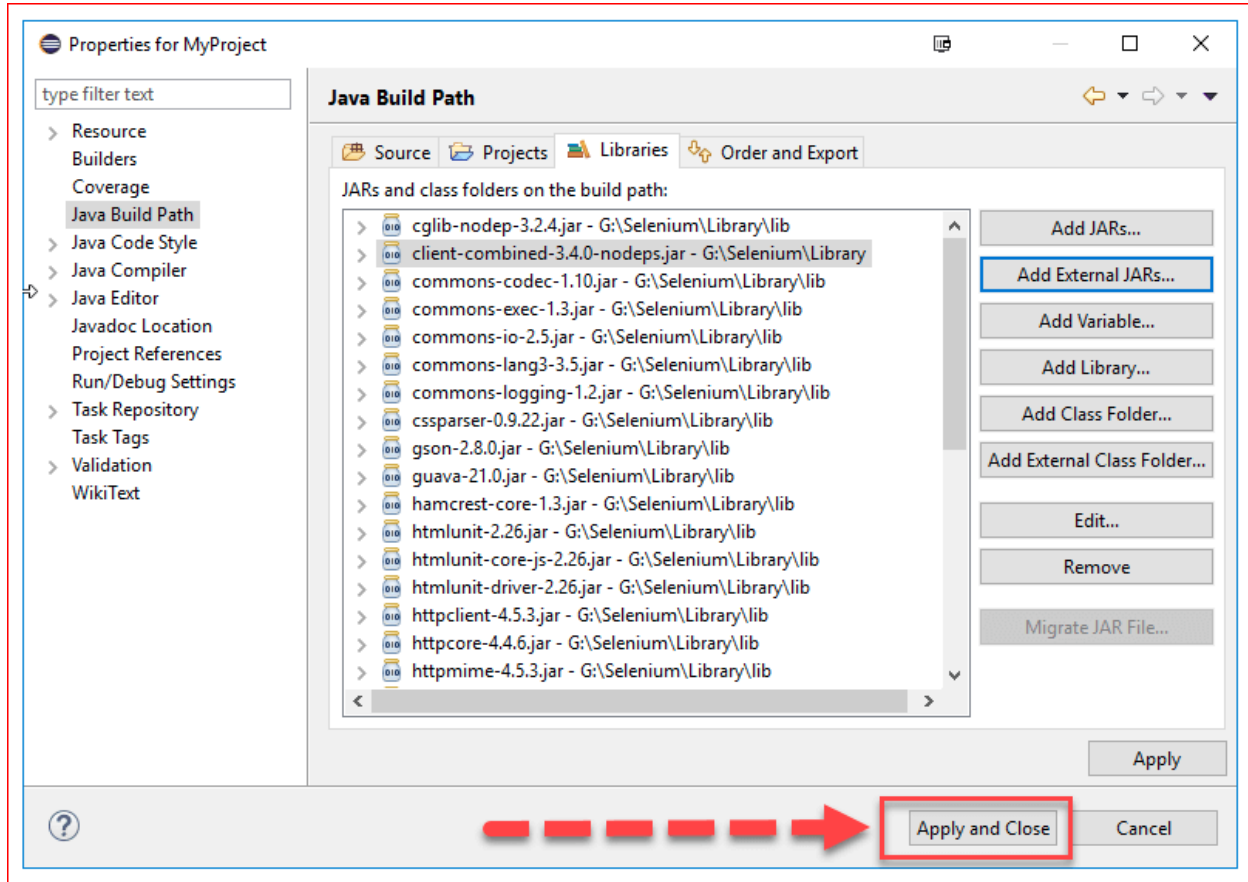
6. Add all the JAR files inside and outside the "libs" folder. Your Properties dialog should now look similar to the image below.
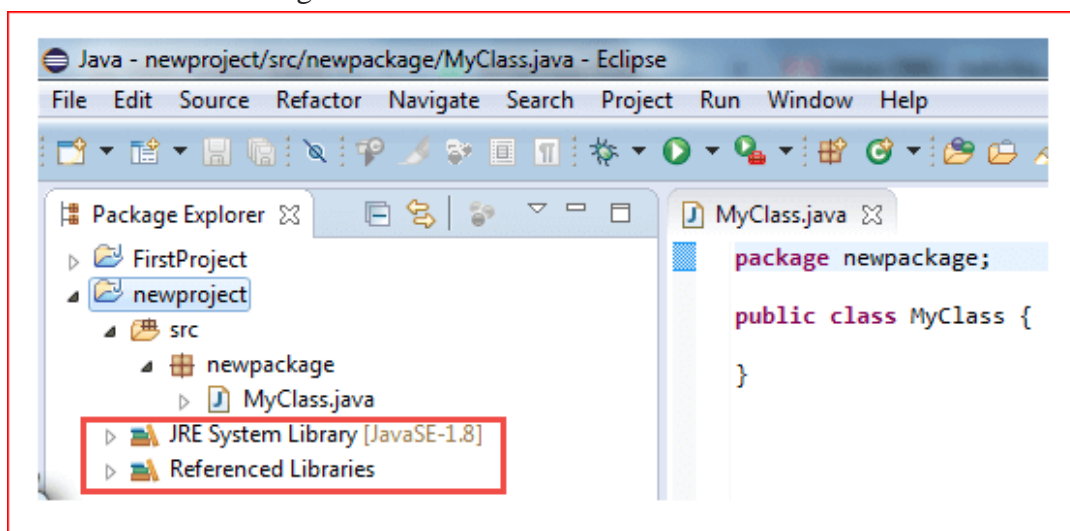
7. Finally, click OK and we are done importing Selenium libraries into our project.

WEEK PROGRAMS

WEEK-1

AIM: Download and Install JAVA, Associate SWD Jars and Browser drivers.

ALGORITHM:
1. How to install JAVA
2. How to create and associate with SWD jars
3. How to select the browser drivers and make them to install.
4. How to convert browser drivers into .exe format.

SOURCE CODE:

Install Java on Windows 10

After downloading the installation file, proceed with installing Java on your Windows system.

Follow the steps below:

Step 1: Run the Downloaded File

Double-click the **downloaded file** to start the installation.

Step 2: Configure the Installation Wizard

After running the installation file, the installation wizard welcome screen appears.

1. Click **Next** to proceed to the next step.
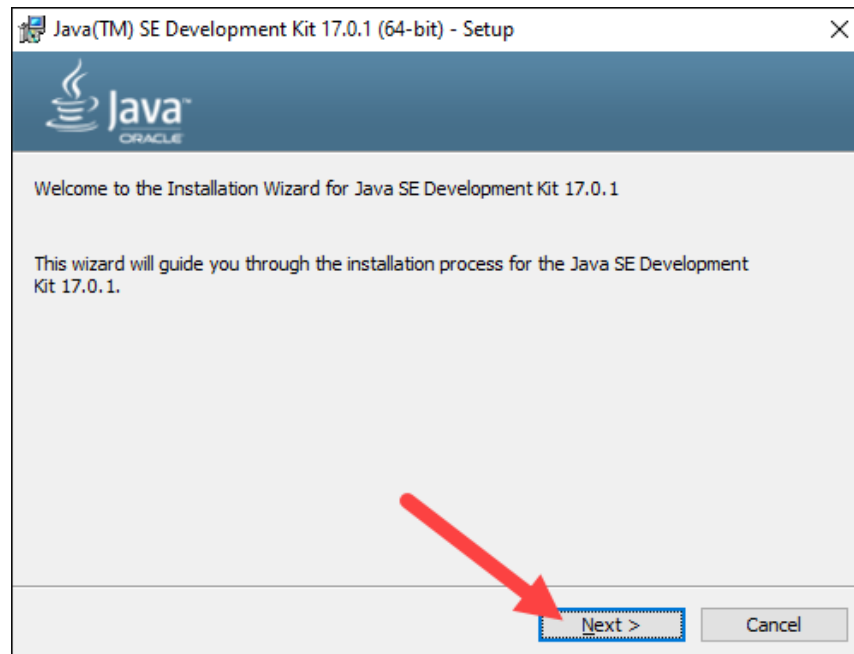
3. Choose the destination folder for the Java installation files or stick to the default path. Click **Next** to proceed.
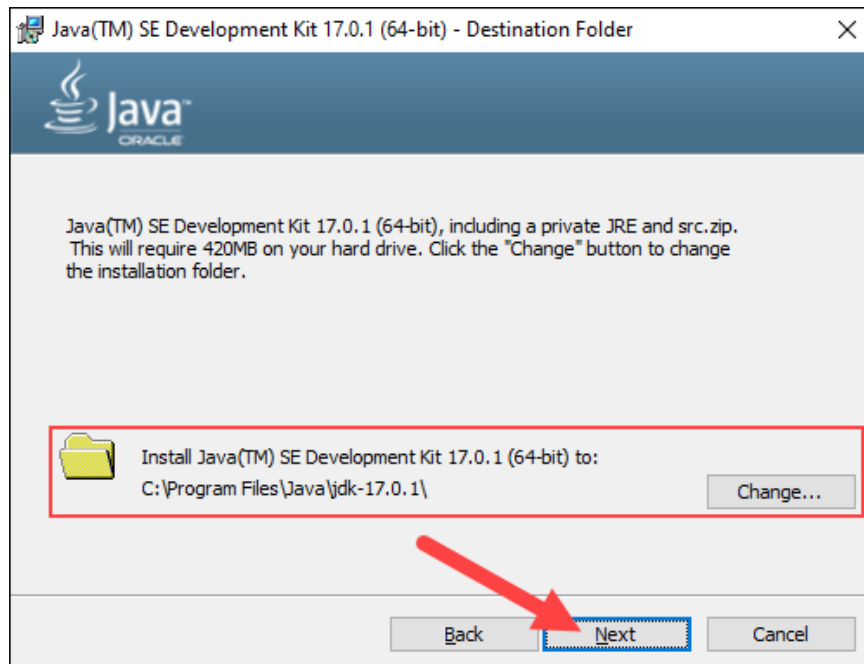
5. Wait for the wizard to finish the installation process until the *Successfully Installed* message appears. Click **Close** to exit the wizard.

<u>Associate SWD Jars : [selenium web driver]</u>

Selenium Server (Grid)

The Selenium Server is needed in order to run Remote Selenium WebDriver (Grid).
Selenium Grid allows the execution of WebDriver scripts on remote machines (virtual or real) by routing commands sent by the client to remote browser instances. It aims to provide an easy way to run tests in parallel on multiple machines.

Selenium Grid allows us to run tests in parallel on multiple machines, and to manage different browser versions and browser configurations centrally (instead of in each individual test).

Selenium Grid is not a silver bullet. It solves a subset of common delegation and distribution problems, but will for example not manage your infrastructure, and might not suit your specific needs.

Purposes and main functionalities

- Central entry point for all tests
- Management and control of the nodes / environment where the browsers run
- Scaling
- Running tests in parallel
- Cross-platform testing
- Load balancing

Selenium Grid 4

Grid 4 takes advantage of a number of new technologies in order to facilitate scaling up while allowing local execution.

Selenium Grid 4 is a fresh implementation and does not share the codebase the previous version had.

To get all the details of Grid 4 components, understand how it works, and how to set up you own, please browse thorough the following sections.

// Latest stable version <u>4.1.4</u>
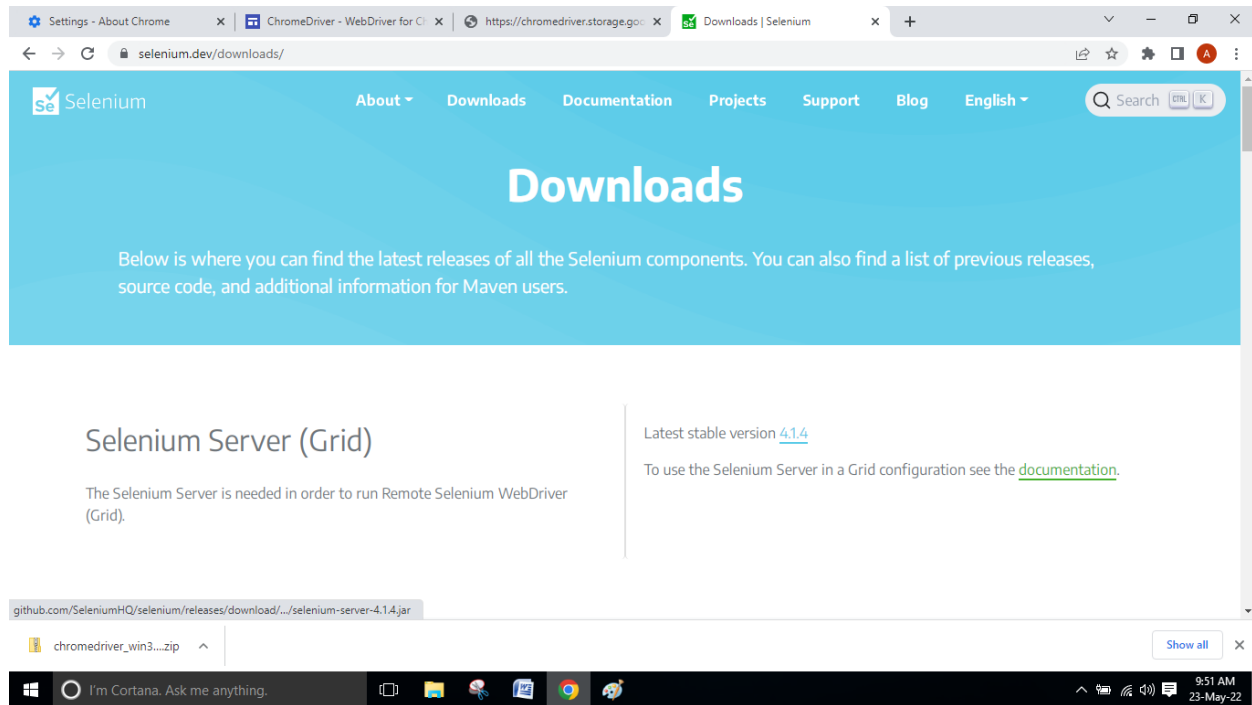
Browser drivers :

Install browser drivers

Setting up your system to allow a browser to be automated.

Through WebDriver, Selenium supports all major browsers on the market such as Chrome/Chromium, Firefox, Internet Explorer, Edge, Opera, and Safari. Where possible, WebDriver drives the browser using the browser's built-in support for automation.

Since all the driver implementations except for Internet Explorer are provided by the browser vendors themselves, they are not included in the standard Selenium distribution. This section explains the basic requirements for getting you started with the different browsers.

Three Ways to Use Drivers

1. Driver Management Software

Most machines automatically update the browser, but the driver does not. To make sure you get the correct driver for your browser, there are many third party libraries to assist you.

1. Import WebDriver Manager

```
import io.github.bonigarcia.wdm.WebDriverManager;
Copy
```

2. Calling setup() automatically puts the correct browser driver where the code will see it:

```
WebDriverManager.chromedriver().setup();
Copy
```

3. Just initialize the driver as you normally would:

```
ChromeDriver driver = new ChromeDriver()
```

2. The PATH Environment Variable

This option first requires manually downloading the driver.

This is a flexible option to change location of drivers without having to update your code, and will work on multiple machines without requiring that each machine put the drivers in the same place.

You can either place the drivers in a directory that is already listed in PATH, or you can place them in a directory and add it to PATH.
To see what directories are already on PATH, open a Command Prompt and execute:

```
echo %PATH%
Copy
```
If the location to your driver is not already in a directory listed, you can add a new directory to PATH:

```
setx PATH "%PATH%;C:\WebDriver\bin"
Copy
```
You can test if it has been added correctly by starting the driver:

```
chromedriver.exe
Copy
```
If your PATH is configured correctly above, you will see some output relating to the startup of the driver:

```
Starting ChromeDriver 95.0.4638.54 (d31a821ec901f68d0d34ccdbaea45b4c86ce543e-refs/branch-heads/4638@{#871})
on port 9515
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
```

3. Hard Coded Location

Similar to Option 2 above, you need to manually download the driver Specifying the location in the code itself has the advantage of not needing to figure out Environment Variables on your system, but has the drawback of making the code much less flexible.

```
System.setProperty("webdriver.chrome.driver","/path/to/chromedriver");
ChromeDriver driver = new ChromeDriver();
```

STEPS TO INSTALL CHROME DRIVER :
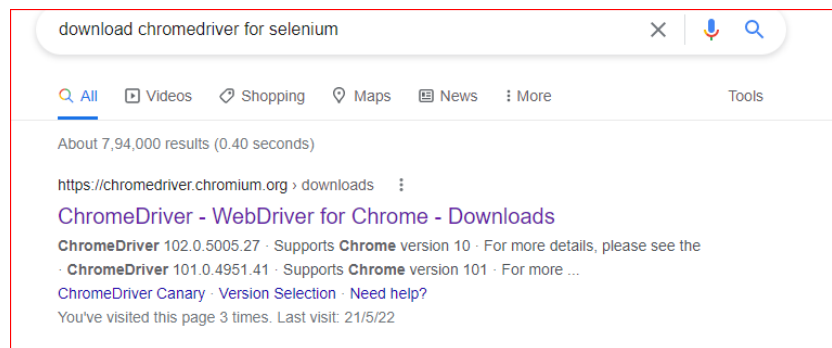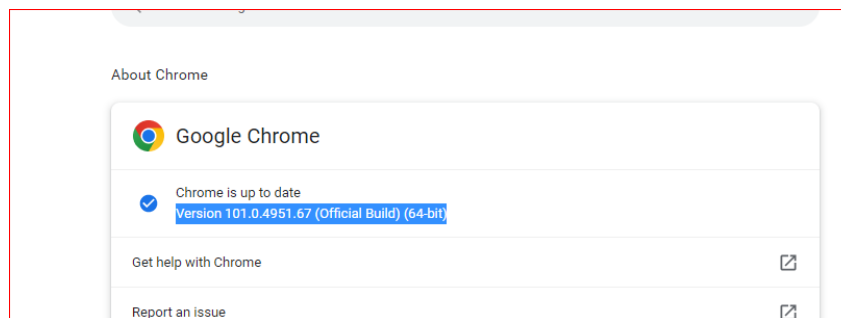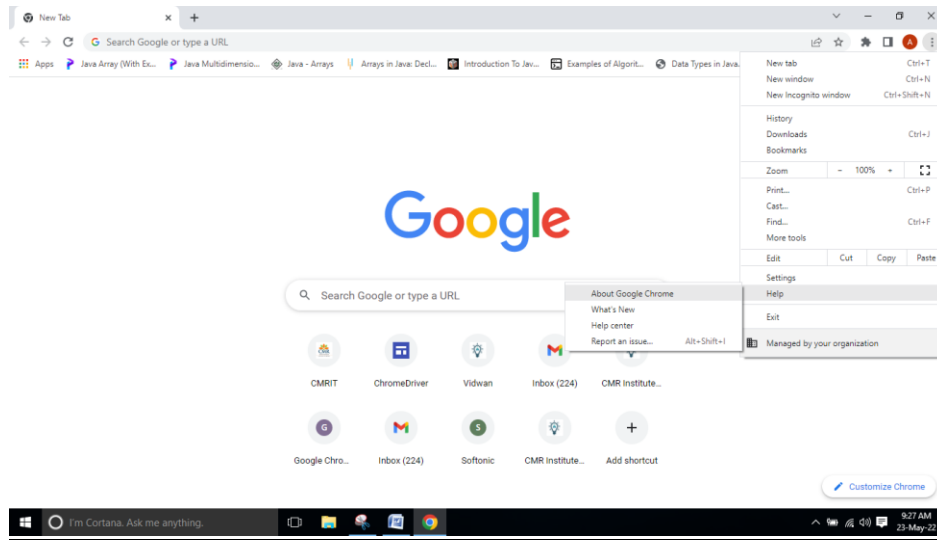
Current Releases

- If you are using Chrome version 102, please download ChromeDriver 102.0.5005.27

- If you are using Chrome version 101, please download ChromeDriver 101.0.4951.41

- If you are using Chrome version 100, please download ChromeDriver 100.0.4896.60

Index of /101.0.4951.41/

Name	Last modified	Size	ETag
Parent Directory		-	
chromedriver_linux64.zip	2022-04-27 07:02:29	5.92MB	57fc88db21f5d009cdf526480378
chromedriver_mac64.zip	2022-04-27 07:02:31	7.88MB	1589eb6b65c5a6848d44dd43c881
chromedriver_mac64_m1.zip	2022-04-27 07:02:34	7.19MB	d6d6cfbd06ca5139f3663d2e68a8
chromedriver_win32.zip	2022-04-27 07:02:37	6.05MB	594669544f54e61c3762252d1a85
notes.txt	2022-04-27 07:02:42	0.00MB	c63873505b72aa1911a152618e25

chromedriver_win3....zip

3.7/6.1 MB, 6 secs left

chromedriver_wi n32 (5)
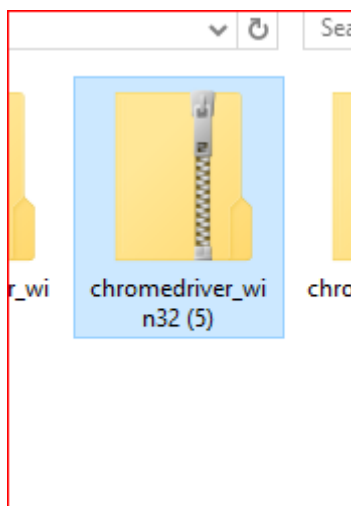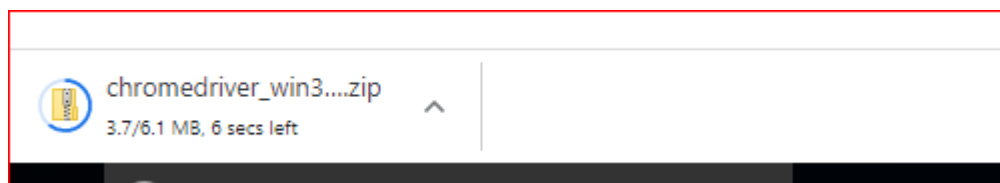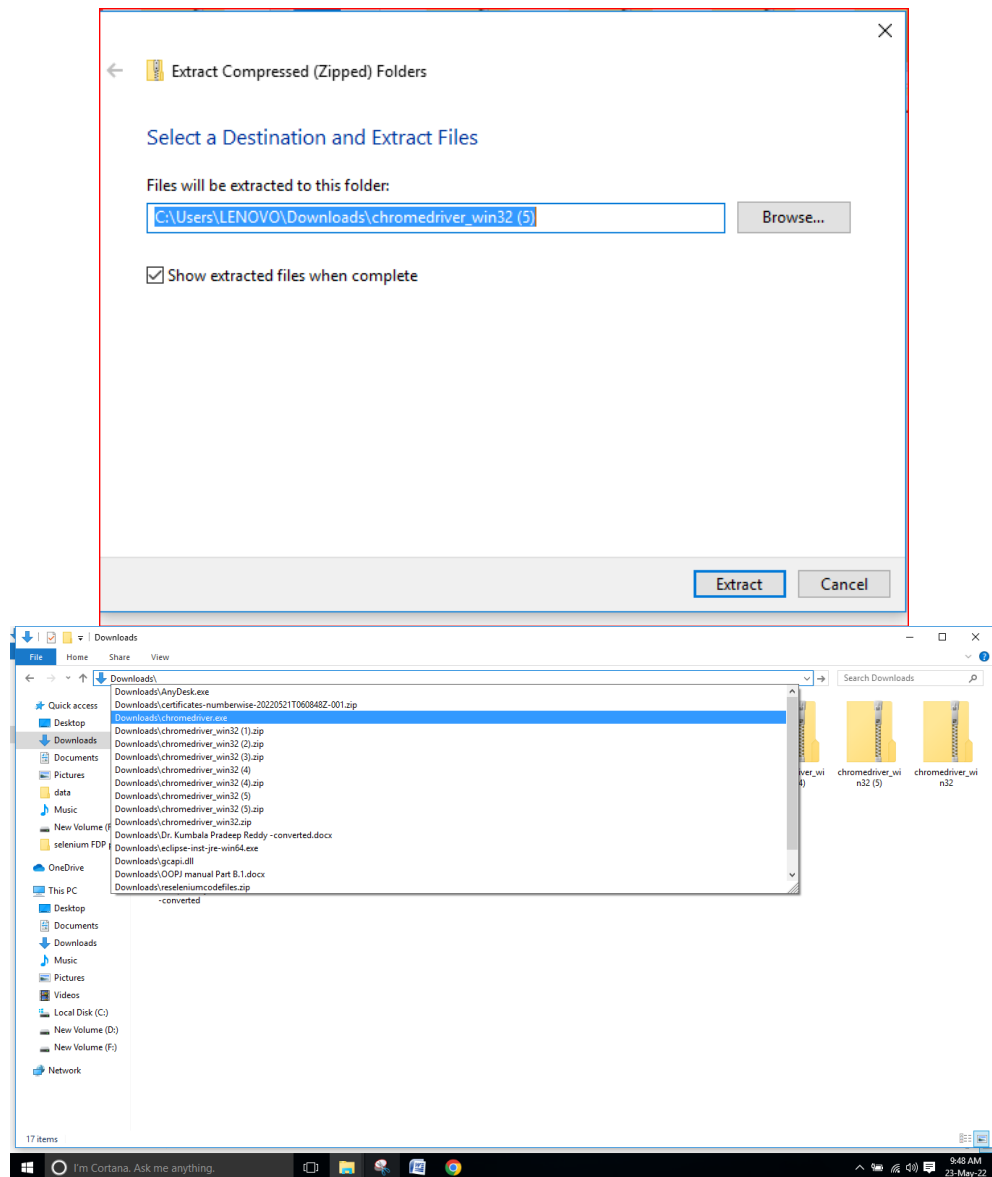
Viva Questions:

1) **What is automation testing?**

   Test automation is the process of simulating a set of (Manual) actions on the
   given system using anautomation tool or utility
   Note: Automation is not a replacement for manual testing.
      Automation is not intended to find bugs.
      100% testing can't be automated.

2) **Why to automate?**

   Repetitive tests performed by a tester manually
   - ➢ May results in human errors
   - ➢ consume more time.

3) **What automation helps to?**

   Automation helps to
   - ➢ Reduce repetitive manual interaction with system during testing phase
   - ➢ Improve product reliability & quality by eliminating human errors
   - ➢ Improve test coverage with automation and
   - ➢ Reduce costs of testing (when wisely used).

4) **What to automate? What not to automate?**

   What to automate

   Tests that need to run for every build
   Testes that use multiple data values for same action
   Identical tests that need to be executed using different browsers.

   What not to automate

   Test cases that are infrequently selected for execution
   Test cases that will only be executed once.
   test cases used for Ad-hoc/random testing.
   Test cases that will require manual intervention i.e a task not possible to automate.
   The most important one-based on the intuition and knowledge of application
   Ex: If you find that you cannot escape from manual intervention.

5) **Difference between Automation testing and manual testing ?**

Manual Testing	**Automated Testing**
• Manual testing is not accurate at all times due to human error, hence it is less reliable.	• Automated testing is more reliable, as it is performed by tools and/or scripts.
• Manual testing is time-consuming, taking up human resources.	• Automated testing is executed by software tools, so it is significantly faster than a manual approach.
• Investment is required for human resources.	• Investment is required for testing tools.
• Manual testing is only practical when the test cases are run once or twice, and frequent repetition is not required.	• Automated testing is a practical option when the test cases are run repeatedly over a long time period.
• Manual testing allows for human observation, which may be more useful if the goal is user-friendliness or improved customer experience.	• Automated testing does not entail human observation and cannot guarantee user-friendliness or positive customer experience.

WEEK-2

AIM:
 Launch Mercury Tour website
a) Click Register link to get registration page   b) Fill fields
c) Click submit   d) Close site

ALGORITHM:
1. write the test cases for each .
2. write test case to open google
3. search for mercury tour search
4. click on registration page for new registration
5. write test case to fill all the fields
6. click on submit button
7. test case for close the website if the new registration is successful

SOURCE CODE:

```
package CMRIT.CMRIT_Assignment;

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.interactions.Actions;

public class MercuryTour_Registration {

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

        System.setProperty("webdriver.chrome.driver","C:\\Users\\LENOVO\\Downloads\\chro
medriver.exe");
                WebDriver driver = new ChromeDriver();
                driver.manage().window().maximize();
                driver.get("https://www.mercurytravels.co.in/");
                Thread.sleep(2000);

                Actions builder = new Actions(driver);

                WebElement customerLogin =
        driver.findElement(By.xpath("(//a[@class='dropdown-toggle'][normalize-
space()='Customer Login'])[2]"));
                builder.moveToElement(customerLogin).perform();
                WebElement register = driver.findElement(By.xpath("(//a[@href='#'][normalize-
space()='Register'])[2]"));
```

```
            register.click();
            Thread.sleep(2000);
            WebElement firstName =
driver.findElement(By.xpath("(//input[@id='acc_first_name'])[1]"));
            firstName.sendKeys("ashu");

            WebElement lastName =
driver.findElement(By.xpath("(//input[@id='acc_last_name'])[1]"));
            lastName.sendKeys("kamatam");

            WebElement emailId =
driver.findElement(By.xpath("(//input[@id='acc_user_email'])[1]"));
            emailId.sendKeys("akulasnashwini559@gmail.com");

            WebElement setPassword =
driver.findElement(By.xpath("(//input[@id='acc_user_password'])[1]"));
            setPassword.sendKeys("ashu123");

            WebElement confirmPassword =
driver.findElement(By.xpath("(//input[@id='acc_user_passconf'])[1]"));
            confirmPassword.sendKeys("ashu123");

            WebElement mobileNumber =
driver.findElement(By.xpath("(//input[@id='acc_mobile_no'])[1]"));
            mobileNumber.sendKeys("6305901106");

            WebElement registerBtn = driver.findElement(By.xpath("(//button[normalize-
space()='Register'])[1]"));
            registerBtn.click();
            Thread.sleep(5000);
            driver.close();


    }

    }
```

OUTPUT:
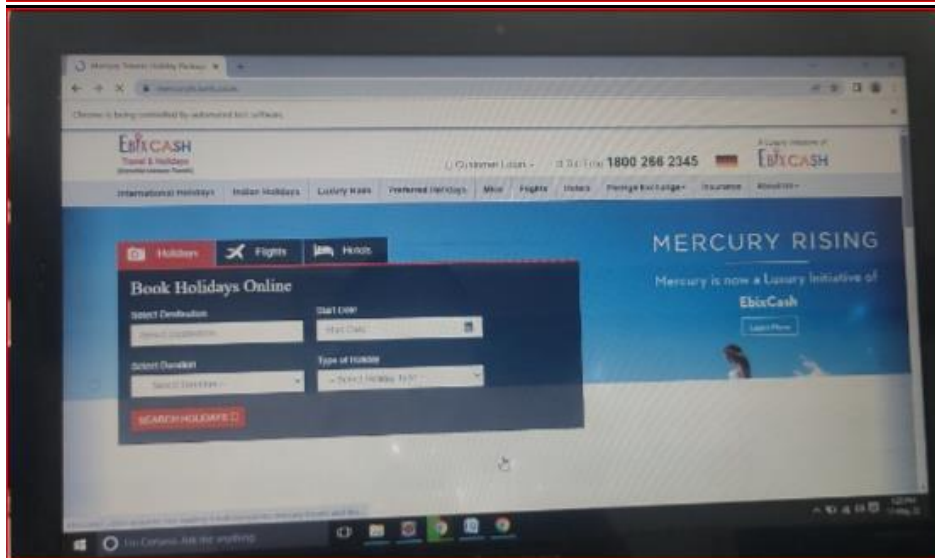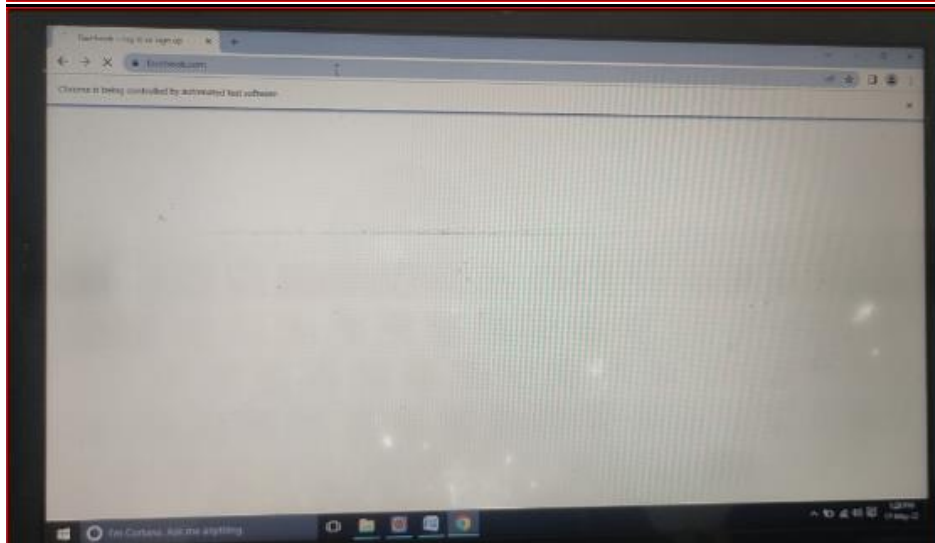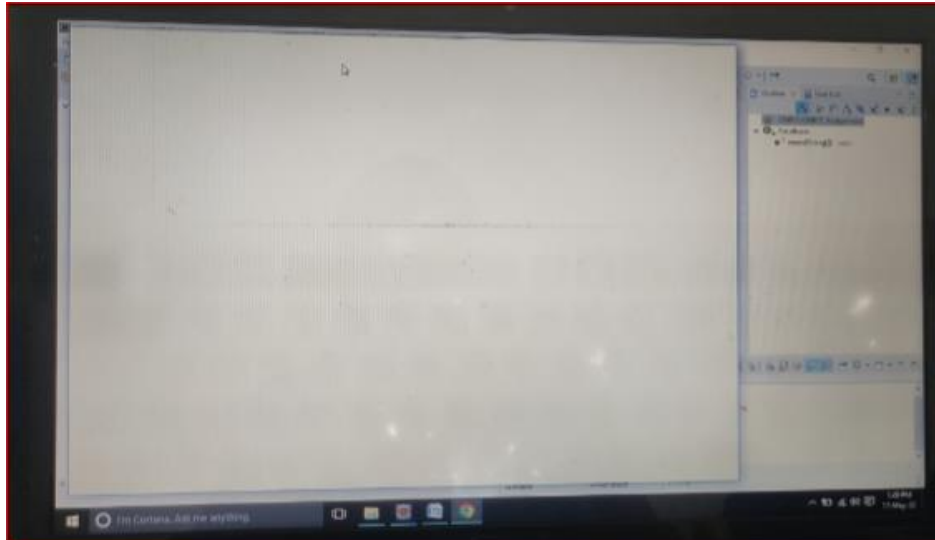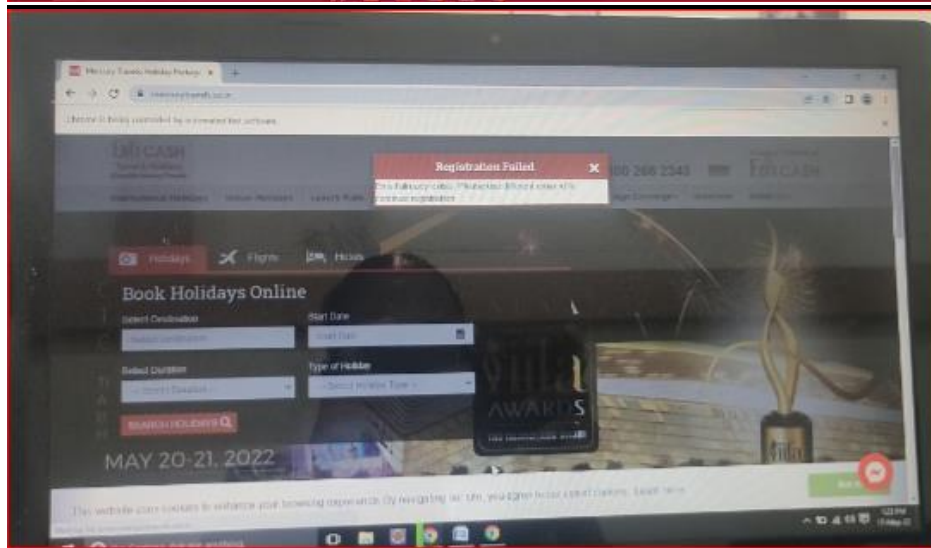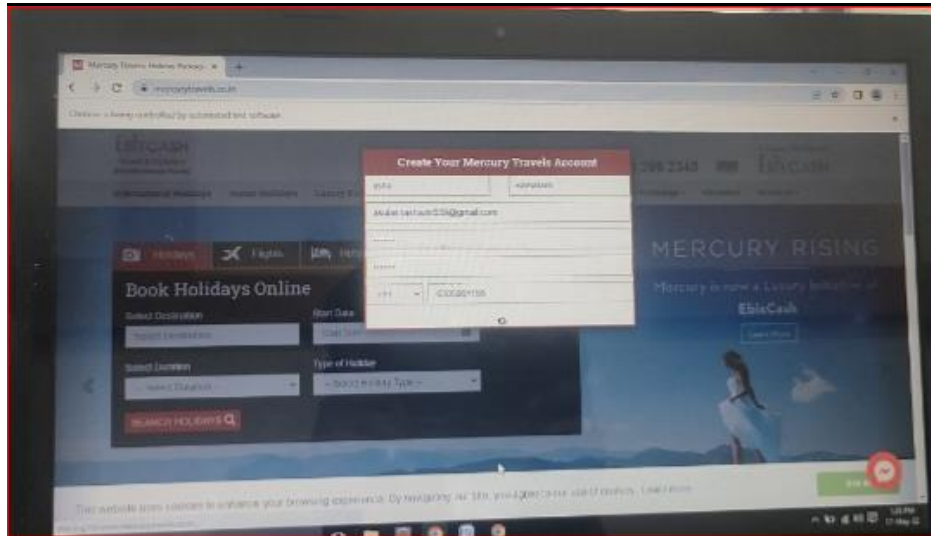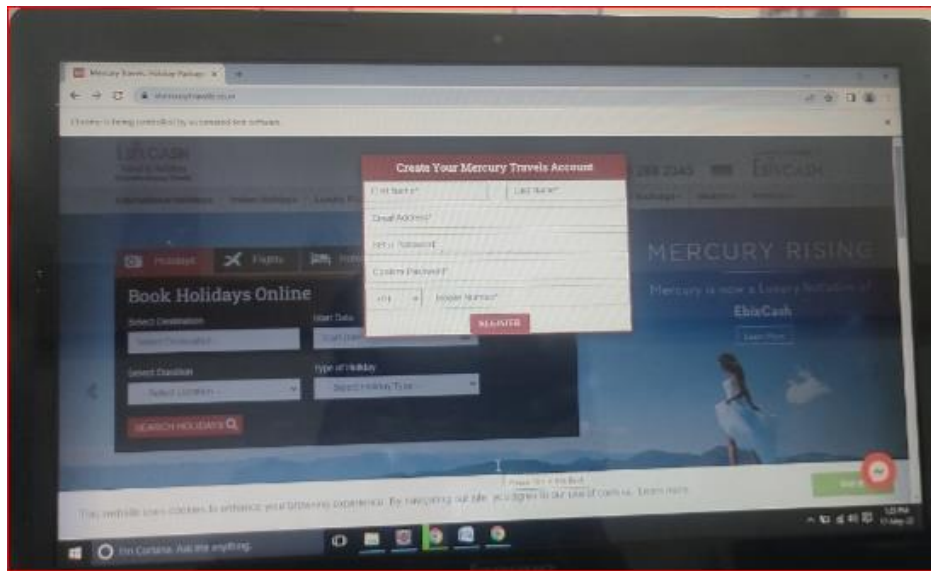
Viva Questions:

1) what are the benefits of automatic testing ?

Fast: Tools runs tests significantly faster than human users

Reliable: Tests perform precisely the same operations each time they are
run, thereby eliminatinghuman error.

Repeatable: You can test how the web site or application reacts after repeated execution
of the same operations.

Programmable: You can program sophisticated tests that bring out hidden information.

Comprehensive: You can build a suite of tests that covers every feature in your
application.

Reusable: You can reuse tests on different versions of a website or appli-
cation.

2) what are the advantages in automation ?

Advantages in Automation

- Less Execution Time and Effort. Tests can be executed on unattended mode and
  24*7.
- Parameterize all the input data through a single global data sheet
  makes it easy to modify theinput data for any future enhancement
  in the application thus reducing the maintenance cost.
- Automation is cost effective.

3) what is selenium?

**Selenium is a free (open source) automated testing suite for web
applications across different browsers and platforms.** It is quite similar to
HP Quick Test Pro (QTP) only that Selenium focuses on automating web-
based applications.

4) whether selenium is a single tool or combination of suite of tools?

yes, it is having different testing needs of an organization. it has four compo-
nents.

- Selenium Integrated Development Environment (IDE)
- Selenium Remote Control (RC)
- WebDriver
- Selenium Grid

5) who developed selenium ?

Primarily, Selenium was created by Jason Huggins in 2004. An engineer at ThoughtWorks, he
was working on a web application that required frequent testing. Having realized that the repeti-

tious Manual Testing of their application was becoming more and more inefficient, he created a JavaScript program that would automatically control the browser's actions. He named this pro-gram as the "JavaScriptTestRunner."

Seeing potential in this idea to help automate other web applications, he made JavaScriptRunner open-source which was later re-named as Selenium Core.

WEEK-3

AIM:

Write a code to search a specific month in the Facebook registration page (Birthday).

ALGORITHM:

1. Search google website
2. Fetching facebook page
3. Sending all the details into required fields
4. Click on login button
5. Exit from the page

SOURCE CODE:

```
package CMRIT.CMRIT_Assignment;

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.Select;

public class Facebook {

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

        System.setProperty("webdriver.chrome.driver","C:\\Users\\LENOVO\\Downloads\\chro
medriver.exe");
                WebDriver driver = new ChromeDriver();
                driver.manage().window().maximize();
                driver.get("https://www.facebook.com/");

                Thread.sleep(2000);

                WebElement createNewAccountButton =
driver.findElement(By.xpath("(//a[normalize-space()='Create New Account'])[1]"));
                createNewAccountButton.click();
                Thread.sleep(5000);
                WebElement firstName = driver.findElement(By.name("firstname"));
                firstName.sendKeys("ashu");

                WebElement surName = driver.findElement(By.name("lastname"));
                surName.sendKeys("kamatam");
```

```
            WebElement mobileNoOrEmailId= driver.findElement(By.name("reg_email__"));
            mobileNoOrEmailId.sendKeys("1234567890");

            WebElement password = driver.findElement(By.name("reg_passwd__"));
            password.sendKeys("123ashu");
            Thread.sleep(2000);
            Select dateDropdown = new
Select(driver.findElement(By.name("birthday_day")));
            dateDropdown.selectByValue("24");
            Thread.sleep(2000);
            Select monthDropdown = new
Select(driver.findElement(By.name("birthday_month")));
            monthDropdown.selectByValue("12");
            Thread.sleep(2000);
            Select yearDropdown = new
Select(driver.findElement(By.name("birthday_year")));
            yearDropdown.selectByValue("1996");
            Thread.sleep(2000);
            WebElement femaleRadioBtn = driver.findElement(By.xpath("(//label[normalize-
space()='Female'])[1]"));
            femaleRadioBtn.click();
            Thread.sleep(2000);
            WebElement signUpBtn = driver.findElement(By.name("websubmit"));
            signUpBtn.click();
            Thread.sleep(2000);
            System.out.println("Performed click action");
            driver.quit();

    }
    }
```

OUTPUT

:
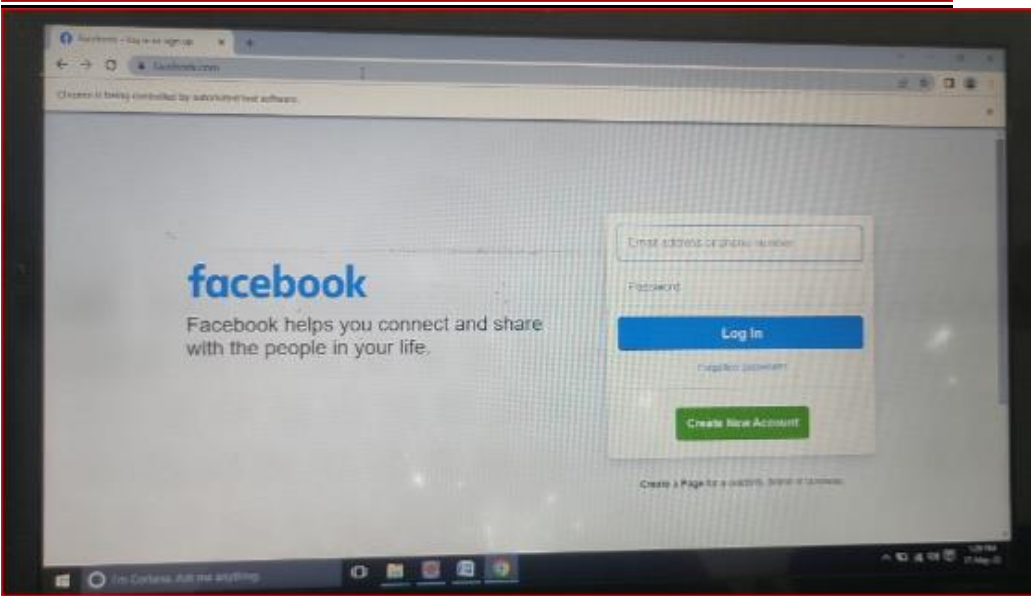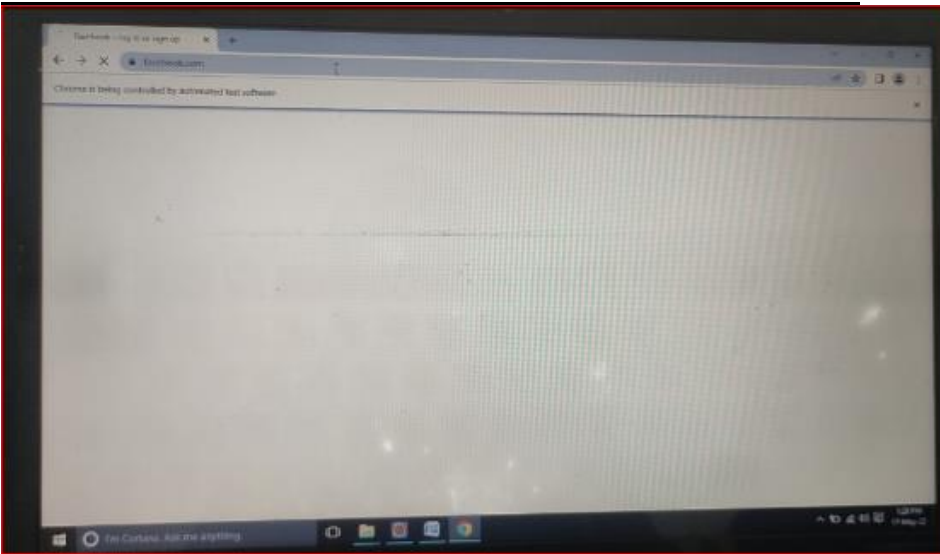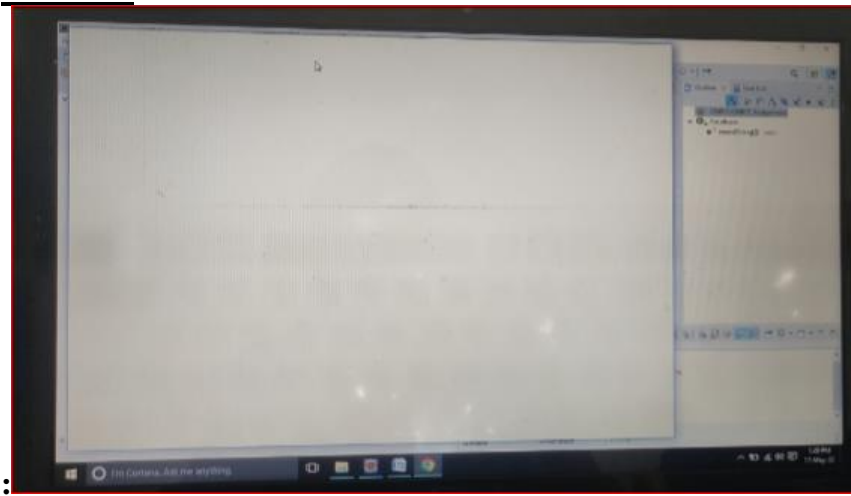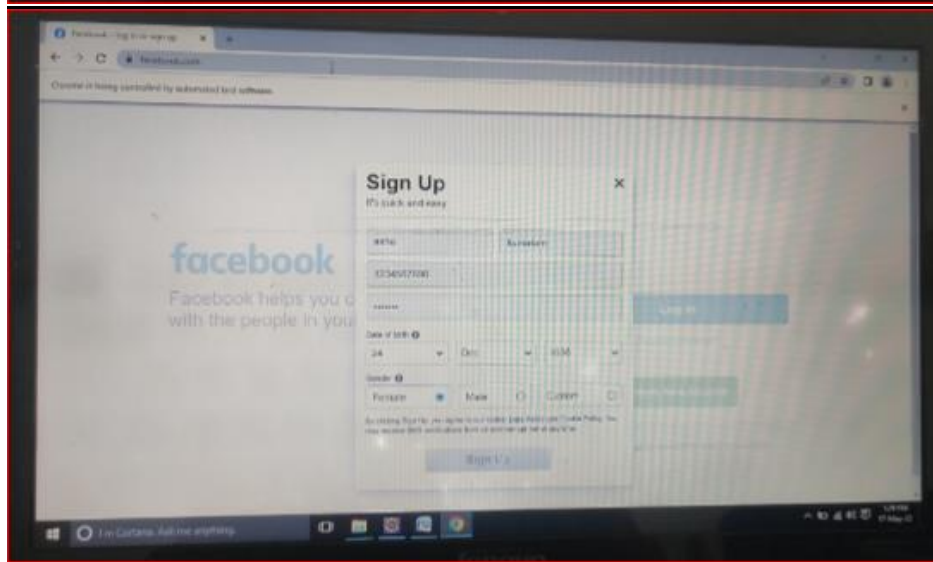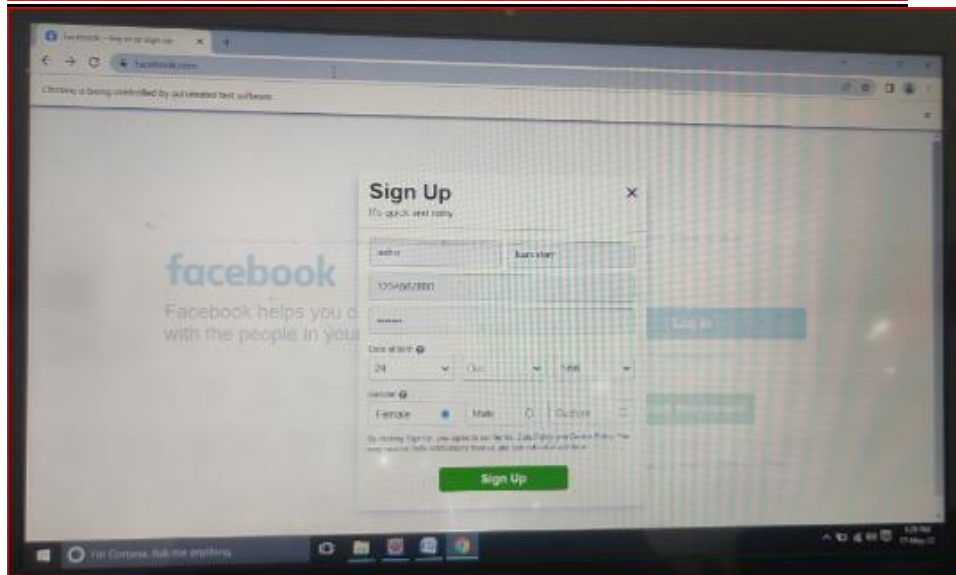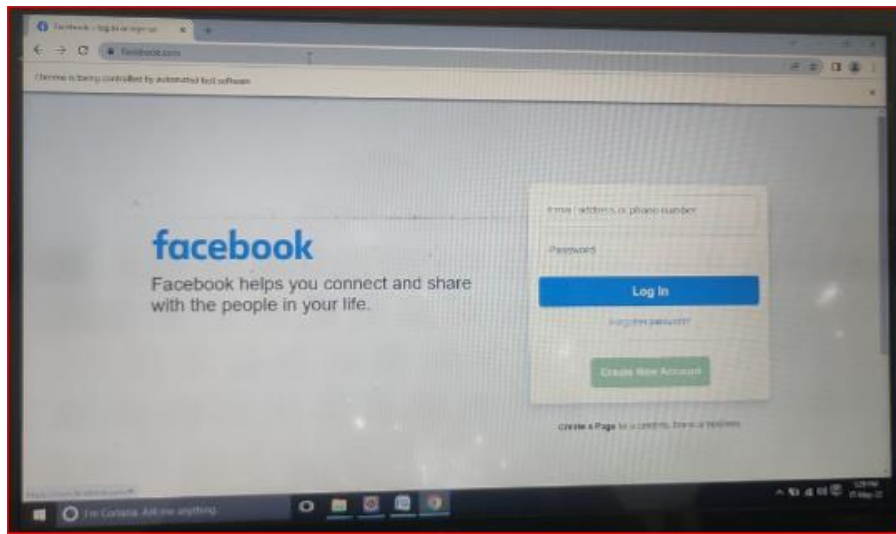
Viva Questions:

1) who introduced selenium RC ?

Unfortunately; testers using Selenium Core had to install the whole application under test and the web server on their own local computers because of the restrictions imposed by the **same origin policy.** So another ThoughtWork's engineer, **Paul Hammant**, decided to create a server that will act as an HTTP proxy to "trick" the browser into believing that Selenium Core and the web application being tested come from the same domain. This system became known as the **Selenium Remote Control** or **Selenium 1**.

2) write about selenium grid ?

Selenium Grid was developed by **Patrick Lightbody** to address the need of minimizing test execution times as much as possible. He initially called the system "**Hosted QA**." It was capable of capturing browser screenshots during significant stages, and also of **sending out Selenium commands to different machines simultaneously.**

3) When to automate ?
When the testing is a long-term project and the AUT does not change very frequently, test automation is a good idea.
And the cost of automation is affordable.

4) write about selenium IDE?
Shinya Kasatani of Japan created Selenium IDE, a Firefox extension that can automate the browser through a record-and-playback feature. He came up with this idea to further increase the speed in creating test cases. He donated Selenium IDE to the Selenium Project in 2006.

5) write about web driver ?

Simon Stewart **created WebDriver circa** 2006 **when browsers and web applications were becoming more powerful and more restrictive with JavaScript programs like Selenium Core.** It was the first cross-platform testing framework that could control the browser from the OS level.

WEEK-4

AIM:
Write a program which pops out an alert message in frame in personal banking login page.

ALGORITHM:
1. open banking website.
2. add key and value to map as follow to switch off browser notification.
   Pass the argument 1 to allow and 2 to block
3. After opening the banking website it gives us one pop-up by using testcase in our code pop-up.

SOURCE CODE:

```
package CMRIT.CMRIT_Assignment;

import java.util.HashMap;
import java.util.Map;

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.chrome.ChromeOptions;

public class Banking_siteAlertPopup {

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

                Map<String, Object> prefs = new HashMap<String, Object>();

                //add key and value to map as follow to switch off browser notification
                //Pass the argument 1 to allow and 2 to block
                prefs.put("profile.default_content_setting_values.notifications", 2);

                //Create an instance of ChromeOptions
                ChromeOptions options = new ChromeOptions();

                // set ExperimentalOption - prefs
                options.setExperimentalOption("prefs", prefs);


        System.setProperty("webdriver.chrome.driver","C:\\Users\\LENOVO\\Downloads\\chro
medriver.exe");
                WebDriver driver = new ChromeDriver(options);
```

```
            driver.manage().window().maximize();
            driver.get("https://www.axisbank.com/");
            Thread.sleep(2000);
            WebElement popupClose =
driver.findElement(By.xpath("/html[1]/body[1]/div[1]/div[1]/div[1]"));
            popupClose.click();
            driver.close();
            Thread.sleep(2000);
        }
}
```

OUTPUT:
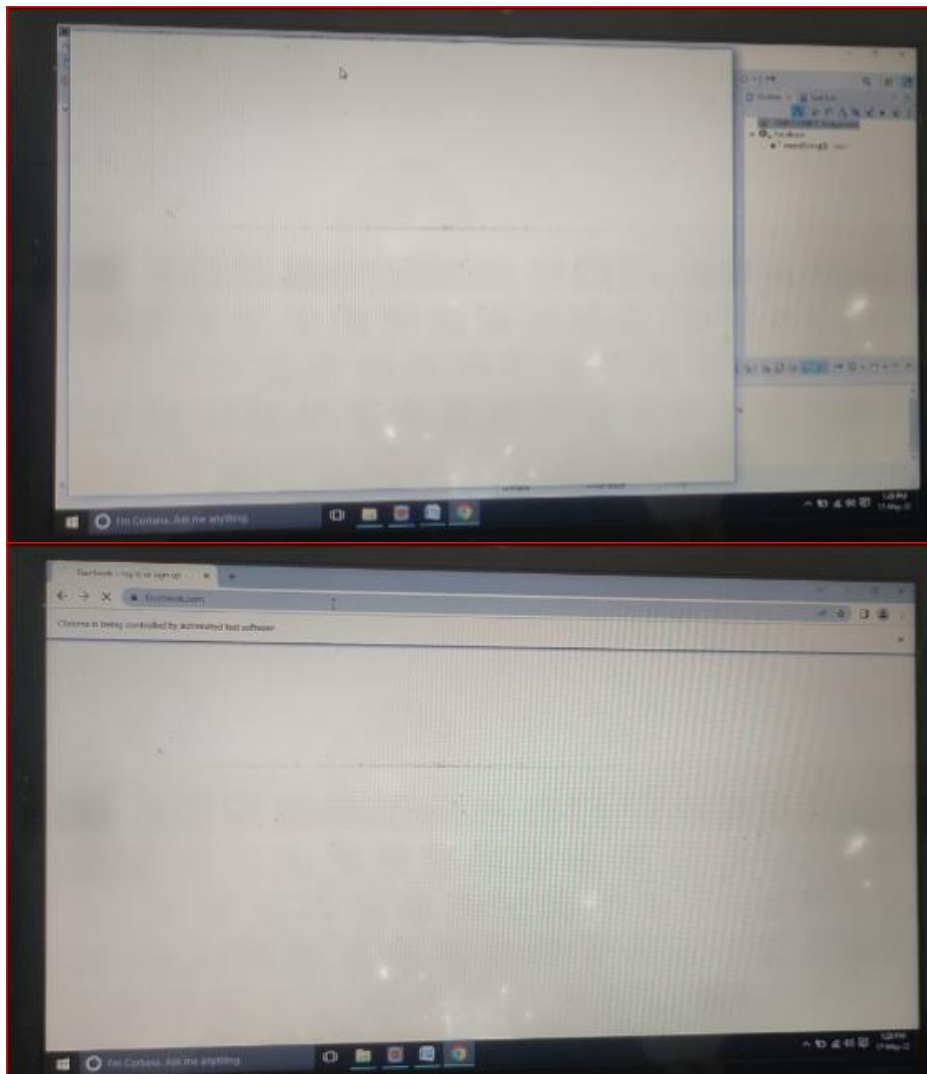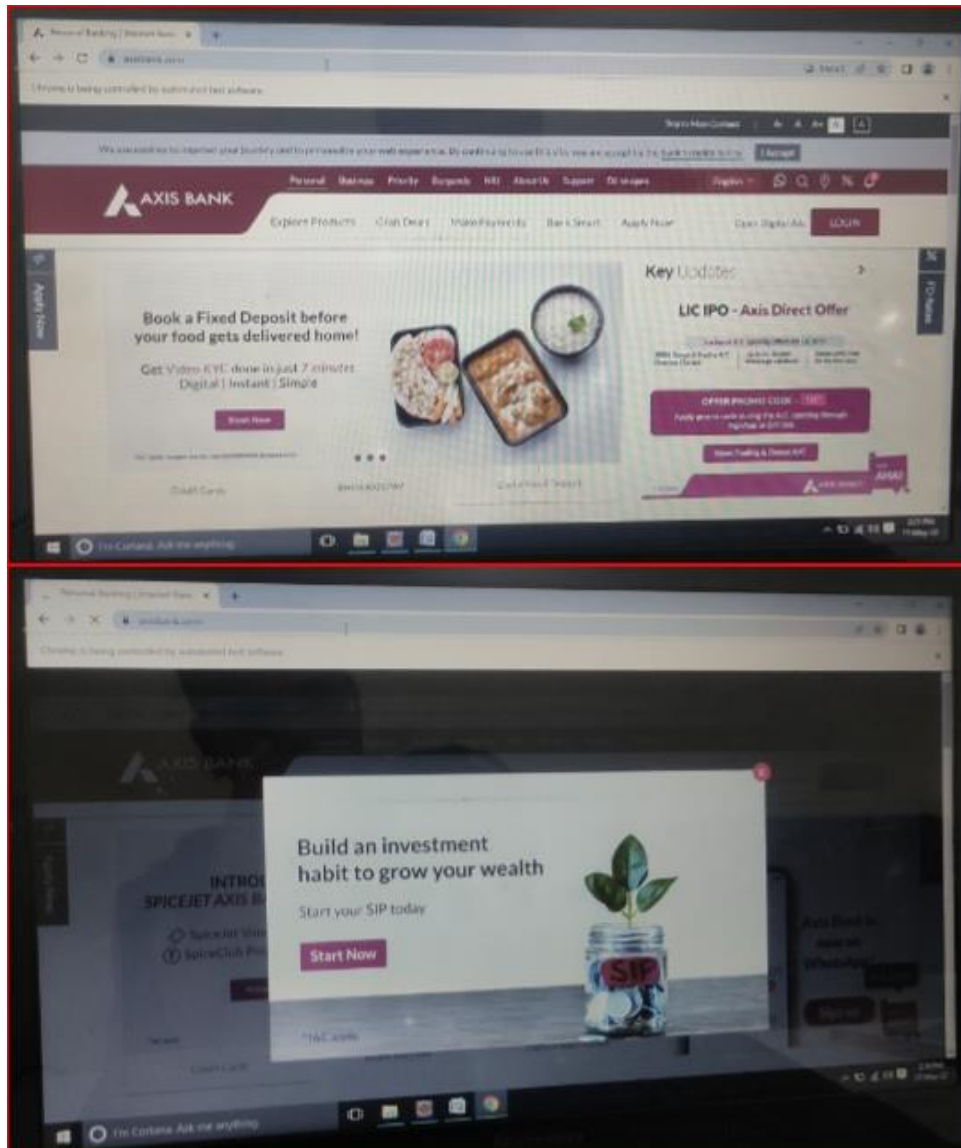
Viva Questions:

1) **What is Selenium 2.0?**

Web Testing tools Selenium RC and WebDriver are consolidated in single tool in Selenium 2.0
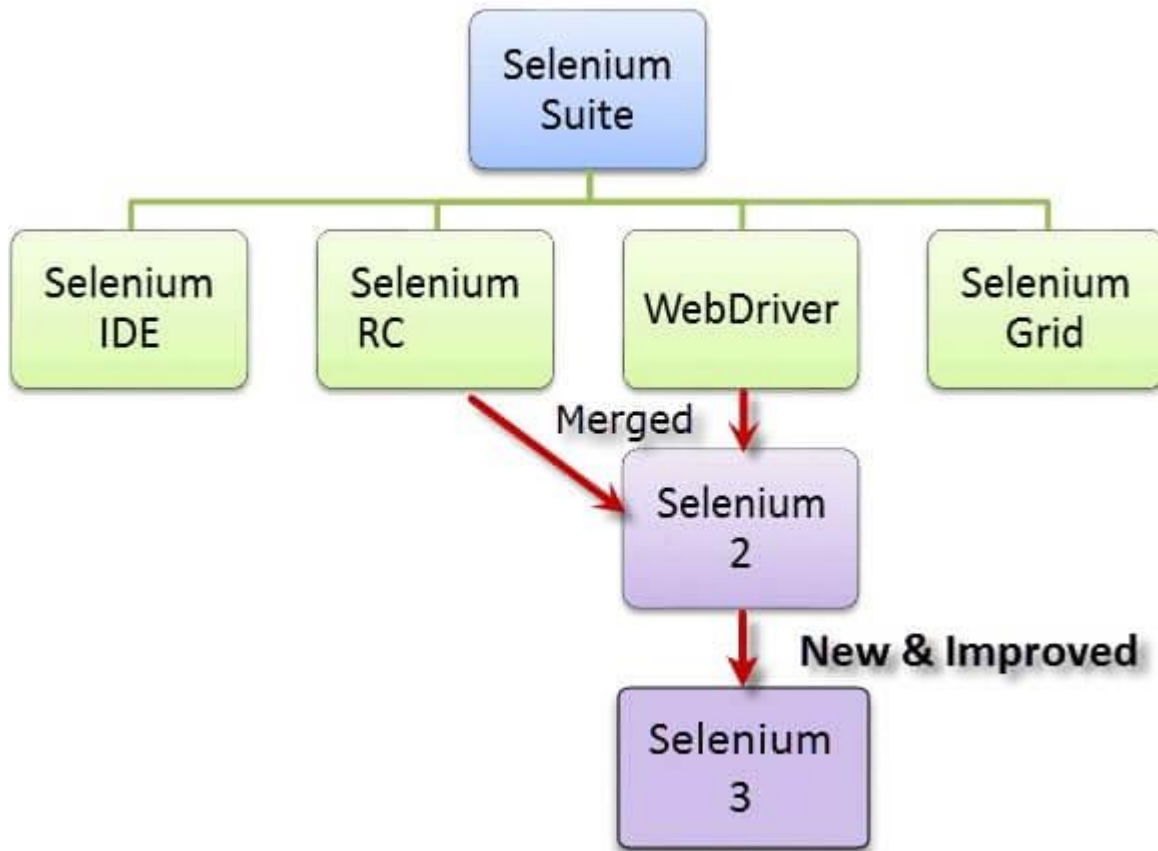
2) **How will you find an element using Selenium?**

In Selenium every object or control in a web page is referred as an elements, there are different ways to find an element in a web page they are

- ID
- Name
- Tag
- Attribute
- CSS
- Linktext
- PartialLink Text
- Xpath etc

3) Mention what is the use of X-path?

X-Path is used to find the WebElement in web pages. It is also useful in identifying the dynamic elements.

4) Explain the difference between single and double slash in X-path?

Single slash '/ '

- Single slash (/) start selection from the document node
- It allows you to create 'absolute' path expressions

Double Slash '// '

- Double slash (//) start selection matching anywhere in the document
- It enables to create 'relative' path expressions

5) What is the difference between verify and assert commands?

Assert: Assert allows to check whether an element is on the page or not. The test will stop on the step failed, if the asserted element is not available. In other words, the test will terminated at the point where check fails.
Verify: Verify command will check whether the element is on the page, if it is not then the test will carry on executing. In verification, all the commands are going to run guaranteed even if any of test fails.

WEEK-5

AIM:
Write a test case to search result section on CMRIT Website.

ALGORITHM:

1. Open google website
2. Search for CMRIT site.
3. Take one official ID of any student in cmrit college.
4. By using that Id we are accessing CMRIT website and giving the info of that student
5. It will redirected to opening the results of that student.

SOURCE CODE:

```java
package CMRIT.CMRIT_Assignment;

import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;

public class CMRIT_StudentResults {

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

        System.setProperty("webdriver.chrome.driver","C:\\Users\\LENOVO\\Downloads\\chro
medriver.exe");
                WebDriver driver = new ChromeDriver();
                driver.manage().window().maximize();
                driver.get("https://www.google.com/");

                Thread.sleep(2000);
                WebElement searchBar = driver.findElement(By.name("q"));
                searchBar.sendKeys("CMRIT");
                searchBar.sendKeys(Keys.ENTER);

                WebElement cmritLink =
driver.findElement(By.xpath("(//h3[contains(text(),'Home - CMR Institute Of Technology ,
Hyderabad, Te')])[1]"));
                cmritLink.click();
```

```
            Thread.sleep(3000);

            Actions builder = new Actions(driver);

            WebElement student =
driver.findElement(By.xpath("/html[1]/body[1]/div[1]/div[1]/section[3]/div[1]/div[1]/div[1]/div[
1]/div[1]/nav[1]/ul[1]/li[9]/a[1]"));
            builder.moveToElement(student).perform();
            student.click();
            Thread.sleep(2000);
            WebElement studentLogin =
driver.findElement(By.xpath("/html[1]/body[1]/div[1]/div[1]/section[3]/div[1]/div[1]/div[1]/div[
1]/div[1]/nav[1]/ul[1]/li[9]/ul[1]/li[1]/a[1]"));
            studentLogin.click();
            WebElement userName =
driver.findElement(By.xpath("(//input[@id='txtUserName'])[1]"));
            Thread.sleep(2000);
            userName.sendKeys("20R01A05K6p");

            WebElement nxtBtn = driver.findElement(By.name("btnNext"));
            nxtBtn.click();

            WebElement password =
driver.findElement(By.xpath("(//input[@placeholder='Enter Password'])[1]"));
            password.sendKeys("20R01A05K6P");

            WebElement submit =
driver.findElement(By.xpath("(//input[@name='btnSubmit'])[1]"));
            submit.click();
            }
}
```

OUTPUT:
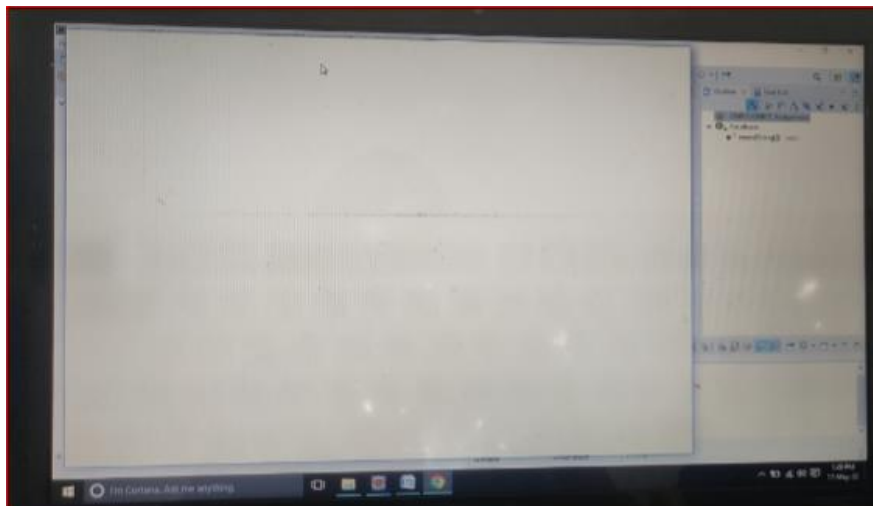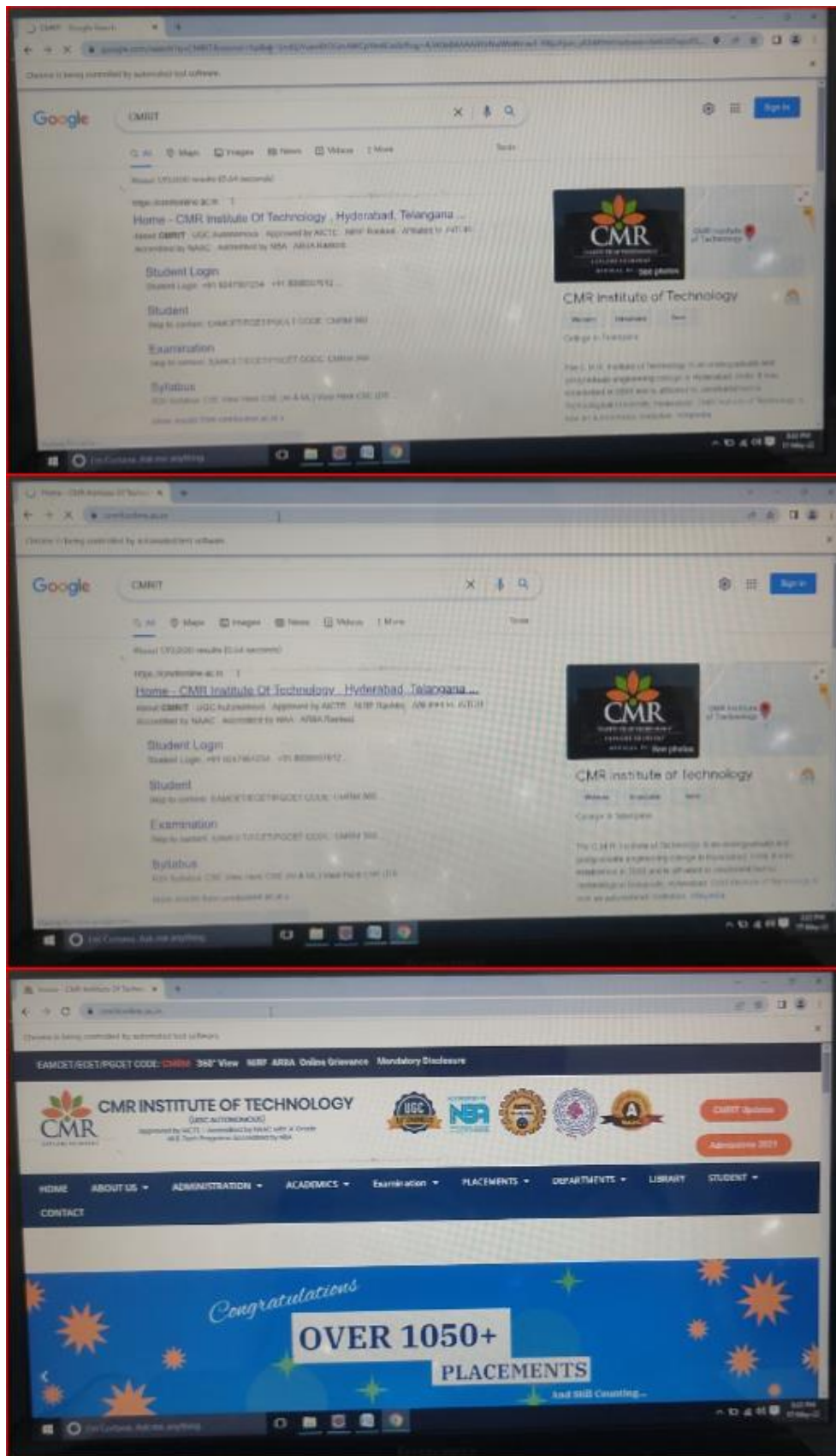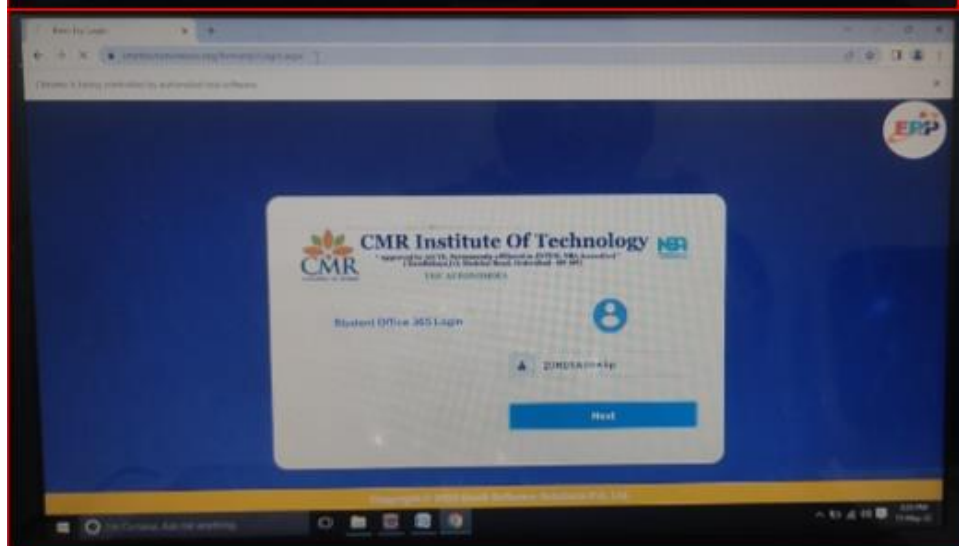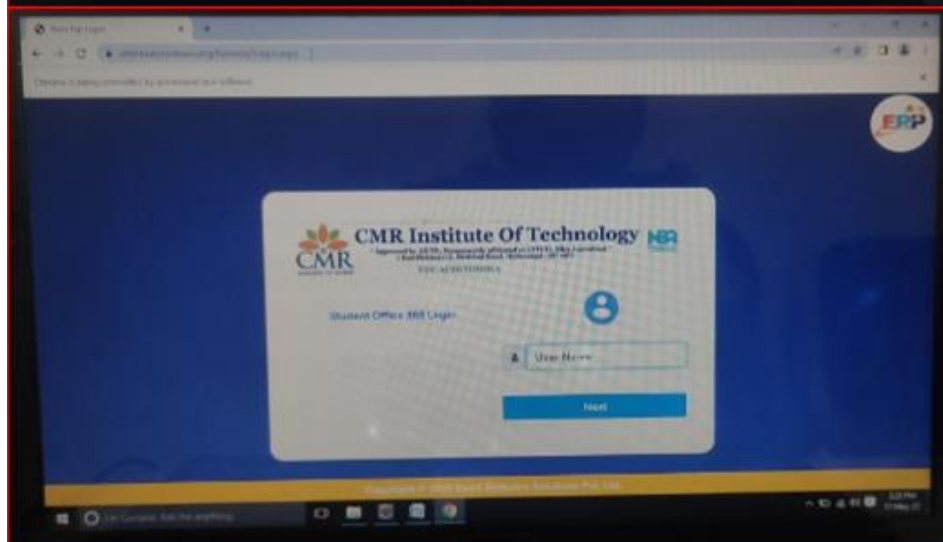
Viva Questions:

1) What are the advantages of Selenium?

- It supports C#, PHP, Java, Perl, Phython
- It supports different OS like Windows, Linux and Mac OS
- It has got powerful methods to locate elements (Xpath, DOM , CSS)
- It has highly developer community supported by Google

2) Why testers should opt for Selenium and not QTP?

Selenium is more popular than QTP as

- Selenium is an open source whereas QTP is a commercial tool
- Selenium is used specially for testing web based applications while QTP can be used for testing client server application also
- Selenium supports Firefox, IE, Opera, Safari on operating systems like Windows, Mac, Linux etc. however QTP is limited to Internet Explorer on Windows.
- Selenium supports many programming languages like Ruby, Perl, Python whereas QTP supports only VB script

3) What are the four parameter you have to pass in Selenium?

Four parameters that you have to pass in Selenium are

- Host
- Port Number
- Browser
- URL

4) What is the difference between setSpeed() and sleep() methods?

Both will delay the speed of execution.

Thread.sleep () : It will stop the current (java) thread for the specified period of time. Its done only once

- It takes a single argument in integer format

Ex: thread.sleep(2000)- It will wait for 2 seconds

- It waits only once at the command given at sleep

SetSpeed () : For specific amount of time it will stop the execution for every selenium command.

- It takes a single argument in integer format

Ex: selenium.setSpeed("2000")- It will wait for 2 seconds

- Runs each command after setSpeed delay by the number of milliseconds mentioned in set Speed

This command is useful for demonstration purpose or if you are using a slow web application

5) What is same origin policy? How you can avoid same origin policy?

The **"Same Origin Policy"** is introduced for security reason, and it ensures that content of your site will never be accessible by a script from another site. As per the policy, any code loaded within the browser can only operate within that website's domain.

To avoid "Same Origin Policy" proxy injection method is used, in proxy injection mode the Selenium Server acts as a client configured **HTTP proxy** , which sits between the browser and application under test and then masks the AUT under a fictional URL

WEEK-6

AIM:
Write a test case to perform automation on Ajio shopping website.

ALGORITHM:
1. Accessing shopping website
2. create an account for that Ajio shopping website.
3. Enter the fields through testcases information we have provided.
4. close the acivity after successfully created account.

SOURCE CODE:

```
package CMRIT.CMRIT_Assignment;

import java.util.Iterator;
import java.util.Set;

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

public class Ajio {

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

        System.setProperty("webdriver.chrome.driver","C:\\Users\\LENOVO\\Downloads\\chro
medriver.exe");
                WebDriver driver = new ChromeDriver();
                driver.get("https://www.ajio.com/");
                driver.manage().window().maximize();
                Thread.sleep(2000);
                WebElement ajioLink = driver.findElement(By.xpath("//span[normalize-
space()='Sign In / Join AJIO']"));
                ajioLink.click();
                Thread.sleep(2000);
                WebElement facebookBtn = driver.findElement(By.xpath("//span[normalize-
space()='Facebook']"));
                facebookBtn.click();
                Thread.sleep(2000);
                Set<String> parentWindow = driver.getWindowHandles();
                Iterator<String> iterator = parentWindow.iterator();
                while(iterator.hasNext())
                        {
                                String childWindow = iterator.next();
```

```java
                                if(!parentWindow.equals(childWindow))
                                {
                                        driver.switchTo().window(childWindow);
                                }
                        }
                WebElement emailOrMobileNo =
driver.findElement(By.xpath("//input[@id='email']"));
                emailOrMobileNo.sendKeys("martin@gmail.com");
                WebElement pwd = driver.findElement(By.xpath("//input[@id='pass']"));
                pwd.sendKeys("Martin@123");
                WebElement loginBtn = driver.findElement(By.cssSelector("body > div:nth-
child(2) > div:nth-child(2) > div:nth-child(2) > form:nth-child(1) > div:nth-child(4) > div:nth-
child(16) > label:nth-child(2) > input:nth-child(1)"));
                loginBtn.click();

        }

}
```

OUTPUT :
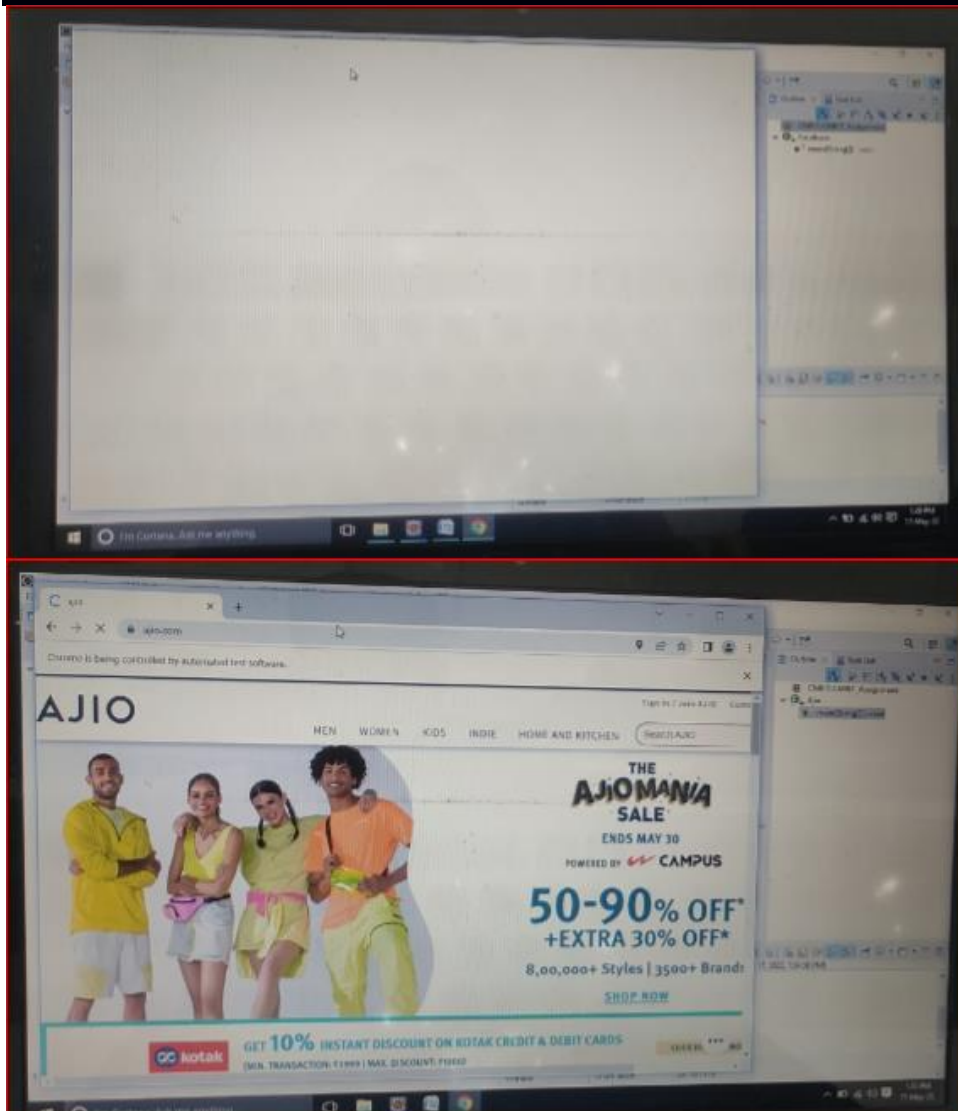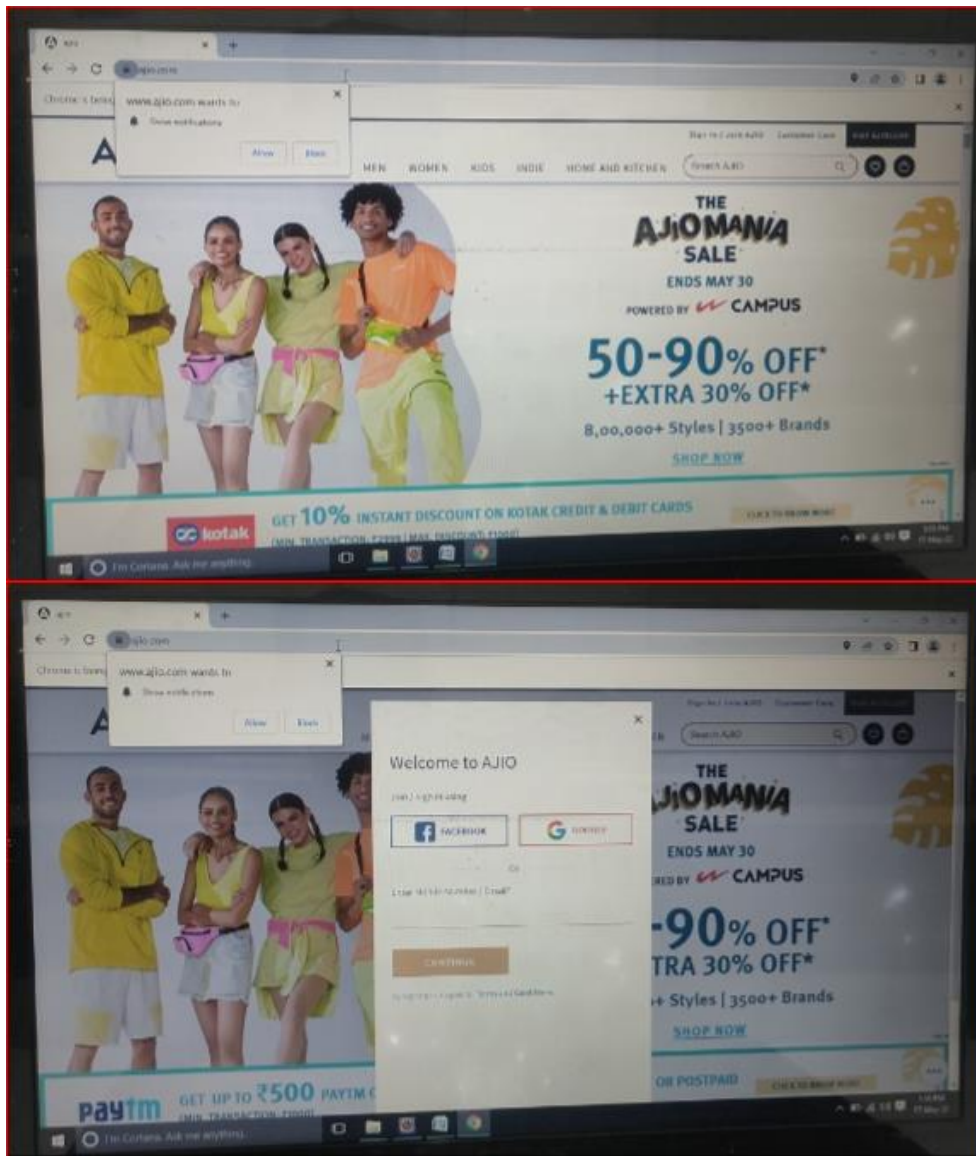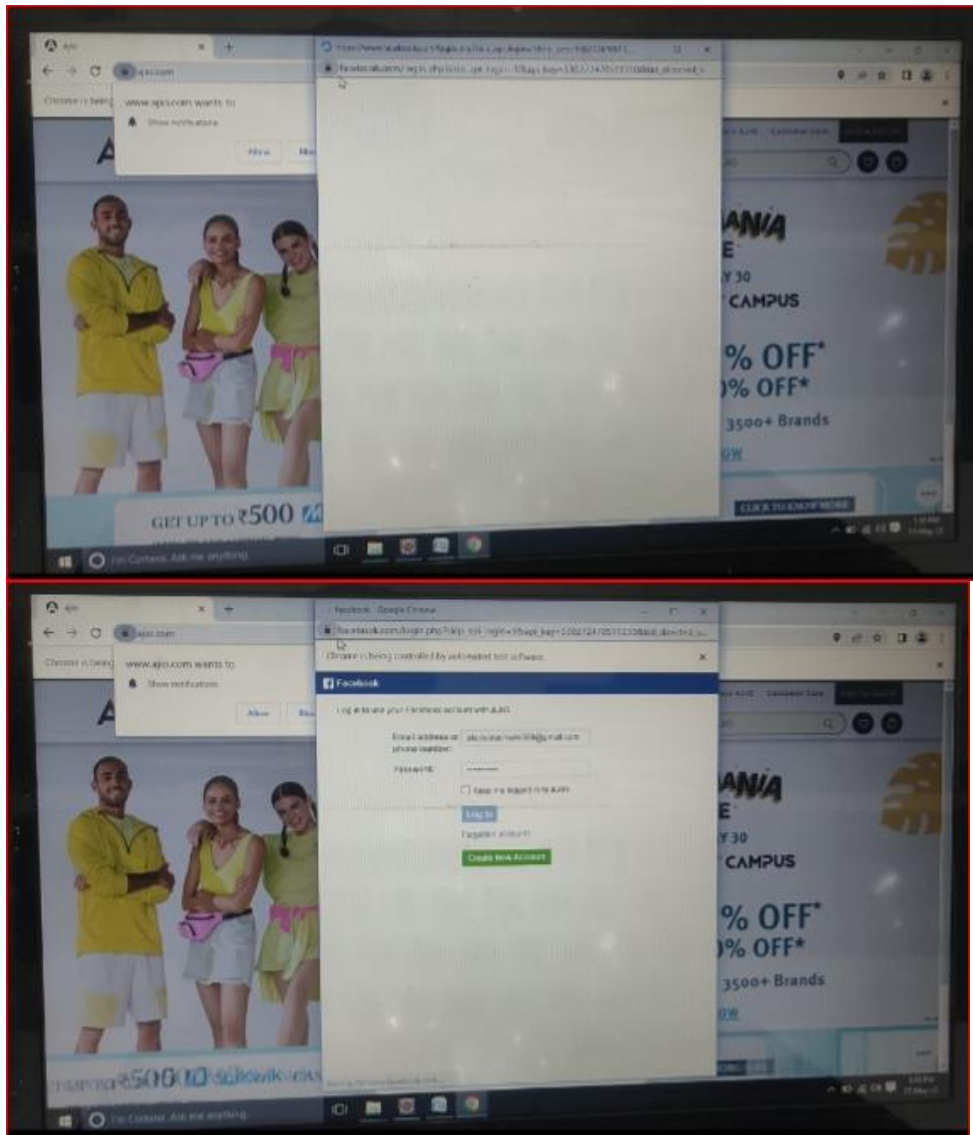
Viva Questions:

1) How you can use "submit" a form using Selenium ?

You can use "submit" method on element to submit form-

element.submit () ;

Alternatively you can use click method on the element which does form submission

2) What are the features of TestNG and list some of the functionality in TestNG which makes it more effective?

TestNG is a testing framework based on JUnit and NUnit to simplify a broad range of testing needs, from Unit Testing to Integration Testing. And the functionality which makes it efficient testing framework are

- Support for annotations
- Support for data-driven testing
- Flexible test configuration
- Ability to re-execute failed test cases

3) Explain what is the difference between find elements () and find element () ?

find element ():

It finds the first element within the current page using the given "locating mechanism". It returns a single WebElement

findElements () : Using the given "locating mechanism" find all the elements within the current page. It returns a list of web elements.

4) Which attribute you should consider throughout the script in frame for "if no frame Id as well as no frame name"?

You can use…..driver.findElements(By.xpath("//iframe"))….

This will return list of frames.

You will need to switch to each and every frame and search for locator which we want.

Then break the loop

5) Explain what are the JUnits annotation linked with Selenium?

The JUnits annotation linked with Selenium are

- @Before public void method() – It will perform the method () before each test, this method can prepare the test

- @Test public void method() – Annotations @Test identifies that this method is a test method environment
- @After public void method()- To execute a method before this annotation is used, test method must start with test@Before

WEEK-7

AIM:
Write a program in web driver to open Google and search CMRIT.

ALGORITHM:

1. Search google website
2. Fetching the google search bar
3. Search CMRIT
4. Exit from page.

SOURCE CODE:

```
package CMRIT.CMRIT_Assignment;

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

public class GoogleSearch_CMRIT {

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

        System.setProperty("webdriver.chrome.driver","C:\\Users\\LENOVO\\Downloads\\chro
medriver.exe");
                WebDriver driver = new ChromeDriver();
                driver.get("https://www.google.com/");
                driver.manage().window().maximize();
                Thread.sleep(2000);
                WebElement searchBar = driver.findElement(By.name("q"));
                searchBar.sendKeys("CMRIT");
                searchBar.sendKeys(Keys.ENTER);
                Thread.sleep(15000);
                driver.quit();

        }

}
```

OUTPUT:
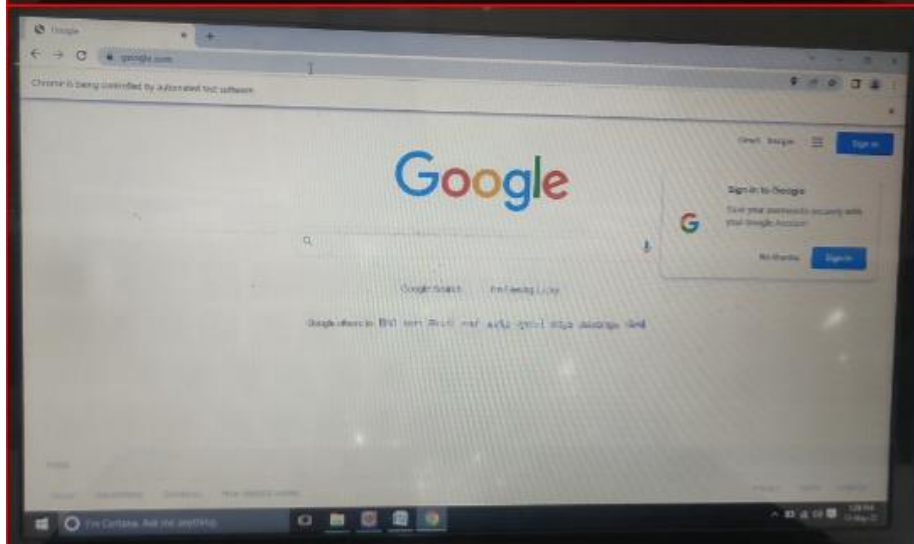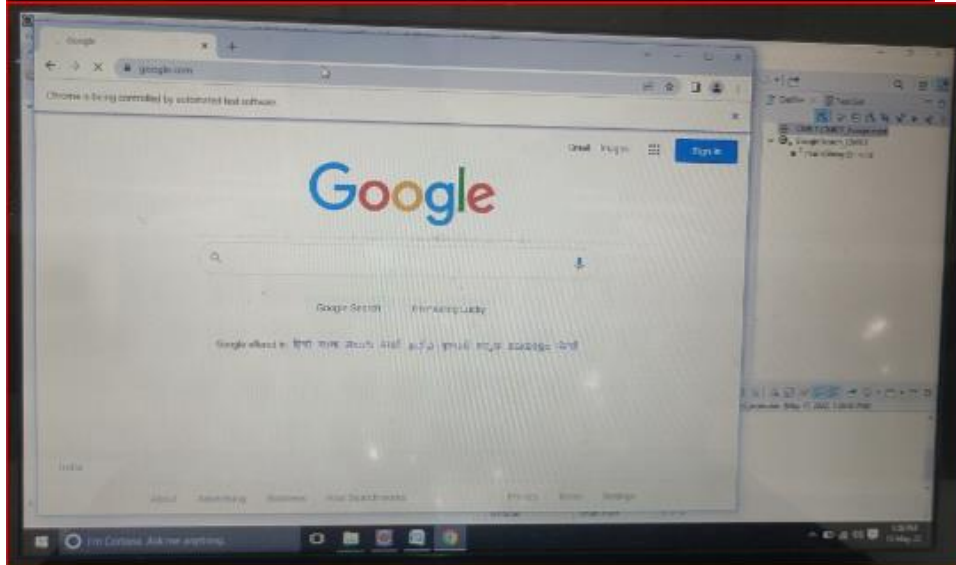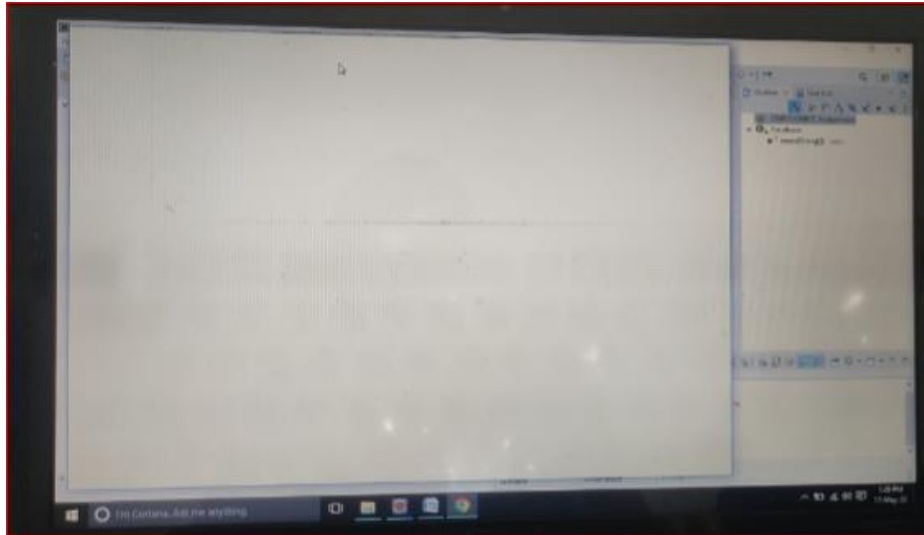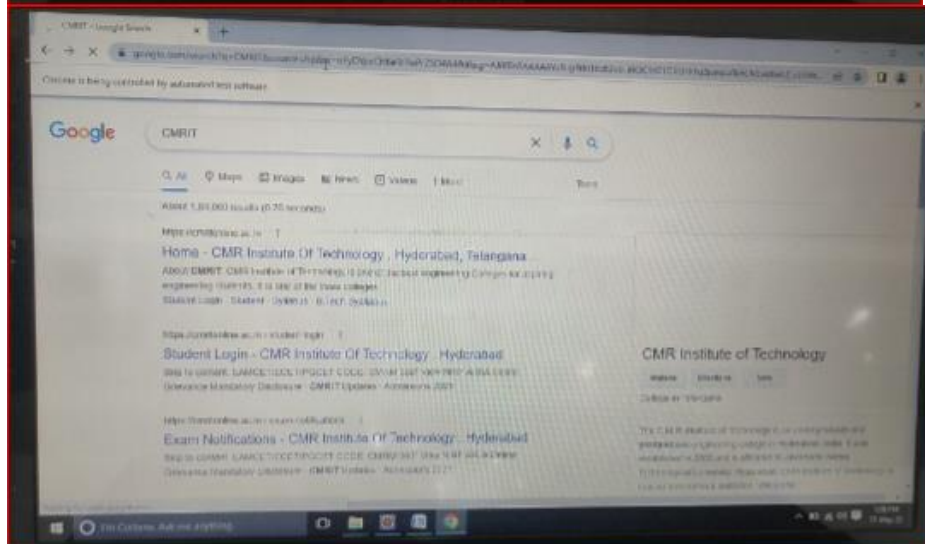
Viva Questions:

1) Explain what is Datadriven framework and Keyword driven?

Datadriven framework: In this framework, the test data is separated and kept outside the Test Scripts, while Test Case logic resides in Test Scripts. Test data is read from the external files (Excel Files) and are loaded into the variables inside the Test Script. Variables are used for both for input values and for verification values.

Keyworddriven framework: The keyword driven frameworks requires the development of data tables and keywords, independent of the test automation. In a keyword driven test, the functionality of the application under test is documented in a table as well as step by step instructions for each test.

2) Explain how you can login into any site if it's showing any authentication popup for password and username?

Pass the username and password with url

- Syntax-http://username:password@url
- ex- http://creyate:tom@www.gmail.com

3) Explain how to assert text of webpage using selenium 2.0 ?

WebElement el = driver.findElement(By.id("ElementID"))

//get test from element and stored in text variable

String text = el.getText();

//assert text from expected

Assert.assertEquals("Element Text", text);

4) Explain how Selenium Grid works?

Selenium Grid sent the tests to the hub. These tests are redirected to Selenium Webdriver, which launch the browser and run the test. With entire test suite, it allows for running tests in parallel.

5) Can we use Selenium grid for performance testing?

Yes. But not as effectively as a dedicated **Performance Testing** tool like Loadrunner.

WEEK- 8

AIM:

Write test case to open google and download a image from google images of cmrit website.

ALGORITHM:
1. search google website
2. Fetching the google searchBar
3. Sending CMRIT in searchbar
4. Opening the images page
5. Select one image and download that image into our system downloads.

SOURCE CODE:

```java
package CMRIT.CMRIT_Assignment;

import java.awt.Robot;
import java.awt.event.KeyEvent;

import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;

public class GoogleImageDownload {
        public static void main(String[] args) throws Exception {

        System.setProperty("webdriver.chrome.driver","C:\\Users\\LENOVO\\Downloads\\chromedriver.exe");
                WebDriver driver = new ChromeDriver();
                driver.manage().window().maximize();
                driver.get("https://www.google.com/");
                Thread.sleep(2000);
                WebElement searchBar = driver.findElement(By.name("q"));
                searchBar.sendKeys("CMRIT");
                searchBar.sendKeys(Keys.ENTER);
                Thread.sleep(2000);
                driver.findElement(By.xpath("//a[normalize-space()='Images']")).click();

                WebElement Image = driver.findElement(By.xpath("//img[@alt='The CMRIT Campus - CMR
IT']"));

                Actions action = new Actions(driver);
                action.contextClick(Image).build().perform();
                Robot robot = new Robot();
                robot.keyPress(KeyEvent.VK_DOWN);
                Thread.sleep(500);
                robot.keyPress(KeyEvent.VK_DOWN);
                Thread.sleep(500);
                robot.keyPress(KeyEvent.VK_DOWN);
                Thread.sleep(500);
                robot.keyPress(KeyEvent.VK_DOWN);
                Thread.sleep(500);
```

II-B.Tech. II-Semester   AUTOMATED TESTING TOOLS (SELENIUM) LAB MANUAL

58 | P a g e

```
                    robot.keyPress(KeyEvent.VK_DOWN);
                    Thread.sleep(500);
                    robot.keyPress(KeyEvent.VK_DOWN);
                    Thread.sleep(500);
                    robot.keyPress(KeyEvent.VK_DOWN);
                    Thread.sleep(500);
                    robot.keyPress(KeyEvent.VK_DOWN);
                    Thread.sleep(500);
                    robot.keyPress(KeyEvent.VK_ENTER);
                    Thread.sleep(3000);
                    robot.keyPress(KeyEvent.VK_ENTER);
                    System.out.println("downloaded");
                    driver.close();

        }
}
```

OUTPUT:
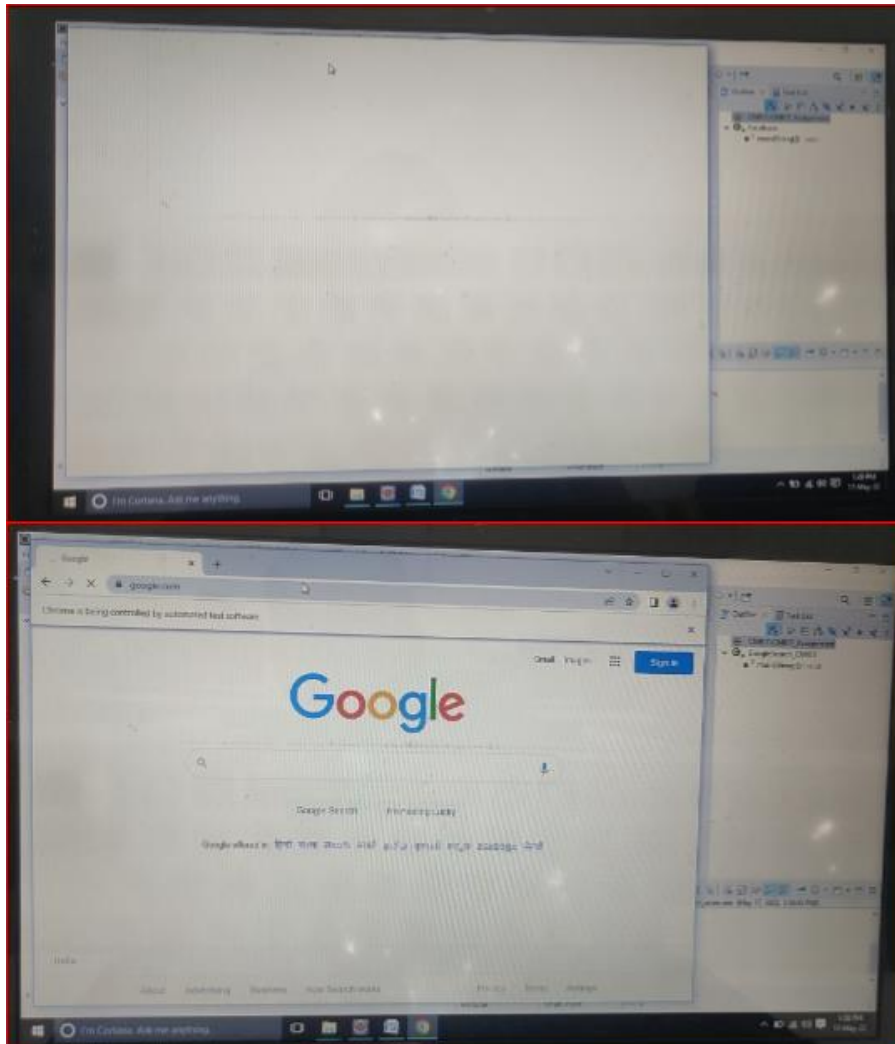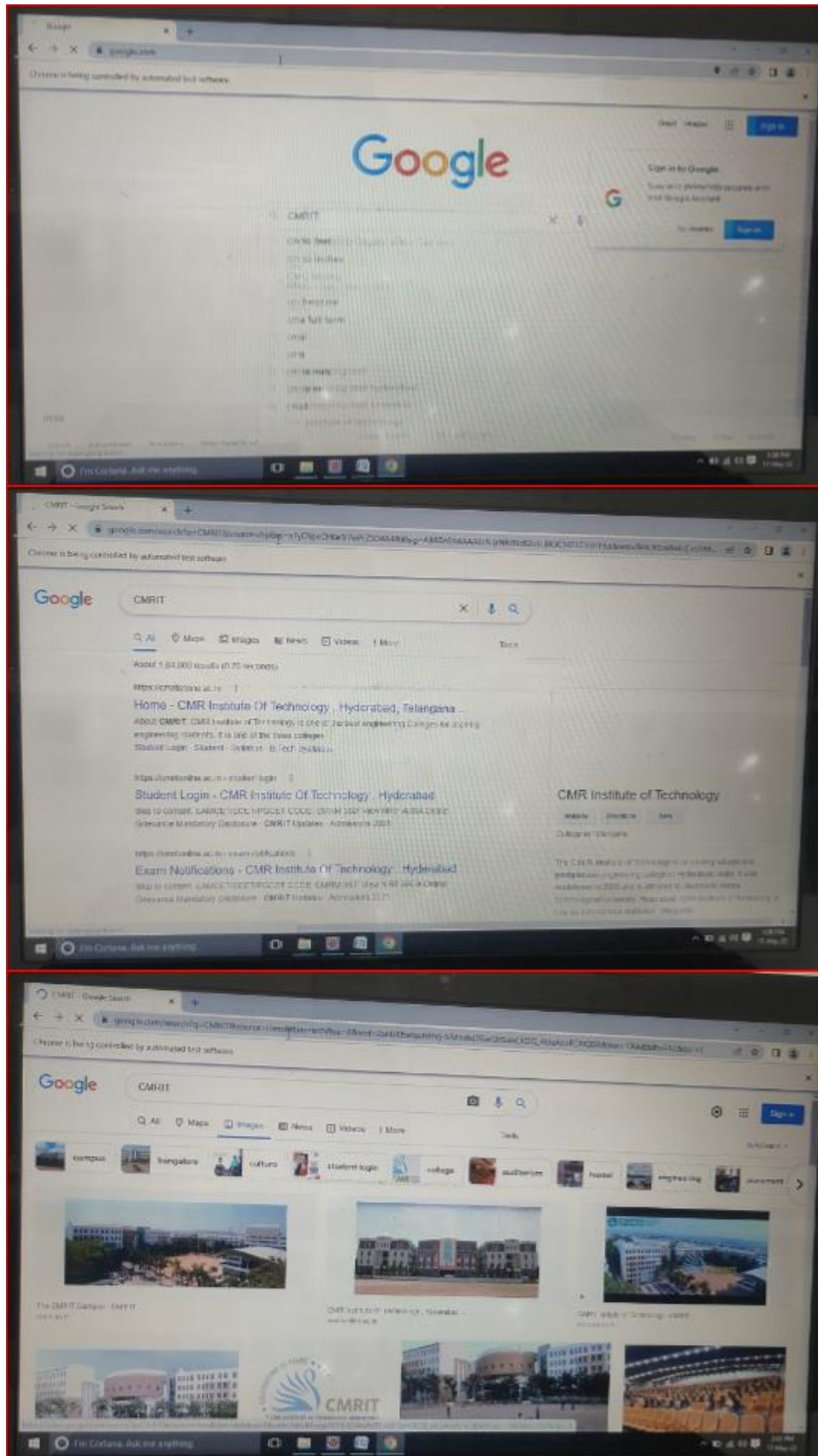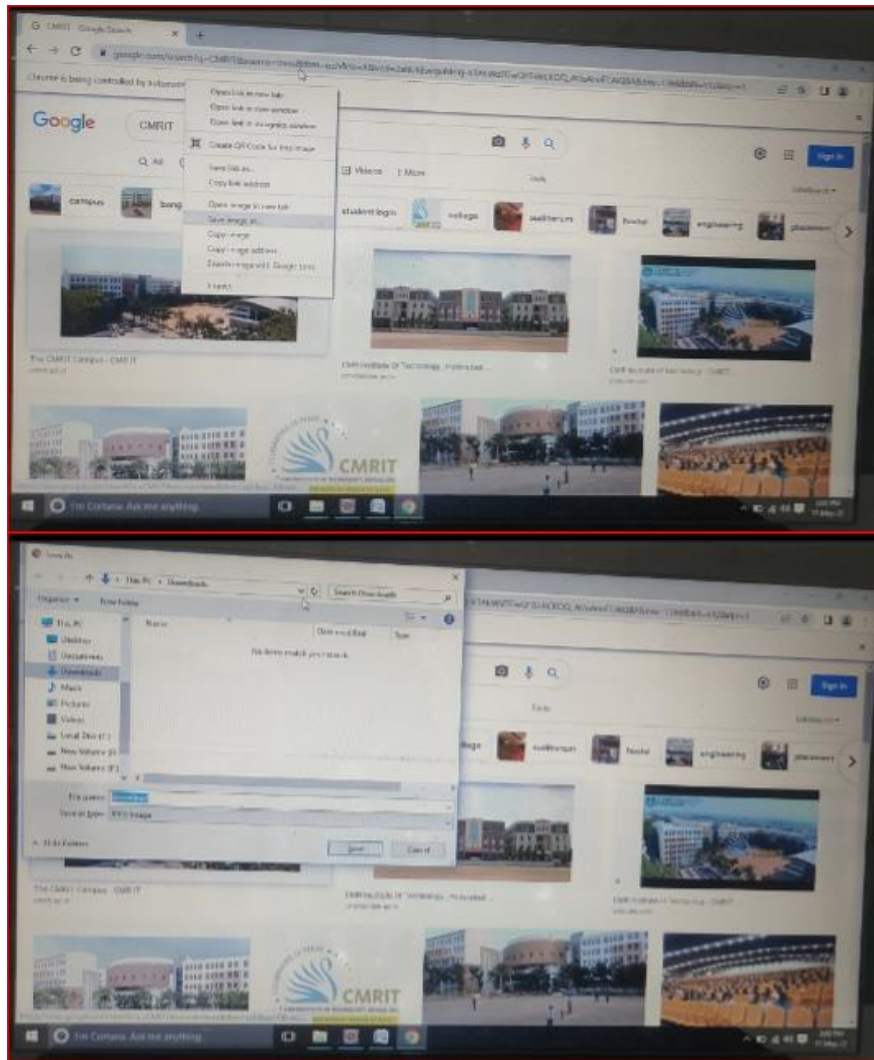
Viva Questions:

1) List the advantages of Webdriver over Selenium Server?

- If you are using Selenium-WebDriver, you don't need the Selenium Server as it is using totally different technology
- Selenium Server provides Selenium RC functionality which is used for Selenium 1.0 backwards compatibility
- Selenium Web driver makes direct calls to browser using each browsers native support for automation, while Selenium RC requires selenium server to inject Javascript into the browser

2) Mention what are the capabilities of Selenium WebDriver or Selenium 2.0 ?

WebDriver should be used when requiring improvement support for

- Handling multiple frames, pop ups , multiple browser windows and alerts
- Page navigation and drag & drop
- Ajax based UI elements
- Multi browser testing including improved functionality for browser not well supported by Selenium 1.0

3) Explain how you can find broken images in a page using Selenium Web driver ?

To find the broken images in a page using Selenium web driver is

- Get XPath and get all the links in the page using tag name
- In the page click on each and every link
- Look for 404/500 in the target page title

4) Explain how you can handle colors in web driver?

To handle colors in web driver you can use

Use getCssValue(arg0) function to get the colors by sending 'color' string as an argument

5) Using web driver how you can store a value which is text box?

You can use following command to store a value which is text box using web driver

driver.findElement(By.id("your Textbox")).sendKeys("your keyword");

WEEK-9

AIM: Write test case to get number of list items in a list.

ALGORITHM:

1. Search google website.
2. Fetching google searchbar
3. Sending JUSTDIAL in searchbar
4. Automatically out IDE reads the List Of Items and shown in our console IDE.

SOURCE CODE:

```
package CMRIT.CMRIT_Assignment;

import java.util.List;
import java.util.concurrent.TimeUnit;

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

public class listItems
{
        public static void main(String[] args)
        {

        System.setProperty("webdriver.chrome.driver","C:\\Users\\LENOVO\\Downloads\\chro
medriver.exe");
                WebDriver driver = new ChromeDriver();
                driver.manage().window().maximize();
                driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
            //URL launch
            driver.get("https://www.justdial.com/Bangalore/Bakeries");
            // identify elements list with same class nam5e
            List<WebElement> m = driver.findElements(By.xpath("//h2[@class='store-name']"));
            // iterate over list
            for(int i = 0; i< m.size(); i++) {
              //obtain text
              String s = m.get(i).getText();
              System.out.println("Text is: " + s);
            }
            driver.quit();
        }
}
```

II-B.Tech. II-Semester   AUTOMATED TESTING TOOLS (SELENIUM) LAB MANUAL

63 | P a g e

OUTPUT:
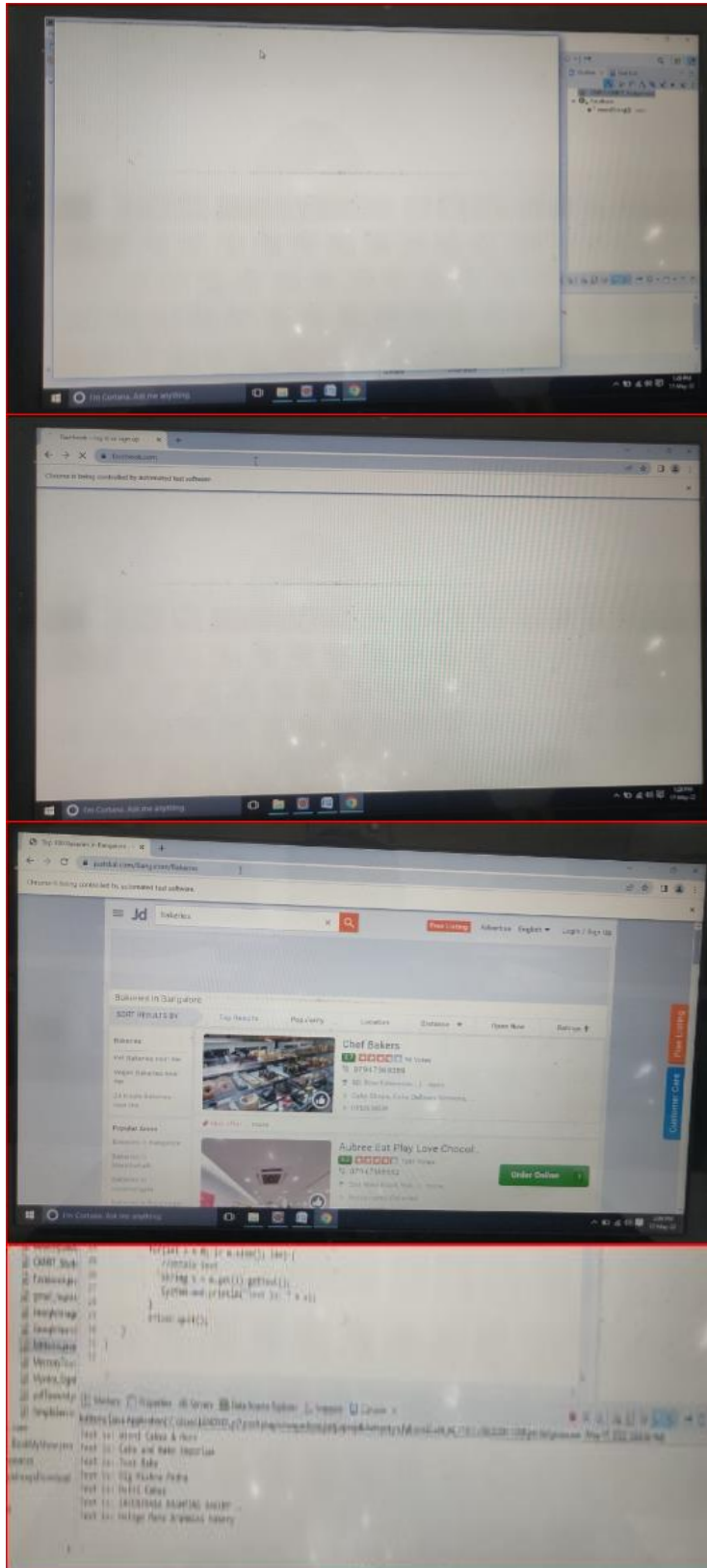
Viva Questions:

1) **Explain how you can switch between frames?**

To switch between frames webdrivers **[driver.switchTo().frame()]** method takes one of the three possible arguments

- A number: It selects the number by its (zero-based) index
- A name or ID: Select a frame by its name or ID
- Previously found WebElement: Using its previously located WebElement select a frame

2) **Mention 5 different exceptions you had in Selenium web driver?**

The 5 different exceptions you had in Selenium web drivers are

- WebDriverException
- NoAlertPresentException
- NoSuchWindowException
- NoSuchElementException
- TimeoutException

3) **Explain using Webdriver how you can perform double click ?**

You can perform double click by using

- **Syntax- Actions act = new Actions (driver);**
- **act.doubleClick(webelement);**

4) **How will you use Selenium to upload a file ?**

You can use "type"command to type in a file input box of upload file. Then, you have to use "Robot" class in JAVA to make file upload work.

5) **Explain how you can handle frames using Selenium 2.0 ?**

To bring control on HTML frame you can use "SwitchTo" frame method-

driver.switchTo().frame("frameName");

To specify a frame you can use index number

driver.switchTo().frame("parentFrame.4.frameName");

This would bring control on frame named- "frameName" of the 4th sub frame names "parentFrame"

WEEK-10

AIM: Write test case for validation in Gmail registration page.

ALGORITHM:

1. search google website.
2. find Gmail website
3. fetching gmail link from google search
4. create a new account
5. fill the fields and click on login button
6. exit from page.

SOURCE CODE:

```java
package CMRIT.CMRIT_Assignment;

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.Select;

public class gmail_register {

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

System.setProperty("webdriver.chrome.driver","C:\\Users\\LENOVO\\Downloads\\chromedriver.exe");
                WebDriver driver = new ChromeDriver();
                driver.get("http://gmail.com/");
                driver.manage().window().maximize();
                Thread.sleep(2000);
                WebElement createAccount = driver.findElement(By.xpath("(//span[normalize-space()='Create
account'])[1]"));
                createAccount.click();
                WebElement mySelft = driver.findElement(By.xpath("(//span[normalize-space()='For
myself'])[1]"));
                mySelft.click();

                WebElement firstName = driver.findElement(By.name("firstName"));
                firstName.sendKeys("abcd");

                WebElement lastName = driver.findElement(By.name("lastName"));
                lastName.sendKeys("efgh");

                WebElement username = driver.findElement(By.name("Username"));
                username.sendKeys("minnukittusuhas123");

                WebElement password = driver.findElement(By.name("Passwd"));
                password.sendKeys("zxcasd@123");
```

```java
                WebElement confirmPassword = driver.findElement(By.name("ConfirmPasswd"));
                confirmPassword.sendKeys("zxcasd@123");

                WebElement nxtButton = driver.findElement(By.xpath("(//span[normalize-space()='Next'])[1]"));
                nxtButton.click();

                Thread.sleep(3000);
                WebElement phoneNumber =
driver.findElement(By.xpath("(//input[@id='phoneNumberId'])[1]"));
                phoneNumber.sendKeys("9876534538");

                /*
                 * Select month = new
                 * Select(driver.findElement(By.xpath("(//select[@id='month'])[1]")));
                 * month.selectByValue("8");
                 *
                 *
                 * WebElement day = driver.findElement(By.xpath("(//input[@id='day'])[1]"));
                 * day.sendKeys("24");
                 *
                 * WebElement year = driver.findElement(By.xpath("(//input[@id='year'])[1]"));
                 * year.sendKeys("1996");
                 *
                 * Select gender = new
                 * Select(driver.findElement(By.xpath("(//select[@id='gender'])[1]")));
                 * gender.selectByValue("1");
                 */

                WebElement nxtButton2 = driver.findElement(By.xpath("(//span[normalize-space()='Next'])[1]"));
                nxtButton2.click();

        }

}
```
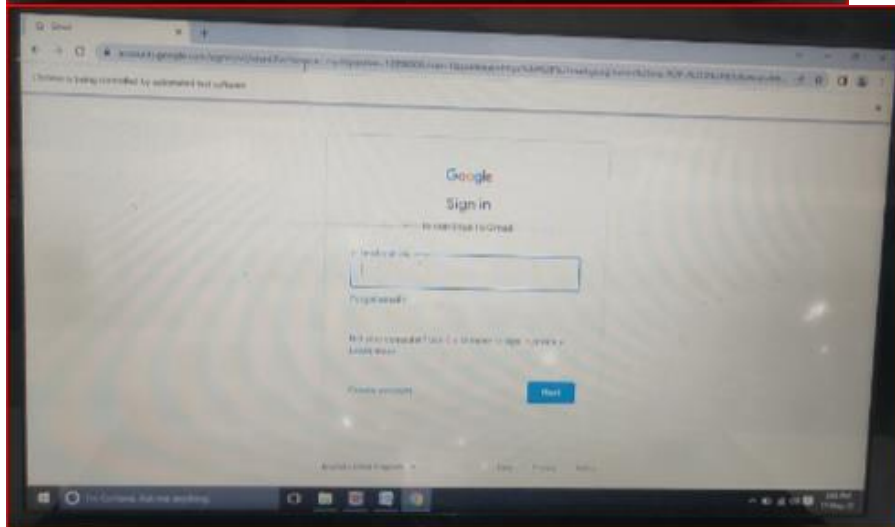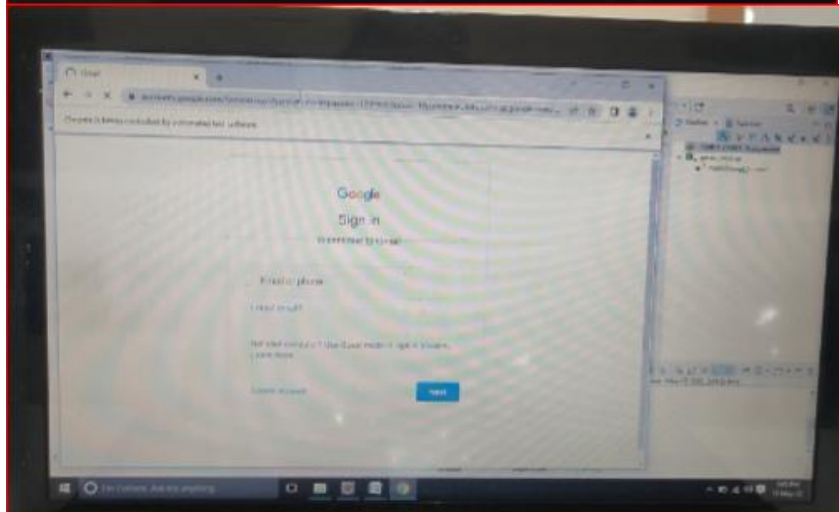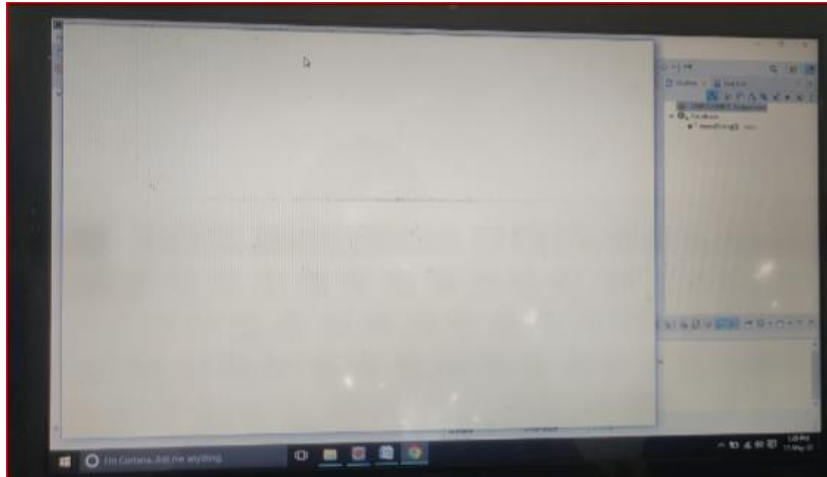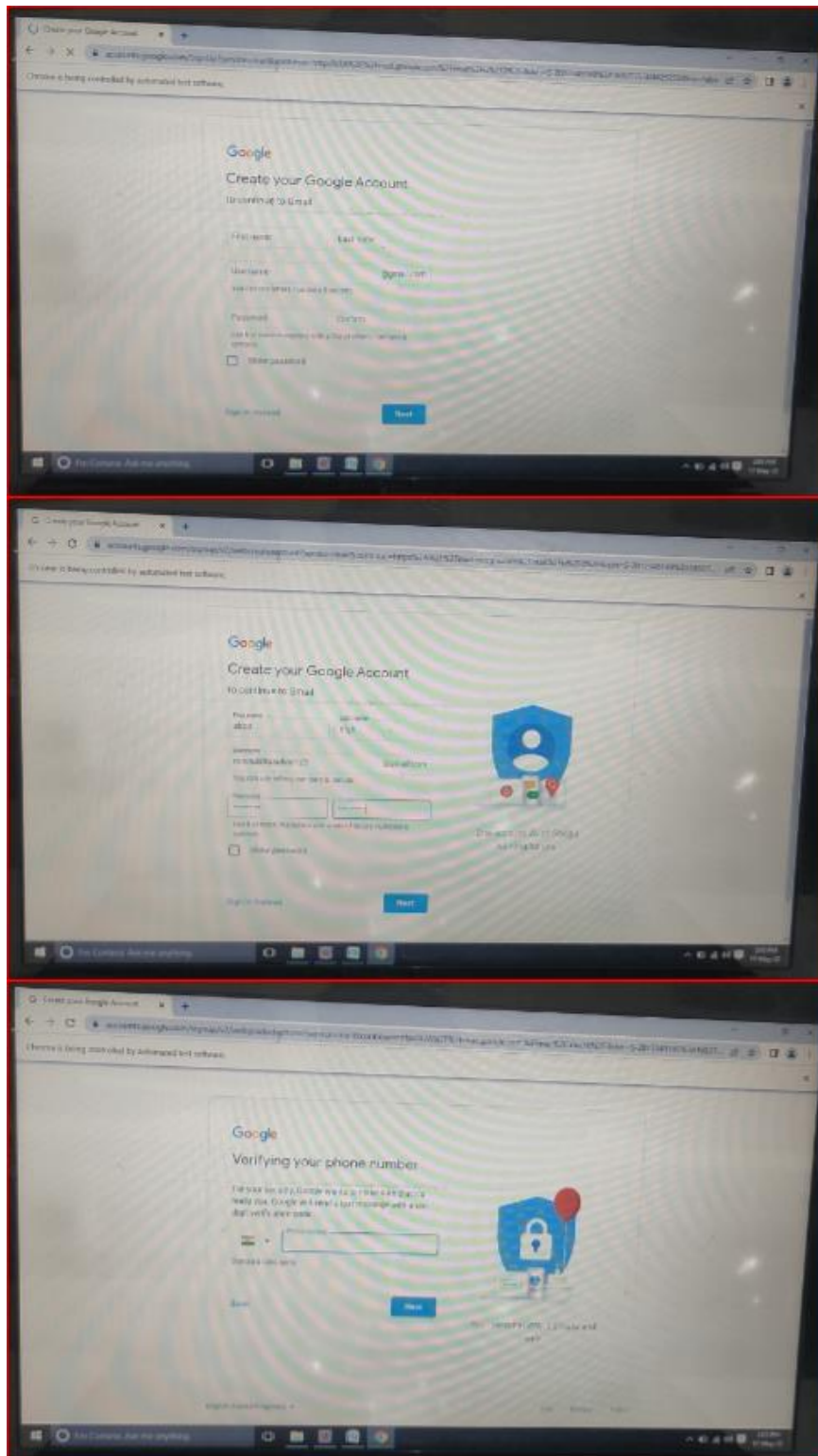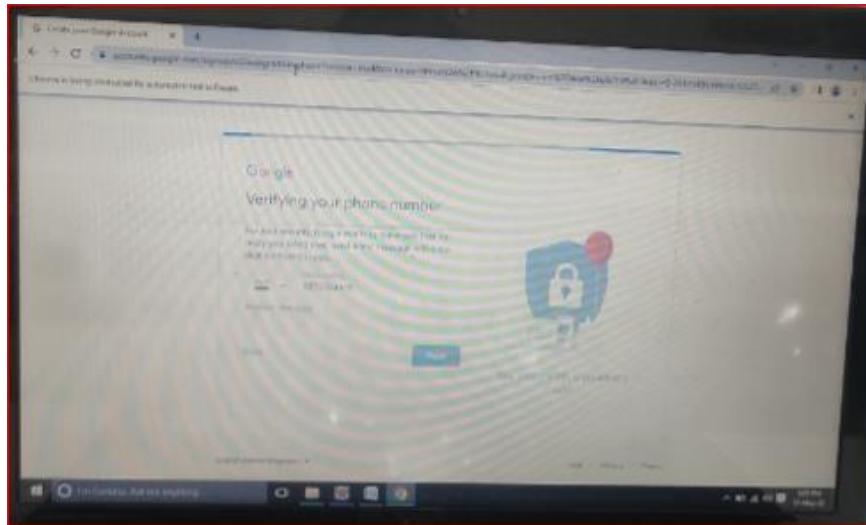
II-B.Tech. II-Semester   AUTOMATED TESTING TOOLS (SELENIUM) LAB MANUAL

67 | P a g e

OUTPUT:

Viva Questions:

1) **Explain how you can switch back from a frame?**

To switch back from a frame use method defaultContent()

Syntax-driver.switchTo().defaultContent();

2) **List out different types of locators?**

Different types of locators are

- By.id()
- By.name()
- By.tagName()
- By.className()
- By.linkText()
- By.partialLinkText()
- By.xpath
- By.cssSelector()

3) **What is the command that is used in order to display the values of a variable into the output console or log?**

- In order to display a constant string, command can be used is echo <constant string>
- If order to display the value of a variable you can use command like echo ${variable name>>

Above is using PHP. If you are using Java, replace echo with System.out.println

4) **Explain how you can use recovery scenario with Selenium?**

Recovery scenarios depends upon the programming language you use. If you are using Java then you can use exception handling to overcome same. By using "Try Catch Block" within your Selenium WebDriver Java tests

5) **Explain what are the limitations of Selenium IDE?**

The limitations of Selenium IDE

- Exceptional handling is not present
- Selenium IDE uses only HTML languages
- External databases reading is not possible with IDE
- Reading from the external files like .txt, .xls is not possible
- Conditional or branching statements execution like if,else, select statements is not possible

WEEK-11

AIM: Write test case for myntra sign in page.

ALGORITHM:

1. Search google website
2. Fetching the google search bar
3. Sending myntra in search bar
4. Submitting the data
5. Fetching myntralink from google search
6. Fill the fields and click on login button
7. Exit from page.

SOURCE CODE:

```
package CMRIT.CMRIT_Assignment;

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

public class Myntra_SignIn {

        public static void main(String[] args) throws InterruptedException
        {
                // System property for chrome driver

        System.setProperty("webdriver.chrome.driver","C:\\Users\\LENOVO\\Downloads\\chro
medriver.exe");
                        //Using driver ref variable creating new obj for chromedriver
                        WebDriver driver = new ChromeDriver();

                        //Maximizing the chrome browser
                        driver.manage().window().maximize();

                        //MouseActions
                        //Actions builder = new Actions(driver);

                        // Launch google website
                        driver.get("https://www.google.co.in/");
```

```
//Fetching the google searchBar
WebElement searchBar = driver.findElement(By.name("q"));

//Sending myntra in searchbar
searchBar.sendKeys("Myntra");
Thread.sleep(1000);

//Submitting the data
searchBar.submit();

//Fetching myntralink from google search
WebElement myntraLink =
driver.findElement(By.xpath("(//h3[contains(text(),'Myntra: Online Shopping for Women, Men,
Kids Fashi')])[1]"));
myntraLink.click();

WebElement profileHyperLink =
driver.findElement(By.xpath("(//span[normalize-space()='Profile'])[1]"));
profileHyperLink.click();

WebElement loginBtn =
driver.findElement(By.xpath("(//a[normalize-space()='login / Signup'])[1]"));
loginBtn.click();

WebElement mobileNo =
driver.findElement(By.xpath("(//input[@type='tel'])[1]"));
mobileNo.sendKeys("9948621019");

WebElement continueBtn =
driver.findElement(By.xpath("(//div[@class='submitBottomOption'])[1]"));
continueBtn.click();
Thread.sleep(1000);

WebElement otp01 =
driver.findElement(By.xpath("(//input[@name='otp0'])[1]"));
otp01.sendKeys("1");

WebElement otp02 =
driver.findElement(By.xpath("(//input[@name='otp1'])[1]"));
otp02.sendKeys("2");

WebElement otp03 =
driver.findElement(By.xpath("(//input[@name='otp2'])[1]"));
otp03.sendKeys("3");
```
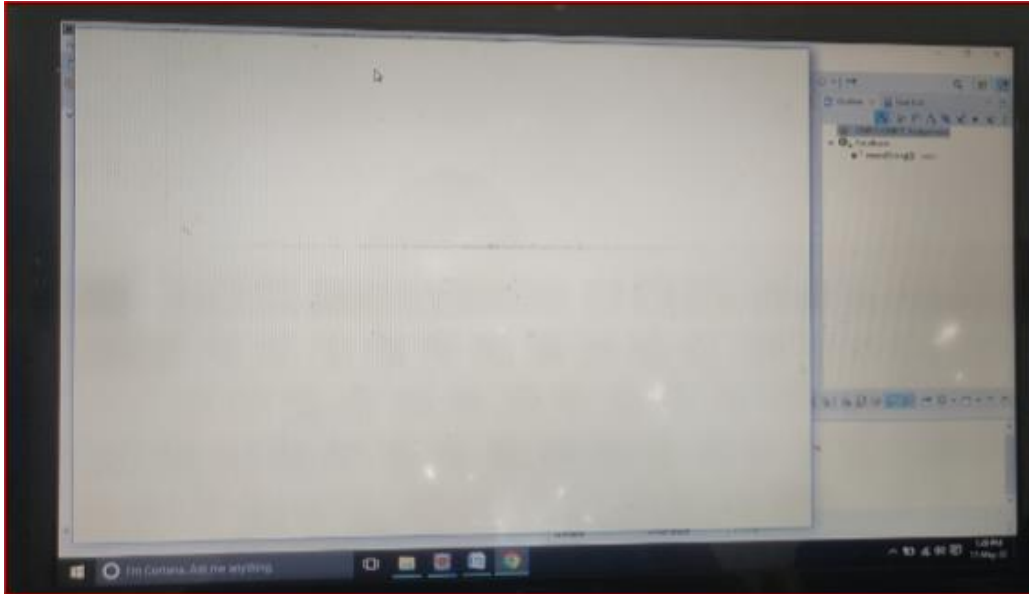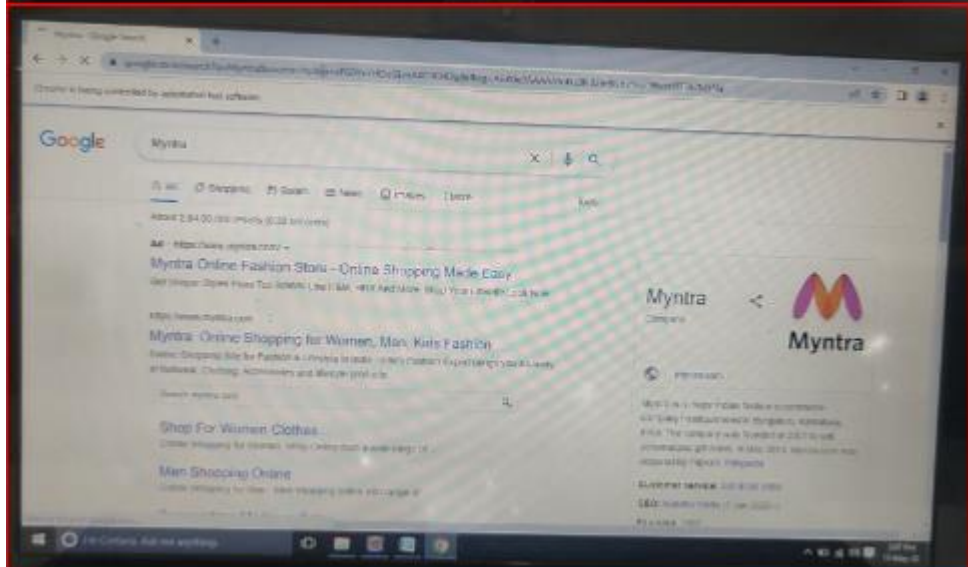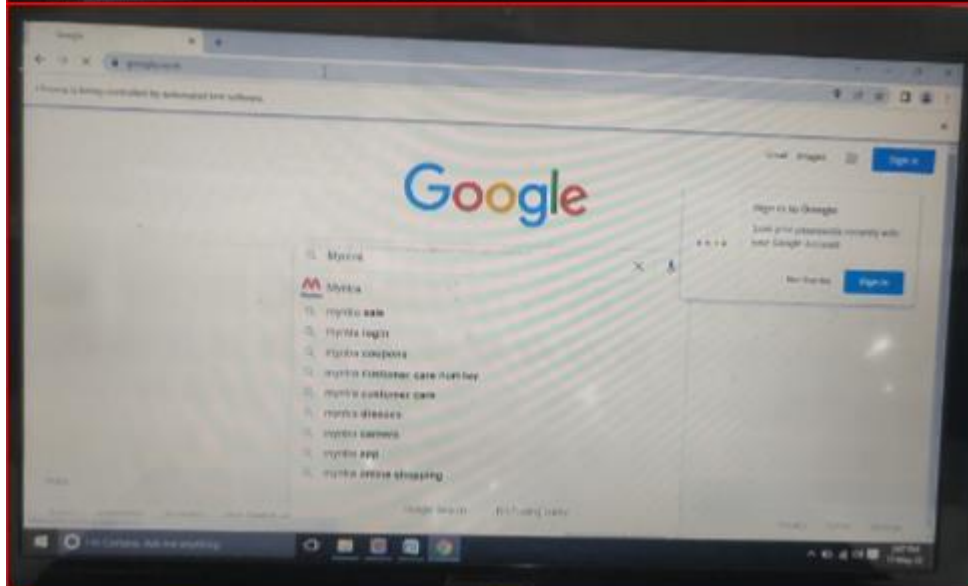
```
                WebElement otp04 =
driver.findElement(By.xpath("(//input[@name='otp3'])[1]"));
                otp04.sendKeys("4");

                driver.close();
        }
}
```
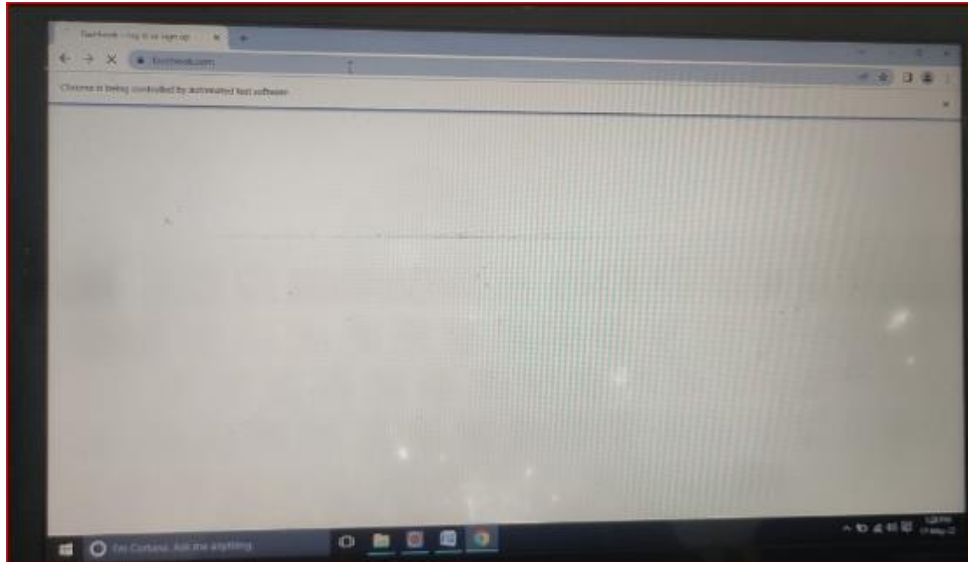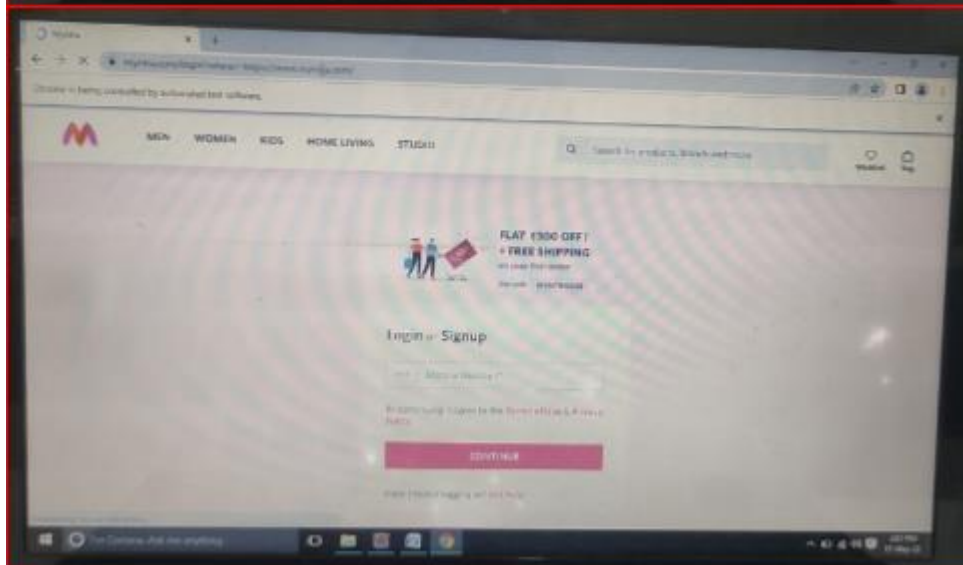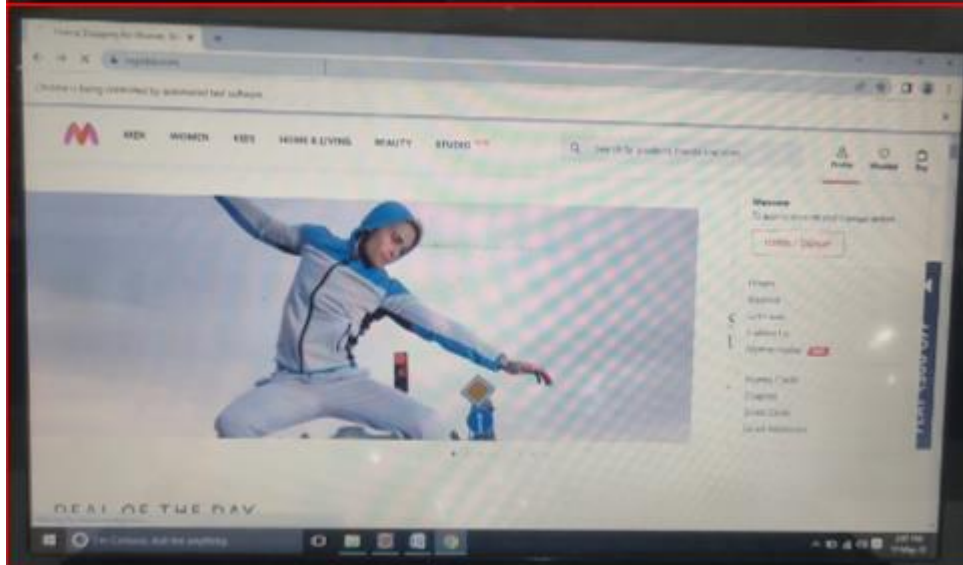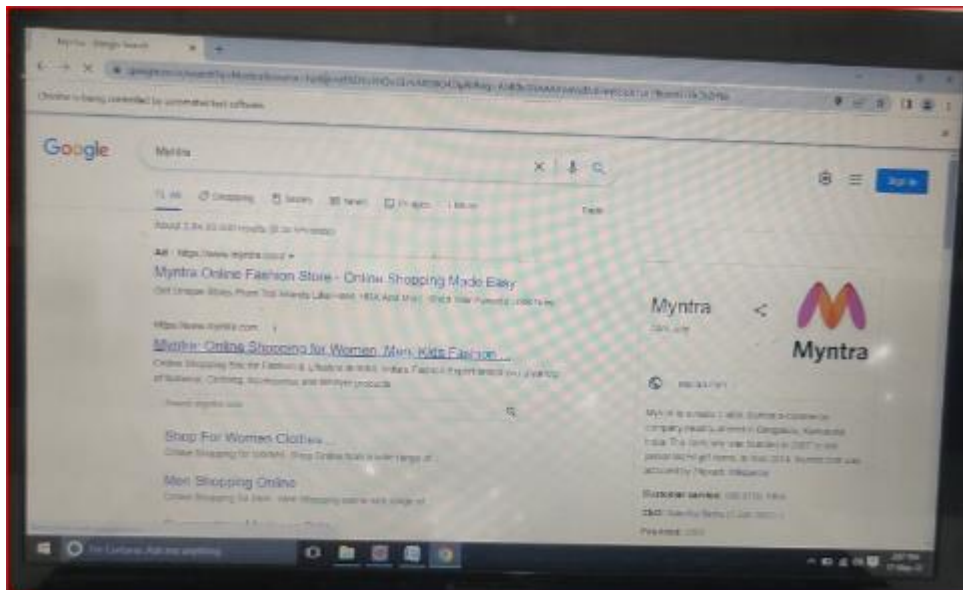
OUTPUT:

Viva Questions:

1) In selenium IDE what are the element locators that can be used to locate elements on web page?

In selenium there are mainly 4 locators that are used

- X-path locators
- Css locators
- Html id
- Html name

2) From Selenium IDE how you can execute a single line?

From Selenium IDE single line command can be executed in two ways

- Select "Execute this command" by right clicking on the command in Selenium IDE
- Press "X" key on the keyboard after selecting the command in Selenium IDE

3) In which format does source view shows your script in Selenium IDE ?

In Selenium IDE source view shows your script in XML format

4) What is selenium RC (Remote Control)?

Selenium IDE have limitations in terms of browser support and language support. By using Selenium RC limitation can be diminished.

- On different platforms and different web browser for automating web application se-lenium RC is used with languages like Java, C#, Perl, Python
- Selenium RC is a java based and using any language it can interact with the web ap-plication
- Using server you can bypass the restriction and run your automation script running against any web application

5) Why Selenium RC is used?

Selenium IDE does not directly support many functions like condition statements, Iteration, logging and reporting of test results, unexpected error handling and so on as IDE supports only HTML language. To handle such issues Selenium RC is used it supports the language like Perl, Ruby, Python, PHP using these languages we can write the program to achieve the IDE issues.

WEEK-12

AIM: Write test case to convert PDF from word.

ALGORITHM:

1. Write test case for finding google search
2. Then find element to search convert pdf from word.
3. Here write test case statement to upload a pdf file.
4. Now include test case for converting file word
5. Then add test case for download element to download word file into your system.

SOURCE CODE:

```
package CMRIT.CMRIT_Assignment;


import java.awt.AWTException;
import java.awt.Robot;
import java.awt.Toolkit;
import java.awt.datatransfer.Clipboard;
import java.awt.datatransfer.StringSelection;
import java.awt.event.KeyEvent;

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

public class pdfToword
{
        public static void main(String[] args) throws Exception
        {

        System.setProperty("webdriver.chrome.driver","C:\\Users\\LENOVO\\Downloads\\chro
medriver.exe");
                WebDriver driver = new ChromeDriver();
                driver.manage().window().maximize();

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

                Thread.sleep(2000);
                WebElement searchBar = driver.findElement(By.name("q"));
                searchBar.sendKeys("Convert pdf to word online");
                searchBar.sendKeys(Keys.ENTER);
```
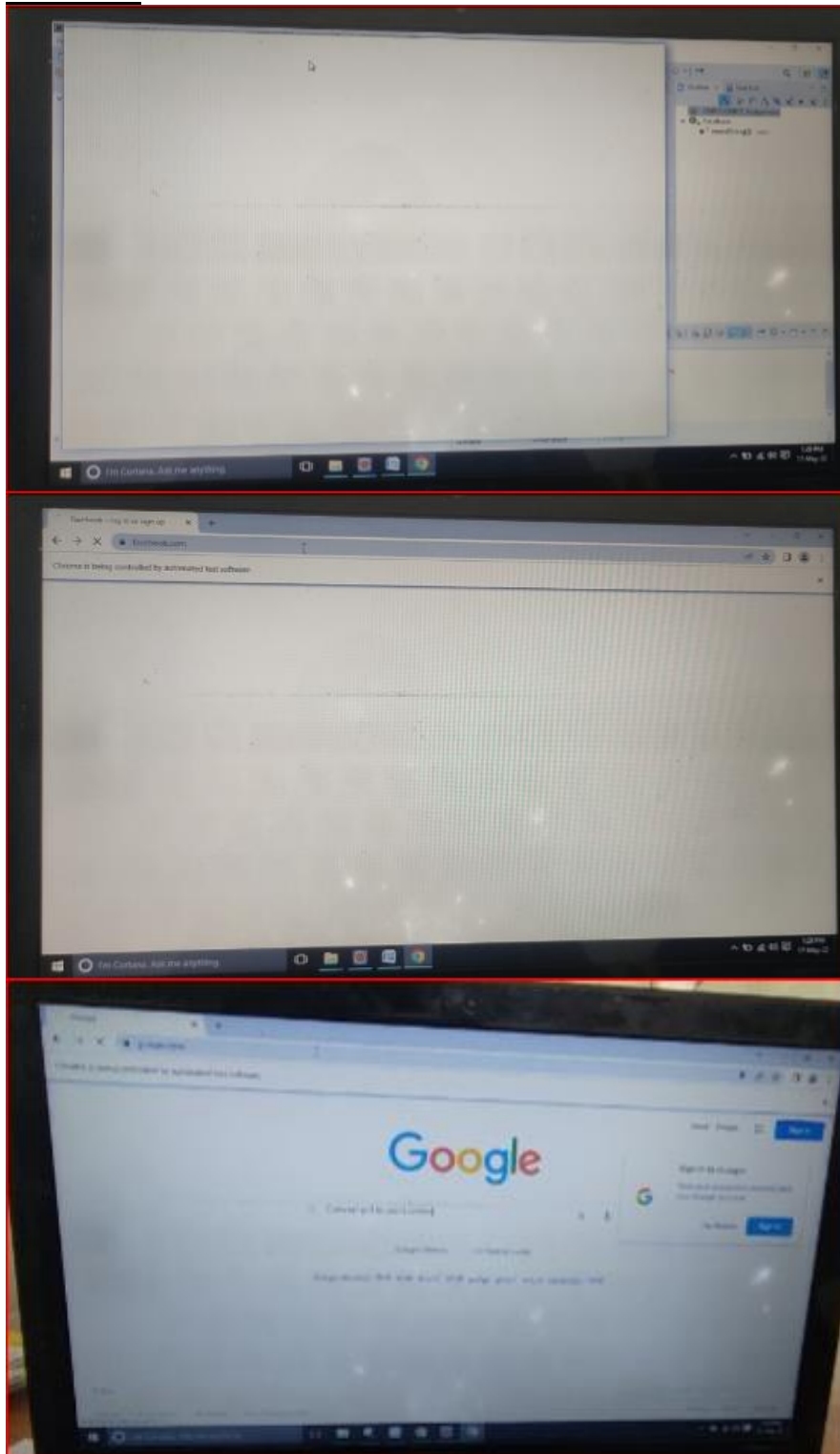
```java
                WebElement pdfToWordHyperLink = driver.findElement(
                                By.xpath("(//h3[normalize-space()='PDF to Word Converter -
100% Free - Smallpdf.com'])[1]"));
                pdfToWordHyperLink.click();
                Thread.sleep(2000);
                WebElement popupClose = driver.findElement(By.xpath("(//span[normalize-
space()='Got it'])[1]"));
                popupClose.click();


                WebElement chooseFileBtn = driver.findElement(By.xpath("(//span[normalize-
space()='Choose Files'])[1]"));
                //chooseFileBtn.sendKeys("C:\\Users\\user\\Desktop\\maneesh
docs\\ManeeshResume.pdf");
                chooseFileBtn.click();


                 Clipboard clipboard = Toolkit.getDefaultToolkit().getSystemClipboard();
                    StringSelection str = new
StringSelection("C:\\Users\\LENOVO\\Documents\\sample.pdf");
                    clipboard.setContents(str, null);

                    Robot robot = new Robot();
                    robot.keyPress(KeyEvent.VK_ENTER);
                    robot.keyRelease(KeyEvent.VK_ENTER);
                    robot.keyPress(KeyEvent.VK_CONTROL);
                    robot.keyPress(KeyEvent.VK_V);
                    robot.keyRelease(KeyEvent.VK_V);
                    robot.keyRelease(KeyEvent.VK_CONTROL);
                    robot.keyPress(KeyEvent.VK_ENTER);
                    robot.keyRelease(KeyEvent.VK_ENTER);
                    Thread.sleep(5000);
                    WebElement convertToWord = driver.findElement(By.xpath("(//div)[197]"));
                    convertToWord.click();
                    WebElement choosePlan =
driver.findElement(By.xpath("(//button[@type='submit'])[2]"));
                    choosePlan.click();
                    Thread.sleep(7000);
                    WebElement download = driver.findElement(By.xpath("(//div)[212]"));
                    download.click();
        }
}
```
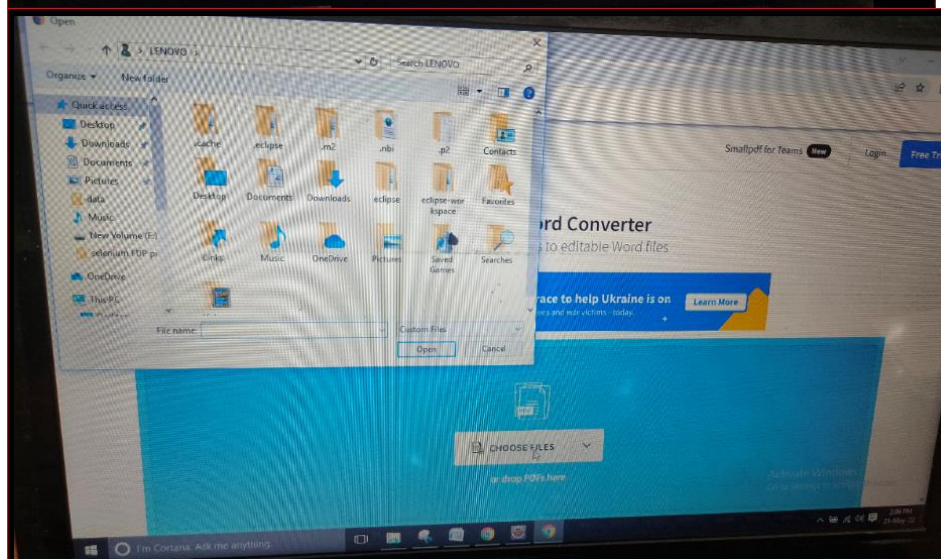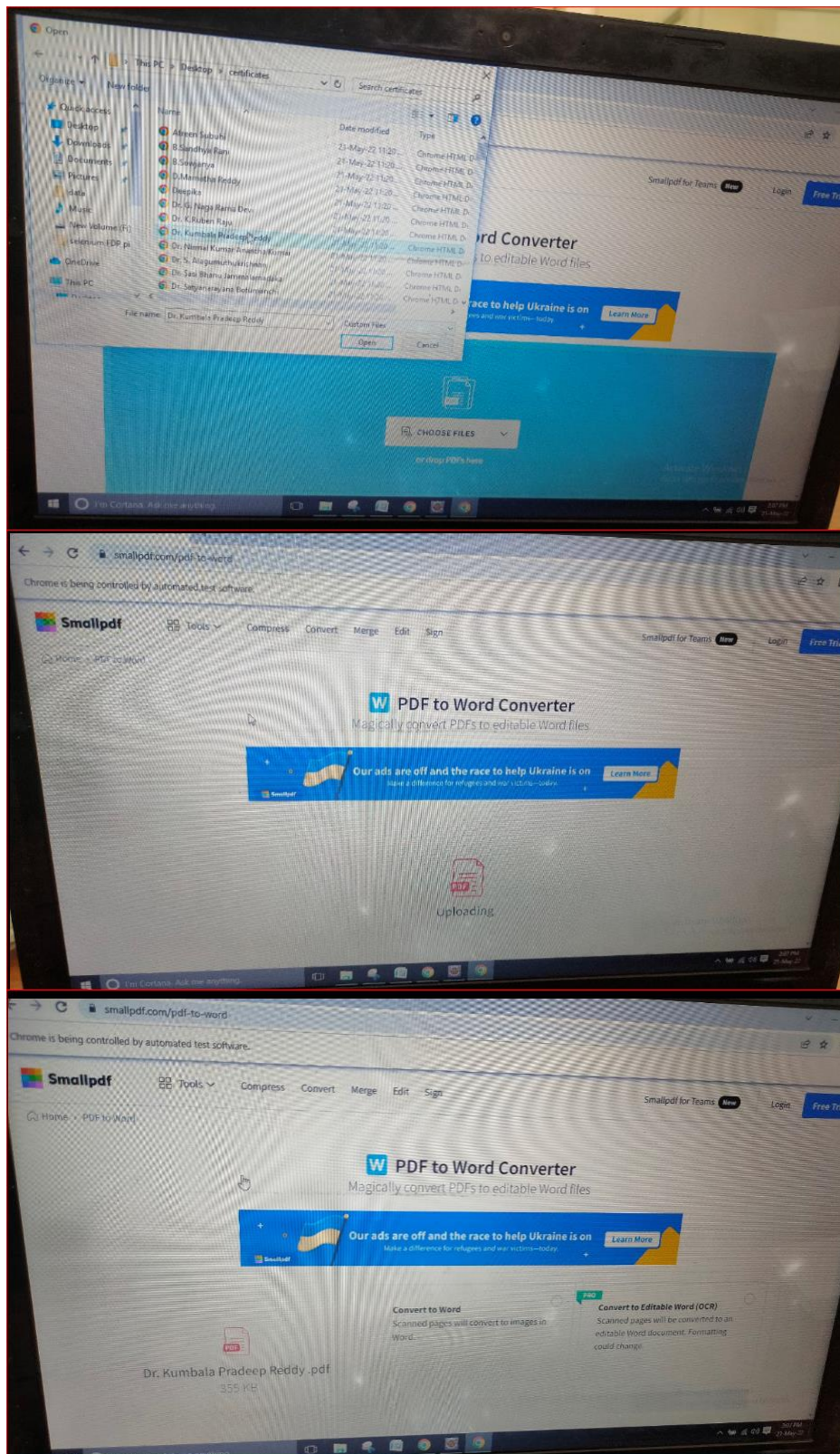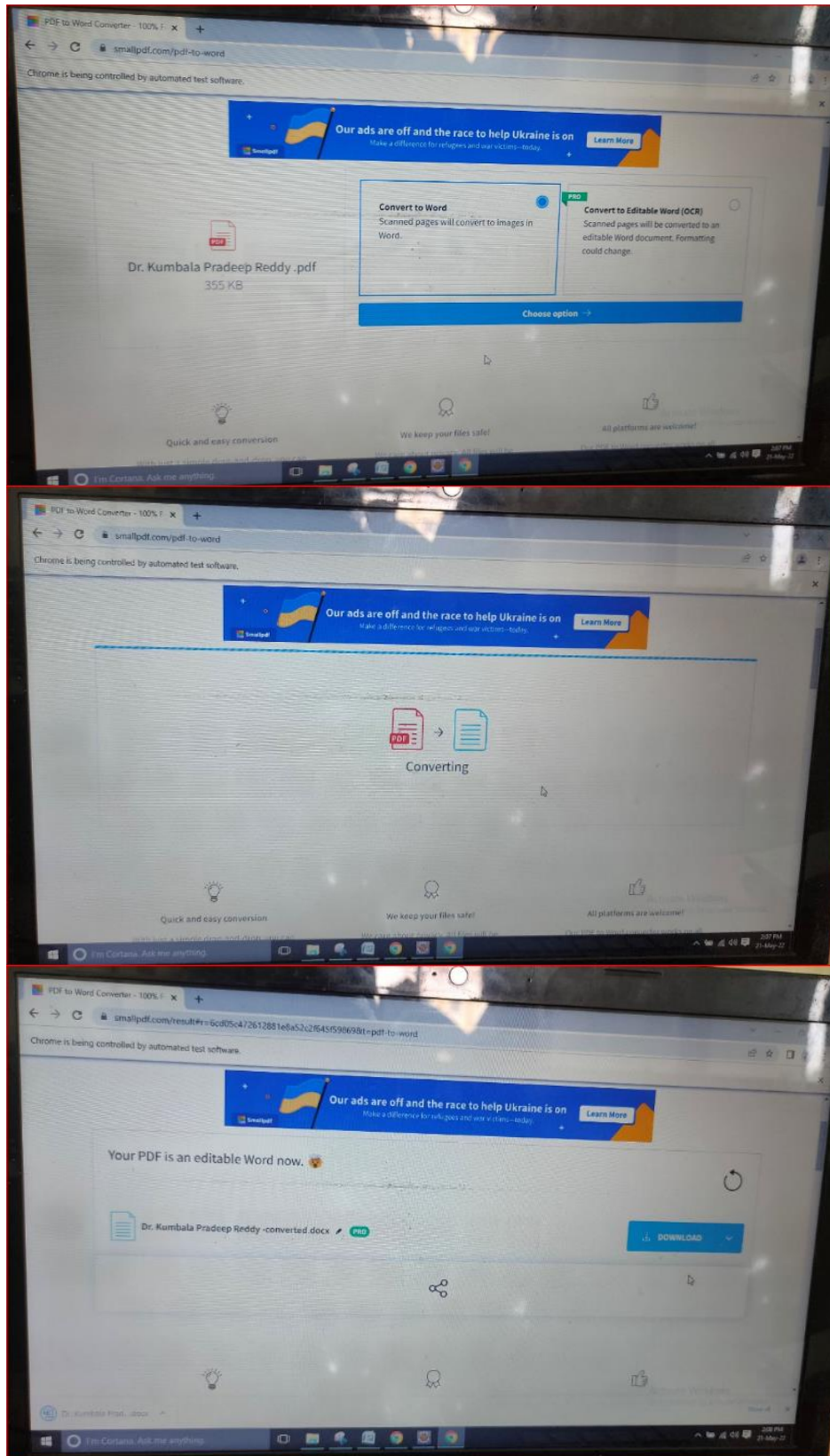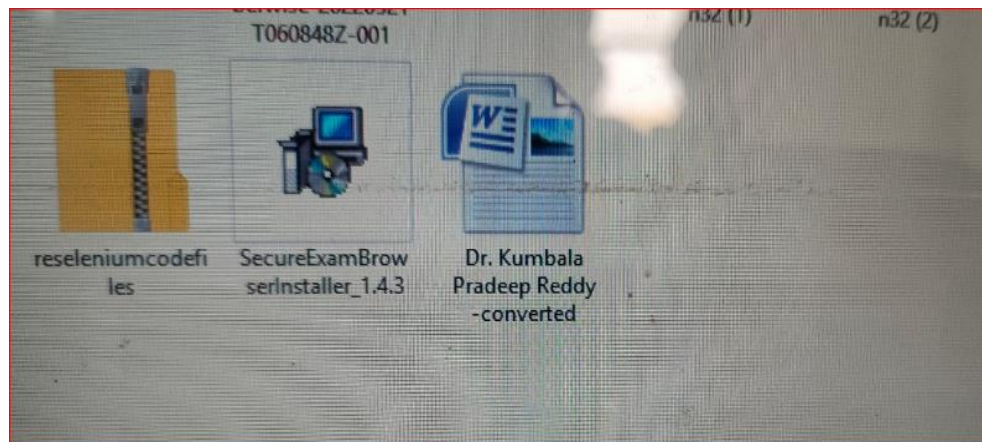
OUTPUT:

OUTPUT:

Viva Questions:

1) Explain what is the main difference between web-driver and RC ?

The main difference between Selenium RC and Webdriver is that, selenium RC injects javascript function into browsers when the page is loaded. On the other hand, Selenium Webdriver drives the browser using browsers built in support

2) What are the advantages of RC?

Advantages of RC are

- Can read or write data from/ to .xls, .txt, etc
- It can handle dynamic objects and Ajax based UI elements
- Loops and conditions can be used for better performance and flexibility
- Support many Programming languages and Operating Systems
- For any JAVA script enabled browser Selenium RC can be used

3) What are the technical limitations while using Selenium RC?

Apart from "same origin policy" restriction from js, Selenium is also restricted from exercising anything that is outside browser.

4) Can we use Selenium RC to drive tests on two different browsers on one operating system without Selenium Grid?

Yes, it is possible when you are not using JAVA testing framework. Instead of using Java testing framework if you are using java client driver of selenium then TestNG allows you to do this. By using "parallel=test" attribute you can set tests to be executed in parallel and can define two different tests, each using different browser.

5) How do you identify an object using selenium?

To identify an object using Selenium you can use

isElementPresent(String locator)

isElementPresent takes a locator as the argument and if found returns a Boolean.

References For Downloading IDE :

install web driver

https://www.selenium.dev/downloads/

install jdk

https://www.oracle.com/java/technologies/downloads/

install eclipse

https://www.eclipse.org/downloads/packages/

windows chrome driver

https://chromedriver.chromium.org/downloads