# EASWARI ENGINEERING COLLEGE

**(Autonomous)**

**Bharathi Salai, Ramapuram, Chennai-600 089**

Department: Master of Computer Applications

Name of the Lab: 244MCE317J/SOFTWARE QUALITY AND TESTING

| | |
|---|---|
| Name | : |
| Reg No. | : |
| Semester | : |
| Year | : |
| Branch | : |

EASWARI ENGINEERING COLLEGE

**(Autonomous)**

**Bharathi Salai, Ramapuram, Chennai-600 089**

**Department:** Master of Computer Applications

**PRACTICAL EXAMINATIONS NOVEMBER 2025**

# BONAFIDE CERTIFICATE

This is to certify that this practical work titled

244MCE317J /SOFTWARE QUALITY AND TESTING is the Bonafide work of Mr./Miss.______________________________

With Registration Number________________________________

in __________Semester of ____________Year in the Department of

MASTER OF COMPUTER APPLICATIONS during the academic year 2025-2026

**Faculty Incharge** **Head of the Department**

Submitted for Practical Examination held on ____/__________/___at
Easwari Engineering College, Ramapuram, Chennai-89

**Internal Examiner** **External Examiner**

## CONTENTS

**Exercise 1: Perform data flow testing for any C program to verify the def-use variables (Ex:largest of two numbers)**

**Aim:**

To find the largest of two numbers using Selenium

**Procedures:**

Step 1: Start the program.

Step 2: Read the first number from the keyboard.

Step 3: Read the second number from the keyboard.

Step 4: Compare the two numbers.

Step 5: If the first number is greater than the second number, assign first number as the largest.

Step 6: Otherwise, assign the second number as the largest.

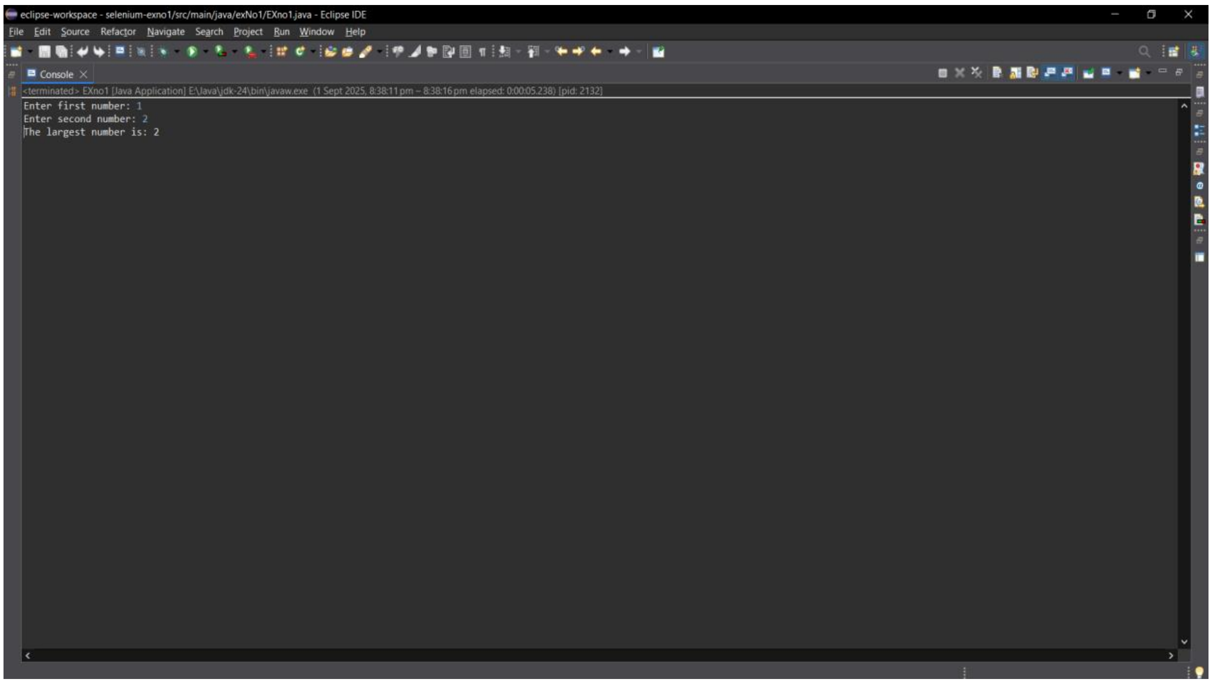
Step 7: Display the largest number.

Step 8: Stop the program.

**Program**

```
package exNo1;
import java.util.Scanner;
public class EXno1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter first number: ");
        int a = sc.nextInt();
        System.out.print("Enter second number: ");
        int b = sc.nextInt();
        int largest = (a > b) ? a : b;
        System.out.println("Largest number is: " + largest);
        sc.close();
    }
}
```

## Output:


eclipse-workspace - selenium-exno1/src/main/java/exNo1/EXno1.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console

```
<terminated> EXno1 [Java Application] E:\Java\jdk-24\bin\javaw.exe (1 Sept 2025, 8:38:11 pm – 8:38:16 pm elapsed: 0:00:05.238) [pid: 2132]
Enter first number: 1
Enter second number: 2
The largest number is: 2
```

## Result:

The output has been successfully implemented.

**Exercise 2: Using Selenium IDE, write a test suite containing minimum 4 testcases for any simple C program**

**(Ex: To check Adam number)**

**Aim:**

To write a test suite containing a minimum of 4 test cases using Selenium

**Procedures:**

Step 1: Start the program.
Step 2: Set the path for ChromeDriver.
Step 3: Open Chrome browser.
Step 4: Create a Scanner to read input.
Step 5: Start an empty HTML page string.
Step 6: Repeat 4 times:
    (a) Ask the user to enter a number.
    (b) Check whether it is an Adam number or not.
    (c) Add the result to the HTML string.
Step 7: Close the Scanner.
Step 8: Complete the HTML page with closing tags.
Step 9: Display the HTML page in Chrome.
Step 10: Stop the program.

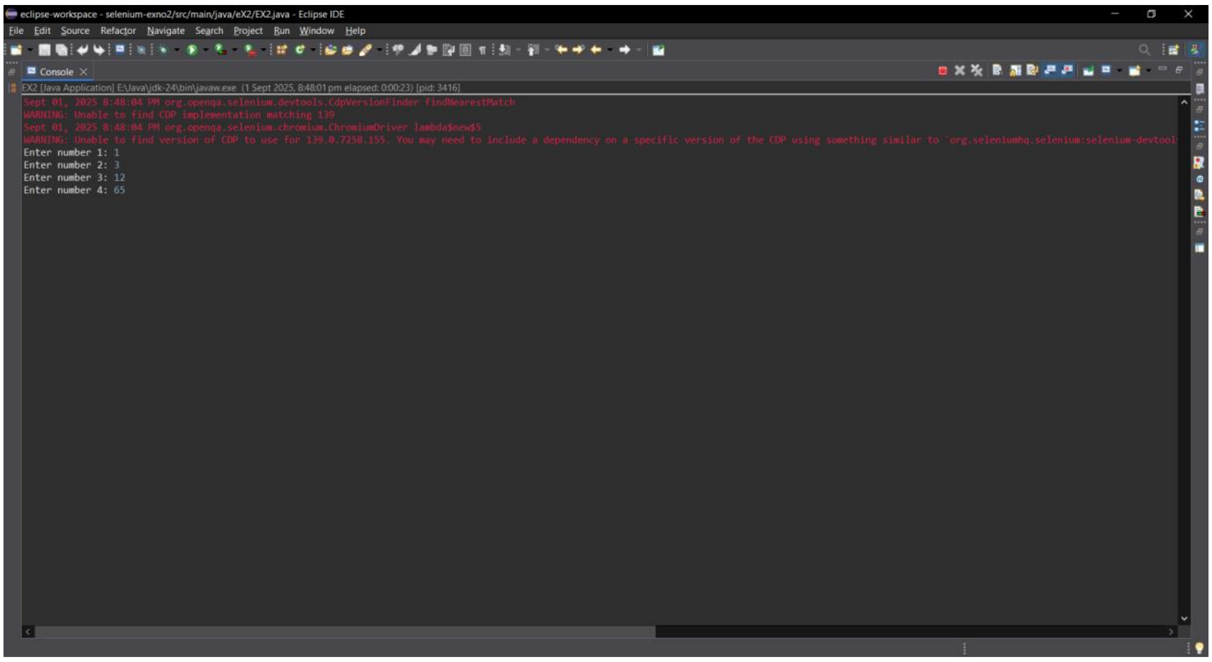
**Program**

```
package eX2;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.Scanner;

public class EX2 {

static boolean isAdam(int n) {

int r = Integer.parseInt(new StringBuilder(n+"").reverse().toString());

return n*n == Integer.parseInt(new StringBuilder(r*r+"").reverse().toString());

}

public static void main(String[] a) {

System.setProperty("webdriver.chrome.driver","E:\\Java\\eclipse-workspace\\selenium-
exno2\\driver\\chromedriver.exe");

WebDriver d = new ChromeDriver();

Scanner sc = new Scanner(System.in);

String h = "<html><meta charset='UTF-8'><body style='font-family:Arial;'>";

for (int i=1;i<=4;i++) {

System.out.print("Enter number "+i+": ");

String in = sc.nextLine(), r;

try { r = isAdam(Integer.parseInt(in))?"Adam Number":"Not Adam Number"; }

catch(Exception e){ r="Invalid Input"; }

h += "<p>Test Case "+i+": "+in+" → "+r+"</p>";
}

sc.close();

d.get("data:text/html,"+h+"</body></html>");

}

}
```

**Apache Maven**

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                             https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>selenium</groupId>
  <artifactId>selenium-exno2</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <dependencies>
    <!-- Selenium Java -->
    <dependency>
      <groupId>org.seleniumhq.selenium</groupId>
      <artifactId>selenium-java</artifactId>
      <version>4.24.0</version>
    </dependency>
  </dependencies>
</project>
```

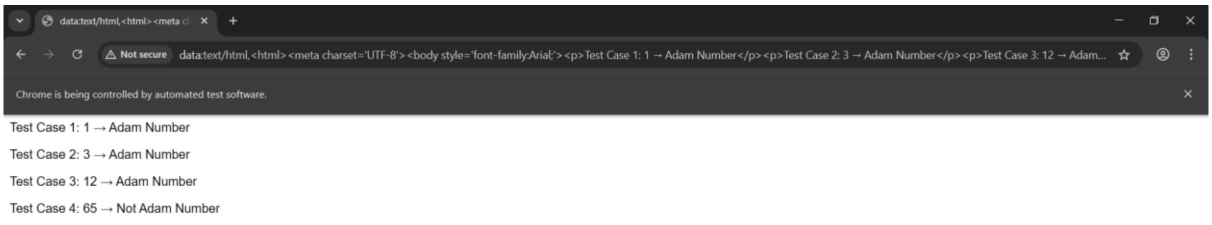
**Output:**



eclipse-workspace - selenium-exno2/src/main/java/eX2/EX2.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help
Console

```
EX2 [Java Application] E:\Java\jdk-24\bin\javaw.exe  (1 Sept 2025, 8:48:01 pm elapsed: 0:00:23) [pid: 3416]
Sept 01, 2025 8:48:04 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find CDP implementation matching 139
Sept 01, 2025 8:48:04 PM org.openqa.selenium.chromium.ChromiumDriver lambda$new$5
WARNING: Unable to find version of CDP to use for 139.0.7258.155. You may need to include a dependency on a specific version of the CDP using something similar to `org.seleniumhq.selenium:selenium-devtool
Enter number 1: 1
Enter number 2: 3
Enter number 3: 12
Enter number 4: 65
```



```
data:text/html,<html><meta charset='UTF-8'><body style='font-family:Arial;'><p>Test Case 1: 1 → Adam Number</p><p>Test Case 2: 3 → Adam Number</p><p>Test Case 3: 12 → Adam...
```

Chrome is being controlled by automated test software.

Test Case 1: 1 → Adam Number

Test Case 2: 3 → Adam Number

Test Case 3: 12 → Adam Number

Test Case 4: 65 → Not Adam Number

**Result:**

The output has been successfully implemented.

**Exercise 3: Write and test a program to update 10 student records into tables into excel file. (Selenium)**

**Aim:**

To update 10 student records into tables into excel file. (Selenium)

**Procedures:**

**Step1:**Start a program

**Step2:**Import files to upload

**Step3:**Import panda library as pd

**Step4:**Import io library

**Step5:**Assigning the statement as ‘df = pd.read_csv(io.BytesIO(uploaded['1.csv']))’

**Step6:**Print the statement with the variable ‘df’.

**Step7:** Import panda library as pd

**Step8:**Reading the csv files ‘df = pd.read_csv("1.csv")’

**Step9:** Updating the column value/data ‘df.loc[5, 'Name'] = 'SHIV CHANDRA'’

**Step10:** Writing into the file ‘df.to_csv("1.csv", index=False)’

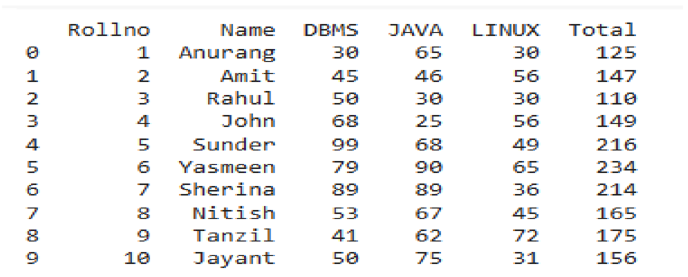
**Step11:** Print the statement with the variable ‘df’.

**Program**

```
from google.colab import files
uploaded = files.upload()
import pandas as pd
import io
df = pd.read_csv(io.BytesIO(uploaded['1.csv']))
print(df)
import pandas as pd
# reading the csv file
df = pd.read_csv("1.csv")
# updating the column value/data
df.loc[5, 'Name'] = 'SHIV CHANDRA'
# writing into the file
df.to_csv("1.csv", index=False)
print(df)
```

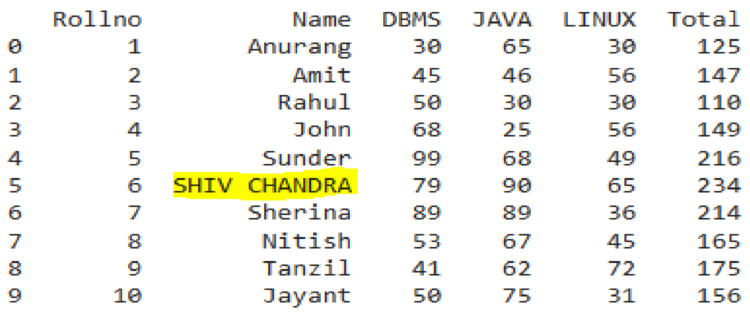
**Output:**

**Read:**



```
   Rollno     Name  DBMS  JAVA  LINUX  Total
0       1  Anurang    30    65     30    125
1       2     Amit    45    46     56    147
2       3    Rahul    50    30     30    110
3       4     John    68    25     56    149
4       5   Sunder    99    68     49    216
5       6  Yasmeen    79    90     65    234
6       7  Sherina    89    89     36    214
7       8   Nitish    53    67     45    165
8       9   Tanzil    41    62     72    175
9      10   Jayant    50    75     31    156
```

**Updated:**



```
   Rollno          Name  DBMS  JAVA  LINUX  Total
0       1       Anurang    30    65     30    125
1       2          Amit    45    46     56    147
2       3         Rahul    50    30     30    110
3       4          John    68    25     56    149
4       5        Sunder    99    68     49    216
5       6  SHIV CHANDRA    79    90     65    234
6       7       Sherina    89    89     36    214
7       8        Nitish    53    67     45    165
8       9        Tanzil    41    62     72    175
9      10        Jayant    50    75     31    156
```

**Result:**

The output has been successfully implemented

**Exercise 4: Write and test a program to select the number of students who have scored more than 60 in any one subject (or all subjects).(Selenium)**

**Aim:**

To select the number of students who have scored more than 60 in any one subject (or all Subjects)

**Procedures:**

1. Start the program.
2. Set up WebDriver path by specifying the location of the ChromeDriver executable.
3. Initialize Chrome browser using new ChromeDriver().
4. Open the HTML file containing the student marks table using driver.get().
5. Locate the student table using its ID (studentTable).
6. Fetch all rows inside the <tbody> of the table.
7. Initialize a counter (count = 0).
8. For each row in the table:
    - Extract all <td> cells.
    - Read the marks for Math, English, and Selenium subjects.
    - Convert the text values into integers.
    - If any subject mark > 60, increment the counter.
9. Display the result – print the number of students who scored more than 60 in at least one subject.
10. Close the browser (optional).
11. End the program.

**Stud.html**

```
<!DOCTYPE html>

<html>

<head>

    <title>Student Scores</title>

</head>

<body>

    <table id="studentTable" border="1">

        <thead>

            <tr>

                <th>Name</th>

                <th>Math</th>

                <th>English</th>

                <th>Selenium</th>

            </tr>

        </thead>

        <tbody>

            <tr>

                <td>Student 1</td>

                <td>85</td>

                <td>72</td>

                <td>55</td>
            </tr>

            <tr>

                <td>Student 2</td>

                <td>60</td>

                <td>78</td>

                <td>68</td>

            </tr>

            <tr>

                <td>Student 3</td>

                <td>70</td>

                <td>45</td>

                <td>75</td>

            </tr>

        </tbody>

    </table>

</body>

</html>
```

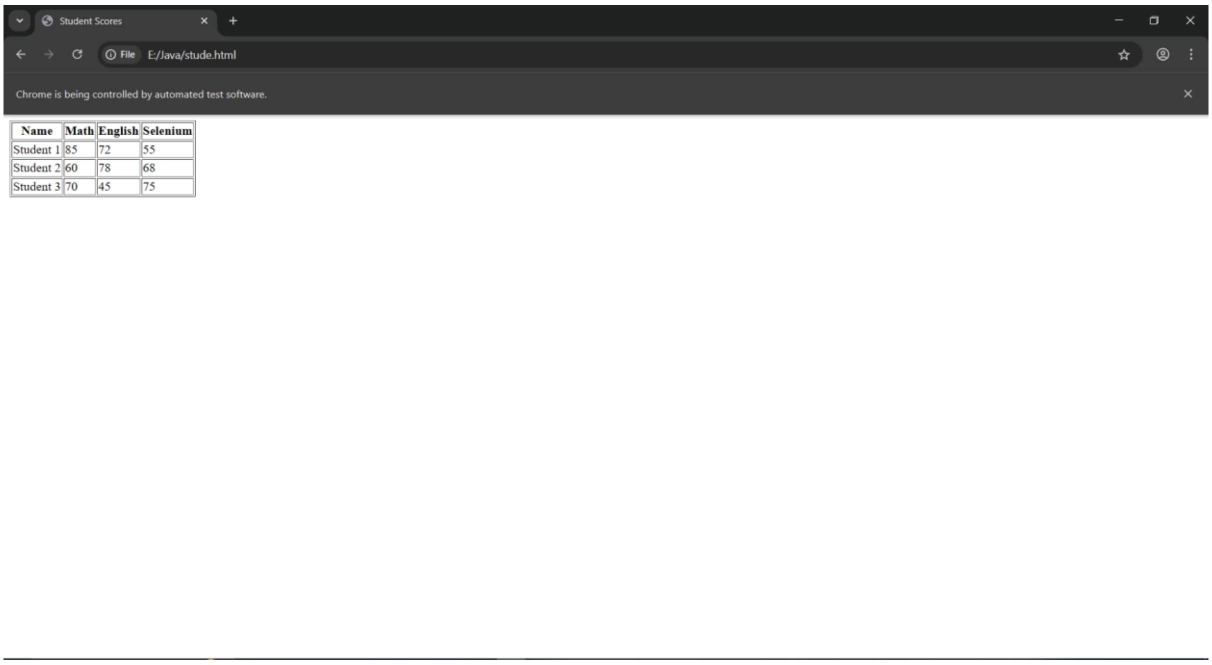
**<u>StudentScoreCounter</u>**

```
package StudentScoreCounter;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;
import java.util.List;

public class StudentScoreCounter {

    public static void main(String[] args) {

        System.setProperty("webdriver.chrome.driver", "E:\\Java\\eclipse-workspace\\selenium-
exno3\\dri\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();

         driver.get("file:///E:/Java/stude.html");

        WebElement table = driver.findElement(By.id("studentTable"));

        List<WebElement> rows = table.findElements(By.xpath(".//tbody/tr"));

        int count = 0;

        for (WebElement row : rows) {

            List<WebElement> cells = row.findElements(By.tagName("td"));

            int math = Integer.parseInt(cells.get(1).getText());

            int english = Integer.parseInt(cells.get(2).getText());

            int selenium = Integer.parseInt(cells.get(3).getText());

            if (math > 60 || english > 60 || selenium > 60) {

                count++;

            }

        }

        System.out.println("Number of students who scored more than 60 in at least one subject: " + count);

    }

}
```

**Xml**

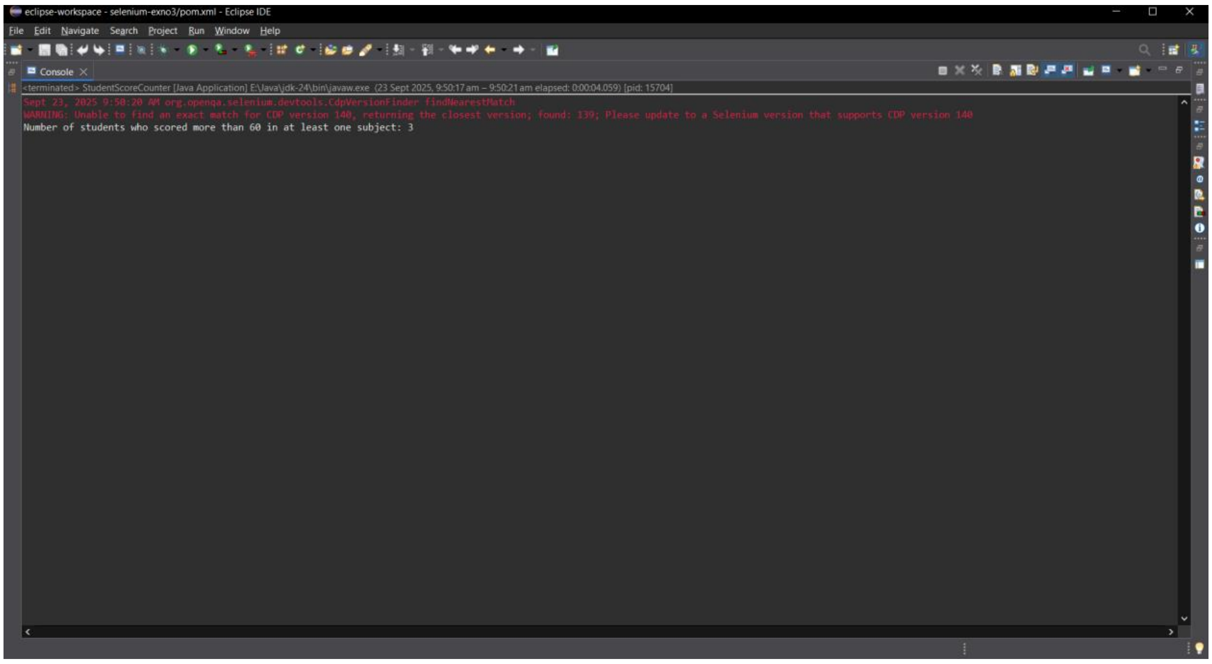
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-
4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>selenium-exno3</groupId>
  <artifactId>selenium-exno3</artifactId>
  <version>0.0.1-SNAPSHOT</version>
</project>
```

## Output



Chrome is being controlled by automated test software.

| Name | Math | English | Selenium |
|---|---|---|---|
| Student 1 | 85 | 72 | 55 |
| Student 2 | 60 | 78 | 68 |
| Student 3 | 70 | 45 | 75 |


eclipse-workspace - selenium-exno3/pom.xml - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Console

```
<terminated> StudentScoreCounter [Java Application] E:\Java\jdk-24\bin\javaw.exe (23 Sept 2025, 9:50:17 am – 9:50:21 am elapsed: 0:00:04.059) [pid: 15704]
Sept 23, 2025 9:50:20 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 140, returning the closest version; found: 139; Please update to a Selenium version that supports CDP version 140
Number of students who scored more than 60 in at least one subject: 3
```

## Result:

The output has been successfully implemented.

**Exercise 5: Write and test a program to login to a specific web page**

**Aim:**

To login to a specific web page.

**Procedures:**

1. Start
2. Set the system property for chromedriver and initialize the WebDriver.
3. Load the login.html page from the local project directory using driver.get().
4. Identify the username and password input fields by their name attributes.
5. Enter the test credentials (username and password) into the respective fields.
6. Locate the login button by its id and perform a click action.
7. Capture the result message from the element with id="message".
8. Print the message to the console and verify it using assertEquals().
9. Close the browser using driver.quit() in the teardown method.
10. Stop.

```
<!DOCTYPE html>

<html>

<head>

    <title>Login Page</title>

</head>

<body>

    <h2>Login Form</h2>

    <form id="loginForm">

        <label for="username">Username:</label>

        <input type="text" id="username" name="username"><br><br>

        <label for="password">Password:</label>

        <input type="password" id="password" name="password"><br><br>

        <button type="submit" id="loginButton">Login</button>

    </form>

    <p id="message" style="color:red;"></p>

    <script>

        const form = document.getElementById('loginForm');

        const message = document.getElementById('message');

        form.addEventListener('submit', function(event) {

            event.preventDefault();

            const username = document.getElementById('username').value;

            const password = document.getElementById('password').value;
            if(username === "Admin" && password === "admin123") {

                message.style.color = "green";

                message.textContent = "Login Successful!";

            } else {

                message.style.color = "red";

                message.textContent = "Invalid credentials!";

            }

        });

    </script>

</body>

</html>
```

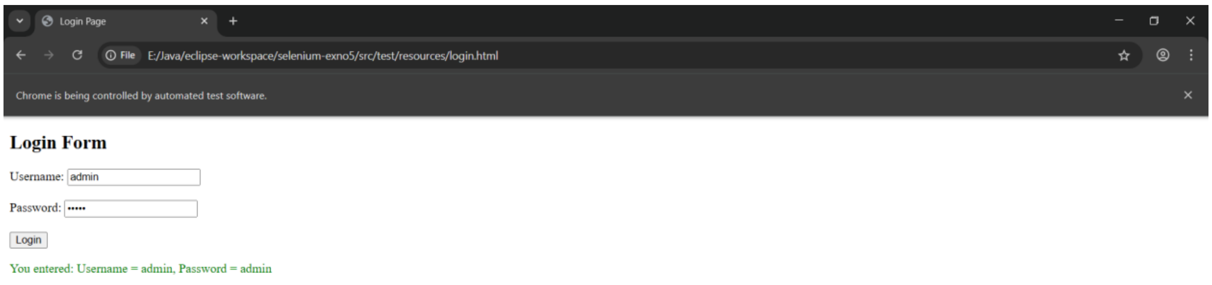
**exno5**

```
package exno5;

import org.junit.Test;

import org.junit.Before;

import org.junit.After;

import static org.junit.Assert.*;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.io.File;

public class exno5 {
    private WebDriver driver;

@Before

    public void setUp() {

        System.setProperty("webdriver.chrome.driver", "E:\\Java\\eclipse-workspace\\selenium-
exno5\\driver\\chromedriver.exe");

        driver = new ChromeDriver();

    }

    @After

    public void tearDown() {

    }

    @Test

    public void loginTest() {

        File file = new File("src/test/resources/login.html");

        driver.get(file.getAbsolutePath());

        String usernameInput = "";

        String passwordInput = "";

        driver.findElement(By.name("username")).sendKeys(usernameInput);

        driver.findElement(By.name("password")).sendKeys(passwordInput);

        driver.findElement(By.id("loginButton")).click();

        WebElement message = driver.findElement(By.id("message"));

        System.out.println("Login message: " + message.getText());

    }

}
```

**xml**

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
   http://maven.apache.org/xsd/maven-4.0.0.xsd">
   <modelVersion>4.0.0</modelVersion>
   <groupId>com.example</groupId>
   <artifactId>SeleniumLoginTest</artifactId>
   <version>1.0-SNAPSHOT</version>
 <dependencies>
      <!-- Selenium -->
      <dependency>
         <groupId>org.seleniumhq.selenium</groupId>
         <artifactId>selenium-java</artifactId>
         <version>4.35.0</version>
      </dependency>
      <!-- JUnit -->
      <dependency>
         <groupId>junit</groupId>
         <artifactId>junit</artifactId>
         <version>4.13.2</version>
      </dependency>
      <!-- WebDriverManager -->
      <dependency>
         <groupId>io.github.bonigarcia</groupId>
         <artifactId>webdrivermanager</artifactId>
         <version>5.5.1</version>
      </dependency>
   </dependencies>
</project>
```

## Output



Login Page

File E:/Java/eclipse-workspace/selenium-exno5/src/test/resources/login.html

Chrome is being controlled by automated test software.

**Login Form**

Username: admin

Password: •••••

Login

You entered: Username = admin, Password = admin

## Result

The output has been successfully implemented.

## Exercise 6: Write and test a program to provide a total number of objects present / available on the page.(Selenium)

**Aim:**

To provide a total number of objects present / available on the page.(Selenium)

**Procedures:**

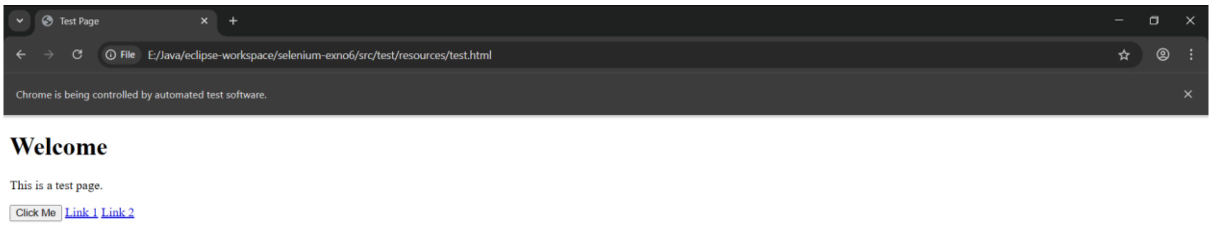
1. Start.
2. Set the system property to specify the path of chromedriver.exe.
3. Create an instance of the ChromeDriver to launch the Chrome browser.
4. Open the local HTML file (test.html) using driver.get("file:///...").
5. Use findElements(By.xpath("//*")) to select all elements present in the page.
6. Store the elements in a List<WebElement>.
7. Find the total number of elements using list.size().
8. Print the total number of objects on the page.
9. Close the browser using driver.quit().
10. Stop.

```html
<!DOCTYPE html>
<html>
<head>
   <title>Test Page</title>
</head>
<body>
   <h1>Welcome</h1>
   <p>This is a test page.</p>
   <button>Click Me</button>
   <a href="#">Link 1</a>
   <a href="#">Link 2</a>
</body>
</html>
```

```java
package exno6;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.List;

public class exno6 {

    public static void main(String[] args) {

        System.setProperty("webdriver.chrome.driver",

            "E:\\Java\\eclipse-workspace\\selenium-exno6\\driver\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();

        try {

             driver.get("file:///E:/Java/eclipse-workspace/selenium-exno6/src/test/resources/test.html");
            List<WebElement> allElements = driver.findElements(By.xpath("//*"));

            System.out.println("Total number of objects on the page: " + allElements.size());

        } catch (Exception e) {

            e.printStackTrace();

        } finally {

        }

    }

}
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>selenium-exno6</groupId>
  <artifactId>selenium-exno6</artifactId>
  <version>0.0.1-SNAPSHOT</version>
</project>
```

## Output


Test Page

File E:/Java/eclipse-workspace/selenium-exno6/src/test/resources/test.html

Chrome is being controlled by automated test software.

Welcome

This is a test page.

Click Me Link 1 Link 2

## Result

The output has been successfully implemented

## Exercise 7: Write and test a program to get the number of list items in a list / combo box. (Selenium)

**Aim:** To get the number of list items in a list/combo box.(selenium)

**Procedures:**

Step 1: Launch the browser and open the local HTML page.

- Set ChromeDriver path, initialize WebDriver, maximize the window, set implicit wait, and navigate to the HTML file.

Step 2: Locate the combo box element on the page.

- Use driver.findElement(By.id("colors")) to find the <select> element.

Step 3: Create a Select object for the combo box.

- Pass the combo box WebElement to the Select constructor to enable dropdown operations.

Step 4: Retrieve and display all options.

- Get all <option> elements using select.getOptions(), print the total count, and display each item's text.

Step 5: Close the browser.

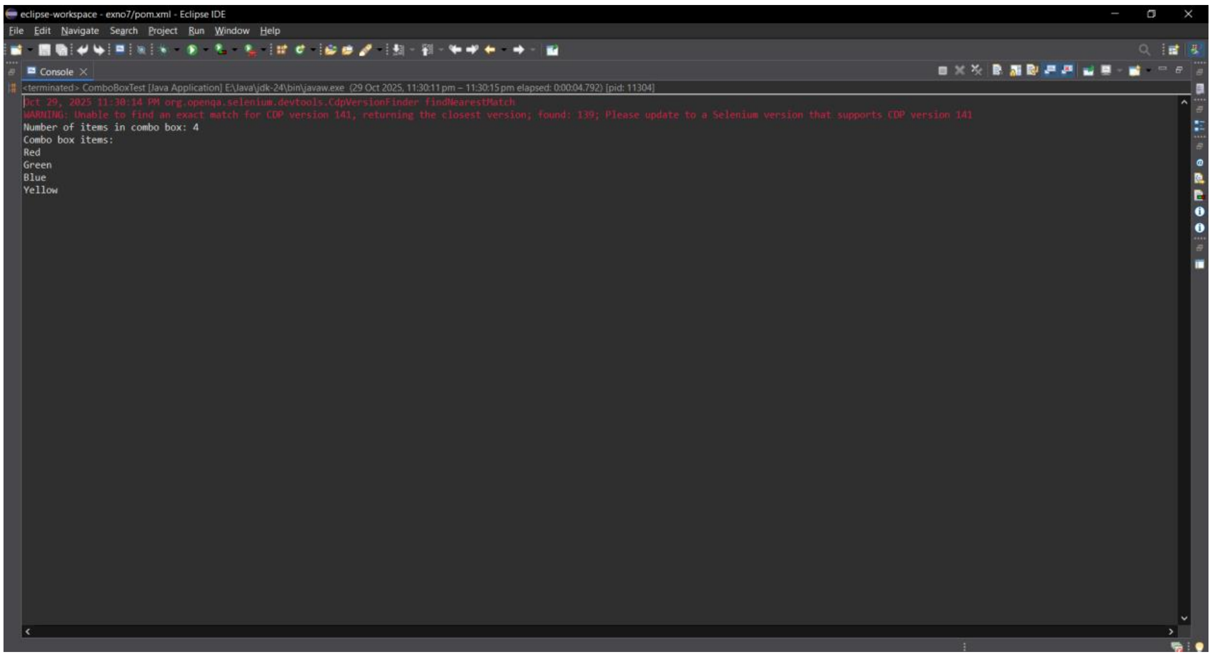
- Call driver.quit() to end the WebDriver session and close the browser.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Simple Combo Box</title>
</head>
<body>
    <h2>Simple Combo Box Example</h2>

    <label for="colors">Choose a color:</label>
    <select id="colors">
        <option value="red">Red</option>
        <option value="green">Green</option>
        <option value="blue">Blue</option>
        <option value="yellow">Yellow</option>
    </select>
</body>
</html>
```

```
package exno7;
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;
import java.time.Duration;
import java.util.List;
public class ComboBoxTest {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver",
                "E:\\Java\\eclipse-workspace\\exno7\\driver\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
        try {
            driver.get("file:///E:/Java/eclipse-workspace/exno7/exno7.html");
            WebElement comboBox = driver.findElement(By.id("colors"));
            Select select = new Select(comboBox);
            List<WebElement> options = select.getOptions();
            System.out.println("Number of items in combo box: " + options.size());
            System.out.println("Combo box items:");
            for (WebElement option : options) {
                System.out.println(option.getText());
            }

        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            driver.quit();
        }
    }
}
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>exno7</groupId>
  <artifactId>exno7</artifactId>
  <version>0.0.1-SNAPSHOT</version>
</project>
```

**Output**


eclipse-workspace - exno7/pom.xml - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Console

```
<terminated> ComboBoxTest [Java Application] E:\Java\jdk-24\bin\javaw.exe (29 Oct 2025, 11:30:11 pm – 11:30:15 pm elapsed: 0:00:04.792) [pid: 11304]
Oct 29, 2025 11:30:14 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 141, returning the closest version; found: 139; Please update to a Selenium version that supports CDP version 141
Number of items in combo box: 4
Combo box items:
Red
Green
Blue
Yellow
```

**Result:**

The output has been successfully implemented.

**Exercise 8: Identify system specification and design test cases to test any application using any one of a testing tool (Selenium/Bugzilla/Test Director)**

**Aim:**

To get the number of list items in a list/combo box.(selenium)

**Procedures:**

**Step 1:** Launch the browser and open the HTML login page.

- Set the ChromeDriver path, initialize WebDriver, maximize the browser window, and navigate to the local HTML file.

**Step 2:** Enter valid credentials and perform login.

- Input "admin" as username and "1234" as password, then click the login button.

**Step 3:** Capture and print the login result.

- Retrieve the message displayed in the <p> tag (id="msg") and print it to the console for Test Case 1 (valid login).

**Step 4:** Repeat login with invalid credentials.

- Clear the fields, enter "admin" and "0000", click login, and print the message for Test Case 2 (invalid password).
- Also test with empty fields for Test Case 3 and print the message.

**Step 5:** Close the browser.

- Quit the WebDriver session to end the test and close the browser.

**HTML Program**

```
<!DOCTYPE html>
<html>
<head>
  <title>Login Page</title>
</head>
<body>
  <h2>Simple Login Application</h2>
  <label>Username:</label>
  <input type="text" id="username"><br><br>
  <label>Password:</label>
  <input type="password" id="password"><br><br>
  <button id="loginBtn">Login</button>
  <p id="msg"></p>

  <script>
    document.getElementById("loginBtn").onclick = function() {
      var u = document.getElementById("username").value;
      var p = document.getElementById("password").value;
      if(u === "admin" && p === "1234")
        document.getElementById("msg").innerText = "Login Successful!";
      else
        document.getElementById("msg").innerText = "Invalid Username or Password!";
    }
  </script>
</body>
</html>
```

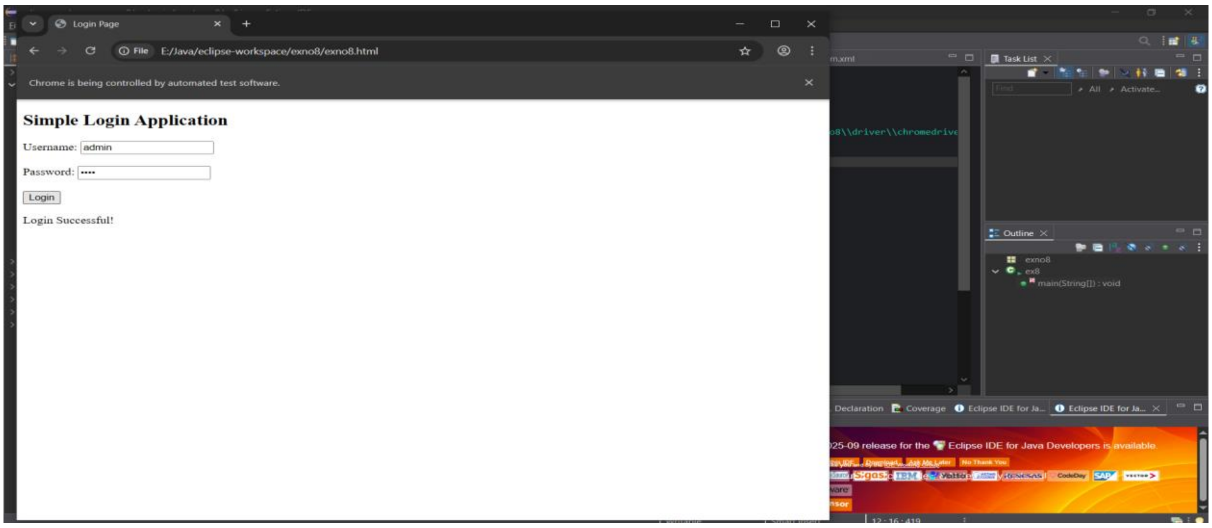
**Selenium**

```
package exno8;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class ex8 {
    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver", "E:\\Java\\eclipse-
workspace\\exno8\\driver\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("file:///E:/Java/eclipse-workspace/exno8/exno8.html");
        driver.manage().window().maximize();
        driver.findElement(By.id("username")).sendKeys("admin");
        driver.findElement(By.id("password")).sendKeys("1234");
        driver.findElement(By.id("loginBtn")).click();
        Thread.sleep(1000);
        String msg1 = driver.findElement(By.id("msg")).getText();
        System.out.println("TC1 Result: " + msg1);
        driver.findElement(By.id("username")).clear();
        driver.findElement(By.id("password")).clear();
        driver.findElement(By.id("username")).sendKeys("admin");
        driver.findElement(By.id("password")).sendKeys("0000");
        driver.findElement(By.id("loginBtn")).click();
        Thread.sleep(1000);
        String msg2 = driver.findElement(By.id("msg")).getText();
        System.out.println("TC2 Result: " + msg2);
        driver.findElement(By.id("username")).clear();
        driver.findElement(By.id("password")).clear();
        driver.findElement(By.id("loginBtn")).click();
        Thread.sleep(1000);
        String msg3 = driver.findElement(By.id("msg")).getText();
        System.out.println("TC3 Result: " + msg3);
        driver.quit();
    }
}
```

**Configuration**

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>exno8</groupId>
  <artifactId>exno8</artifactId>
  <version>0.0.1-SNAPSHOT</version>
</project>
```

## Output



Chrome is being controlled by automated test software.

Simple Login Application

Username: admin

Password: ••••

Login

Login Successful!

## Result:

The output has been successfully implemented.

**Exercise 9: Automate the test cases of the above system using any test automation tool.**

**Aim:**

Automate the test cases of the above system using any test automation tool.

**Procedures:**

Step1: Go to **https://bugzilla.org**

Step2: Create an account in this website.

Step3: Click new bug

Step4: Type issues and click 'find similar issues'.

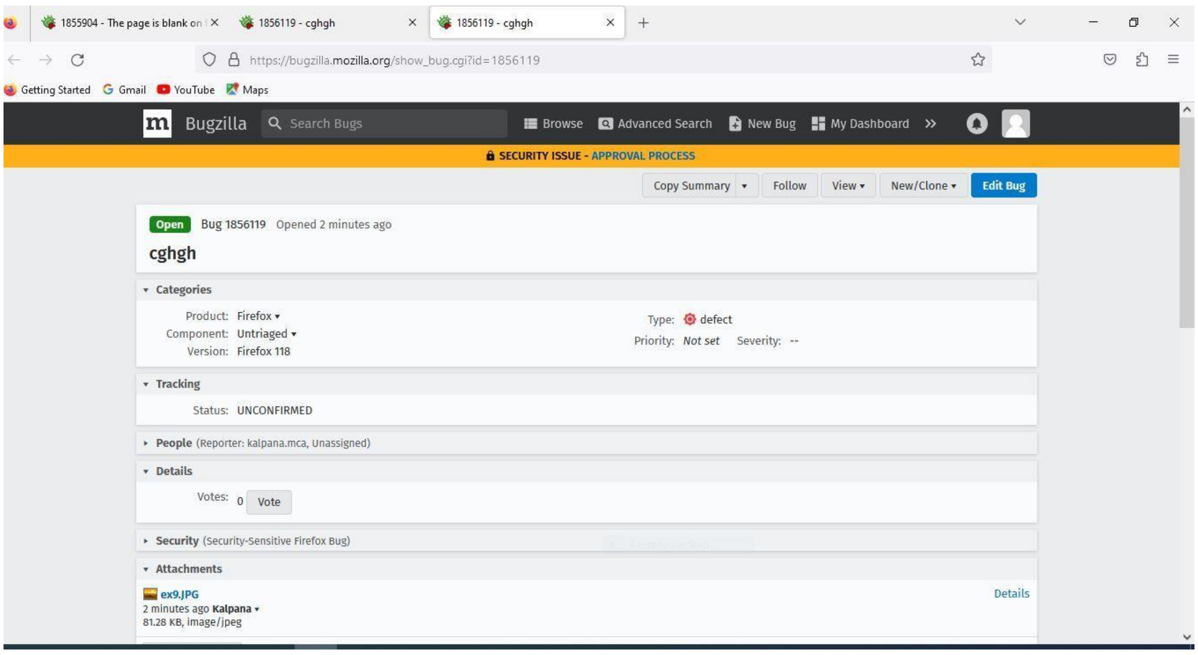
Step5: If your issues is new means click 'my issue is not listed'.

Step6: Enter a bug details and click 'submit a bug'.

Step7: Then Bug page is opened.

Step8: If you want, you can click 'new/clone' and 'edit bug' options in bug page.

**Output**



1855904 - The page is blank on
1856119 - cghgh
1856119 - cghgh

https://bugzilla.mozilla.org/show_bug.cgi?id=1856119

Getting Started | Gmail | YouTube | Maps

Bugzilla | Search Bugs | Browse | Advanced Search | New Bug | My Dashboard

SECURITY ISSUE - APPROVAL PROCESS

Copy Summary | Follow | View | New/Clone | Edit Bug

Open Bug 1856119 Opened 2 minutes ago

cghgh

Categories

Product: Firefox
Component: Untriaged
Version: Firefox 118
Type: defect
Priority: Not set
Severity: --

Tracking

Status: UNCONFIRMED

People (Reporter: kalpana.mca, Unassigned)

Details

Votes: 0 Vote

Security (Security-Sensitive Firefox Bug)

Attachments

ex9.JPG
2 minutes ago Kalpana
81.28 KB, image/jpeg
Details

**Result:**

The output has been successfully implemented.

**Exercise 10: Design test cases for web pages to test any websites.**

**Aim:**

To design test cases for web pages to test any websites.

**Procedures:**

Step1: Type webpage test in the search engine.

Step2: Paste particular url in the webpage test website.

Step3: Click test button.

Step4: Finally the web performance analysis report is generated.

**Output:** Webpage Test ( Web Performance Analyzer)

URL: https://www.tamildailycalendar.com/tamil_daily_calendar.php

From: Virginia USA - EC2 - Chrome - Emulated Motorola G (gen 4) - 4G

Page Performance Metrics(Run 1)

First View (Run 1)

**TTFB Start Render DC Time DC Requests DC Bytes Total Time Total Requests Page Weight**

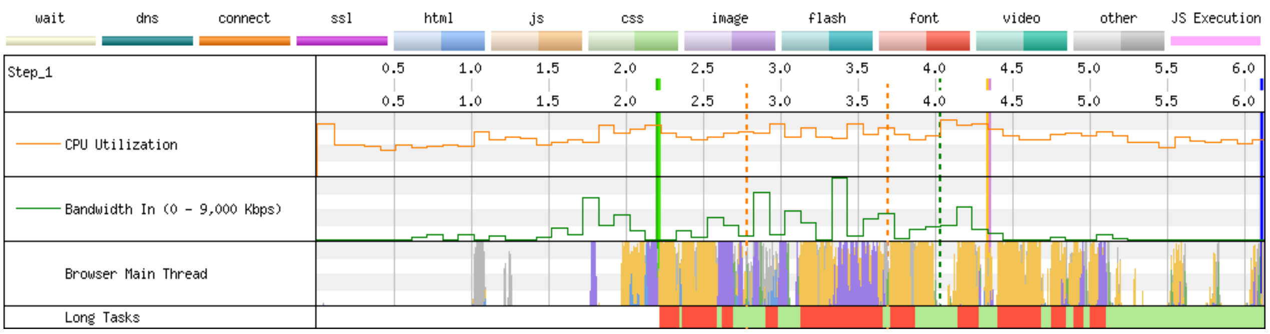
**.000S .000S .000S - - .000S - -**

Waterfall View

Start Render Document Complete

Render Blocking Resource Insecure Request 3xx response 4xx+ response

Doesn't Belong to Main Doc



**Result:**

The output has been successfully implemented.