UNIVERSITY OF THE FUTURE

University of Petroleum and Energy Studies

Test Automation Lab

CSDV3017

Name: Aanchal Tailwal **Faculty: Ms. J. Dhiviya Rose**

Branch: BTech CSE

Batch: DevOps B-4 (NH)

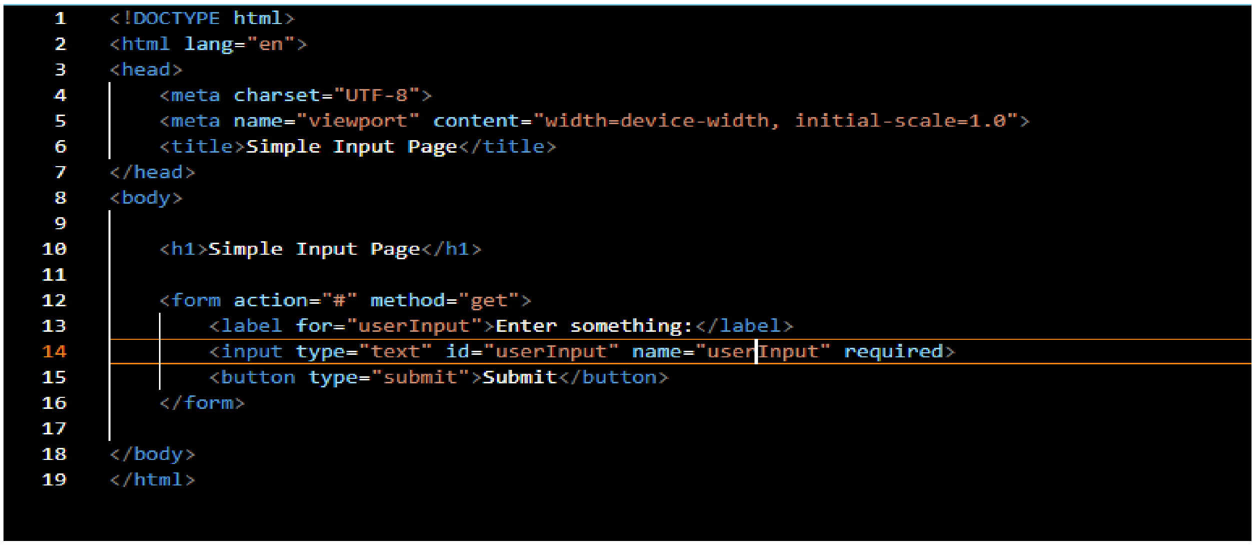
Sap Id: 500097386

INDEX

Experiment 1

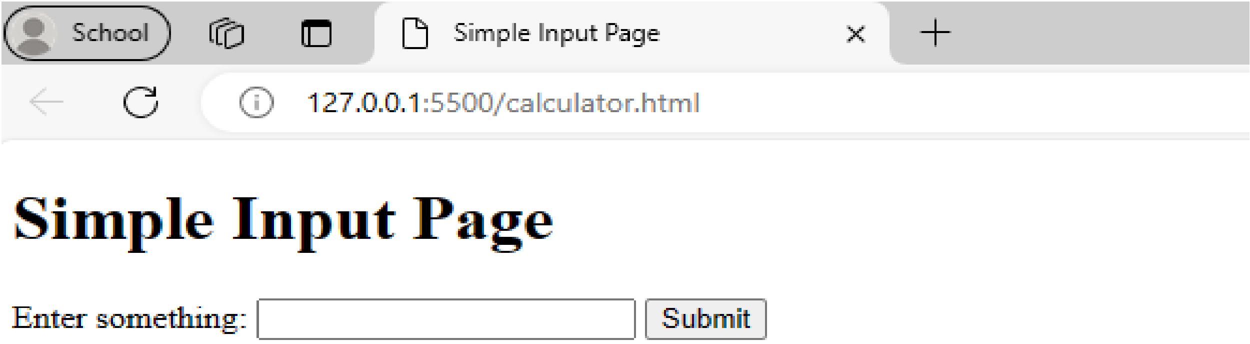
Invoking the WebPage via WebDriver in Java Demonstration

➔ Html code for an application



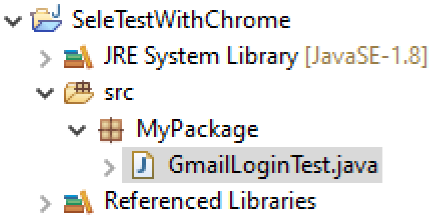
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simple Input Page</title>
</head>
<body>

    <h1>Simple Input Page</h1>

    <form action="#" method="get">
        <label for="userInput">Enter something:</label>
        <input type="text" id="userInput" name="userInput" required>
        <button type="submit">Submit</button>
    </form>

</body>
</html>
```

➔ Output



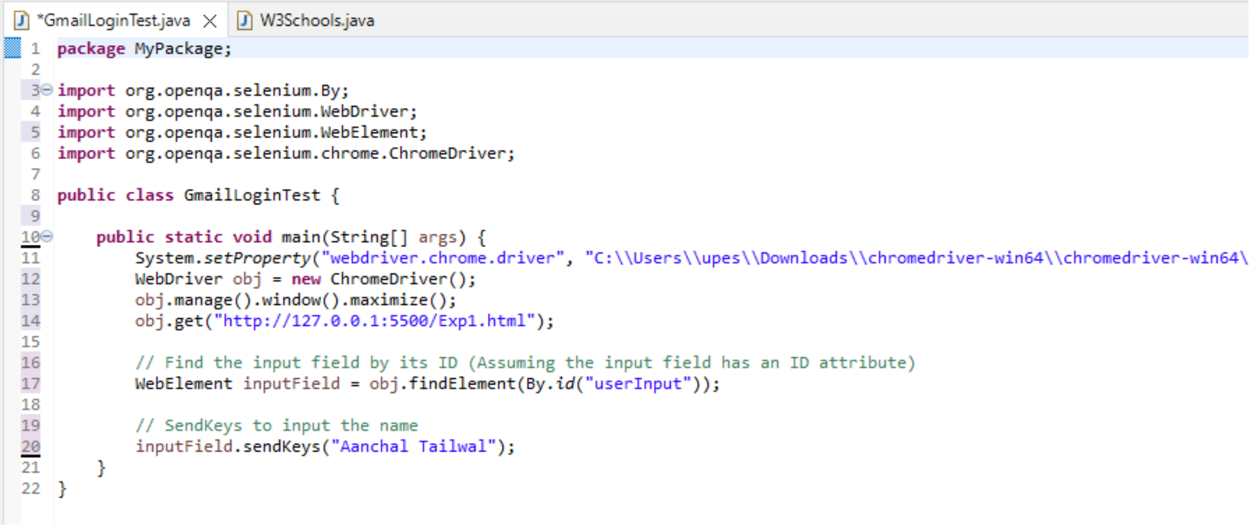
Simple Input Page

Enter something: [   ] Submit



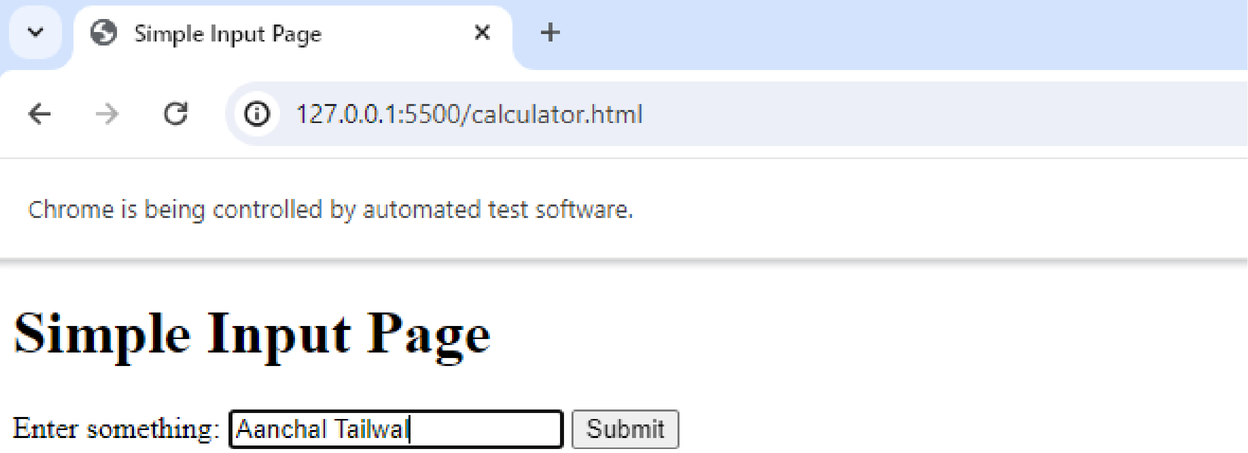
➔

➔ Java code to invoke webpage via WebDriver



```java
package MyPackage;

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

public class GmailLoginTest {

    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\upes\\Downloads\\chromedriver-win64\\chromedriver-win64\
        WebDriver obj = new ChromeDriver();
        obj.manage().window().maximize();
        obj.get("http://127.0.0.1:5500/Exp1.html");

        // Find the input field by its ID (Assuming the input field has an ID attribute)
        WebElement inputField = obj.findElement(By.id("userInput"));

        // SendKeys to input the name
        inputField.sendKeys("Aanchal Tailwal");
    }
}
```

➔ Output



Chrome is being controlled by automated test software.

Simple Input Page

Enter something: Aanchal Tailwal Submit

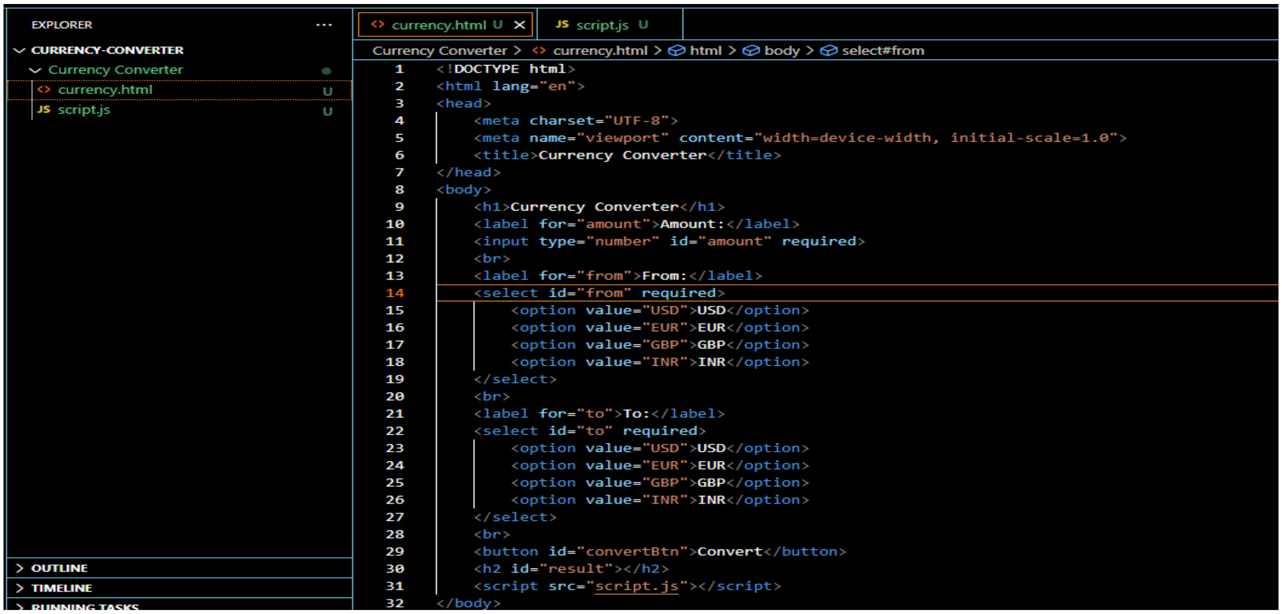
Experiment 2

Usability Testing Questionnaire

Task 1: Currency Converter

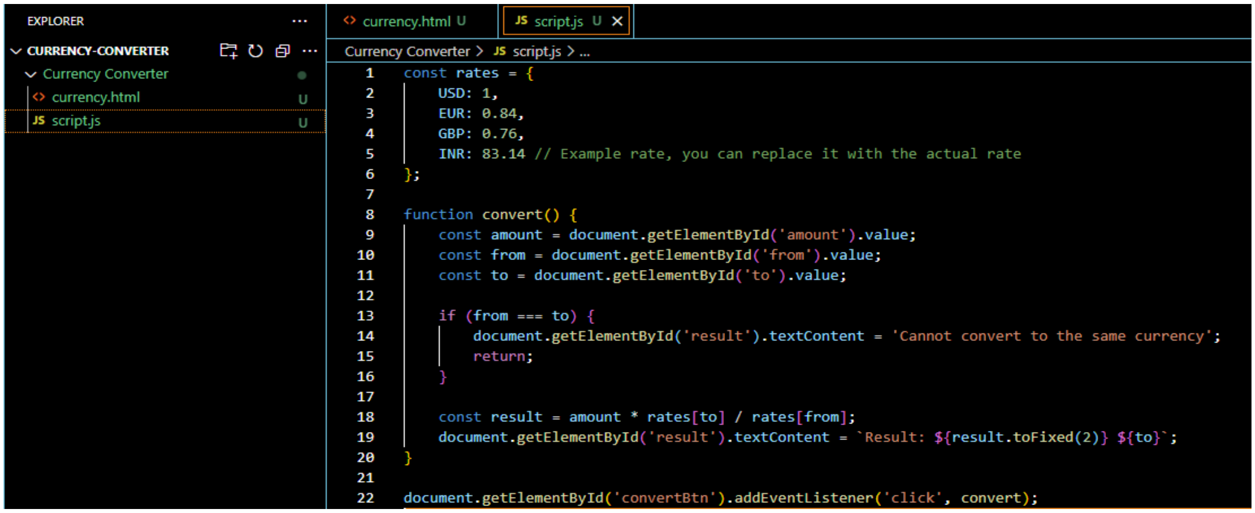
Step:1 Creating a Web application Currency Converter

- **HTML file**



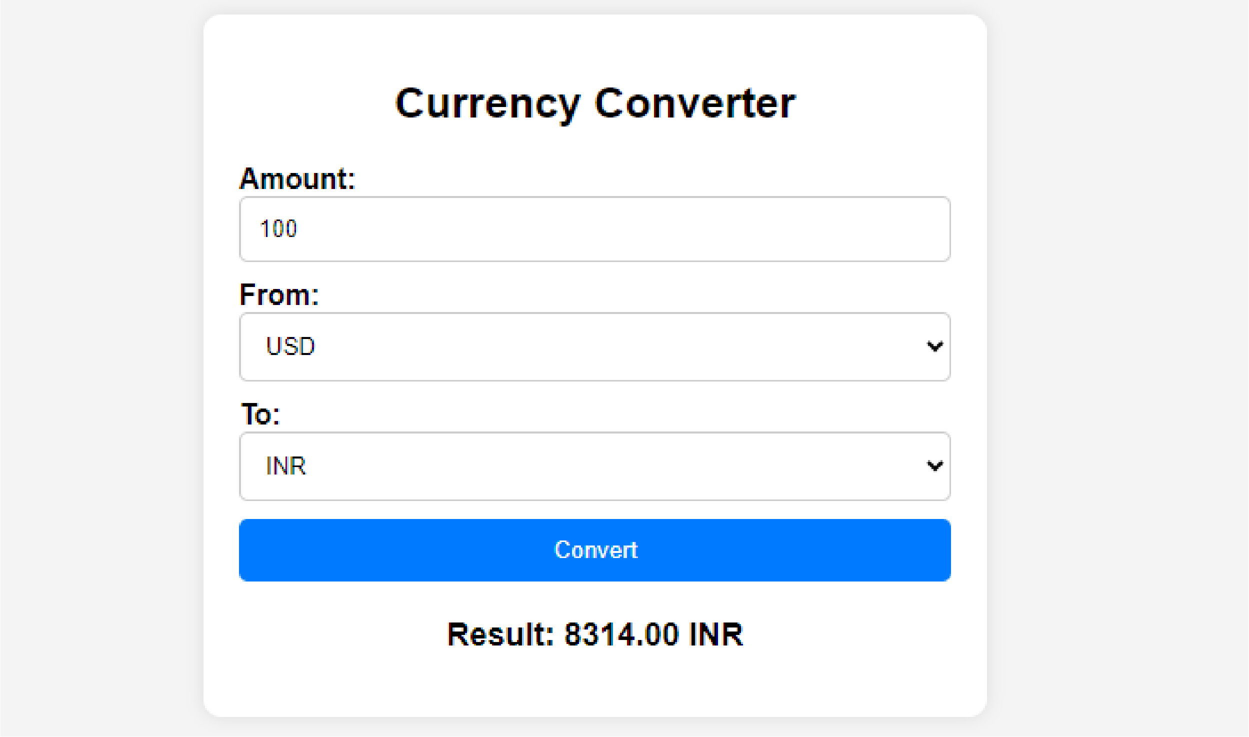
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Currency Converter</title>
</head>
<body>
    <h1>Currency Converter</h1>
    <label for="amount">Amount:</label>
    <input type="number" id="amount" required>
    <br>
    <label for="from">From:</label>
    <select id="from" required>
        <option value="USD">USD</option>
        <option value="EUR">EUR</option>
        <option value="GBP">GBP</option>
        <option value="INR">INR</option>
    </select>
    <br>
    <label for="to">To:</label>
    <select id="to" required>
        <option value="USD">USD</option>
        <option value="EUR">EUR</option>
        <option value="GBP">GBP</option>
        <option value="INR">INR</option>
    </select>
    <br>
    <button id="convertBtn">Convert</button>
    <h2 id="result"></h2>
    <script src="script.js"></script>
</body>
```

- **JavaScript**



```javascript
const rates = {
    USD: 1,
    EUR: 0.84,
    GBP: 0.76,
    INR: 83.14 // Example rate, you can replace it with the actual rate
};

function convert() {
    const amount = document.getElementById('amount').value;
    const from = document.getElementById('from').value;
    const to = document.getElementById('to').value;

    if (from === to) {
        document.getElementById('result').textContent = 'Cannot convert to the same currency';
        return;
    }

    const result = amount * rates[to] / rates[from];
    document.getElementById('result').textContent = `Result: ${result.toFixed(2)} ${to}`;
}

document.getElementById('convertBtn').addEventListener('click', convert);
```

Step:2 Output



Currency Converter

Amount: 100

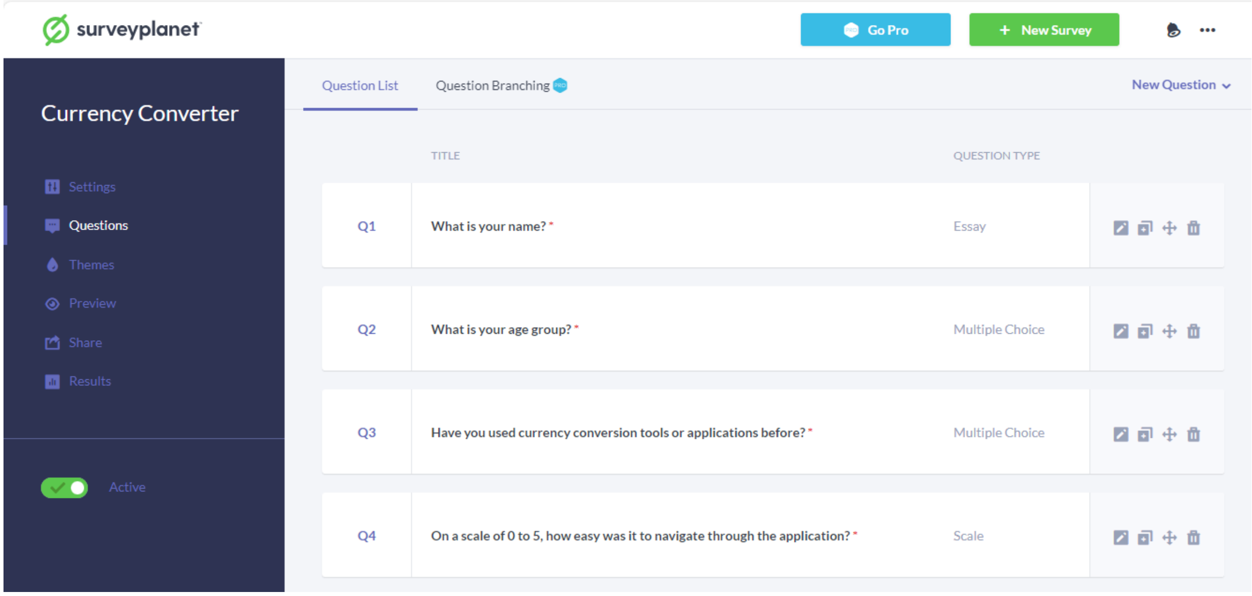
From: USD

To: INR

Result: 8314.00 INR

Task 2: Usability Testing

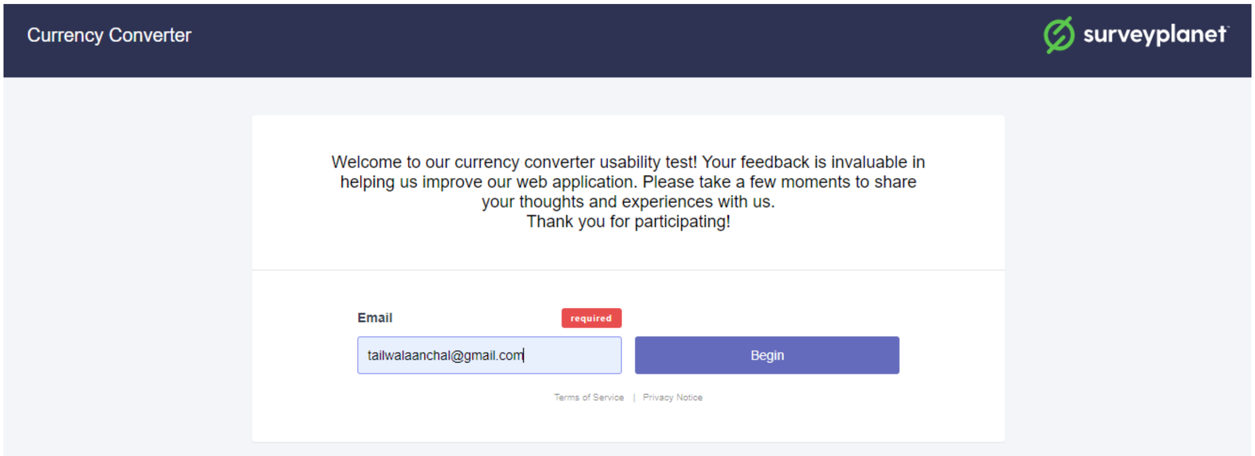
Step:1 Creating a Feedback Form using Survey Planet



Currency Converter

	TITLE	QUESTION TYPE
Q1	What is your name? *	Essay
Q2	What is your age group? *	Multiple Choice
Q3	Have you used currency conversion tools or applications before? *	Multiple Choice
Q4	On a scale of 0 to 5, how easy was it to navigate through the application? *	Scale

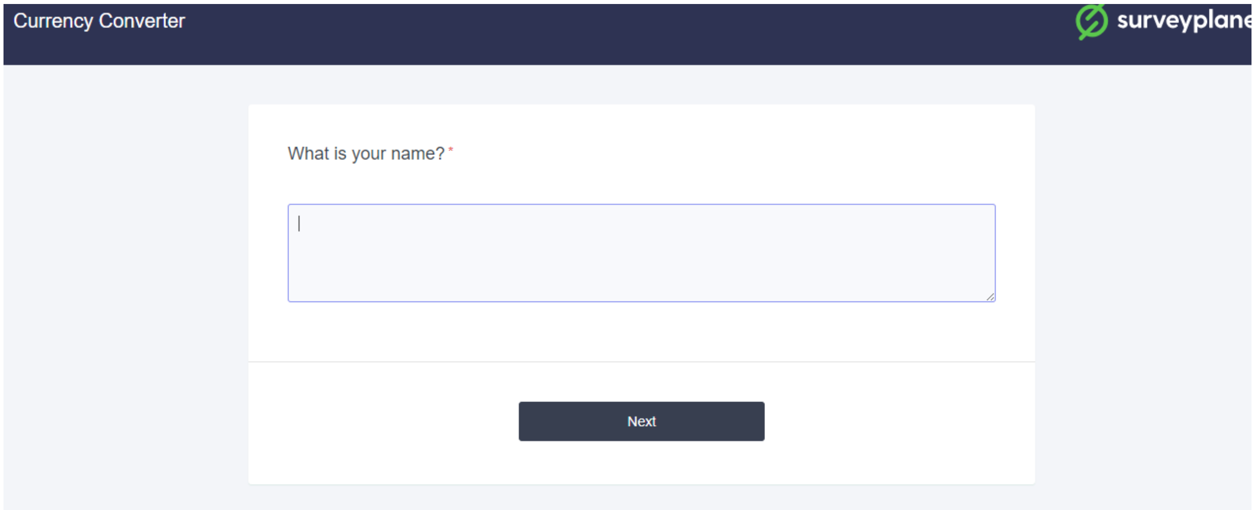
Step:2 Form Template



Welcome to our currency converter usability test! Your feedback is invaluable in helping us improve our web application. Please take a few moments to share your thoughts and experiences with us.
Thank you for participating!

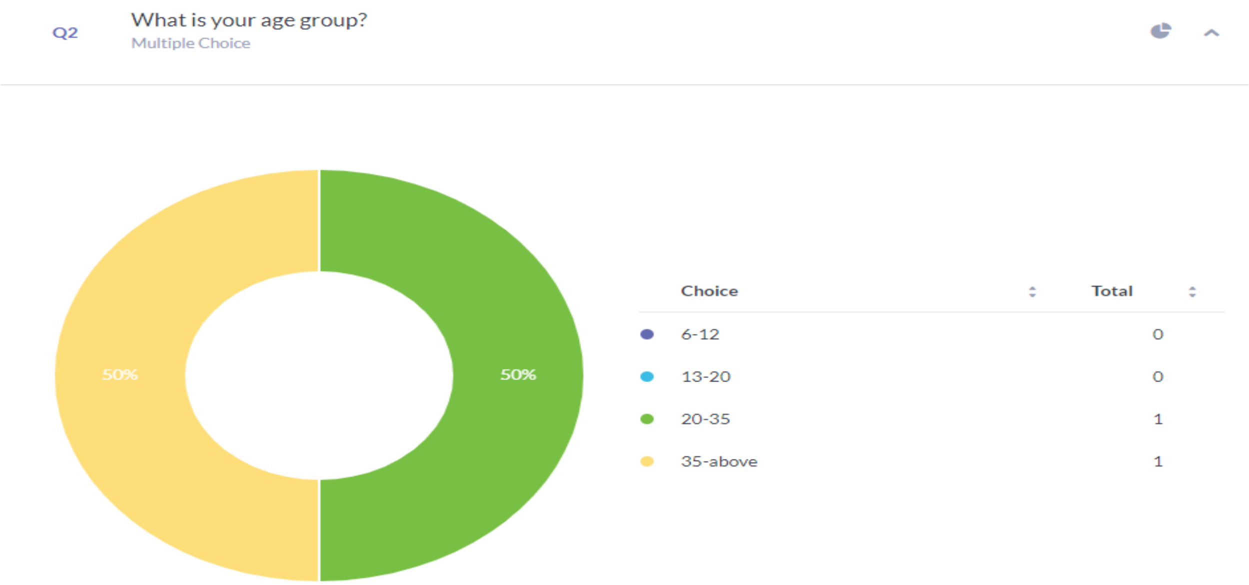
Email required

tailwalaanchal@gmail.com



What is your name? *

Step:3 Results



Q2 What is your age group?
Multiple Choice

	Choice	Total
●	6-12	0
●	13-20	0
●	20-35	1
●	35-above	1

Experiment 3

Usability Testing Report Generation

Currency Converter Test

Aanchal Tailwal

5th Feb 2024

Table of Contents

Introduction

Introduction:

Our currency converter web application serves as a user-friendly tool for quickly converting between different currencies. Its purpose is to provide users with a convenient way to perform currency conversions accurately and efficiently. This usability test aims to evaluate how effectively our interface facilitates users in completing routine currency conversion tasks. Participants will be asked to perform a series of common tasks using the application. Session recordings will be analyzed to identify areas for improvement and enhance the user experience of our web application.

The usability test for our currency converter web application was conducted online by the moderator, Aanchal Tailwal. There were six participants involved in the session. The test was conducted remotely using web conferencing software. The moderator guided participants through a series of tasks while observing their interactions with the application. Session data included participants' navigational choices, task completion rates, comments, overall satisfaction ratings, questions, and feedback.

Executive Summary

The usability test for our currency converter web application was conducted online by the moderator on 5th Feb 2024. Six participants took part in the session, which was conducted remotely via web conferencing software. The purpose of the test was to evaluate the usability of the currency converter interface and identify any areas for improvement in terms of user experience.

Each session lasted approximately for 10-15 min, during which participants were guided through a series of tasks to assess their interaction with the application. Overall, participants found the currency converter web application to be user-friendly, with 80% reporting that the application was easy to use.

Key findings from the usability test included:

- Ease of navigation was rated highly.
- The currency conversion process was found to be straightforward.
- Participants expressed satisfaction with the design and layout of the application.
- Some participants reported encountering technical issues.
- All participants found the provided currency exchange rates to be accurate.
- Most participants indicated they would recommend the currency converter application to others.

- Feedback for improvement was generally positive, with specific suggestions not provided by most participants. However, one participant suggested shutting down the app.

This document contains detailed feedback from participants, including satisfaction ratings, task completion rates, ease or difficulty of completion ratings, time on task, errors, and recommendations for improvements. Additionally, scenarios and questionnaires used during the test are included in the Attachments section for reference.

Methodology

Sessions

Participants for the usability testing of the currency converter web application were selected by our teacher for our class assessment. Each individual session lasted approximately 10-15 minutes. Before the session, I explained to participants how the test would proceed and what they would be asked to do.

During the session, participants were asked to use the currency converter web application and perform various tasks such as converting currencies, navigating through the application, and providing feedback on their experience.

After the session, participants were asked a series of questions to gather feedback on their experience using the application. I prepared a form with a set of questions, and participants were asked to respond to these questions based on their interaction with the application during the testing session.

The post-session questions included aspects such as ease of use, navigation, satisfaction with the design/layout, any technical issues encountered, accuracy of exchange rates, likelihood of recommending the application to others, and any additional feedback or suggestions for improvement.

Participants were encouraged to provide detailed feedback to help identify areas for improvement in the currency converter application.

Participants

All participants were students enrolled in BTech CSE/ DevOps B4.

Six participants were scheduled for the usability testing sessions. All six participants completed the test.

Role

Participant	Name	Age Group	Gender	Previous Experience with Currency Conversion Tools	
1	Shashank Saurabh	20-35	Male	Yes	-
2	Aakshita Singh	20-35	Female	Yes	-
3	Abhinav Kumar	20-35	Male	No	-
4	Yadrishi Dixit	20-35	Female	Yes	-
5	Shivam Singh	20-35	Male	No	-
6	Shivam Kumar	20-35	Male	Yes	-

Evaluation Tasks/Scenarios

The task scenarios were created by Moderator Aanchal Tailwal. Participants were asked to complete the following tasks:

- Convert USD to EUR.
- Convert EUR to GBP.
- Convert GBP to INR.
- Convert USD to GBP.
- Convert USD to INR.
- Convert GBP to EUR.
- Convert INR to EUR.

These tasks were designed to evaluate the participants' ability to use the currency converter web application to perform currency conversions accurately and efficiently.

Results

Task Completion Success Rate

All participants successfully completed Task 1 (convert USD to EUR) and Task 2 (convert EUR to INR). Six out of six participants (100%) completed Task 3 (convert

INR to GBP), Task 4 (convert GBP to USD), and Task 5 (convert USD to EUR). Task 6 (convert EUR to GBP) also had a completion rate of 100%.

Task Ratings:

Ease in Finding Information:

Participants found it easy to complete all tasks, with a mean agreement rating ranging from 4.0 to 4.7.

Keeping Track of Location in Site:

All participants found it easy to keep track of their location while completing all tasks, with a mean agreement rating ranging from 4.0 to 4.7.

Predicting Information Section:

Participants found it easy to predict which section of the website contained the information for all tasks, with a mean agreement rating ranging from 4.3 to 4.7.

Test 1 – Mean Task Ratings & Percent Agree

Task	Ease – Finding Info	Location in Site	Predict Section	Overall
1 – Convert USD to EUR	4.7 (100%)	4.7 (100%)	4.7 (100%)	4.7
2 – Convert EUR to INR	4.7 (100%)	4.7 (100%)	4.7 (100%)	4.7
3 – convert INR to GBP	4.7 (100%)	4.7 (100%)	4.7 (100%)	4.7
4 – convert GBP to USD	4.7 (100%)	4.7 (100%)	4.7 (100%)	4.7
5 – convert USD to EUR	4.7 (100%)	4.7 (100%)	4.7 (100%)	4.7
6 – convert EUR to GBP	4.7 (100%)	4.7 (100%)	4.7 (100%)	4.7

**Percent Agree (%) = Agree & Strongly Agree Responses combined*

Time on Task:

- Task 1 (Convert USD to EUR): Participants spent an average of 82 seconds completing this task, with completion times ranging from 50 to 310 seconds.
- Task 2 (Convert EUR to INR): Participants spent an average of 200 seconds completing this task, with completion times ranging from 50 to 390 seconds.
- Task 3 (Convert INR to GBP): Participants spent an average of 67 seconds completing this task, with completion times ranging from 15 to 215 seconds.

- Task 4 (Convert GBP to USD): Participants spent an average of 129 seconds completing this task, with completion times ranging from 55 to 240 seconds.

- Task 5 (Convert USD to EUR): Participants spent an average of 69 seconds completing this task, with completion times ranging from 29 to 127 seconds.

- Task 6 (Convert EUR to GBP): Participants spent an average of 210 seconds completing this task, with completion times ranging from 110 to 465 seconds.

Errors:

The number of errors participants made while trying to complete the task scenarios was captured by the moderator.

Task 4 (Convert GBP to USD) had the most errors, with participants making a total of 9 errors. Task 2 (convert EUR to GBP) also had a high number of errors, with participants making a total of 10 errors. However, all errors made were non-critical and did not prevent successful completion of the scenarios.

Task	Task Completion	Errors	Time on Task	Satisfaction*
1	6	2	87	4.33
2	6	10	102	6.37
3	6	3	75	4.00
4	6	9	120	3.33
5	6	2	90	4.33
6	6	4	150	6.00

Overall Metrics

After completing the task sessions, participants rated the currency converter web application for eight overall measures. The highest percent of 'agreed' satisfaction ratings were as follows:

Most participants (85%) agreed that they would use the currency converter application frequently.

The majority of participants (77%) agreed that they could get currency conversion information quickly.

Participants generally agreed (62%) that the website was well-organized.

However, the variables that received the lowest satisfaction ratings were:

Only 54% of participants agreed that the homepage's content made them want to explore the site.

A low percentage (8%) of participants found it difficult to keep track of where they were in the application.

**Percent Agree (%) = Agree & Strongly Agree Responses combined*

Recommendations for Improvement

No need to improve

No

No

Shut your app down

No all good

Recommendations

The recommendations section offers suggestions for enhancing the user experience based on participant feedback, behaviors, and success rates. Each recommendation is accompanied by a severity rating. These recommendations aim to enhance overall usability and address areas where participants encountered difficulties or confusion with the interface and information architecture.

For example:

Find Organizational or Individual Funding Information (Task 2)

Task 2 required participants to find organization funding (Test 1) or individual funding (Test 2).

Change	Justification	Severity
• Add categories to funding pages	Participants rated the ease of finding funding information poorly, indicating a lack of clarity and organization. Categorizing funding opportunities will enhance navigation.	High
• Add additional descriptive text	By adding descriptive text on the funding opportunities homepage, users will have clearer guidance on the types of funding available, improving overall usability.	High

Conclusion

Overall, participants expressed positive feedback regarding the currency converter web application, highlighting its ease of use and accurate currency exchange rates. The majority of participants indicated they would use the application frequently and found it easy to navigate. However, there were areas for improvement identified, particularly regarding the clarity of funding information. Implementing the recommended changes, such as adding categories to funding pages and providing additional descriptive text, can enhance the overall user experience. Continuous collaboration with users will ensure that the application remains user-centered and addresses evolving user needs.

Attachments:

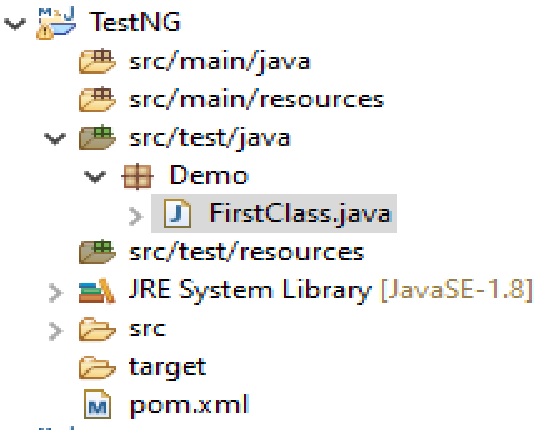
- Attachment A: Background Questionnaire

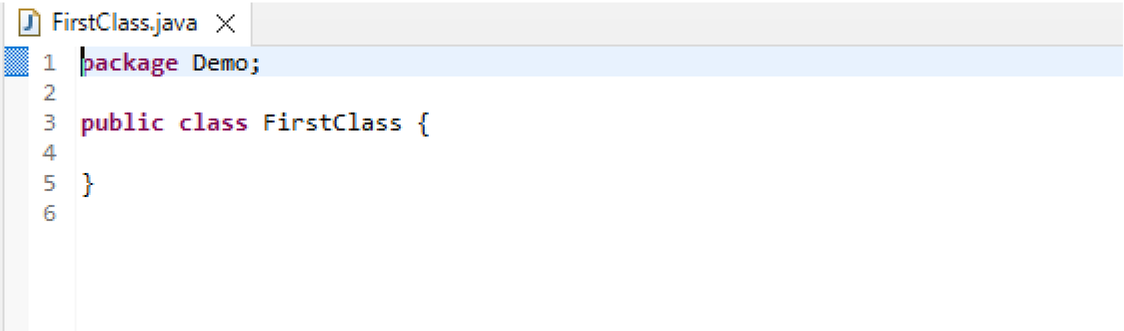
https://s.surveyplanet.com/xpyokxfq

Experiment 4

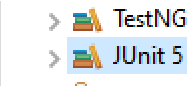
Getting Introduced to Test NG framework

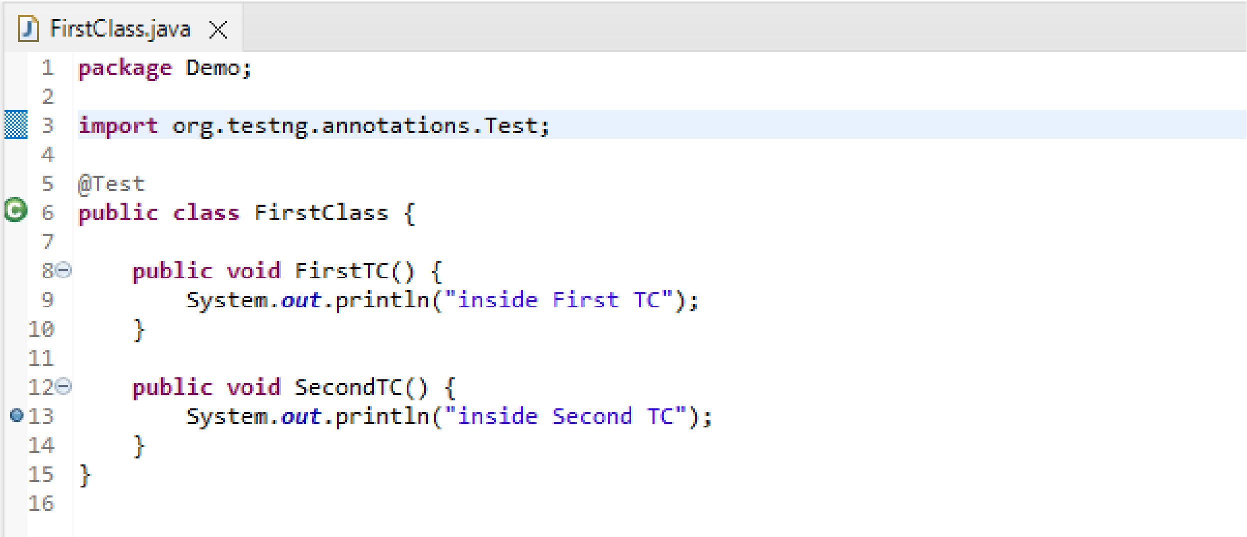
Task1: Creating Test cases and run it using TESTNG Framework



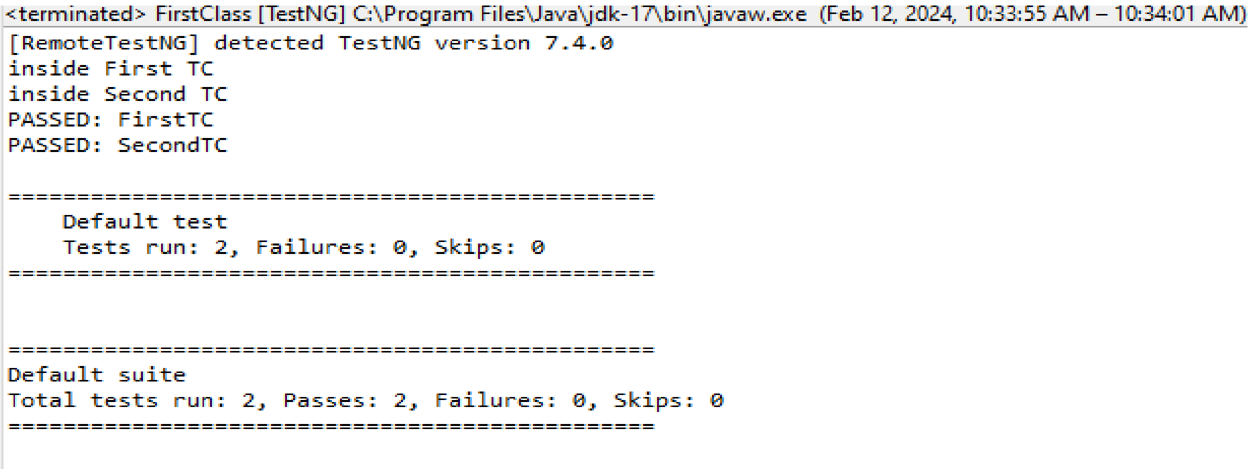

FirstClass.java

```java
package Demo;

public class FirstClass {

}
```



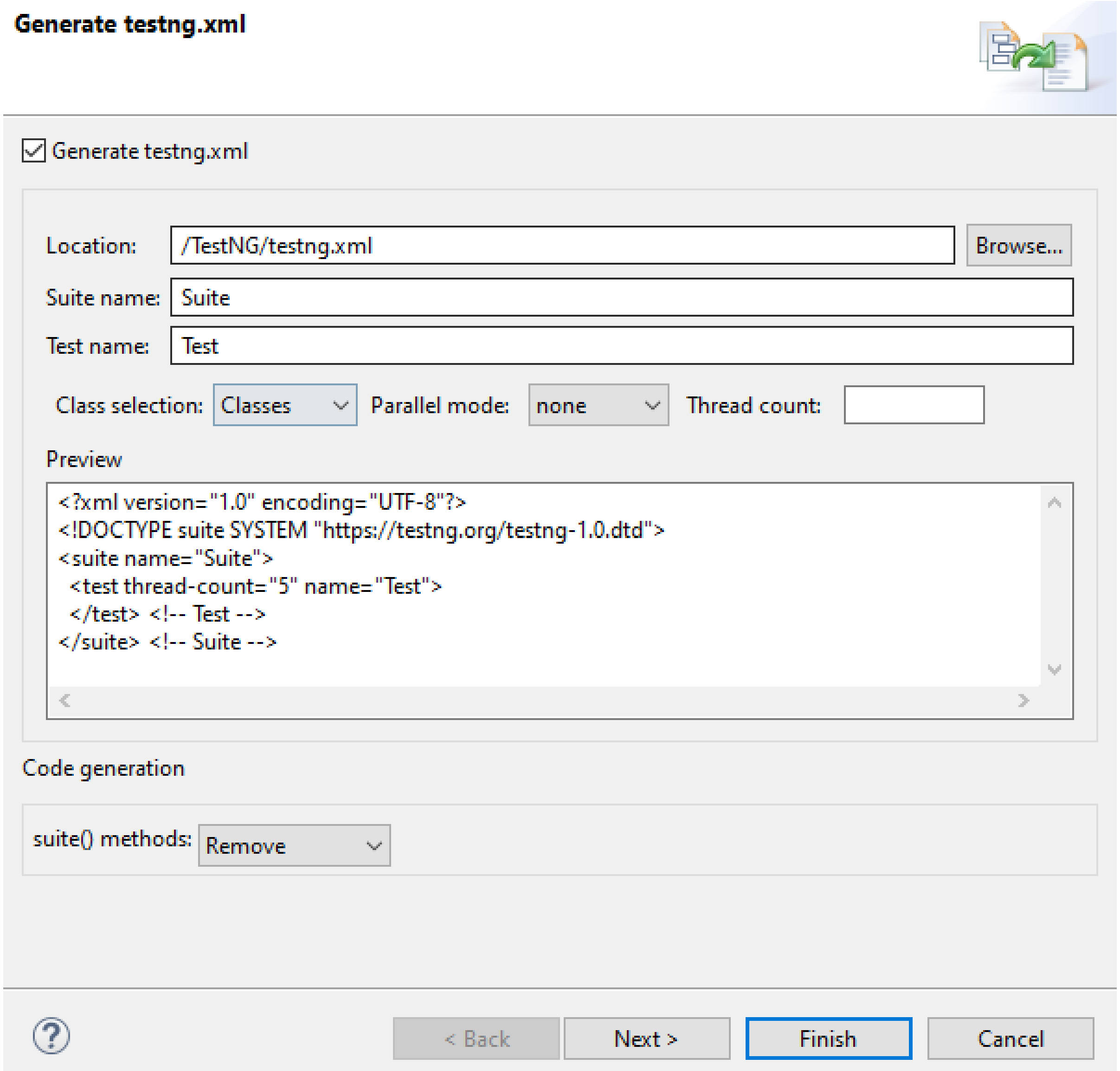


```java
package Demo;

import org.testng.annotations.Test;

@Test
public class FirstClass {

    public void FirstTC() {
        System.out.println("inside First TC");
    }

    public void SecondTC() {
        System.out.println("inside Second TC");
    }
}

```



```
<terminated> FirstClass [TestNG] C:\Program Files\Java\jdk-17\bin\javaw.exe  (Feb 12, 2024, 10:33:55 AM – 10:34:01 AM)
[RemoteTestNG] detected TestNG version 7.4.0
inside First TC
inside Second TC
PASSED: FirstTC
PASSED: SecondTC

===============================================
    Default test
    Tests run: 2, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
===============================================
```

Task2: Creating Testing.XML and running it using TestNG framework



Generate testng.xml

☑ Generate testng.xml

Location: /TestNG/testng.xml

Suite name: Suite

Test name: Test

Class selection: Classes

Parallel mode: none

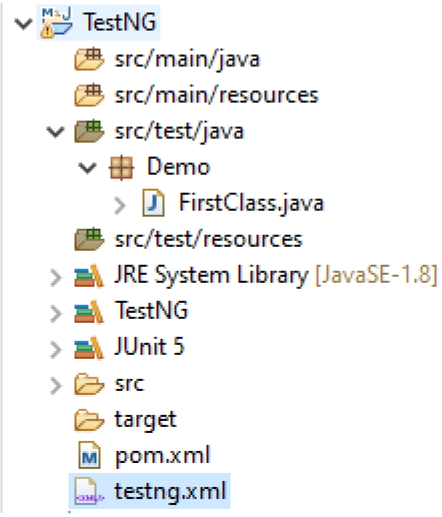
Thread count:

Preview

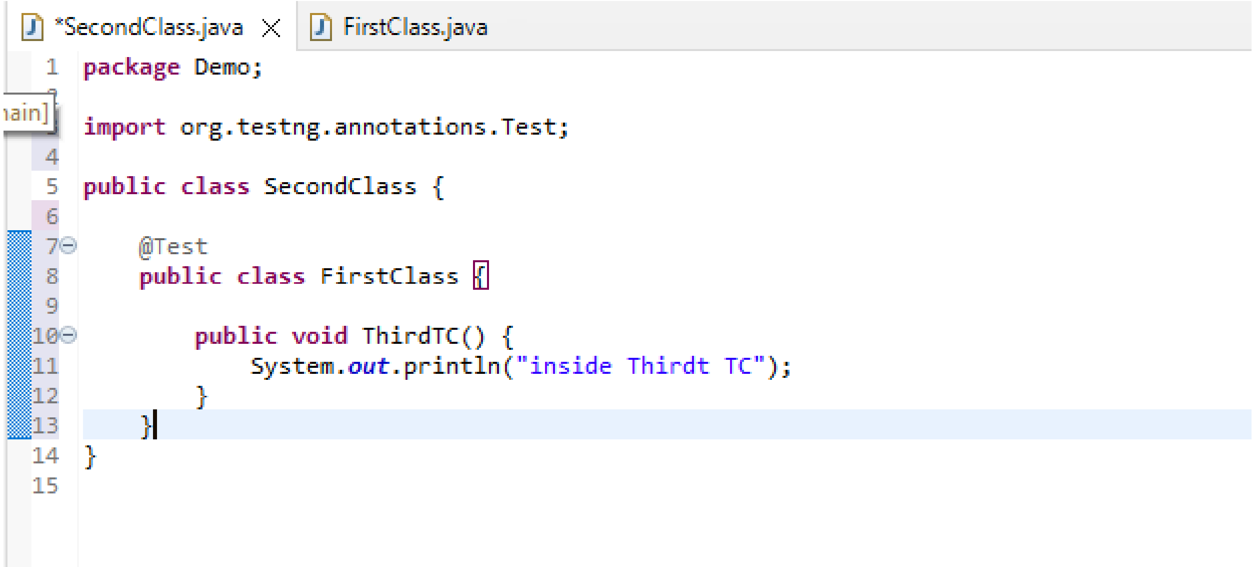
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="Test">
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

Code generation

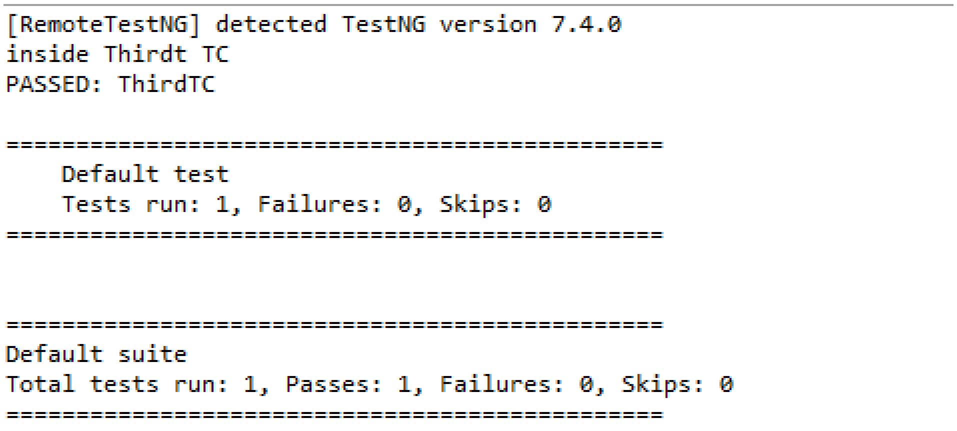
suite() methods: Remove



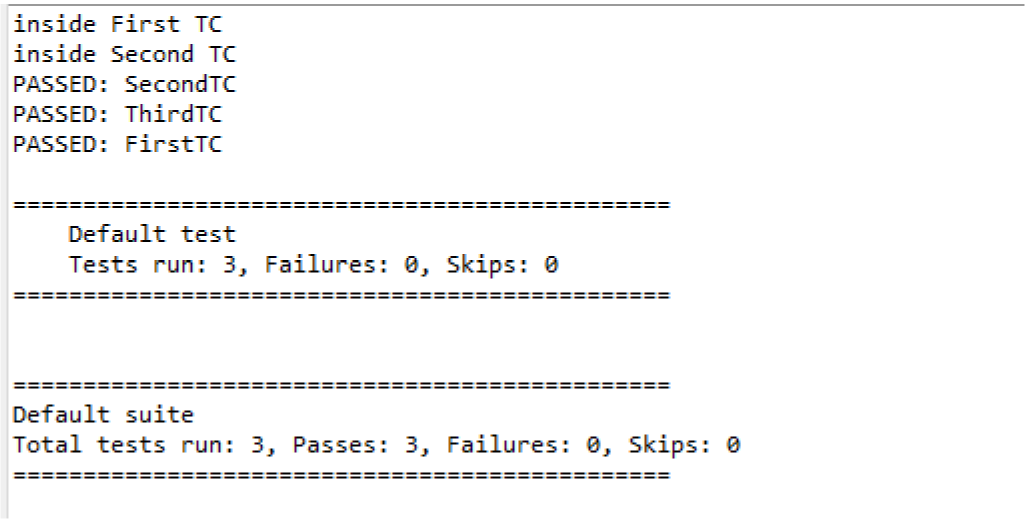
```
TestNG
    src/main/java
    src/main/resources
    src/test/java
        Demo
            FirstClass.java
    src/test/resources
    JRE System Library [JavaSE-1.8]
    TestNG
    JUnit 5
    src
    target
    pom.xml
    testng.xml
```



```java
package Demo;

import org.testng.annotations.Test;

public class SecondClass {

    @Test
    public class FirstClass {

        public void ThirdTC() {
            System.out.println("inside Thirdt TC");
        }
    }
}
```



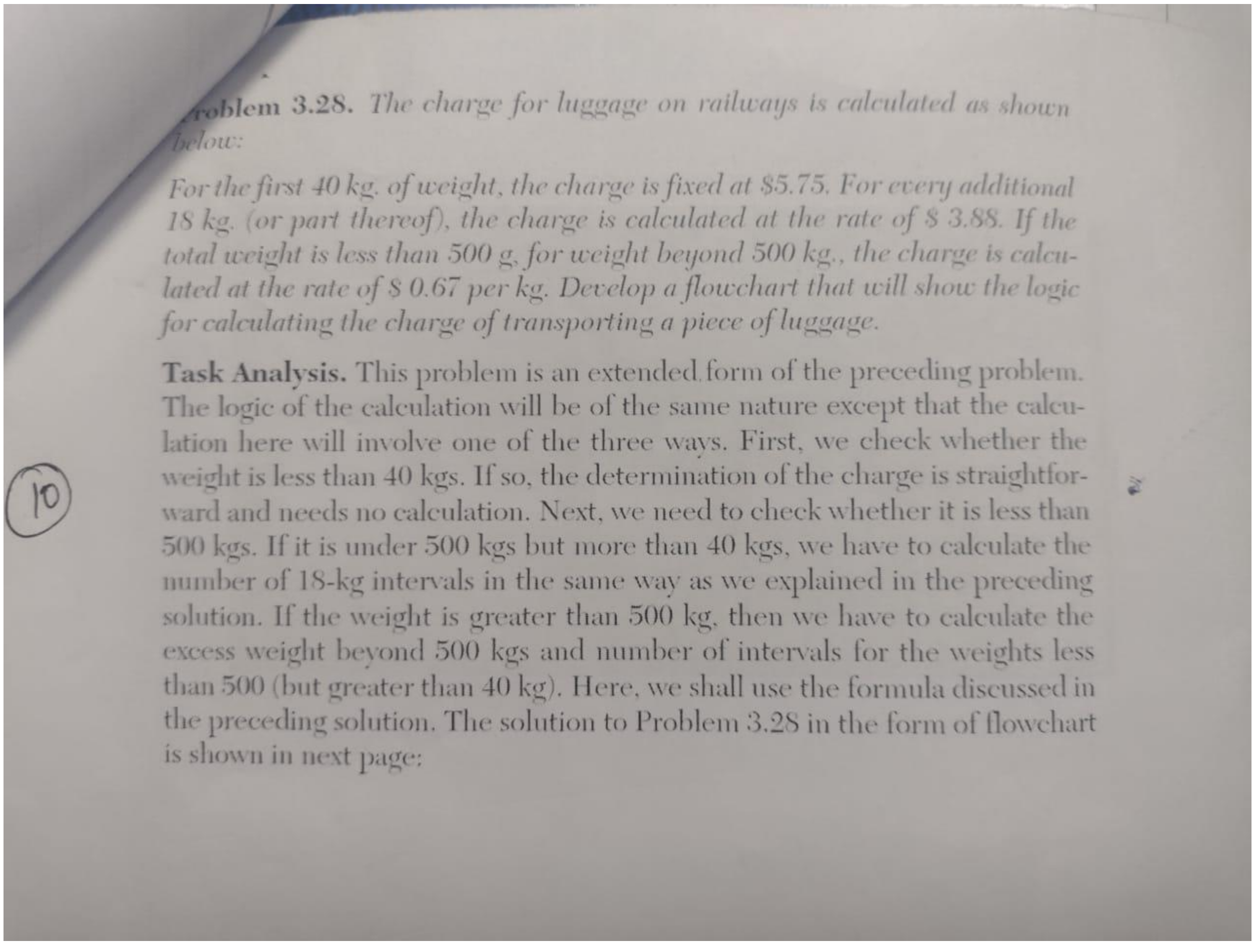
```
[RemoteTestNG] detected TestNG version 7.4.0
inside Thirdt TC
PASSED: ThirdTC

===============================================
    Default test
    Tests run: 1, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
===============================================
```



```
inside First TC
inside Second TC
PASSED: SecondTC
PASSED: ThirdTC
PASSED: FirstTC

===============================================
    Default test
    Tests run: 3, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 3, Passes: 3, Failures: 0, Skips: 0
===============================================
```

Experiment 5

Mid Semester Assessment

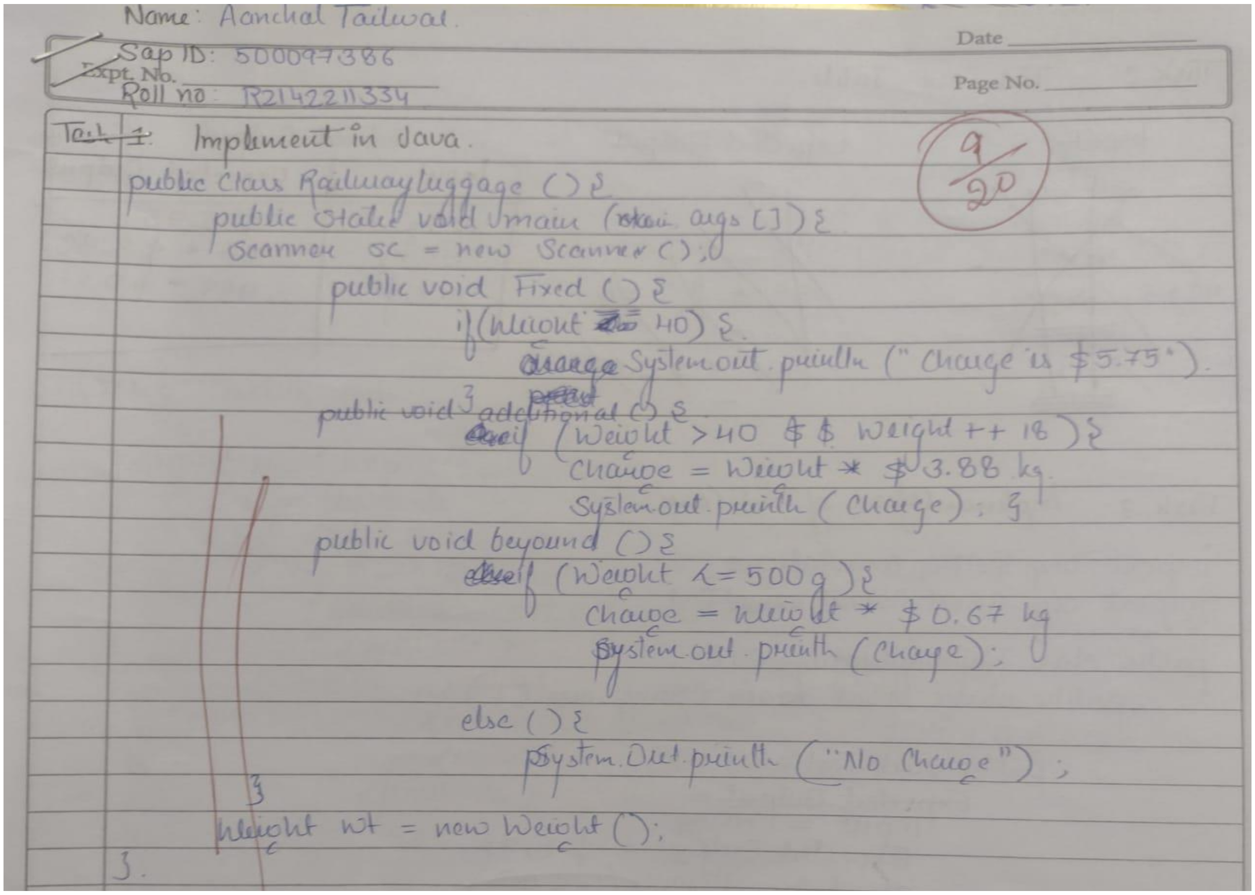


roblem 3.28. *The charge for luggage on railways is calculated as shown below:*

For the first 40 kg. of weight, the charge is fixed at $5.75. For every additional 18 kg. (or part thereof), the charge is calculated at the rate of $ 3.88. If the total weight is less than 500 g. for weight beyond 500 kg., the charge is calculated at the rate of $ 0.67 per kg. Develop a flowchart that will show the logic for calculating the charge of transporting a piece of luggage.

Task Analysis. This problem is an extended form of the preceding problem. The logic of the calculation will be of the same nature except that the calculation here will involve one of the three ways. First, we check whether the weight is less than 40 kgs. If so, the determination of the charge is straightforward and needs no calculation. Next, we need to check whether it is less than 500 kgs. If it is under 500 kgs but more than 40 kgs, we have to calculate the number of 18-kg intervals in the same way as we explained in the preceding solution. If the weight is greater than 500 kg, then we have to calculate the excess weight beyond 500 kgs and number of intervals for the weights less than 500 (but greater than 40 kg). Here, we shall use the formula discussed in the preceding solution. The solution to Problem 3.28 in the form of flowchart is shown in next page:

10



Name: Aanchal Tailwal.
~~Expt. No.~~ Sap ID: 500097386
Roll no: R2142211334

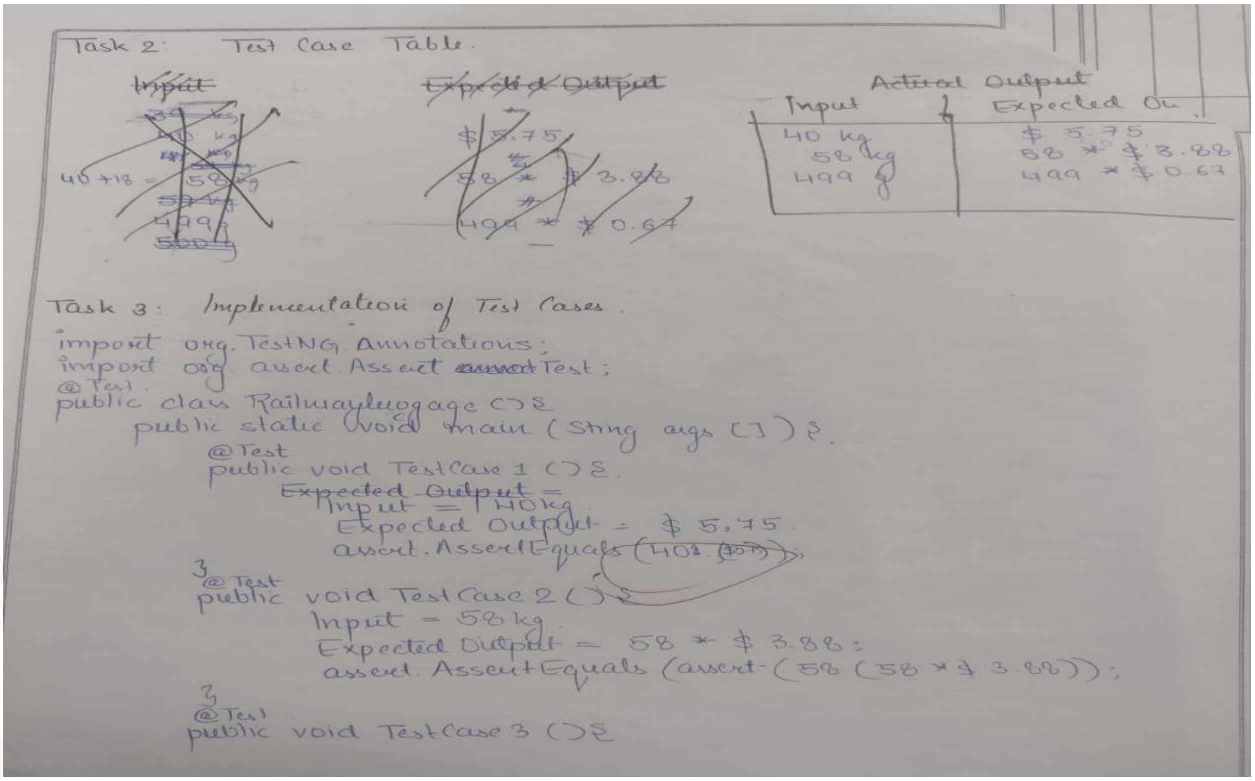
Date

Page No.

9/20

~~Task~~ 1: Implement in Java.

```
public Class Railwayluggage () {
    public Static void main (~~str~~ args []) {
    Scanner sc = new Scanner ();
            public void Fixed () {
                    if(Weight <= 40) {
                        ~~charge~~ System.out.println (" Charge is $5.75").
            public void } ~~print~~ additional () {
                    ~~else~~if (Weight >40 $$ Weight ++ 18 ){
                        Charge = Weight * $ 3.88 kg.
                        System.out.println ( Charge) ; }
            public void beyound () {
                    ~~else~~if (Weight <= 500g ){
                        Charge = Weight * $ 0.67 kg
                        System.out.println (Charge);

                    else () {
                        System.Out.println ("No Charge") ;
        }
    Weight wt = new Weight ();
}.
```



Task 2: Test Case Table.

~~Input~~

~~39 kg~~
~~40 kg~~
~~48 kg~~
~~40 +18 = 58 kg~~
~~59 kg~~
~~499 g~~
~~500 g~~

~~Expected Output~~

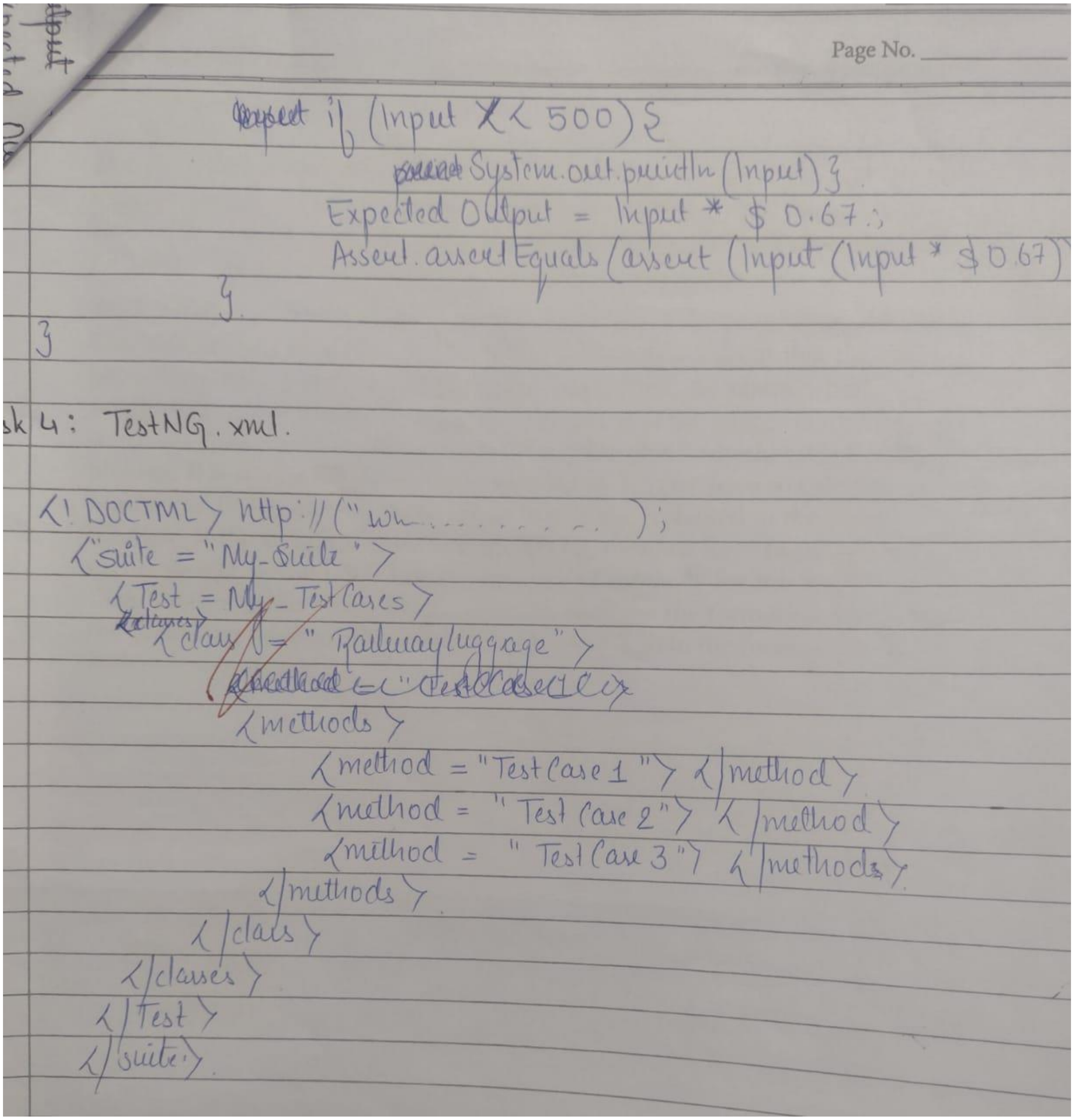
~~$ 5.75~~
~~58 * $ 3.88~~
~~499 * $ 0.67~~

~~Actual Output~~

Input	Expected Ou.
40 kg 58 kg 499 g	$ 5.75 58 * $ 3.88 499 * $ 0.67

Task 3: Implementation of Test Cases.

```
import org.TestNG.Annotations;
import org.assert.Assert ~~assert~~Test;
@Test.
public class Railwayluggage () {
    public static void main (String args []) {
        @Test
        public void TestCase 1 () {
            ~~Expected Output =~~
            Input = 40kg
            Expected Output = $ 5,75.
            assert.AssertEquals (40$. ($5.75));
        }
        @Test
        public void TestCase 2 () {
            Input = 58 kg
            Expected Output = 58 * $ 3.88;
            assert.AssertEquals (assert-(58 (58 * $ 3.88));
        }
        @Test
        public void TestCase 3 () {
```



Page No. ______

```
if (Input ~~X~~< 500){
        System.out.println(Input)}
    Expected Output = Input * $ 0.67;
    Assert.assertEquals(assert (Input (Input * $0.67))
  }.
}
```

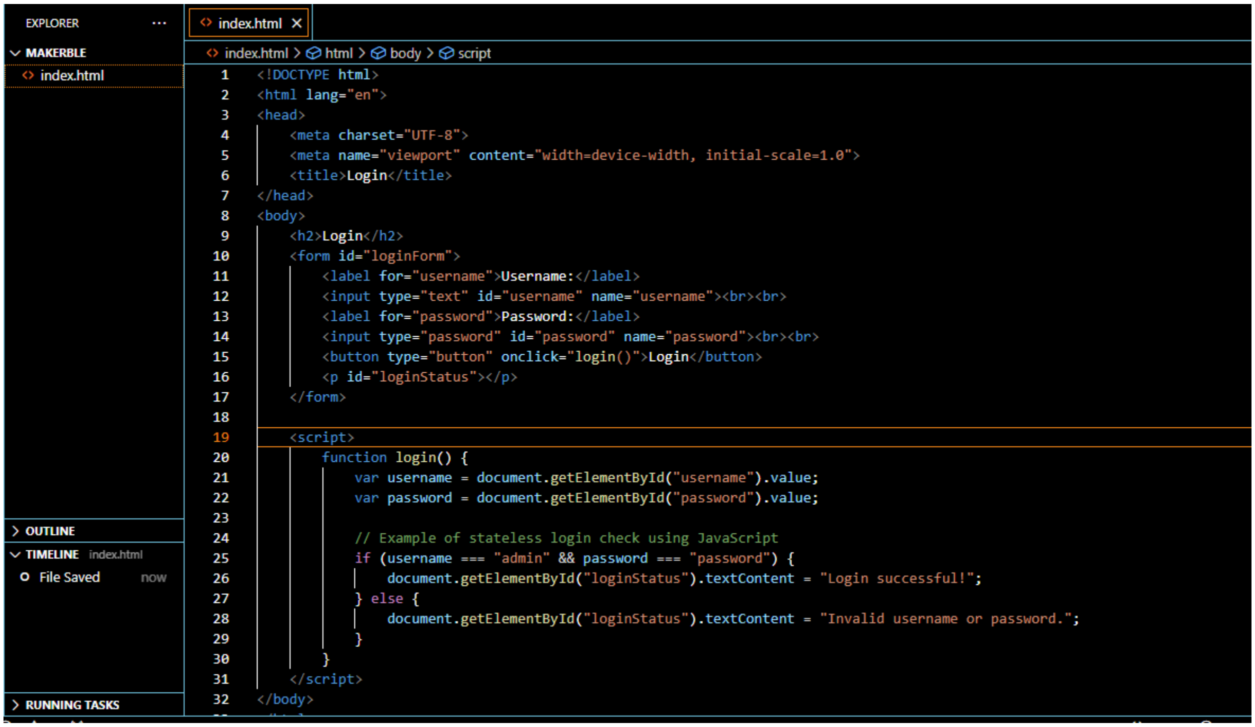
sk 4: TestNG.xml.

```
<! DOCTML> http://("wn........ );
  <"suite = "My-Suite" >
    <Test = My_TestCases>
    <classes>
      <class = "Railwayluggage">
        ~~<method = "TestCase1"<>~~
        <methods>
            <method = "TestCase 1"> </method>
            <method = " Test Case 2"> </method>
            <method = " TestCase 3"> </methods>.
        </methods>
      </class>
    </classes>
  </Test>
</suite>.
```

Experiment 6

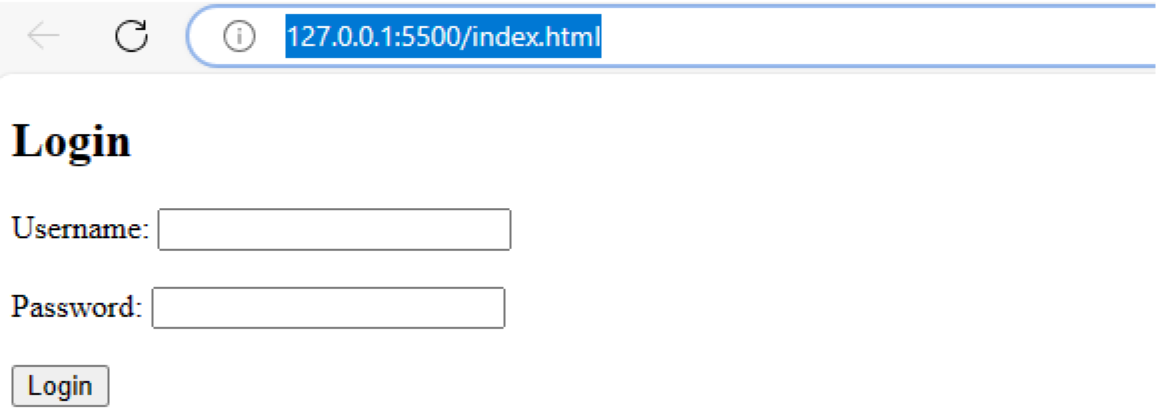
Task1: Create a Web application for stateless Login Checking

➔ Html code for an application



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
</head>
<body>
    <h2>Login</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="button" onclick="login()">Login</button>
        <p id="loginStatus"></p>
    </form>

    <script>
        function login() {
            var username = document.getElementById("username").value;
            var password = document.getElementById("password").value;

            // Example of stateless login check using JavaScript
            if (username === "admin" && password === "password") {
                document.getElementById("loginStatus").textContent = "Login successful!";
            } else {
                document.getElementById("loginStatus").textContent = "Invalid username or password.";
            }
        }
    </script>
</body>
```

➔ output


127.0.0.1:5500/index.html

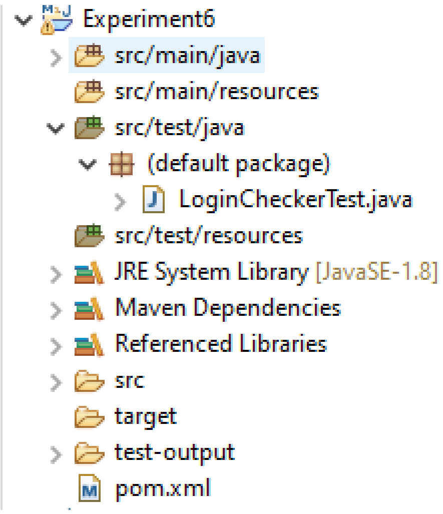
Login

Username:

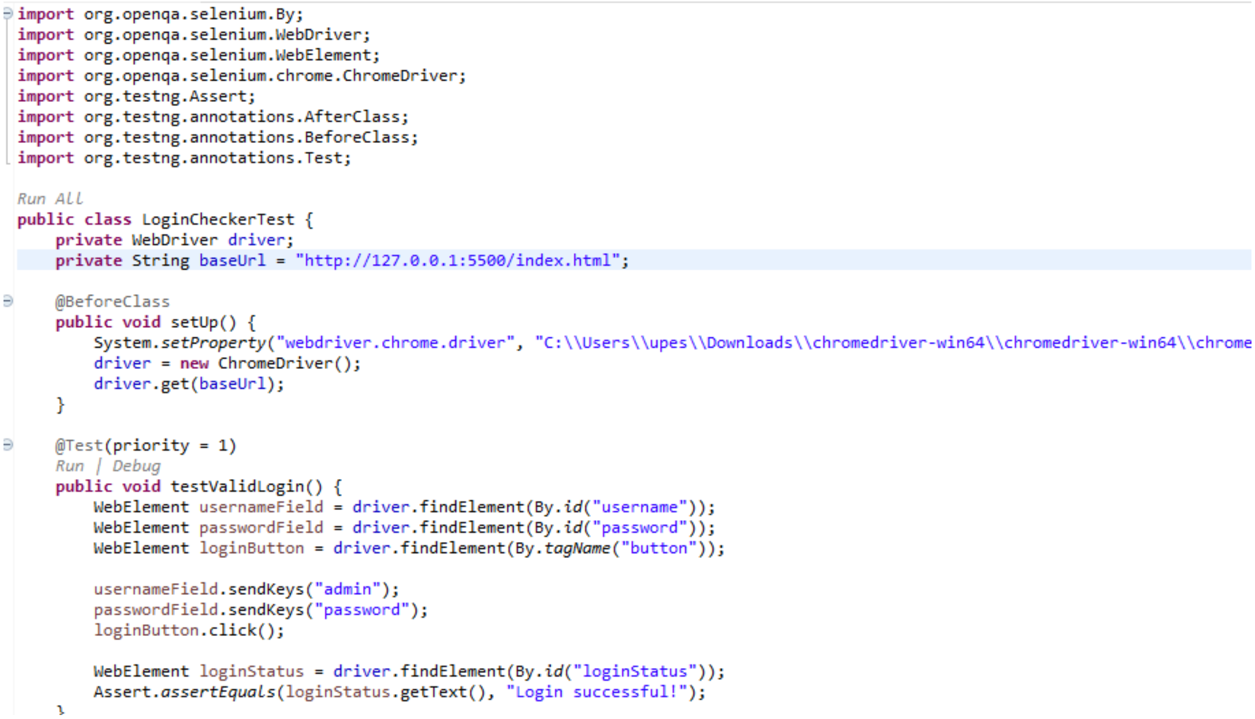
Password:

Task2: Create test-case to check the application using TestNG & Selenium

➔ Java Code with Test cases

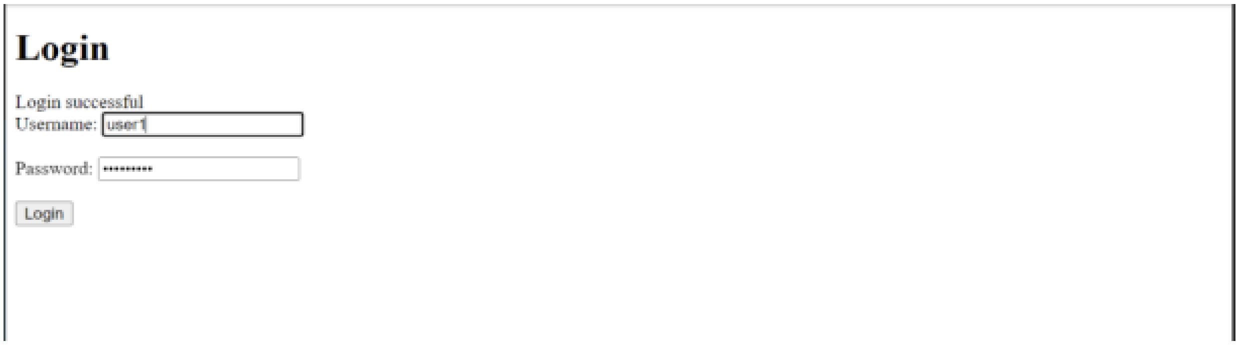


```
Experiment6
  src/main/java
  src/main/resources
  src/test/java
    (default package)
      LoginCheckerTest.java
  src/test/resources
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  Referenced Libraries
  src
  target
  test-output
  pom.xml
```



```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.Assert;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;

Run All
public class LoginCheckerTest {
    private WebDriver driver;
    private String baseUrl = "http://127.0.0.1:5500/index.html";

    @BeforeClass
    public void setUp() {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\upes\\Downloads\\chromedriver-win64\\chromedriver-win64\\chrome
        driver = new ChromeDriver();
        driver.get(baseUrl);
    }

    @Test(priority = 1)
    Run | Debug
    public void testValidLogin() {
        WebElement usernameField = driver.findElement(By.id("username"));
        WebElement passwordField = driver.findElement(By.id("password"));
        WebElement loginButton = driver.findElement(By.tagName("button"));

        usernameField.sendKeys("admin");
        passwordField.sendKeys("password");
        loginButton.click();

        WebElement loginStatus = driver.findElement(By.id("loginStatus"));
        Assert.assertEquals(loginStatus.getText(), "Login successful!");
    }
```

➔ Output

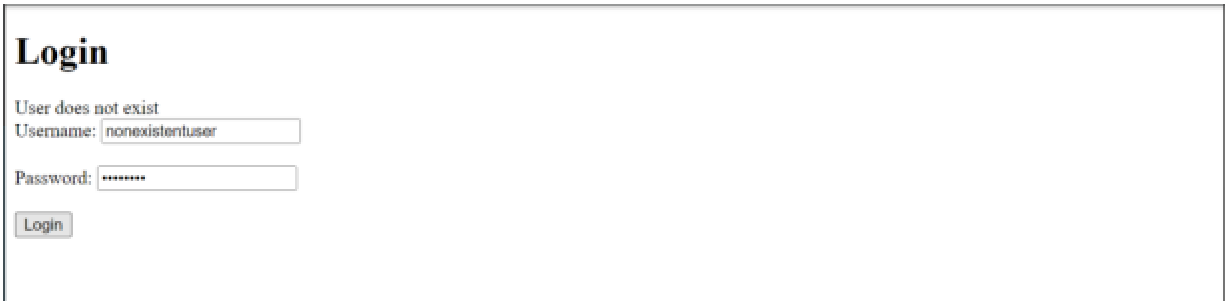


Login

Login successful

Username: user1

Password: ••••••••••

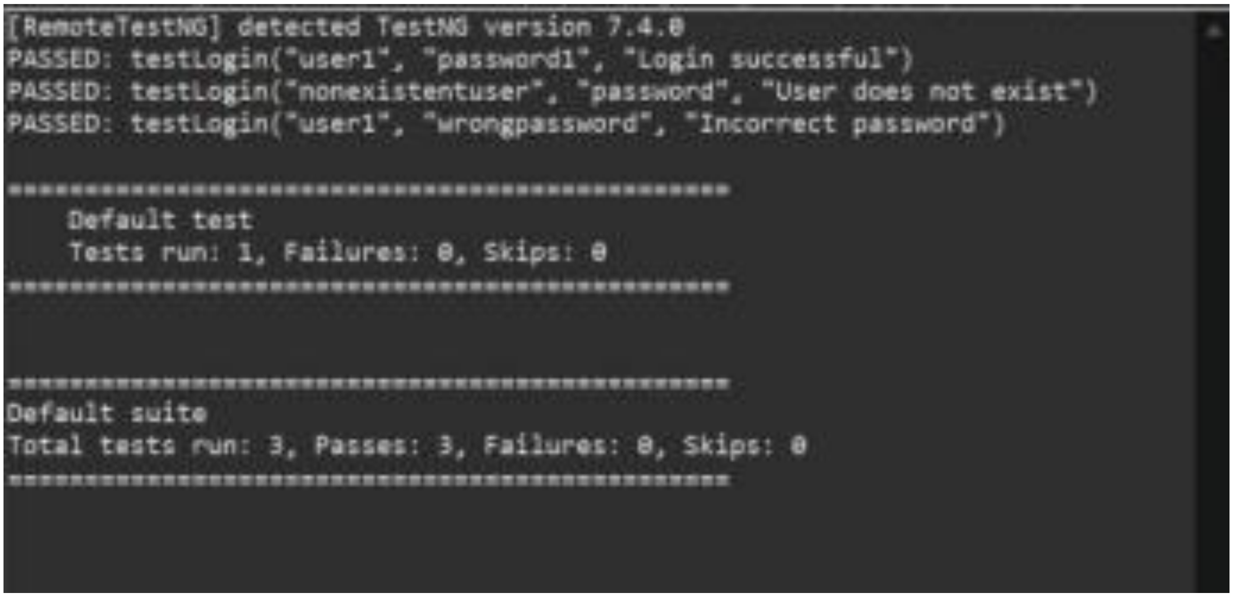


Login

User does not exist

Username: nonexistentuser

Password: ••••••••

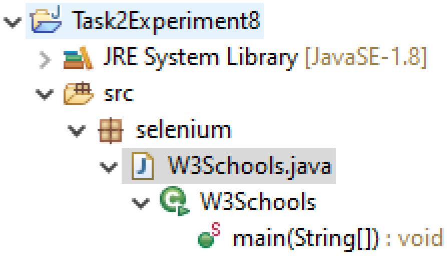


```
[RemoteTestNG] detected TestNG version 7.4.0
PASSED: testLogin("user1", "password1", "Login successful")
PASSED: testLogin("nonexistentuser", "password", "User does not exist")
PASSED: testLogin("user1", "wrongpassword", "Incorrect password")

===============================================
    Default test
    Tests run: 1, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 3, Passes: 3, Failures: 0, Skips: 0
===============================================
```

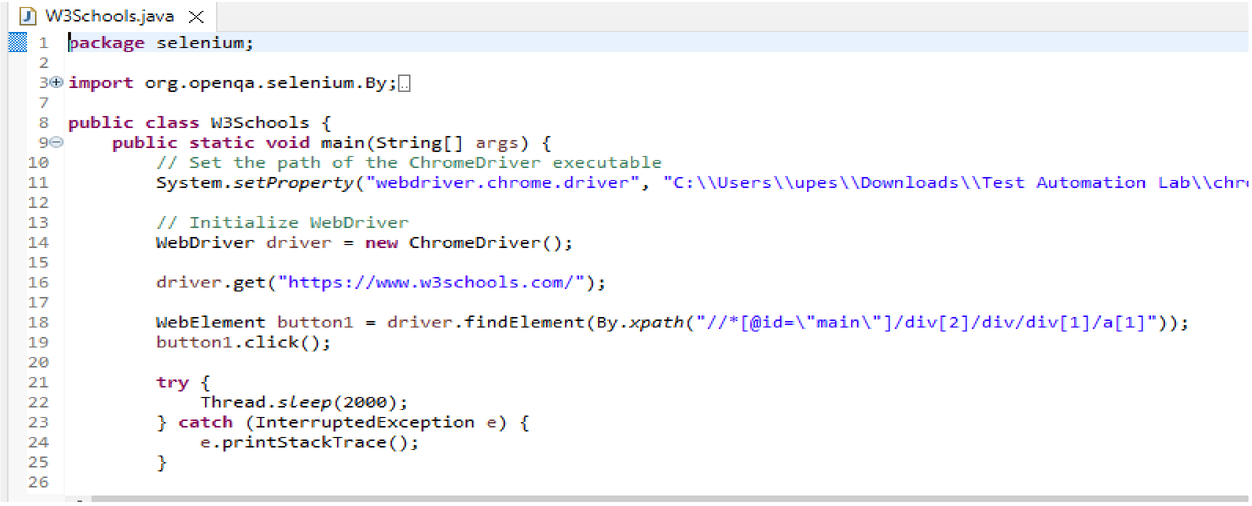
Experiment 7

Traversing through a Tutorial Website

➔ Java Code to traverse trough tutorial Website

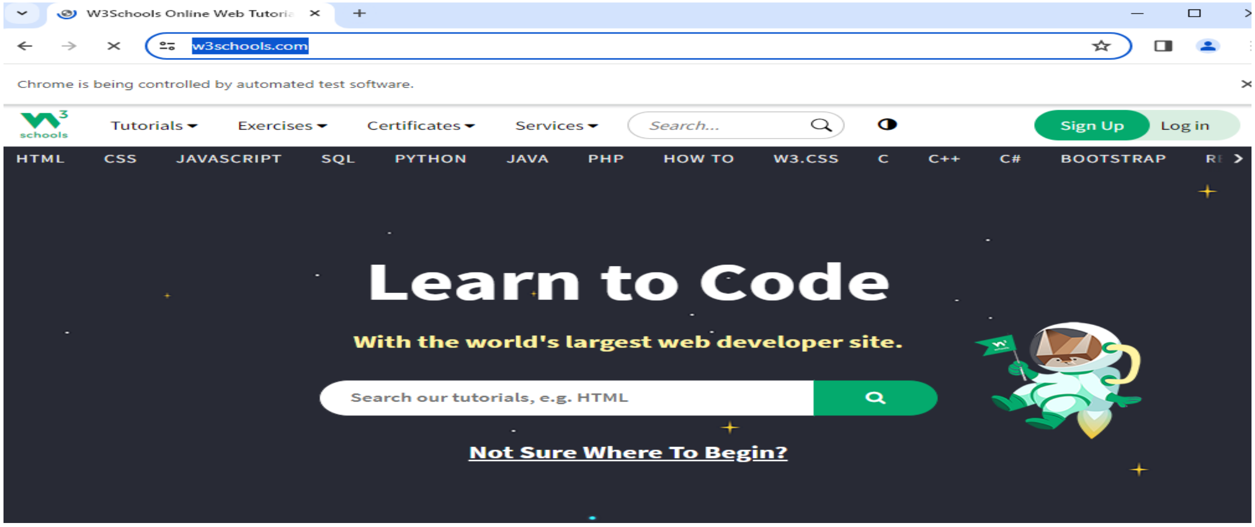


```
Task2Experiment8
  JRE System Library [JavaSE-1.8]
  src
    selenium
      W3Schools.java
        W3Schools
          main(String[]) : void
```



```java
package selenium;

import org.openqa.selenium.By;

public class W3Schools {
    public static void main(String[] args) {
        // Set the path of the ChromeDriver executable
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\upes\\Downloads\\Test Automation Lab\\chr

        // Initialize WebDriver
        WebDriver driver = new ChromeDriver();

        driver.get("https://www.w3schools.com/");

        WebElement button1 = driver.findElement(By.xpath("//*[@id=\"main\"]/div[2]/div/div[1]/a[1]"));
        button1.click();

        try {
            Thread.sleep(2000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

```

➔ **Output**

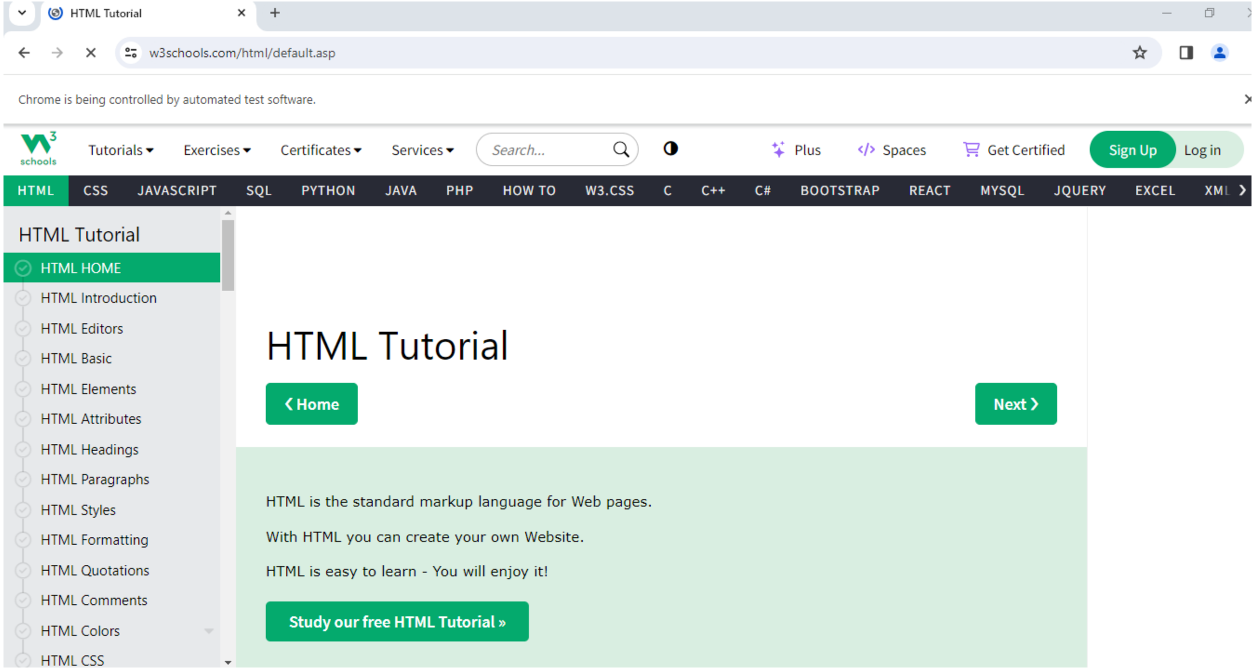


Learn to Code

With the world's largest web developer site.

Not Sure Where To Begin?

Aanchal Tailwal
500097386



Chrome is being controlled by automated test software.

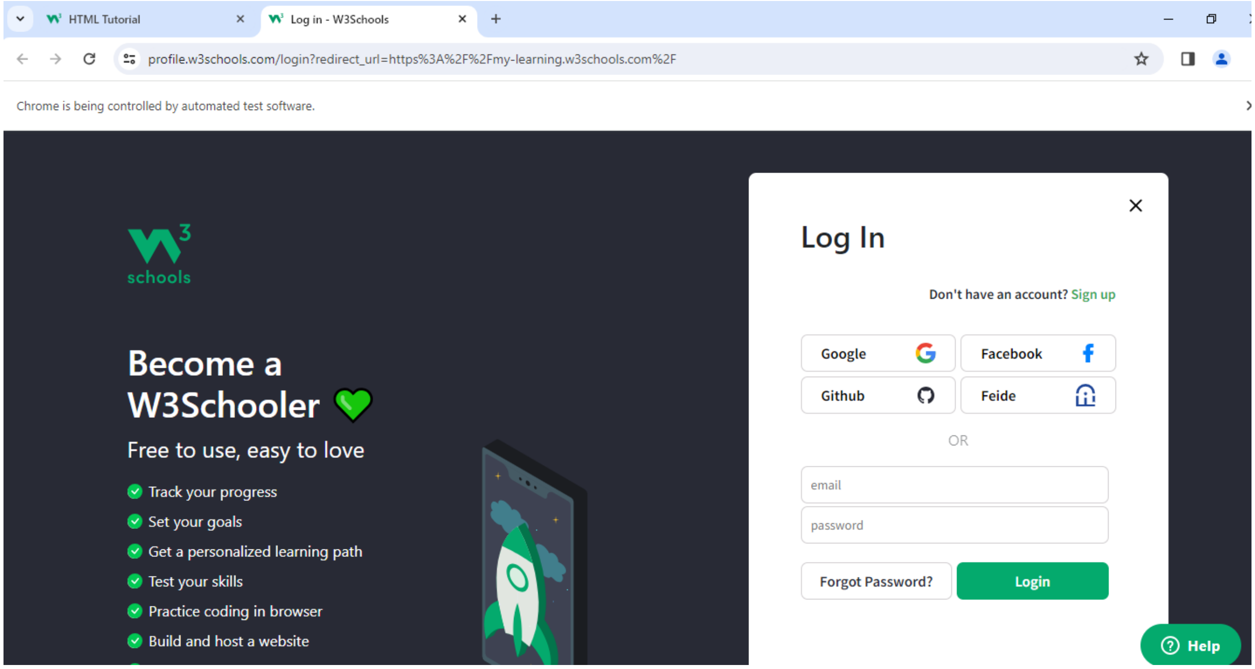
HTML Tutorial

HTML Tutorial

HTML is the standard markup language for Web pages.

With HTML you can create your own Website.

HTML is easy to learn - You will enjoy it!



Chrome is being controlled by automated test software.

Become a W3Schooler 💚

Free to use, easy to love

- Track your progress
- Set your goals
- Get a personalized learning path
- Test your skills
- Practice coding in browser
- Build and host a website

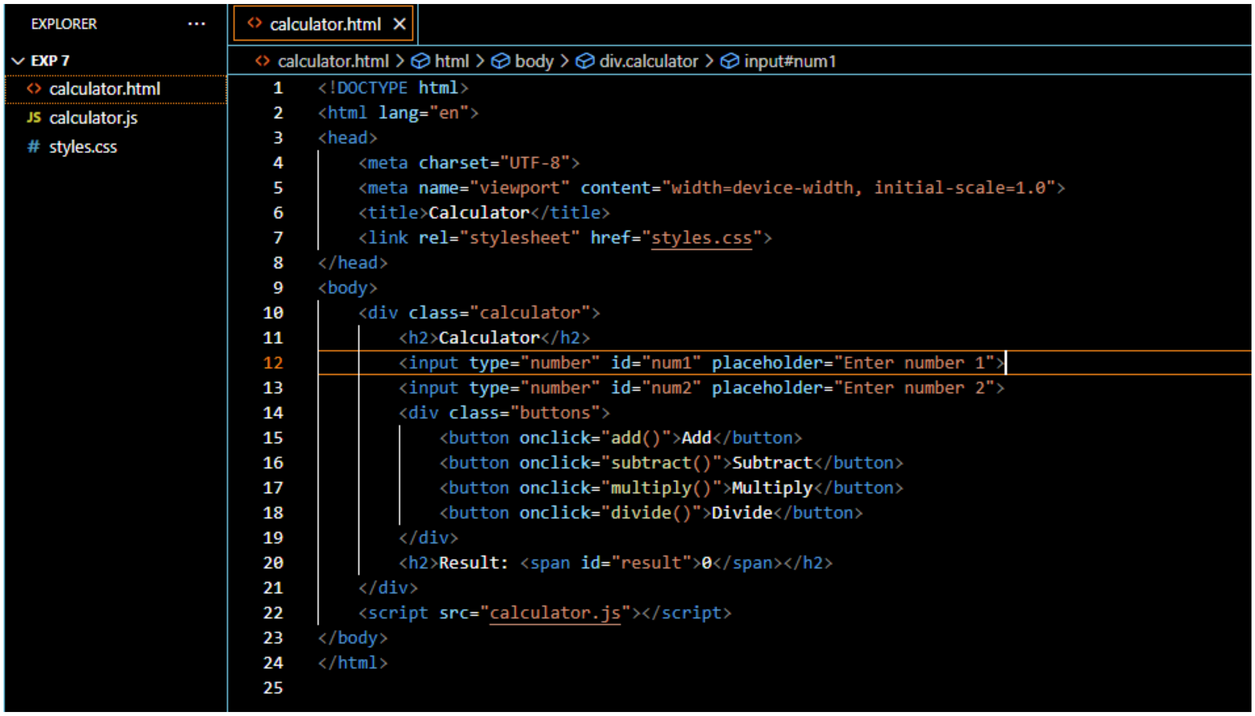
Log In

Don't have an account? Sign up

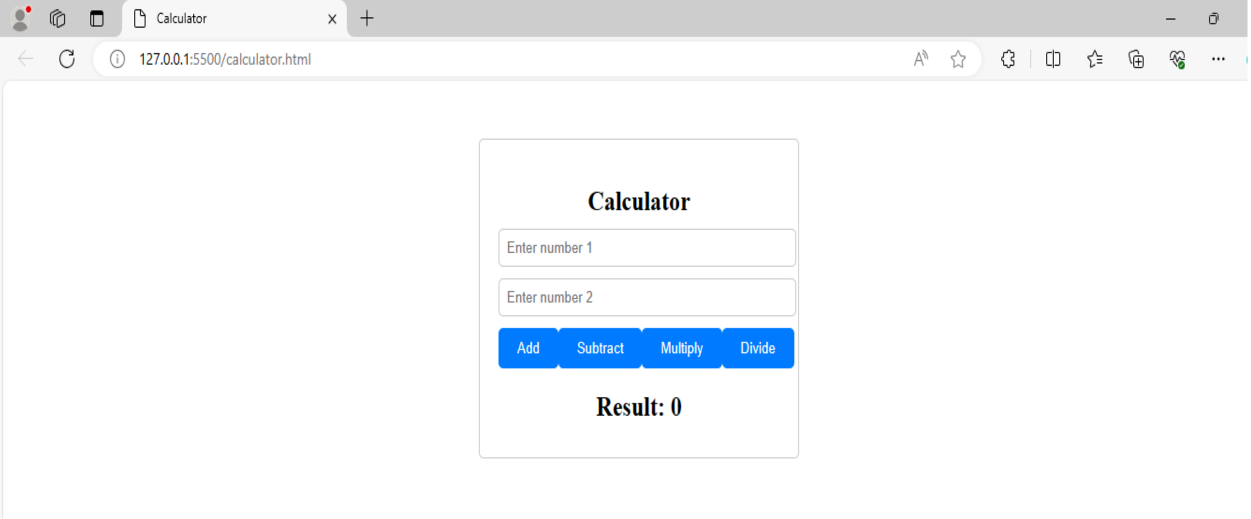
Experiment 8

Test Automating the Calculator Application

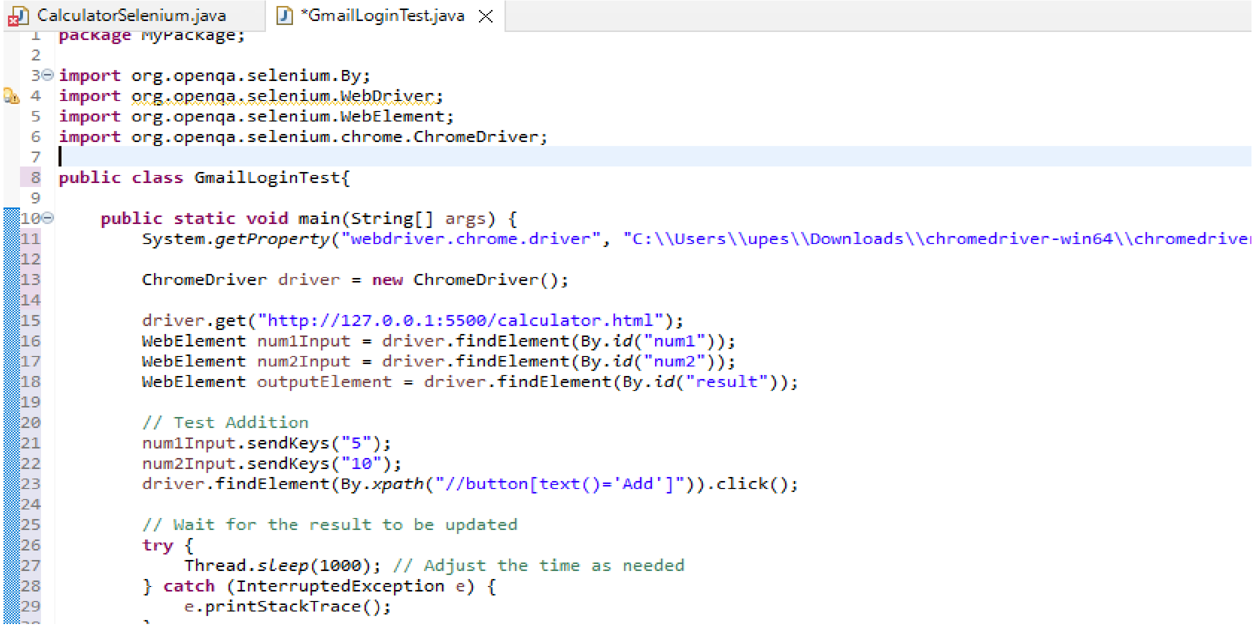
Task 1: Create a web application(home.html) using JS to add/sub/mul/del nos.(2 number boxes, where 2 for input and one for output when a button is clicked)



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Calculator</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="calculator">
        <h2>Calculator</h2>
        <input type="number" id="num1" placeholder="Enter number 1">
        <input type="number" id="num2" placeholder="Enter number 2">
        <div class="buttons">
            <button onclick="add()">Add</button>
            <button onclick="subtract()">Subtract</button>
            <button onclick="multiply()">Multiply</button>
            <button onclick="divide()">Divide</button>
        </div>
        <h2>Result: <span id="result">0</span></h2>
    </div>
    <script src="calculator.js"></script>
</body>
</html>
```

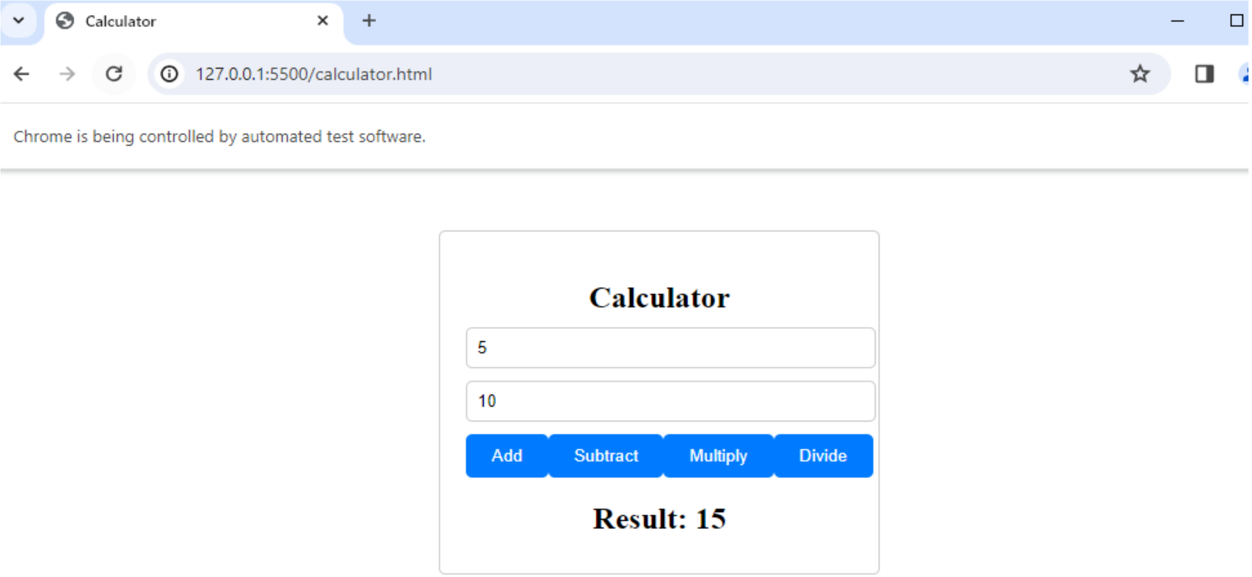


Task 2: Create a maven project where the PSVM will open the web application using selenium



```
package MyPackage;

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

public class GmailLoginTest{

    public static void main(String[] args) {
        System.getProperty("webdriver.chrome.driver", "C:\\Users\\upes\\Downloads\\chromedriver-win64\\chromedrive

        ChromeDriver driver = new ChromeDriver();

        driver.get("http://127.0.0.1:5500/calculator.html");
        WebElement num1Input = driver.findElement(By.id("num1"));
        WebElement num2Input = driver.findElement(By.id("num2"));
        WebElement outputElement = driver.findElement(By.id("result"));

        // Test Addition
        num1Input.sendKeys("5");
        num2Input.sendKeys("10");
        driver.findElement(By.xpath("//button[text()='Add']")).click();

        // Wait for the result to be updated
        try {
            Thread.sleep(1000); // Adjust the time as needed
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
```

Task 3:Now automate the testing process. Send keys to t1 and t2 and obtain the value and check if the values are correct as your test case. without using TestNG



Chrome is being controlled by automated test software.

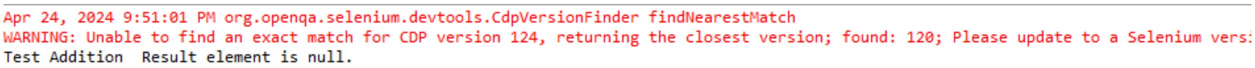
Calculator

5

10

Add | Subtract | Multiply | Divide

Result: 15

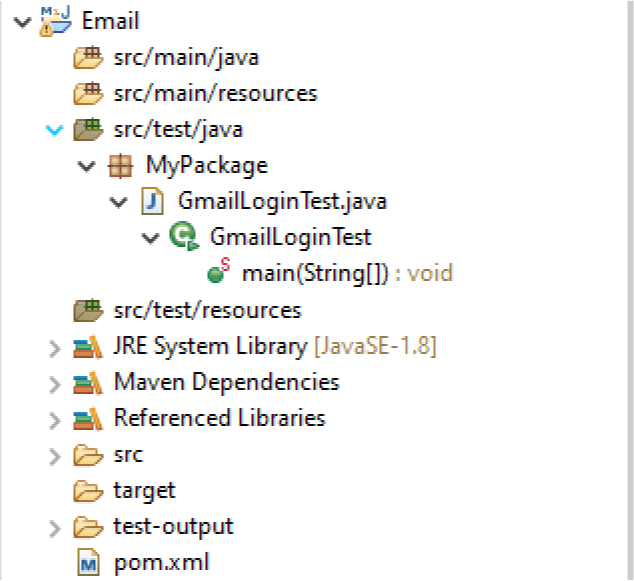


```
Apr 24, 2024 9:51:01 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 124, returning the closest version; found: 120; Please update to a Selenium vers:
Test Addition  Result element is null.
```

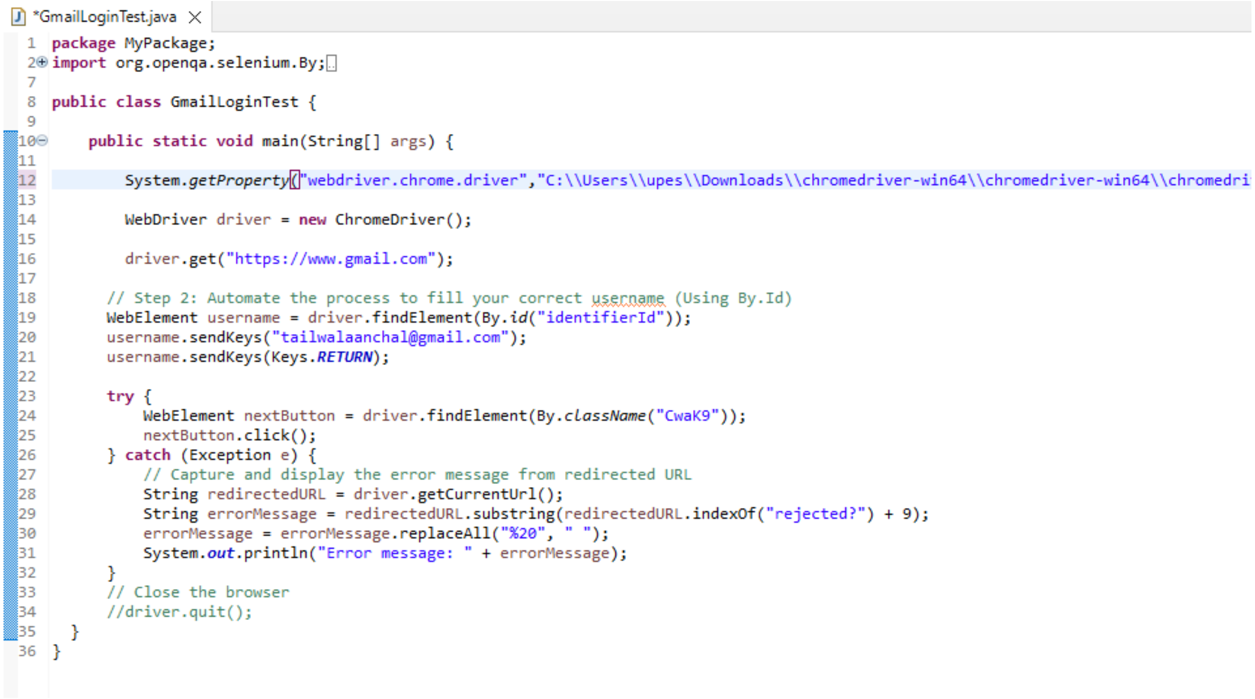
Experiment 9

Working on XPath Selector

Step 1: Open gmail.com

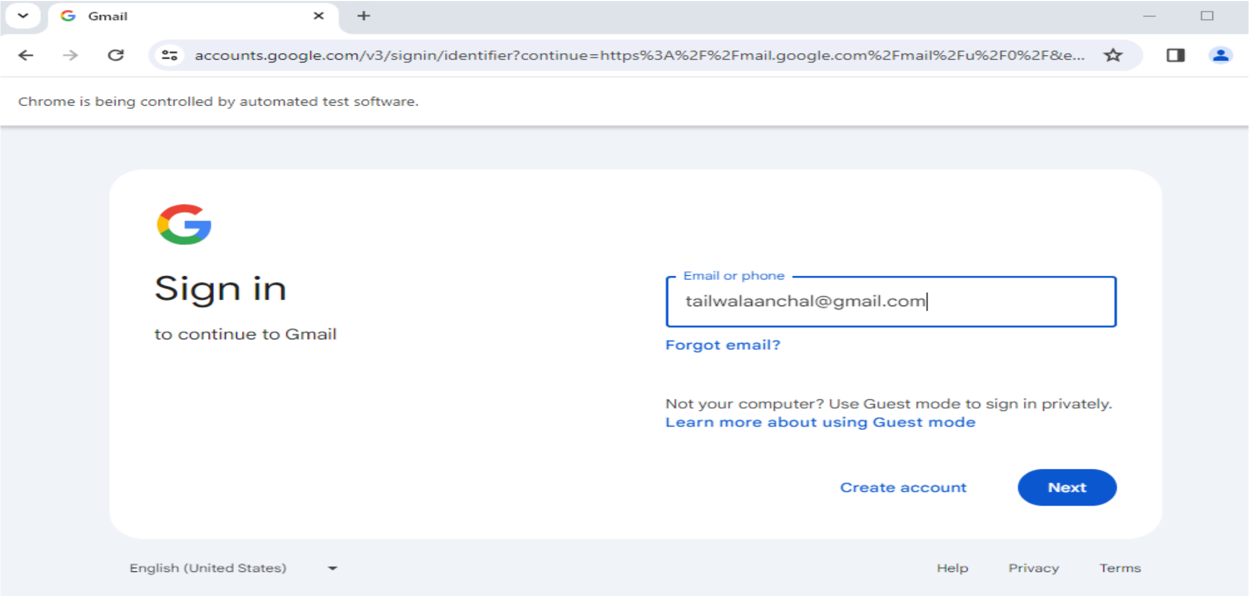


```
Email
    src/main/java
    src/main/resources
    src/test/java
        MyPackage
            GmailLoginTest.java
                GmailLoginTest
                    main(String[]) : void
    src/test/resources
    JRE System Library [JavaSE-1.8]
    Maven Dependencies
    Referenced Libraries
    src
    target
    test-output
    pom.xml
```



```java
package MyPackage;
import org.openqa.selenium.By;

public class GmailLoginTest {

    public static void main(String[] args) {

        System.getProperty("webdriver.chrome.driver","C:\\Users\\upes\\Downloads\\chromedriver-win64\\chromedriver-win64\\chromedri

        WebDriver driver = new ChromeDriver();

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

      // Step 2: Automate the process to fill your correct username (Using By.Id)
      WebElement username = driver.findElement(By.id("identifierId"));
      username.sendKeys("tailwalaanchal@gmail.com");
      username.sendKeys(Keys.RETURN);

      try {
          WebElement nextButton = driver.findElement(By.className("CwaK9"));
          nextButton.click();
      } catch (Exception e) {
          // Capture and display the error message from redirected URL
          String redirectedURL = driver.getCurrentUrl();
          String errorMessage = redirectedURL.substring(redirectedURL.indexOf("rejected?") + 9);
          errorMessage = errorMessage.replaceAll("%20", " ");
          System.out.println("Error message: " + errorMessage);
      }
      // Close the browser
      //driver.quit();
  }
}
```

Step 2: Automate the process to fill an incorrect username(Using By.Id)


Gmail
accounts.google.com/v3/signin/identifier?continue=https%3A%2F%2Fmail.google.com%2Fmail%2Fu%2F0%2F&e...

Chrome is being controlled by automated test software.

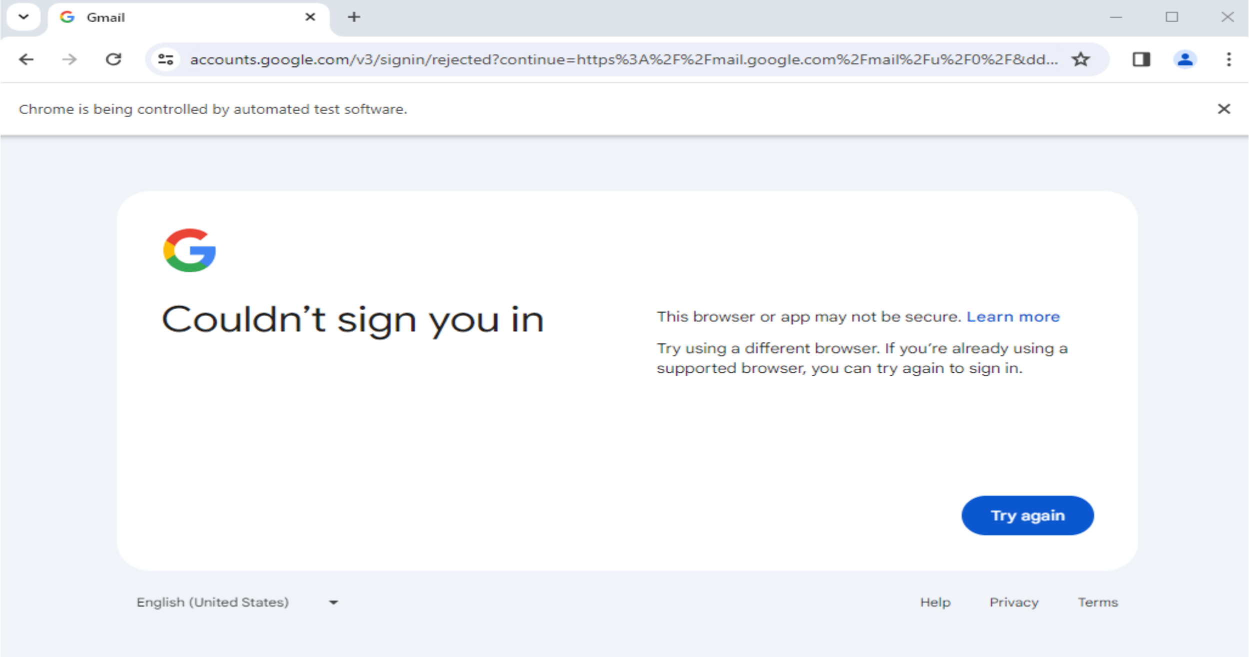
Sign in

to continue to Gmail

Email or phone: tailwalaanchal@gmail.com

Not your computer? Use Guest mode to sign in privately.

Learn more about using Guest mode
Create account
Next
English (United States)
Help
Privacy
Terms


Gmail
accounts.google.com/v3/signin/rejected?continue=https%3A%2F%2Fmail.google.com%2Fmail%2Fu%2F0%2F&dd...

Chrome is being controlled by automated test software.

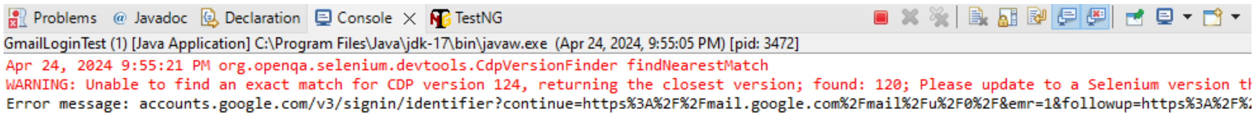
Couldn’t sign you in

This browser or app may not be secure. Learn more

Try using a different browser. If you’re already using a supported browser, you can try again to sign in.

Try again
English (United States)
Help
Privacy
Terms

Step 3: Automated Click(Using By.className)

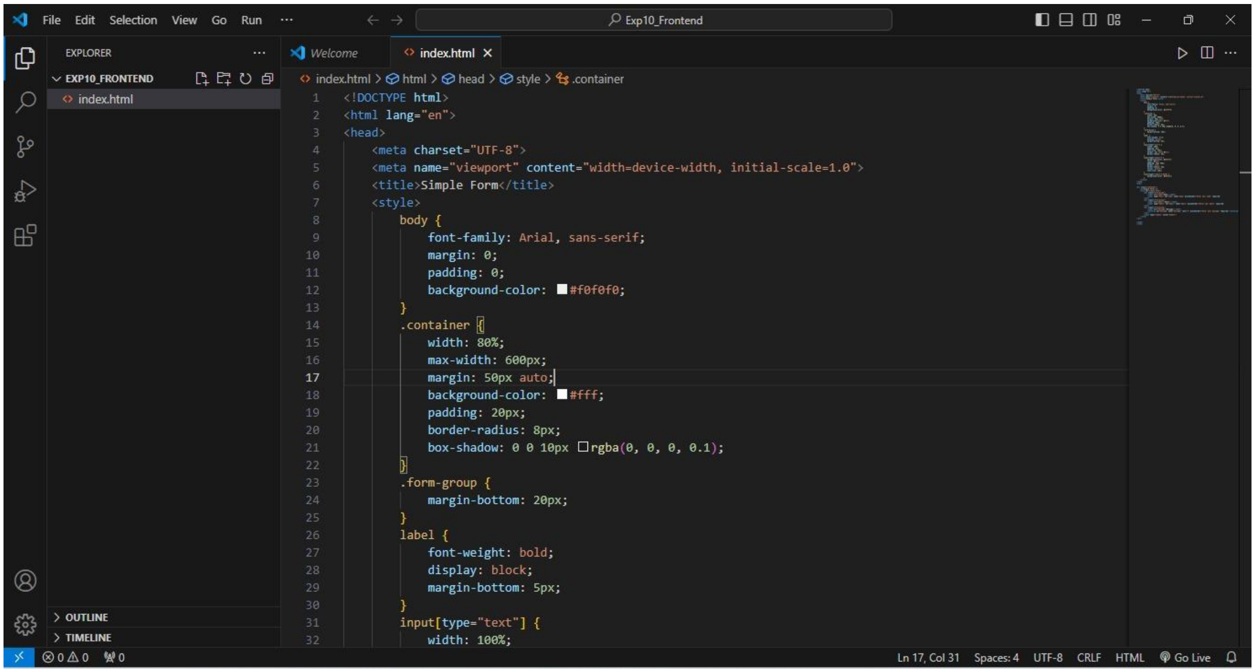


```
GmailLoginTest (1) [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (Apr 24, 2024, 9:55:05 PM) [pid: 3472]
Apr 24, 2024 9:55:21 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 124, returning the closest version; found: 120; Please update to a Selenium version th
Error message: accounts.google.com/v3/signin/identifier?continue=https%3A%2F%2Fmail.google.com%2Fmail%2Fu%2F0%2F&emr=1&followup=https%3A%2F%2
```

Experiment 10

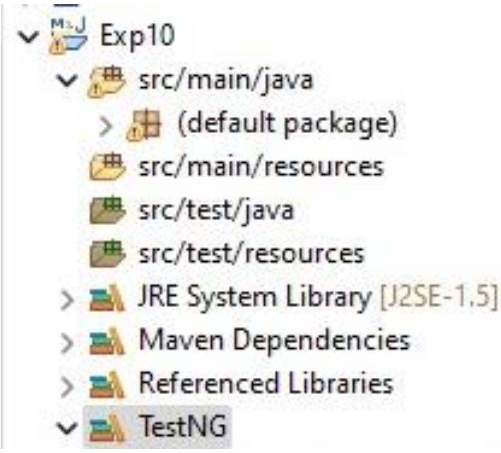
Project with Selenium & TestNG

➔ HTML code for an application

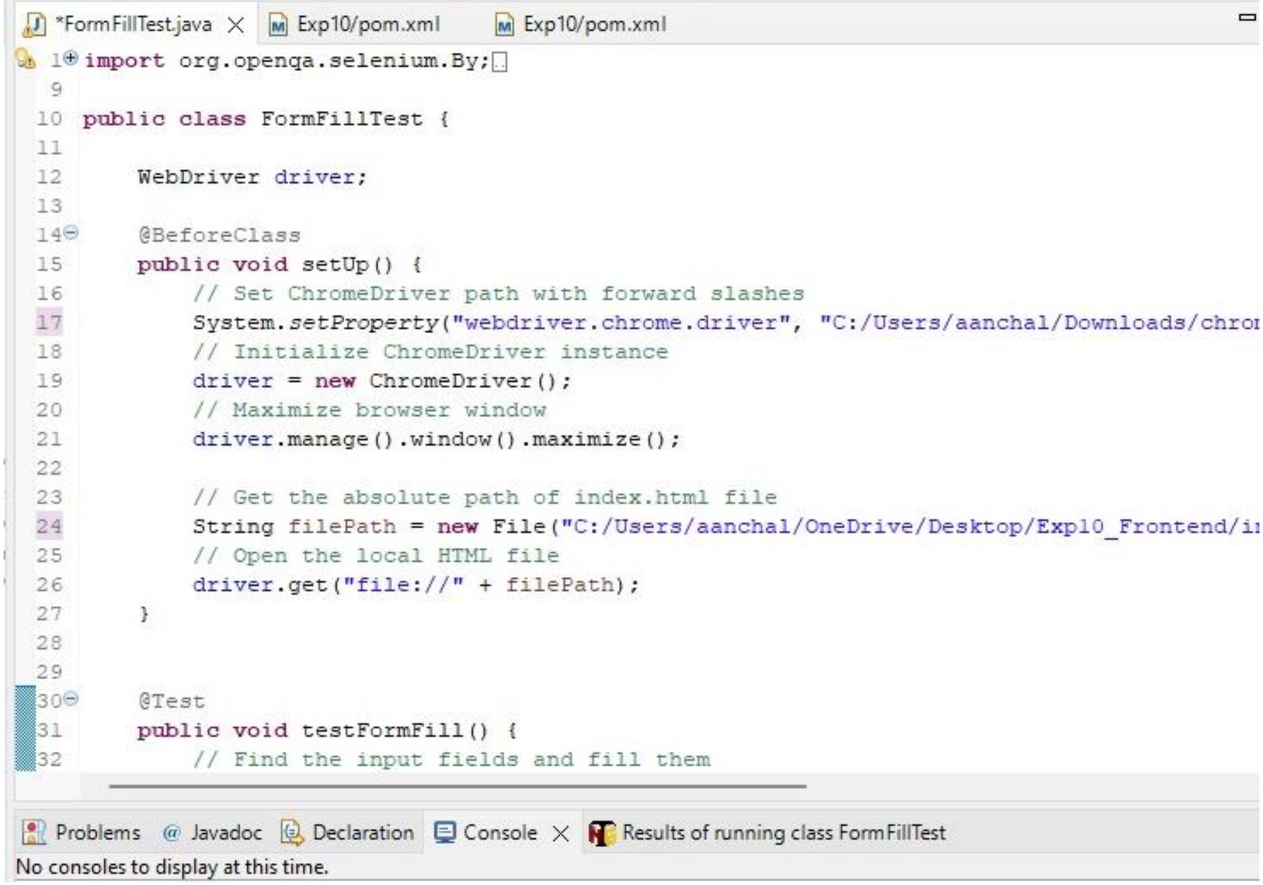


File Edit Selection View Go Run ···
Exp10_Frontend
EXPLORER
EXP10_FRONTEND
index.html
Welcome
index.html
index.html > html > head > style > .container
OUTLINE
TIMELINE
Ln 17, Col 31 Spaces: 4 UTF-8 CRLF HTML Go Live

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simple Form</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f0f0f0;
        }
        .container {
            width: 80%;
            max-width: 600px;
            margin: 50px auto;
            background-color: #fff;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }
        .form-group {
            margin-bottom: 20px;
        }
        label {
            font-weight: bold;
            display: block;
            margin-bottom: 5px;
        }
        input[type="text"] {
            width: 100%;
```

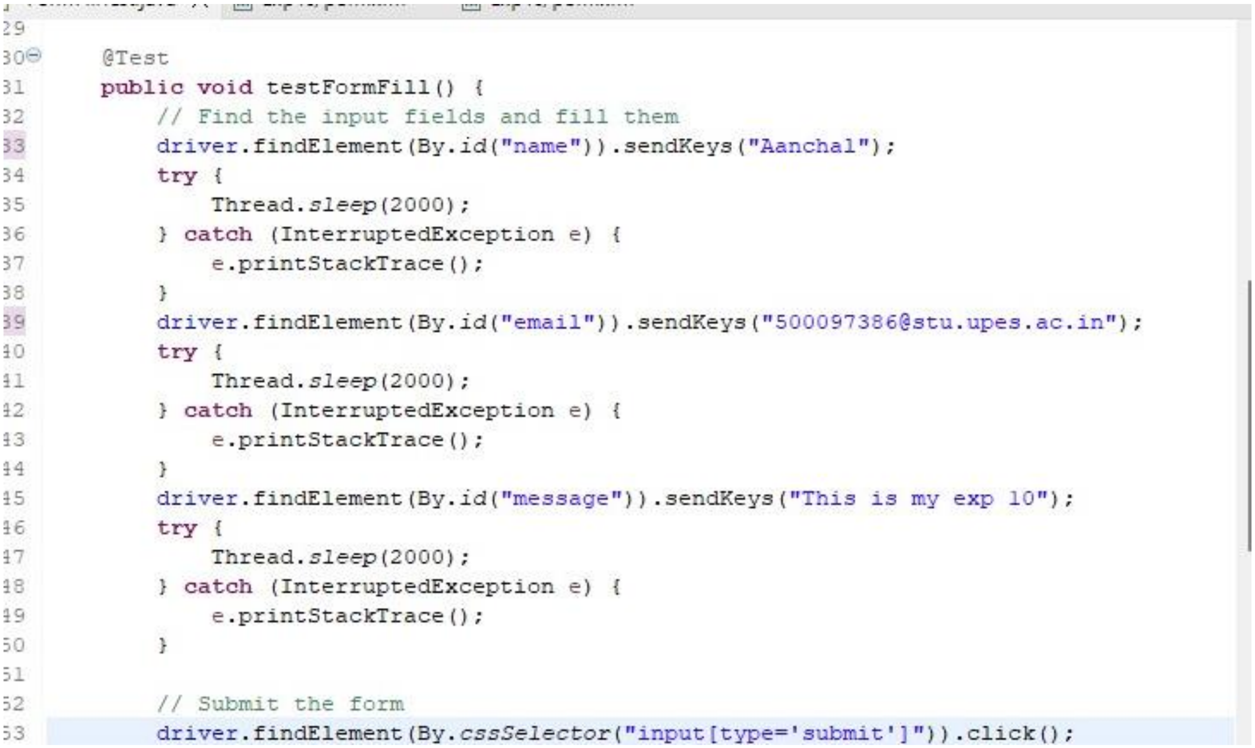


➔ Java Code For Selenium and TestNG



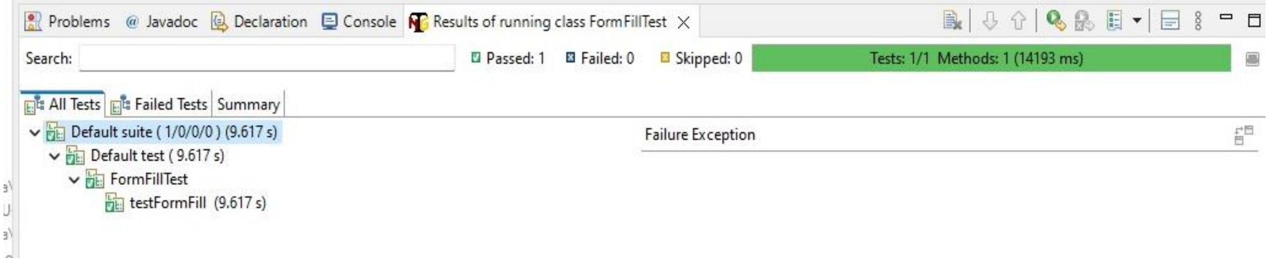
```
import org.openqa.selenium.By;

public class FormFillTest {

    WebDriver driver;

    @BeforeClass
    public void setUp() {
        // Set ChromeDriver path with forward slashes
        System.setProperty("webdriver.chrome.driver", "C:/Users/aanchal/Downloads/chro
        // Initialize ChromeDriver instance
        driver = new ChromeDriver();
        // Maximize browser window
        driver.manage().window().maximize();

        // Get the absolute path of index.html file
        String filePath = new File("C:/Users/aanchal/OneDrive/Desktop/Exp10_Frontend/i
        // Open the local HTML file
        driver.get("file://" + filePath);
    }


    @Test
    public void testFormFill() {
        // Find the input fields and fill them
```

No consoles to display at this time.



```

    @Test
    public void testFormFill() {
        // Find the input fields and fill them
        driver.findElement(By.id("name")).sendKeys("Aanchal");
        try {
            Thread.sleep(2000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        driver.findElement(By.id("email")).sendKeys("500097386@stu.upes.ac.in");
        try {
            Thread.sleep(2000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        driver.findElement(By.id("message")).sendKeys("This is my exp 10");
        try {
            Thread.sleep(2000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        // Submit the form
        driver.findElement(By.cssSelector("input[type='submit']")).click();
```

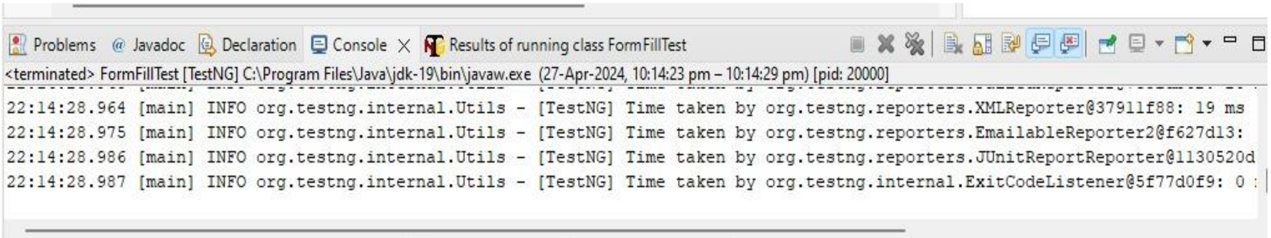
➔ TestNG Output



Passed: 1 Failed: 0 Skipped: 0 Tests: 1/1 Methods: 1 (14193 ms)

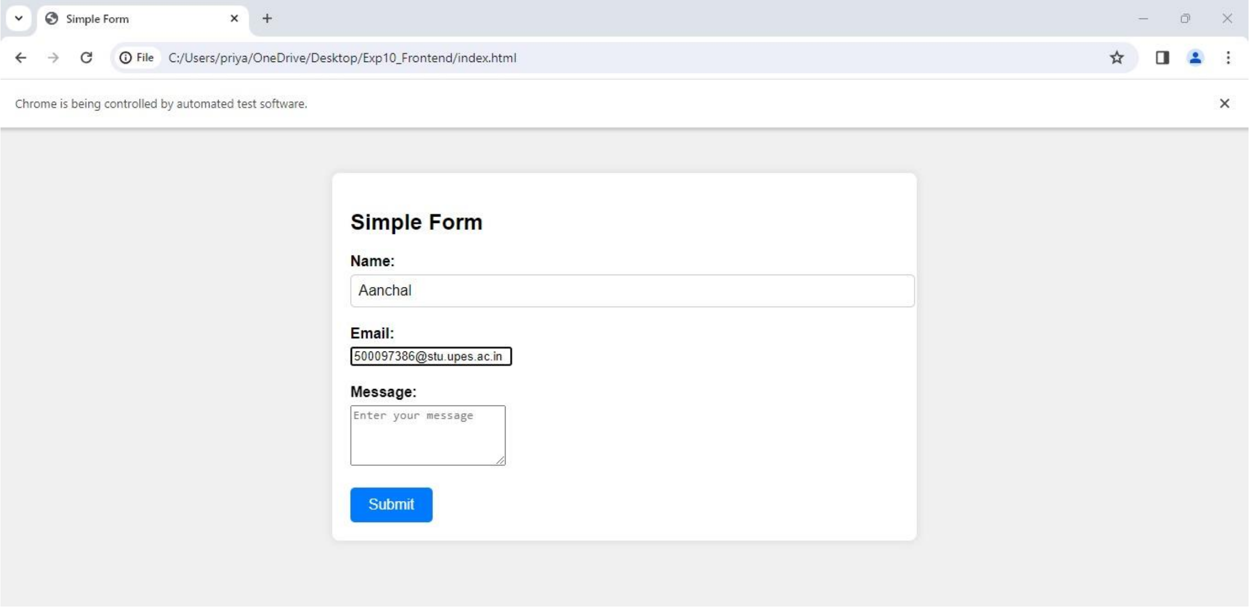
- Default suite (1/0/0/0) (9.617 s)
  - Default test (9.617 s)
    - FormFillTest
      - testFormFill (9.617 s)

➔ Console Output



```
<terminated> FormFillTest [TestNG] C:\Program Files\Java\jdk-19\bin\javaw.exe (27-Apr-2024, 10:14:23 pm – 10:14:29 pm) [pid: 20000]
22:14:28.964 [main] INFO org.testng.internal.Utils - [TestNG] Time taken by org.testng.reporters.XMLReporter@37911f88: 19 ms
22:14:28.975 [main] INFO org.testng.internal.Utils - [TestNG] Time taken by org.testng.reporters.EmailableReporter2@f627d13:
22:14:28.986 [main] INFO org.testng.internal.Utils - [TestNG] Time taken by org.testng.reporters.JUnitReportReporter@1130520d
22:14:28.987 [main] INFO org.testng.internal.Utils - [TestNG] Time taken by org.testng.internal.ExitCodeListener@5f77d0f9: 0
```

➔ Output



Chrome is being controlled by automated test software.

Simple Form

Name: Aanchal

Email: 500097386@stu.upes.ac.in

Message: Enter your message