# 1. Implement below Program using Java Concepts

# a. Create a class name as ReadAnWriteDate

# b. Read the data from Excel Sheet (Use Apache POI) and should have below columns and should contain data

# Name, Courses, Fee

# Kirk, Java, $100

# Dan, Python, $100

# c. Use Collections concept to store the reading data and display it in console.

Path : /comprehensive\_1/src/test/java/org/lUS/comprehensive\_1/Assinment\_1.java

**public** **class** Assinment\_1 {

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

// **TODO** Auto-generated method stub

**try**

{

ArrayList data =**new** ArrayList();

FileInputStream file = **new** FileInputStream("src\\main\\java\\Assignment1.xlsx");

XSSFWorkbook wb = **new** XSSFWorkbook(file);

XSSFSheet ws = wb.getSheetAt(0);

Iterator<Row> rowIterator = ws.iterator();

**while** (rowIterator.hasNext())

{

Row row = rowIterator.next();

Iterator<Cell> cellIterator = row.cellIterator();

**while** (cellIterator.hasNext())

{

Cell cell = cellIterator.next();

**switch** (cell.getCellType())

{

**case** ***NUMERIC***:

System.***out***.print(cell.getNumericCellValue()+"\t");

data.add(cell.getNumericCellValue());

**break**;

**case** ***STRING***:

System.***out***.print(cell.getStringCellValue()+ "\t");

data.add(cell.getStringCellValue());

**break**;

}

}

System.***out***.println();

}

System.***out***.println("Reading File Completed.");

System.***out***.print("data in List:\n" + data.get(0)+"\t"+ data.get(1)+"\t"+data.get(2)+"\n"+data.get(3)+"\t"+data.get(4)+"\t"+data.get(5)+"\n"+data.get(6)+"\t"+data.get(7)+"\t"+data.get(8));

file.close();

}

**catch** (Exception ex)

{

ex.printStackTrace();

}

}

A screenshot of a computer

Description automatically generated

# 2. Implement below program using Python.

# a. Create a class Name as ReadAndWriteData

# b. Read the Data from CSV File and should have below three Columns and should contain data

# Name, SkillSet, Experience

# Kirk, Java, 10 Years

# Dan, Python, 5 Years

# c. Build a Data Structure to display the data, which contains in CSV Fil

Path: /comprehensive\_1/src/test/java/org/lUS/comprehensive\_1/ReadAndWriteData.py

import csv

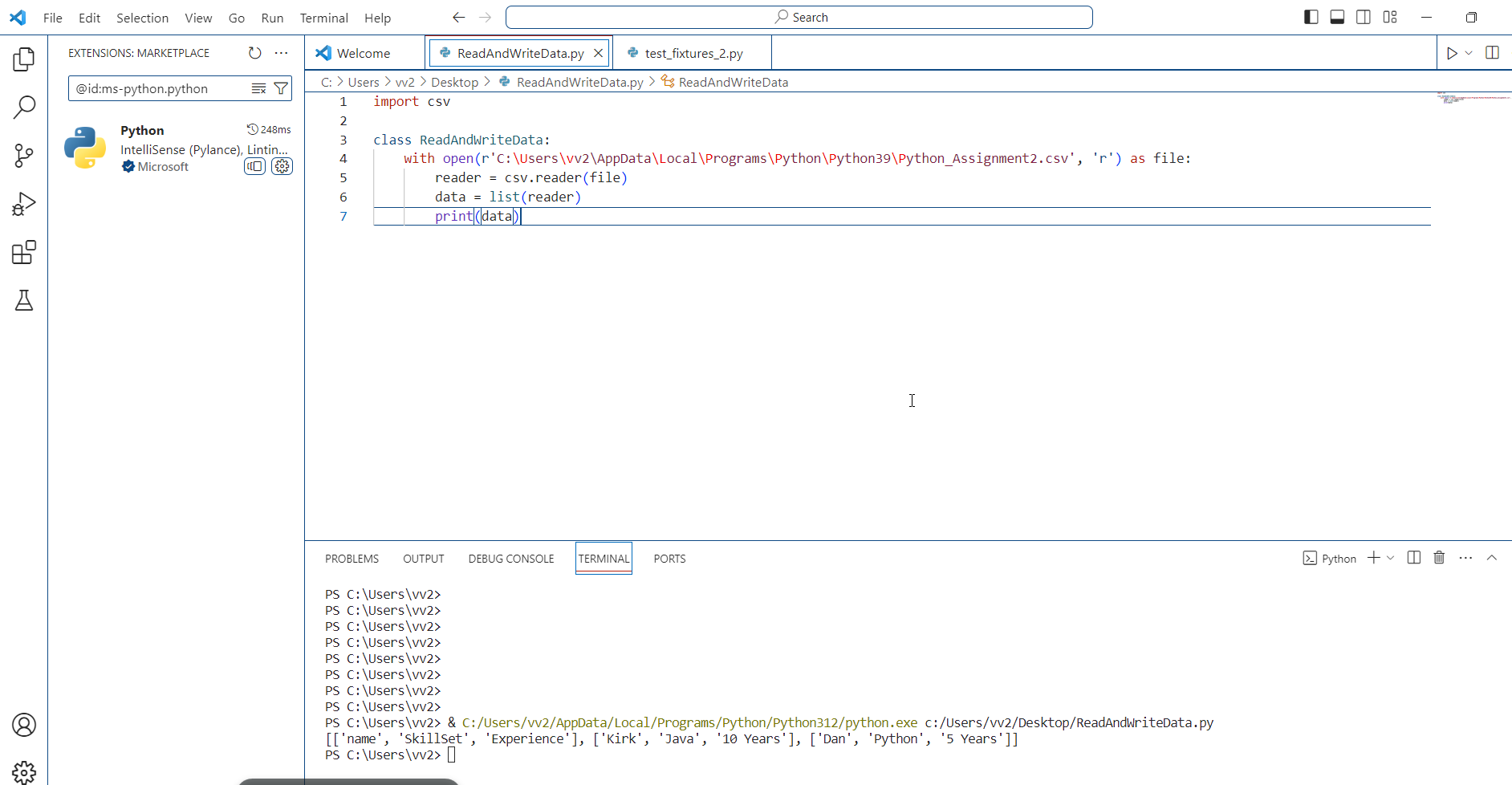
class ReadAndWriteData:

with open('Python\_Assignment2.csv', 'r') as file:

reader = csv.reader(file)

data = list(reader)

print(data)



# 3. Launch a below browser in Chrome and verify below conditions using Selenium (Use XPaths to Identify Web Elements) with Java https://www.automationanywhere.com/

# a. Automation Anywhere logo should be present

# b. Verify Request demo button is present and verify if it is clickable or not

path: /comprehensive\_1/src/test/java/org/lUS/comprehensive\_1/Assignment\_3.java

**public** **class** Assignment\_3 {

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

WebDriver driver = **new** ChromeDriver();

driver.manage().window().maximize();

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

driver.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(5));

**boolean** cookies = driver.findElement(By.*id*("onetrust-accept-btn-handler")).isDisplayed();

**if** (cookies == **true**) {

driver.findElement(By.*id*("onetrust-accept-btn-handler")).click();

}

**boolean** logName = driver

.findElement(By.*xpath*("//img[@src='/sites/default/files/images/default-images/logo-aa-new.svg']"))

.isDisplayed();

System.***out***.println("logo is present " + logName);

**boolean** requestDemo = driver

.findElement(By.*xpath*("//a[@class='coh-link coh-style-solid-orange-button btn-link']")).isDisplayed();

System.***out***.println("Request Demo button is avalilable " + requestDemo);

WebDriverWait wt = **new** WebDriverWait(driver, Duration.*ofSeconds*(6));

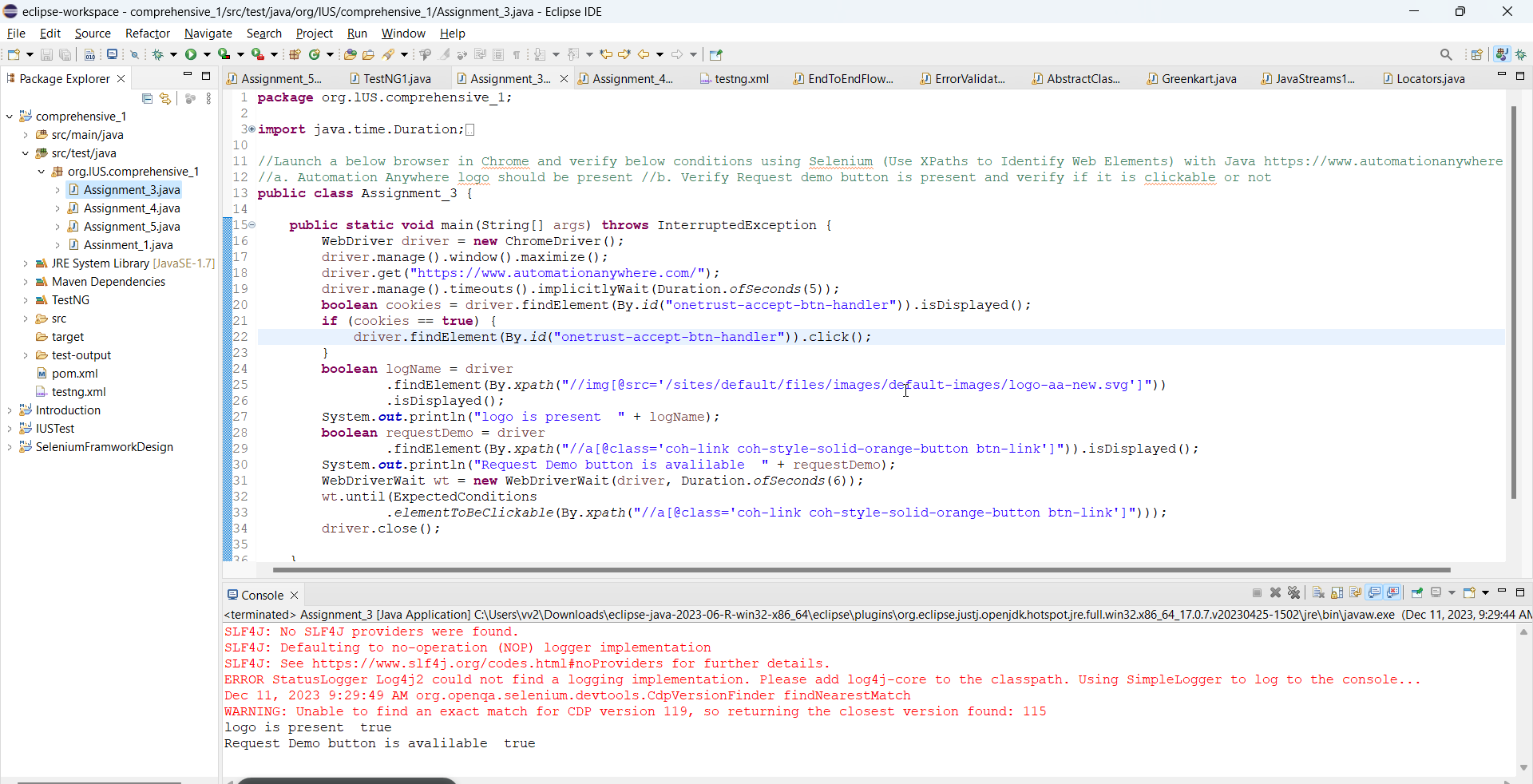
wt.until(ExpectedConditions

.*elementToBeClickable*(By.*xpath*("//a[@class='coh-link coh-style-solid-orange-button btn-link']")));

driver.close();

}

}



# 4. Launch a below browser in Chrome and verify below conditions using Selenium (Use XPaths to Identify Web Elements) with Java https://www.automationanywhere.com/

# Verify below list is present on home page and click on each and every link and verify that it is navigating to the proper page

# Products b. Solutions c. Resources d. Beyond RPA e. Company

Path: /comprehensive\_1/src/test/java/org/lUS/comprehensive\_1/Assignment\_4.java

**public** **class** Assignment\_4 {

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

WebDriver driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(5));

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

**boolean** cookies = driver.findElement(By.*id*("onetrust-accept-btn-handler")).isDisplayed();

**if** (cookies == **true**) {

driver.findElement(By.*id*("onetrust-accept-btn-handler")).click();

}

**for** (**int** i = 0; i <= 4; i++) {

System.***out***.println(driver.findElements(By

.*xpath*("//ul[@class='coh-menu-list-container coh-unordered-list menu-level-1 coh-ce-646fa54d']/li"))

.get(i).getText());

// String clickOnLink = Keys.chord(Keys.CONTROL,Keys.ENTER);

driver.findElements(By

.*xpath*("//ul[@class='coh-menu-list-container coh-unordered-list menu-level-1 coh-ce-646fa54d']/li"))

.get(i).click();

System.***out***.println(driver.getTitle());

driver.navigate().back();

}

driver.close();

}

}

A screenshot of a computer

Description automatically generated

# 5. Implement 3 and 4 Programs using TestNG Concepts and write it using @Test annotation.

# And add TestNG Attributes – priority, enabled and timeOut to run the programs.

# Run the program with enabled attribute value as either true or false

# Run the program with timeOut Attribute values

# And run 3 and 4 Programs from TestNG.xml file and verify the generated test reports.

Path :/comprehensive\_1/src/test/java/org/lUS/comprehensive\_1/Assignment\_5.java

XML file : /comprehensive\_1/testng.xml

**public** **class** Assignment\_5 {

@Test(enabled = **true**)

**public** **void** automationanywhere() **throws** InterruptedException {

WebDriver driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(5));

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

**boolean** cookies = driver.findElement(By.*id*("onetrust-accept-btn-handler")).isDisplayed();

**if** (cookies == **true**) {

driver.findElement(By.*id*("onetrust-accept-btn-handler")).click();

}

**boolean** logName = driver

.findElement(By.*xpath*("//img[@src='/sites/default/files/images/default-images/logo-aa-new.svg']"))

.isDisplayed();

System.***out***.println("logo is present " + logName);

**boolean** requestDemo = driver

.findElement(By.*xpath*("//a[@class='coh-link coh-style-solid-orange-button btn-link']")).isDisplayed();

System.***out***.println("Request Demo button is avalilable " + requestDemo);

WebDriverWait wt = **new** WebDriverWait(driver, Duration.*ofSeconds*(6));

wt.until(ExpectedConditions

.*elementToBeClickable*(By.*xpath*("//a[@class='coh-link coh-style-solid-orange-button btn-link']")));

driver.close();

}

@Test(priority = 1)

**public** **void** AllLinks() **throws** InterruptedException {

WebDriver driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(5));

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

**boolean** cookies = driver.findElement(By.*id*("onetrust-accept-btn-handler")).isDisplayed();

**if** (cookies == **true**) {

driver.findElement(By.*id*("onetrust-accept-btn-handler")).click();

}

**for** (**int** i = 0; i <= 4; i++) {

System.***out***.println(driver.findElements(By

.*xpath*("//ul[@class='coh-menu-list-container coh-unordered-list menu-level-1 coh-ce-646fa54d']/li"))

.get(i).getText());

// String clickOnLink = Keys.chord(Keys.CONTROL,Keys.ENTER);

driver.findElements(By

.*xpath*("//ul[@class='coh-menu-list-container coh-unordered-list menu-level-1 coh-ce-646fa54d']/li"))

.get(i).click();

System.***out***.println(driver.getTitle());

driver.navigate().back();

}

driver.close();

}

@Test(timeOut = 5000)

**public** **void** AllLinks1() **throws** InterruptedException {

WebDriver driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(5));

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

**boolean** cookies = driver.findElement(By.*id*("onetrust-accept-btn-handler")).isDisplayed();

**if** (cookies == **true**) {

driver.findElement(By.*id*("onetrust-accept-btn-handler")).click();

}

**for** (**int** i = 0; i <= 4; i++) {

System.***out***.println(driver.findElements(By

.*xpath*("//ul[@class='coh-menu-list-container coh-unordered-list menu-level-1 coh-ce-646fa54d']/li"))

.get(i).getText());

// String clickOnLink = Keys.chord(Keys.CONTROL,Keys.ENTER);

driver.findElements(By

.*xpath*("//ul[@class='coh-menu-list-container coh-unordered-list menu-level-1 coh-ce-646fa54d']/li"))

.get(i).click();

System.***out***.println(driver.getTitle());

driver.navigate().back();

}

driver.close();

}

@Test(enabled = **false**)

**public** **void** automationanywhere1() **throws** InterruptedException {

WebDriver driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(5));

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

**boolean** cookies = driver.findElement(By.*id*("onetrust-accept-btn-handler")).isDisplayed();

**if** (cookies == **true**) {

driver.findElement(By.*id*("onetrust-accept-btn-handler")).click();

}

**boolean** logName = driver

.findElement(By.*xpath*("//img[@src='/sites/default/files/images/default-images/logo-aa-new.svg']"))

.isDisplayed();

System.***out***.println("logo is present " + logName);

**boolean** requestDemo = driver

.findElement(By.*xpath*("//a[@class='coh-link coh-style-solid-orange-button btn-link']")).isDisplayed();

System.***out***.println("Request Demo button is avalilable " + requestDemo);

WebDriverWait wt = **new** WebDriverWait(driver, Duration.*ofSeconds*(6));

wt.until(ExpectedConditions

.*elementToBeClickable*(By.*xpath*("//a[@class='coh-link coh-style-solid-orange-button btn-link']")));

driver.close();

}

}

A screenshot of a computer

Description automatically generated

XML file

<?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">

<classes>

<class name="org.lUS.comprehensive\_1.Assignment\_5"/>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

## 7. Create a branch name as comprehensive\_basic\_2 in GitHub and implement all the above programs in that branch and push the code into the same branch. Share the GitHub details to evaluate. Write the git commands accordingly.

<https://github.com/veena988/comprehensive_basic_>2

# 8 Create a collection in Postman and create positive and negative cases for below API. And use get method to get valid responses.

# Verify the HTTP Status Codes, while running Positive and Negative Test Cases.

# To validate the HTTP Status Codes and responses use Assertions concept in PostMan <https://restcountries.com/v3.1/name/India>

Path : /comprehensive\_1/src/test/java/org/lUS/comprehensive\_1/Comprehensive\_Assignment.postman\_collection.json

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# 9. Implement below steps in JMeter

# 1. Create a Test Plan

# 2. Create a Thread Group and Add Number of Threads (Count is 2)

# 3. Add HTTP Request (Use https protocol) and URL as https://www.w3schools.com/

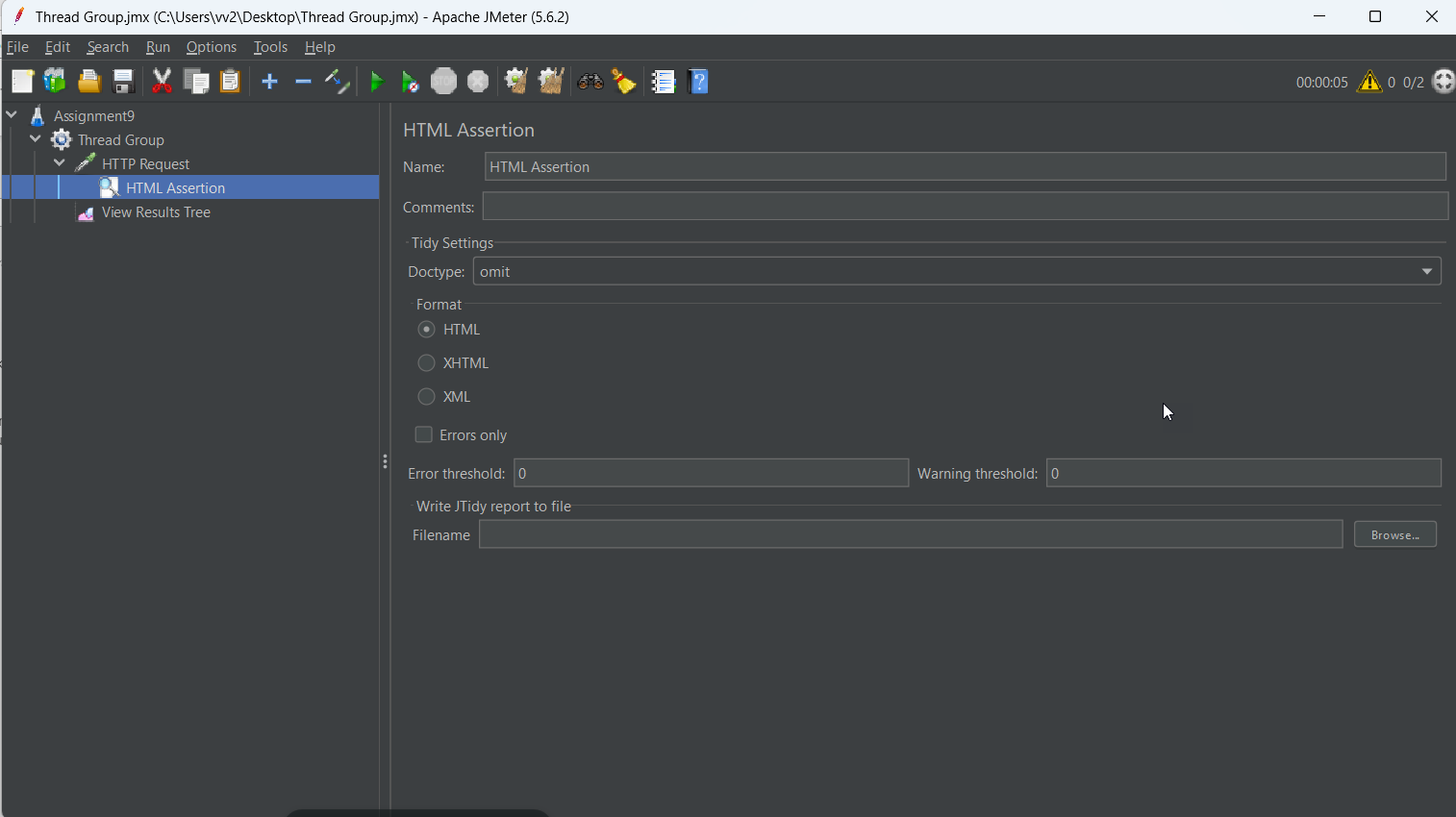
# 4. Add HTML Assertions

# 5. Add Error Threshold as zero

# 6. Add Warning Threshold count

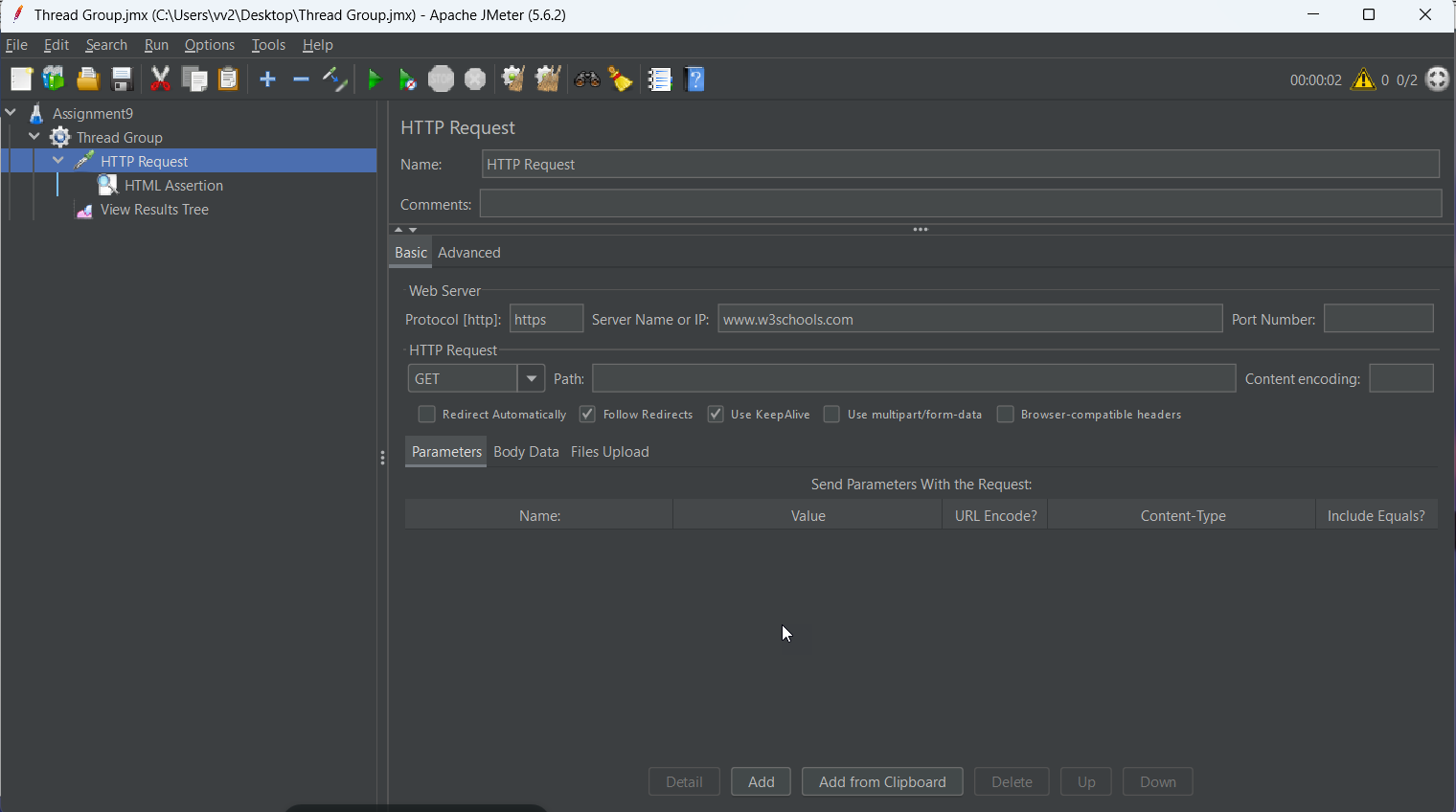
# 7. Run the Script

Path : /comprehensive\_1/src/test/java/org/lUS/comprehensive\_1/Assignment9.jmx



A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated