TestNG Documentation

This document will contain the record of steps performed to complete the TestNG section of the final assessment due 8th October 2022.

Question

TestNG Framework

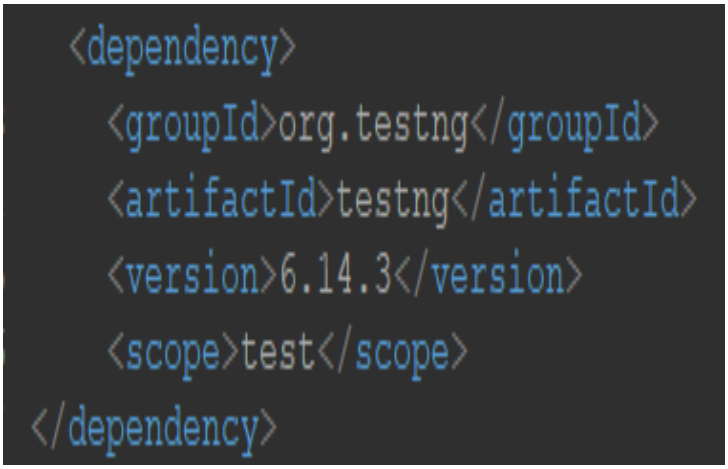
a) Write a program to HRM Login
b) Write program to apply leave
c) User method over loading for searching as discussed in the class assignment

What is TestNG?

TestNG is an open-source testing framework where NG stands for 'Next Generation'. It is architected to simplify a broad range of testing needs starting from unit testing to integrated system testing. Initially, both JUnit and TestNG were designed solely for unit testing. TestNG is a testing framework for the Java programming language. It makes automated tests more structured, readable, maintainable, and user-friendly.

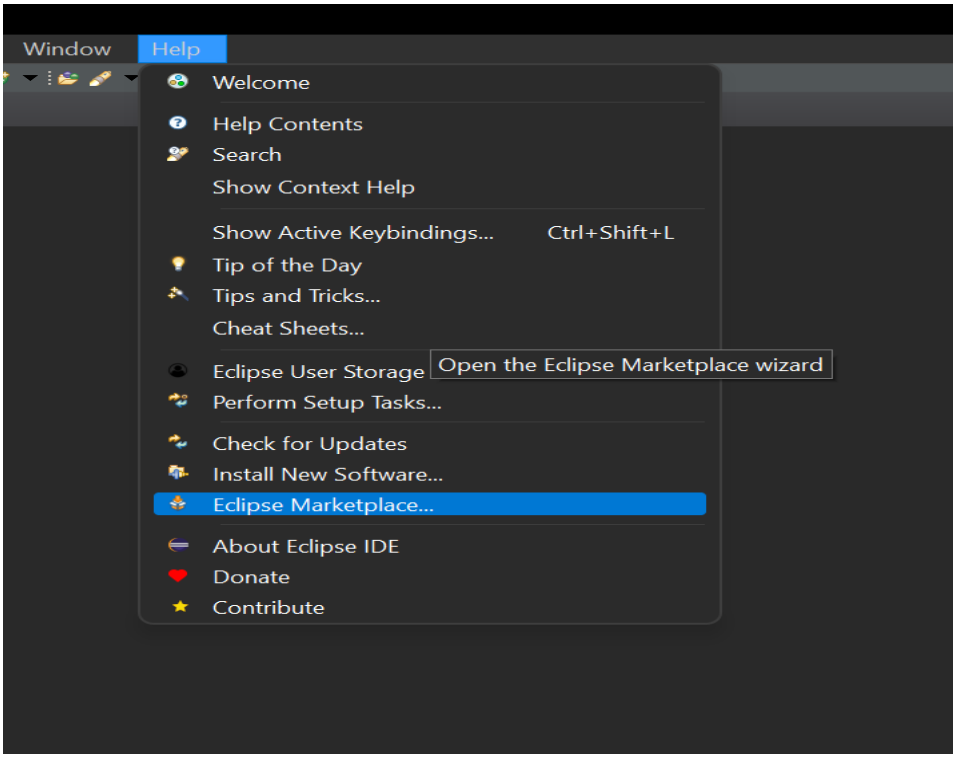
a) Setting up TestNG in our environment

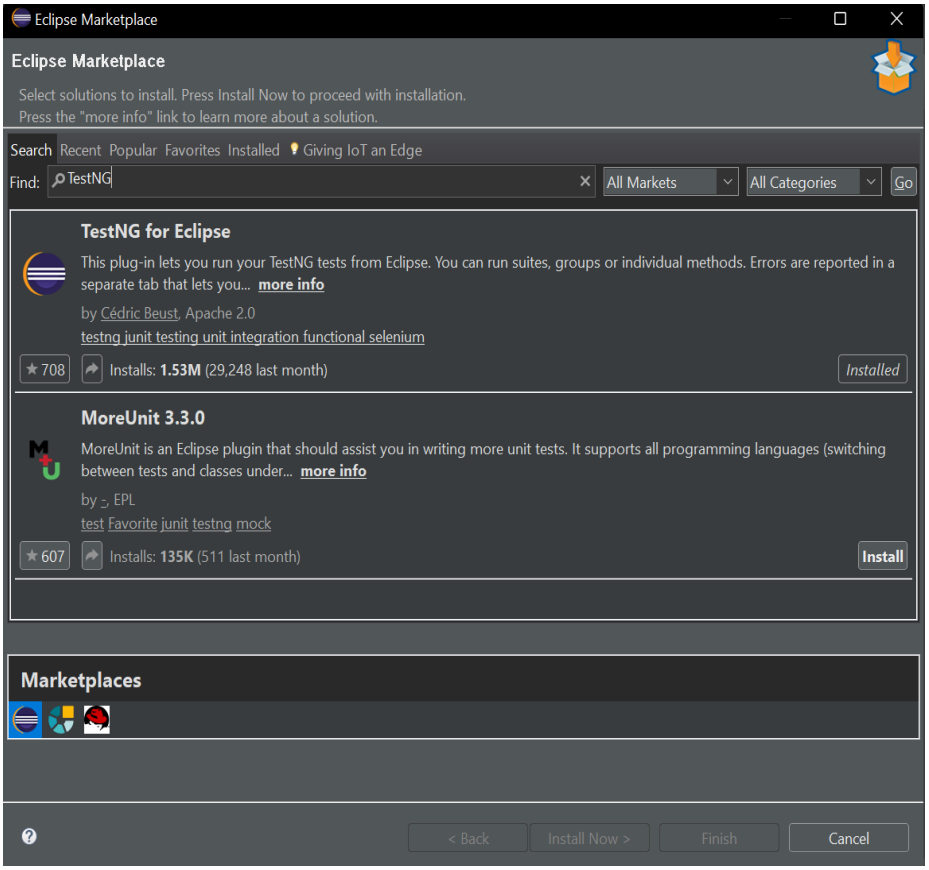
Before creating the test NG class, the below piece of code needs to be appended into the dependencies tag of the pom.xml file of the maven project.



It’s also a good practice to download the TestNG extension for eclipse from the eclipse workspace. Since TestNG is a very common framework, this package will be needed

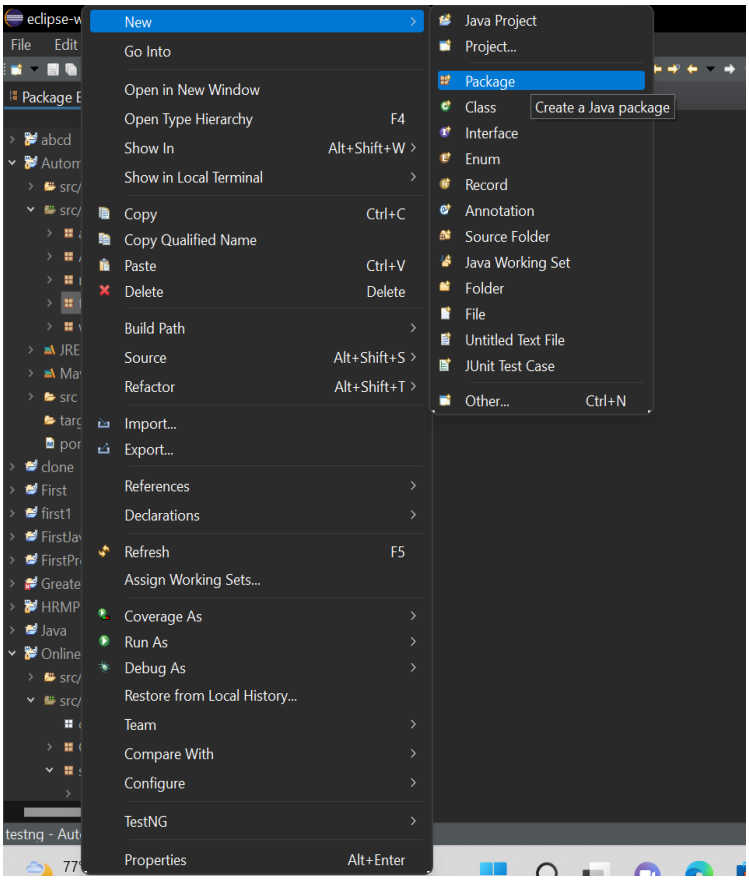
Go to Help -> Eclipse marketplace -> search for TestNG and click install



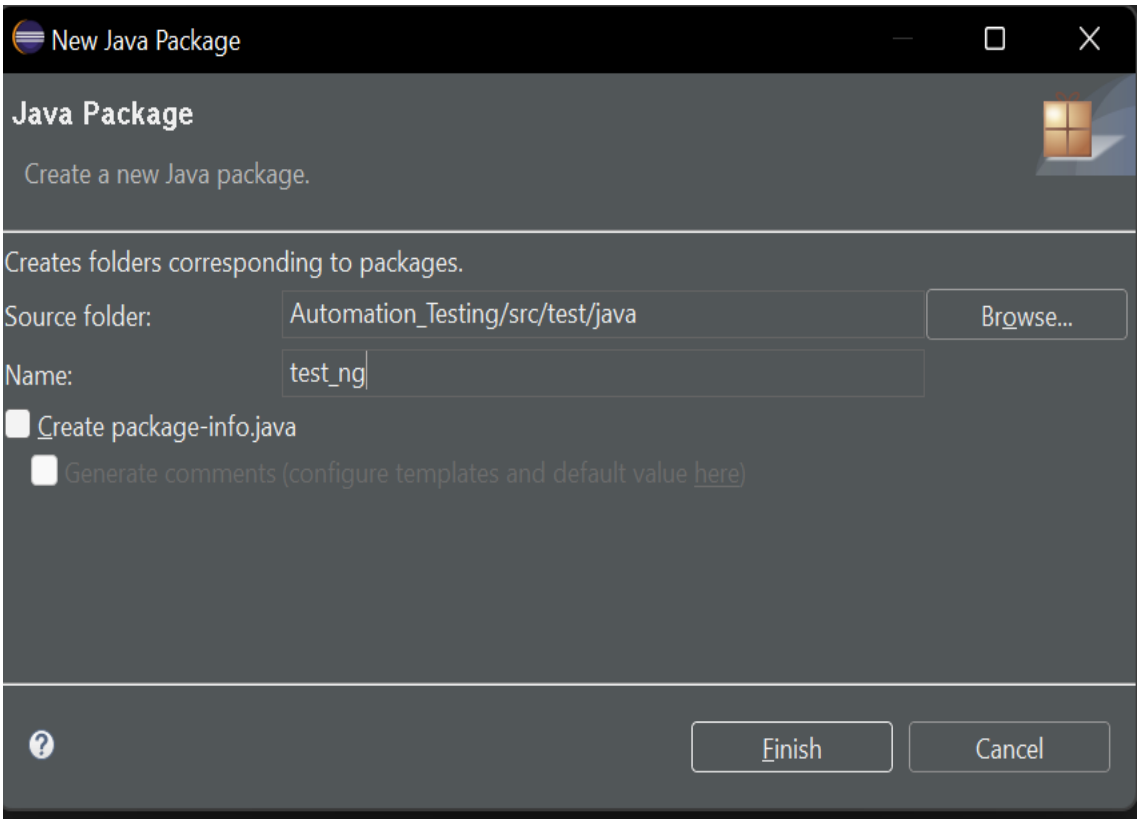


b) Creating a TestNG class

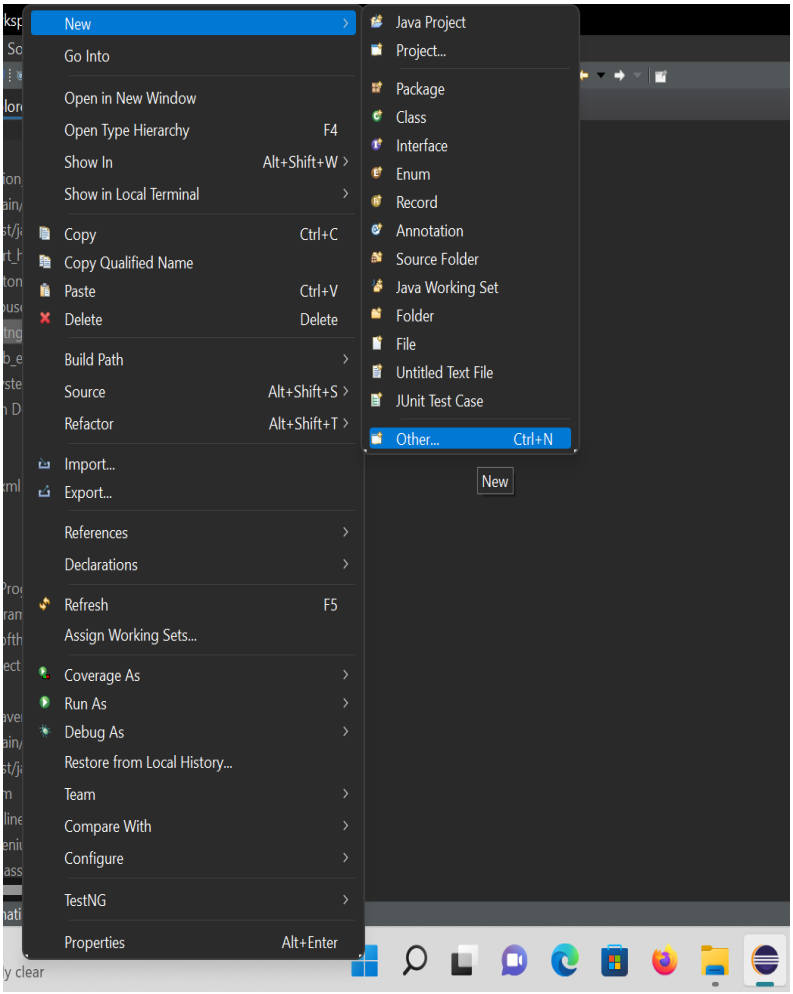
Now that TestNG has been installed, go inside the test java folder in our maven project and create a new package

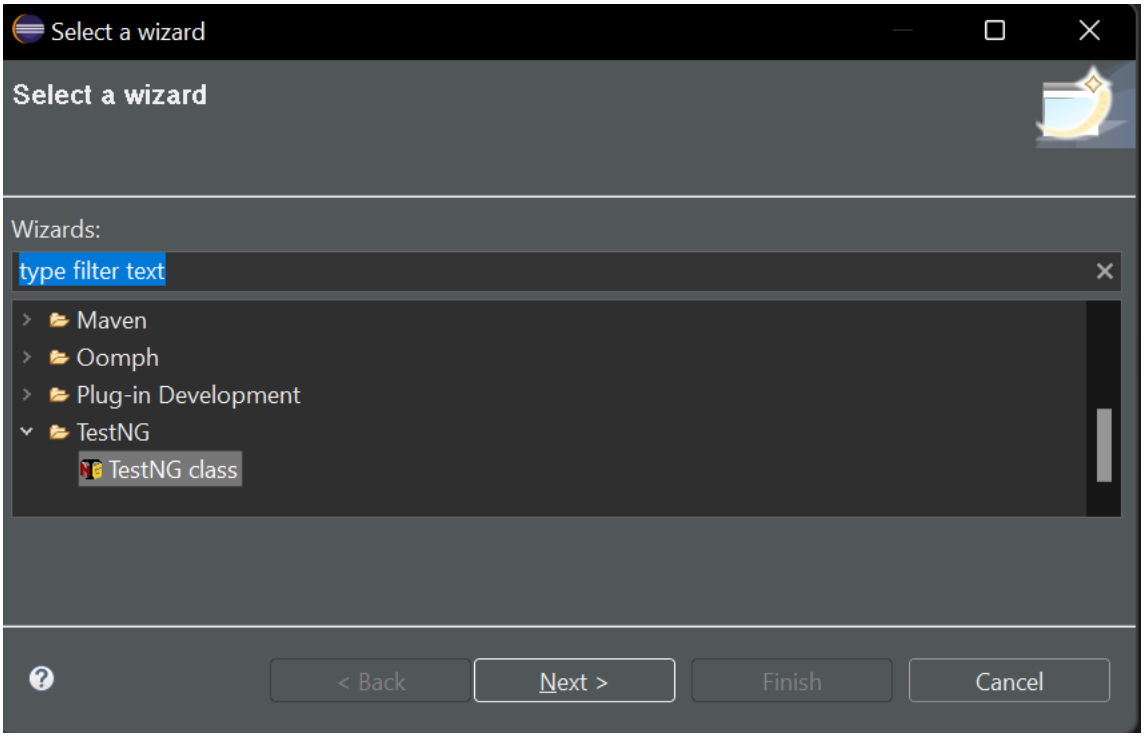


Give the package an appropriate name

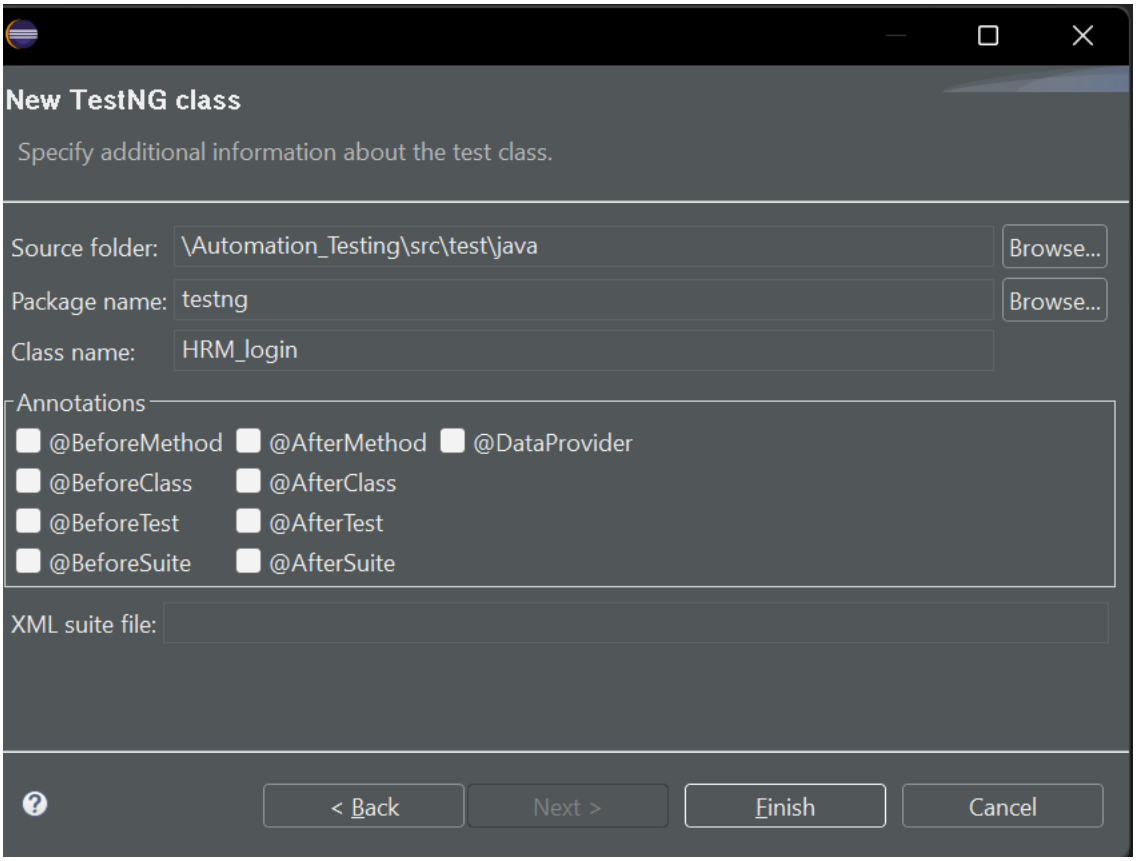


Now that a package has been created, right click on the package and go to Other -> TestNG -> TestNG class

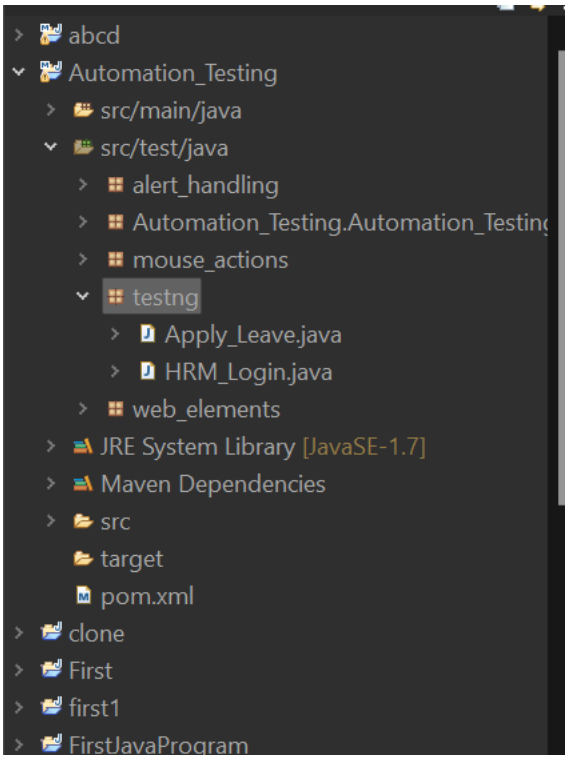




Here, give the name for your TestNG class. You can also select the annotations needed for you in the testing process. For now, I am only considering the default annotation that is @test.



Once the classes have been created, your folder structure will look something like this. I have the Apply leave and HRM login classes inside the testing package. These classes are testNG classes, and the package is inside the maven project named Automation_Testing.



Write a program to HRM Login

This program is for automating the login process of OrangeHRM using selenium. The single test case is run using the testNG framework.

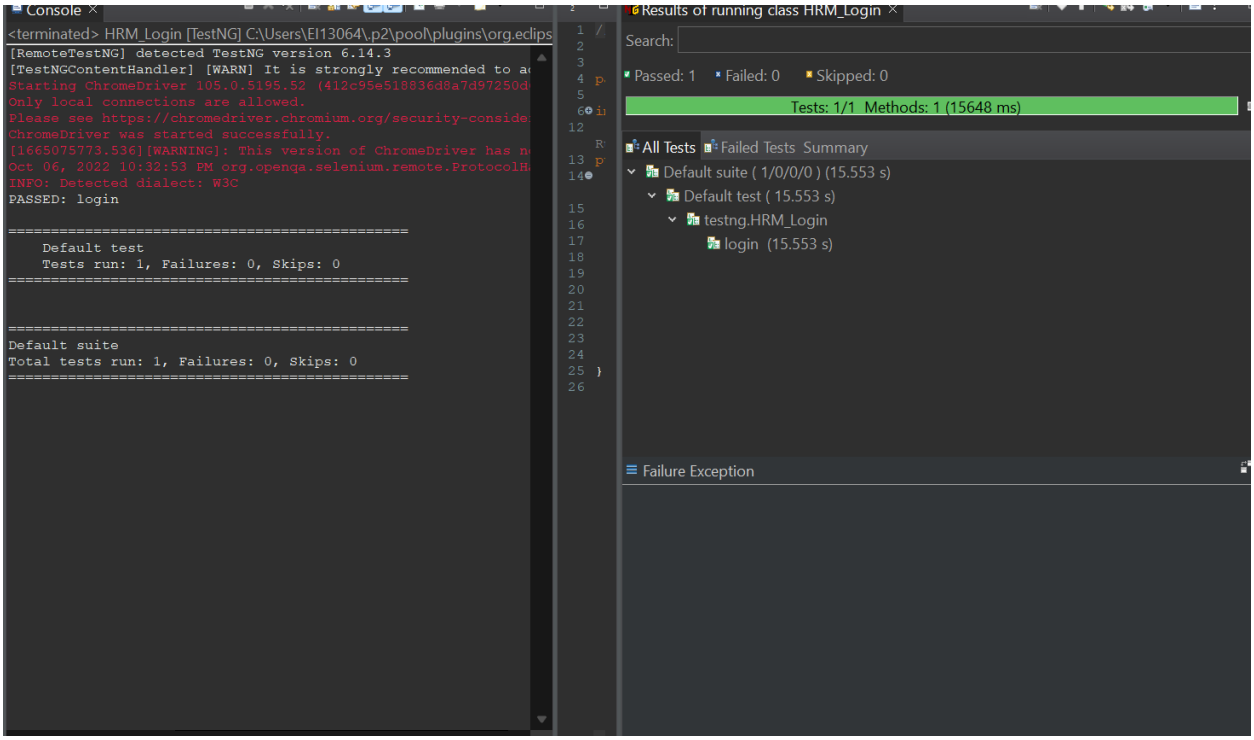
i) In line 4 we import the testing package needed
ii) line 14 denotes the test annotation. Everything under the method under annotation is considered as a single test case. This makes more sense later when the program is executed.
iii) Line 16 and 17 are the default driver code, used to set up the chrome driver.
iv) Line 18 is the implicitly wait method, used to introduce delay in the automation process. This is needed if the website takes time to load. Here I have given a delay of 100 seconds
v) In line 19-21, we are visiting in the OrangeHRM website, sending the username and password using the sendkeys method.
vi) In line 23, we are using the Xpath to locate the login button, and the click method to click it.

HRM_Login.java

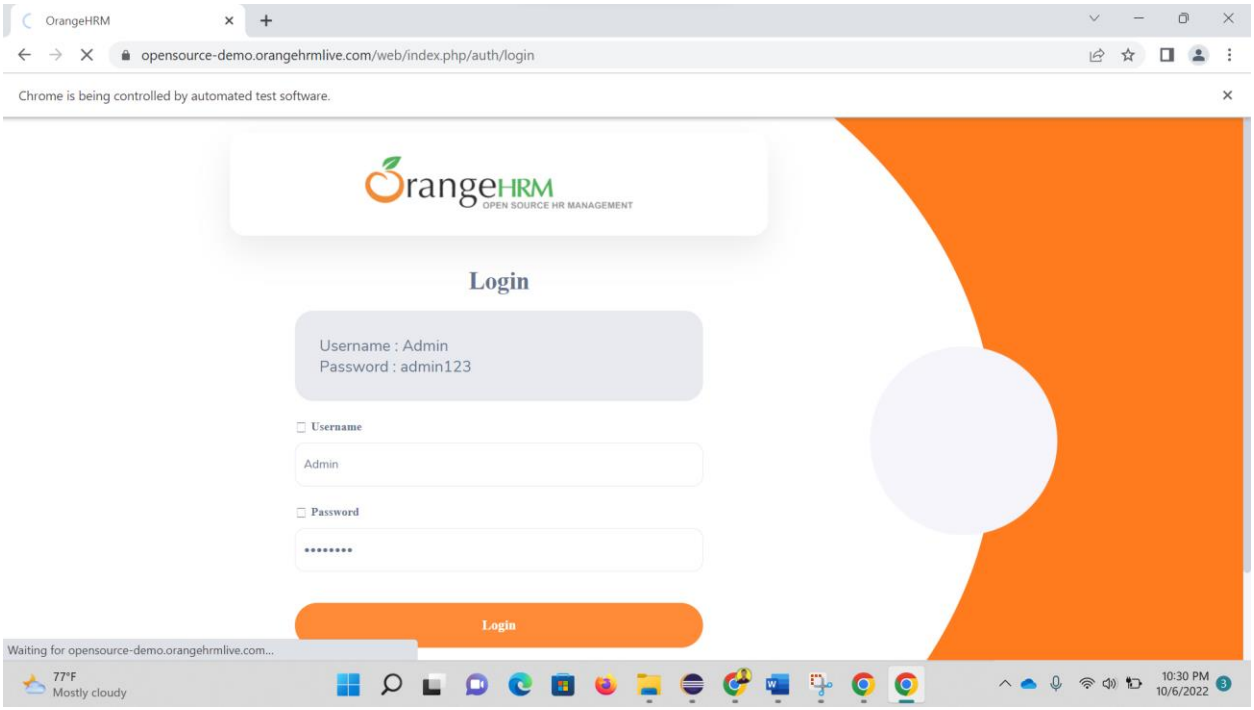
```
1  // This program executes the automation testing on OrangeHRM website for the login functionality
2
3
4  package testng;
5
6  import java.util.concurrent.TimeUnit;
12
   Run All
13 public class HRM_Login {
14   @Test
     Run | Debug
15   public void login() {
16         System.setProperty("webdriver.chrome.driver", "C:\\brayan1\\chromedriver.exe");
17         WebDriver driver = new ChromeDriver(); // Instatiating an object of type Webdriver
18         driver.manage().timeouts().implicitlyWait(100, TimeUnit.SECONDS); //Used to introduce delay for the webpage to load
19         driver.get(" https://opensource-demo.orangehrmlive.com/web/index.php/pim/viewEmployeeList");
20         driver.findElement(By.name("username")).sendKeys("Admin");
21         driver.findElement(By.name("password")).sendKeys("admin123");
22         //driver.findElement(By.name("login")).click();
23         driver.findElement(By.xpath("//*[@id=\"app\"]/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button")).click();
24   }
25 }
26
```

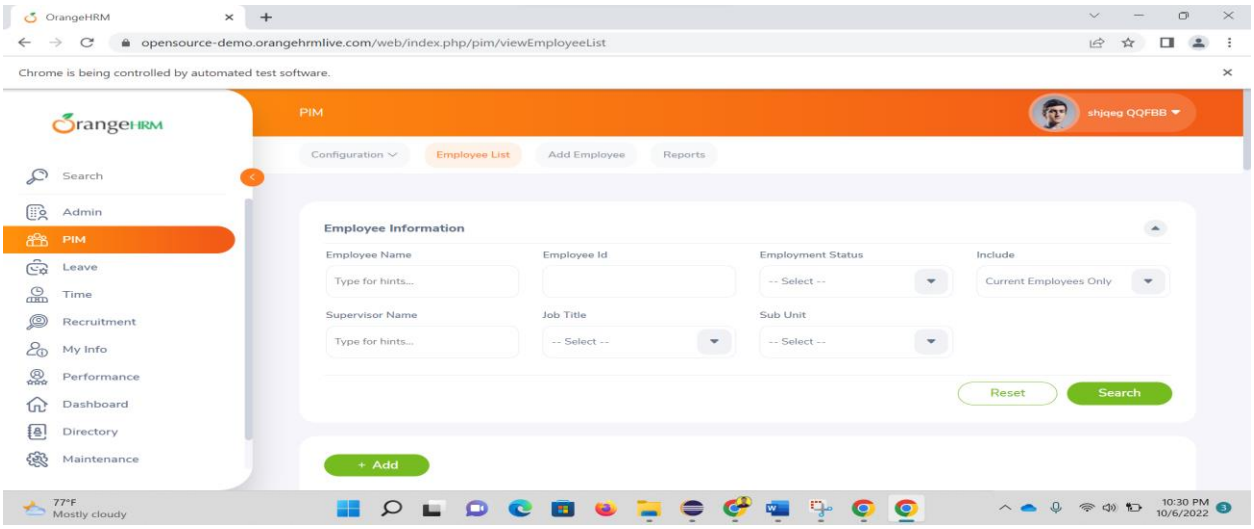
Output console

This output console shows the test cases as passed in a more structured way, thanks to TestNG.



Automation output





Write a program to apply leave

i) In line 3-10, we import all the needed packages including testing and webdriver

ii) Line 15 and 16 are the commands needed to bring the chrome driver file into the program and to make an object of type chrome driver.

iii) Line 17-21 involves logging into the HRM. Here we use Xpath locator to locate and enter the text into the elements we use sendkeys

iv) We then find the sidebar element of the dashboard and the apply button using link text locator

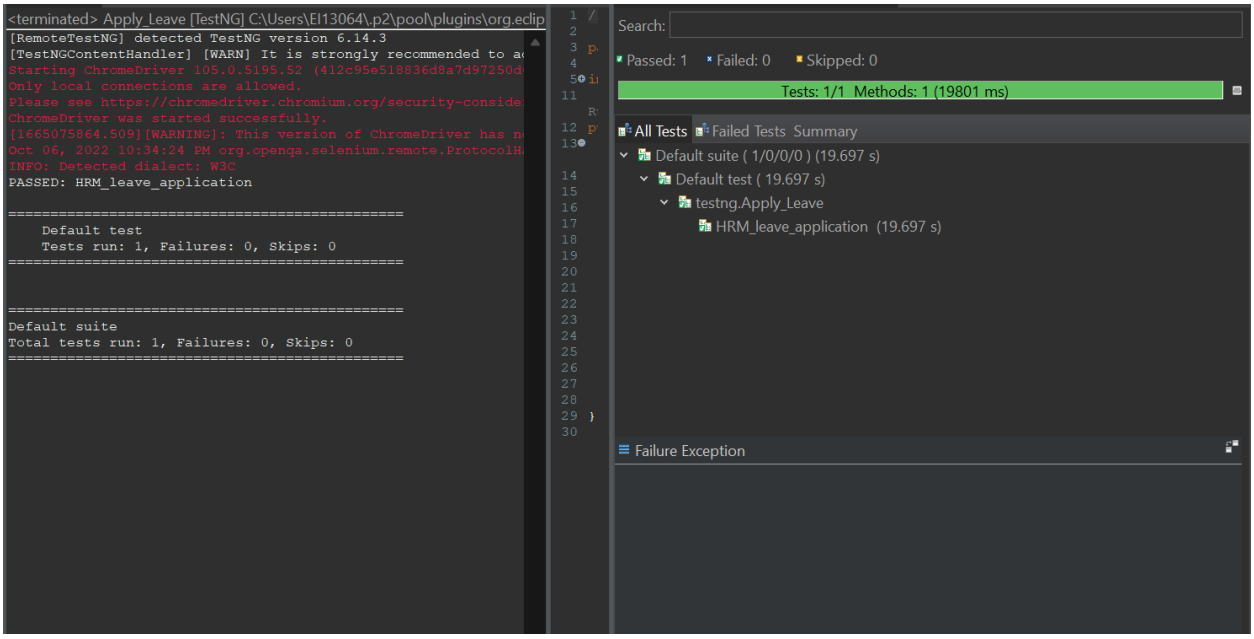
v) line 24 is used to click on the drop-down arrow and line 25 is used to select an option using the. contains filter

vi) We then send the leave end date using the sendkeys into the date field and click on apply

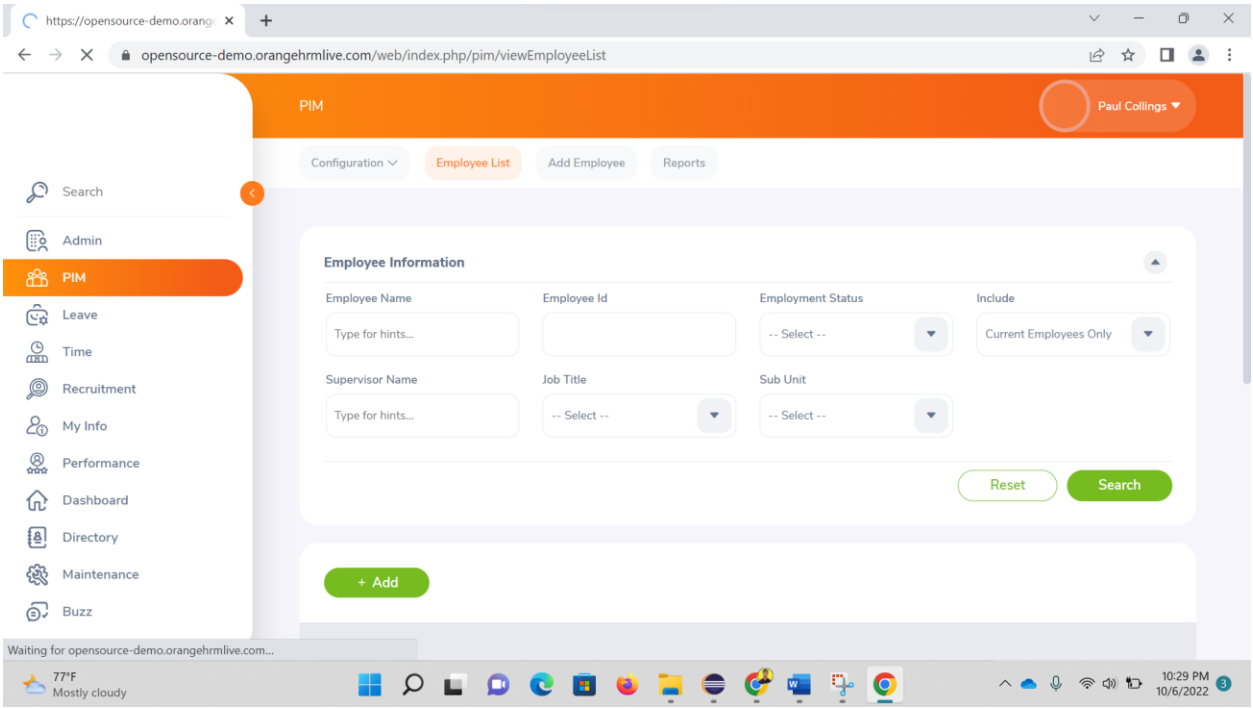
HRM_Login.java | Apply_Leave.java

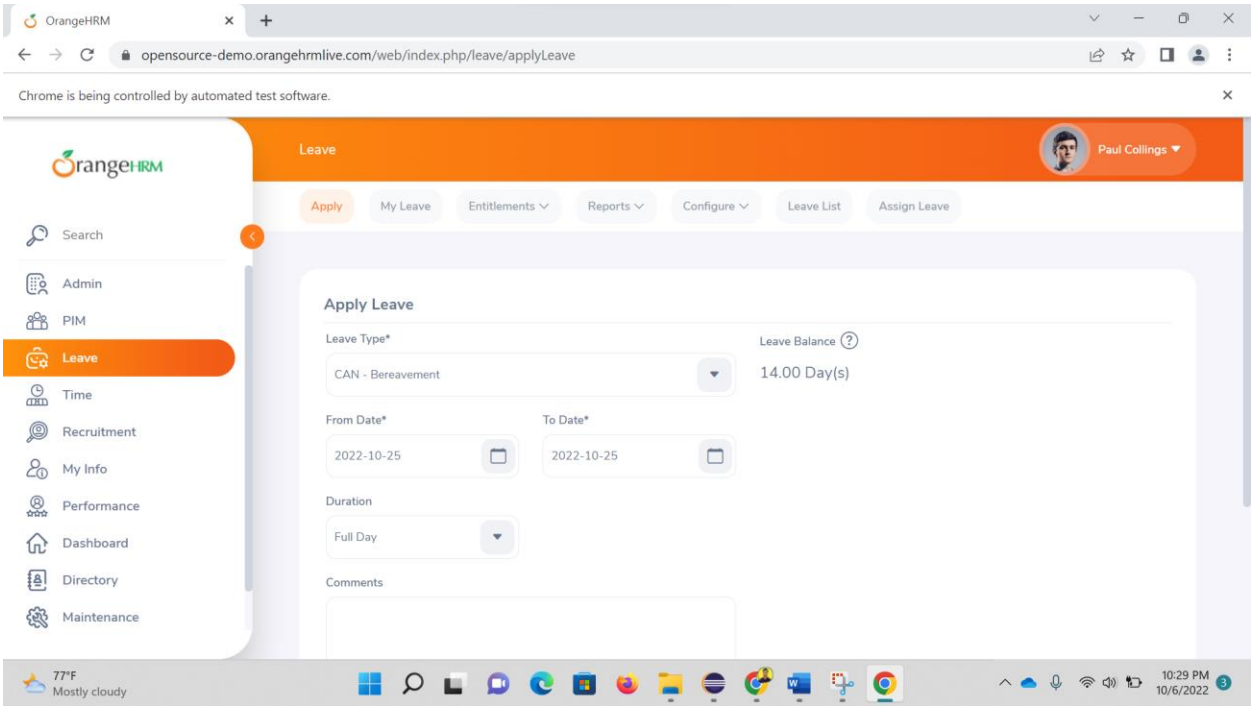
```
1  // This program automates the process of leave application in OrangeHRM website
2
3  package testng;
4
5  import java.util.concurrent.TimeUnit;
6
7  import org.openqa.selenium.By;
8  import org.openqa.selenium.WebDriver;
9  import org.openqa.selenium.chrome.ChromeDriver;
10 import org.testng.annotations.Test;
11
   Run All
12 public class Apply_Leave {
13    @Test //This is the test annotation used in TestNG
      Run | Debug
14    public void HRM_leave_application() {
15          System.setProperty("webdriver.chrome.driver", "C:\\brayan1\\chromedriver.exe");
16          WebDriver driver = new ChromeDriver();
17          driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");
18          driver.manage().timeouts().implicitlyWait(1000, TimeUnit.MILLISECONDS);
19          driver.findElement(By.xpath("//div/input[@name = 'username']")).sendKeys("Admin");
20          driver.findElement(By.xpath("//input[@type = 'password']")).sendKeys("admin123");
21          driver.findElement(By.xpath("//button[@type = 'submit']")).click(); // Clicking on login button
22          driver.findElement(By.linkText("Leave")).click(); // Clicking on the leave element on sidebar
23          driver.findElement(By.linkText("Apply")).click();  // clicking on apply button for leave
24          driver.findElement(By.xpath("//div[@class='oxd-select-text oxd-select-text--active']/div[2]/i")).click();
25          driver.findElement(By.xpath("//*[contains(text(),'CAN')]")).click(); // selecting an option from dropdown using the contains filter
26          driver.findElement(By.xpath("//div/div[2]/div/div/input")).sendKeys("2022-10-25"); // sending the leave date
27          driver.findElement(By.xpath("//button[@type='submit']")).click(); // clicking on submit application
28    }
29 }
30
```

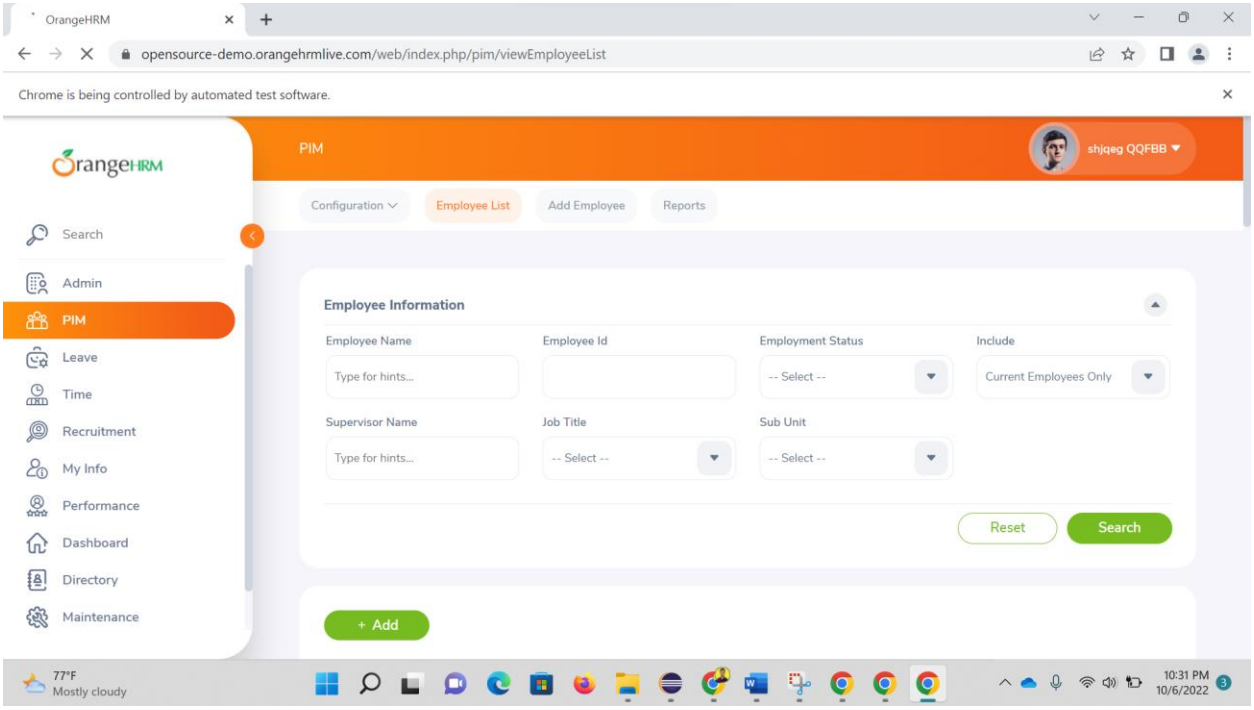
Console output



Automation output




OrangeHRM
opensource-demo.orangehrmlive.com/web/index.php/leave/applyLeave
Chrome is being controlled by automated test software.
Leave
Paul Collings
Apply
My Leave
Entitlements
Reports
Configure
Leave List
Assign Leave
Search
Admin
PIM
Leave
Time
Recruitment
My Info
Performance
Dashboard
Directory
Maintenance
Apply Leave
Leave Type*
CAN - Bereavement
Leave Balance
14.00 Day(s)
From Date*
2022-10-25
To Date*
2022-10-25
Duration
Full Day
Comments
77°F
Mostly cloudy
10:29 PM
10/6/2022


OrangeHRM
opensource-demo.orangehrmlive.com/web/index.php/pim/viewEmployeeList
Chrome is being controlled by automated test software.
PIM
shjqeg QQFBB
Configuration
Employee List
Add Employee
Reports
Search
Admin
PIM
Leave
Time
Recruitment
My Info
Performance
Dashboard
Directory
Maintenance
Employee Information
Employee Name
Type for hints...
Employee Id
Employment Status
-- Select --
Include
Current Employees Only
Supervisor Name
Type for hints...
Job Title
-- Select --
Sub Unit
-- Select --
Reset
Search
+ Add
77°F
Mostly cloudy
10:31 PM
10/6/2022

User method over loading for searching as discussed in the class assignment

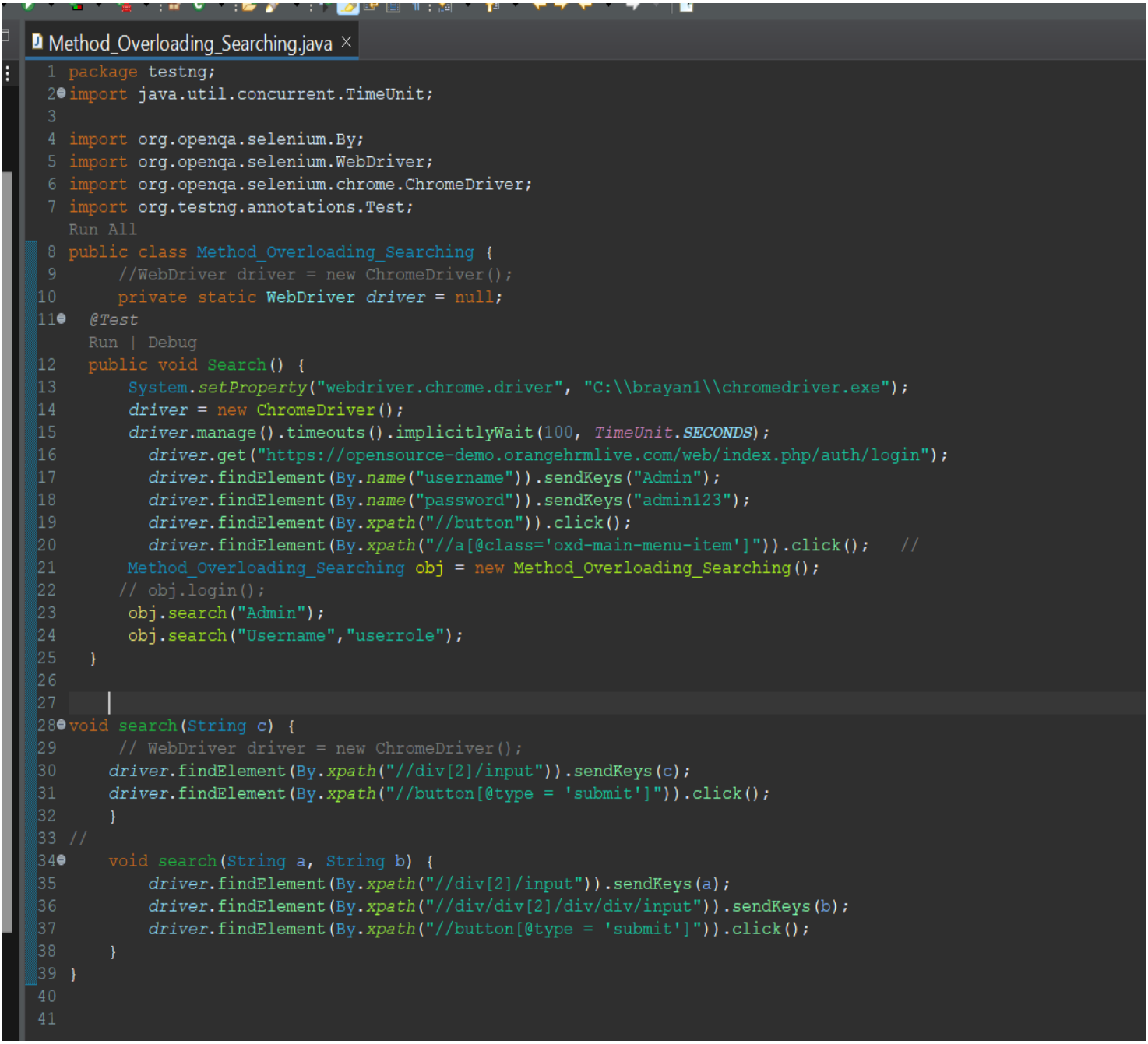
i) In line 2-7, we import the needed packages

ii) Inside the search method in the test annotation, we login into HRM, click on the admin element in the dashboard. We also create an object of the main public class i.e. Method_Overloading_Searcing. All this happens from line 12-21.

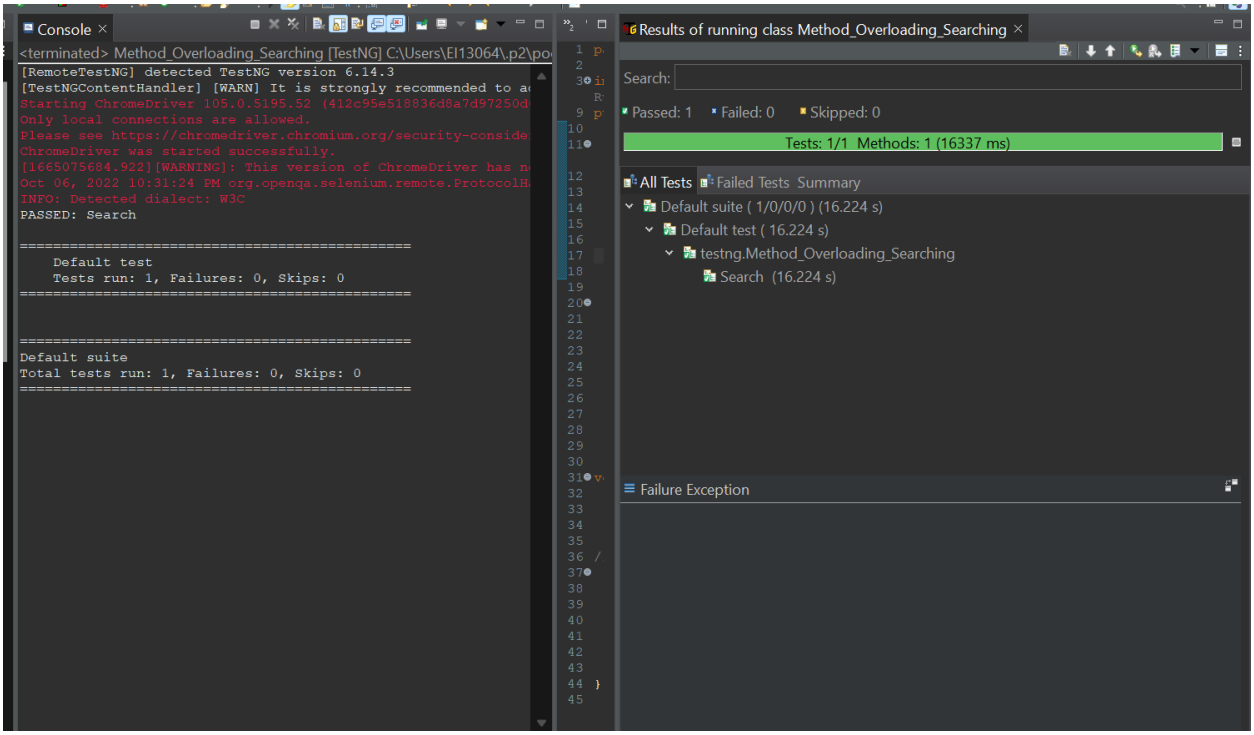
iii) Next in line 23 and 24, we call the same search function using different parameters to perform the method overloading.

iv) The first function searches only based on the name and the second function searches based on the name and user role. So, method overloading takes place in line 23 and 24

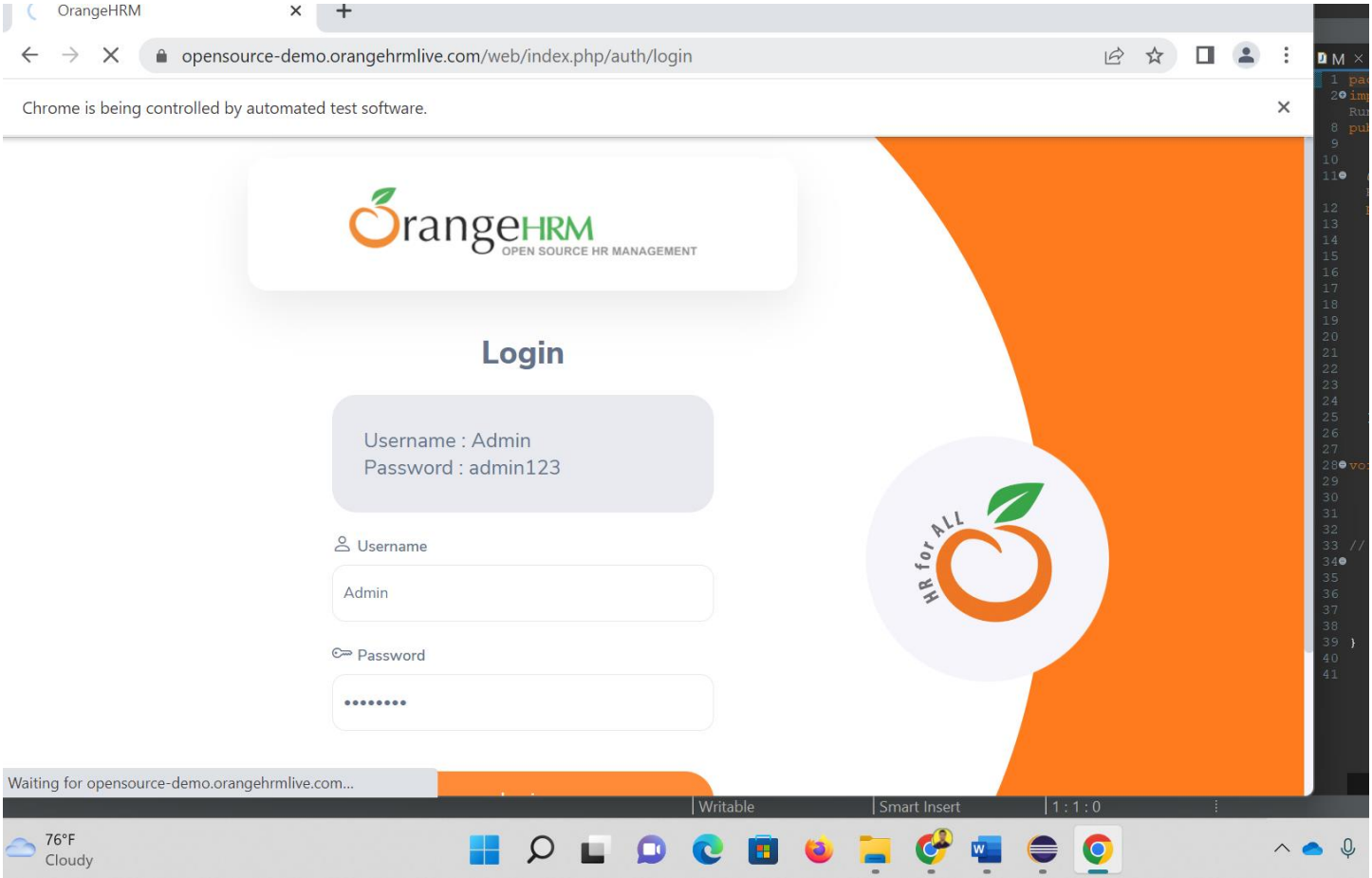
v) Line 28-39 contains the definitions for both these functions. Note that these definitions are not part of the @test annotation.

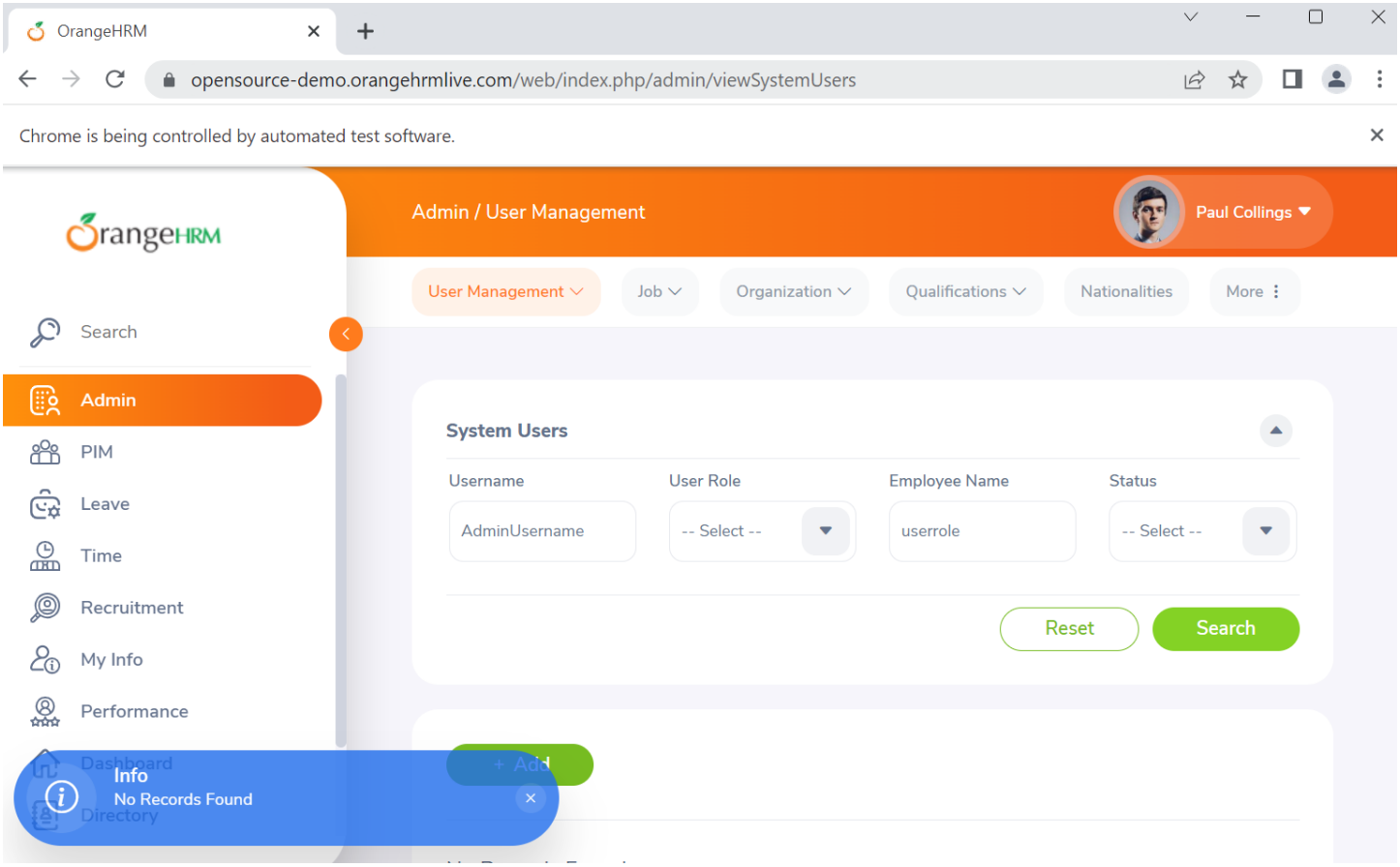


Console Output



Automation output




OrangeHRM
opensource-demo.orangehrmlive.com/web/index.php/admin/viewSystemUsers
Chrome is being controlled by automated test software.
OrangeHRM
Admin / User Management
Paul Collings
User Management
Job
Organization
Qualifications
Nationalities
More
Search
Admin
PIM
Leave
Time
Recruitment
My Info
Performance
Dashboard
Directory
System Users
Username
AdminUsername
User Role
-- Select --
Employee Name
userrole
Status
-- Select --
Reset
Search
+ Add
Info
No Records Found