

HEALTH - APP

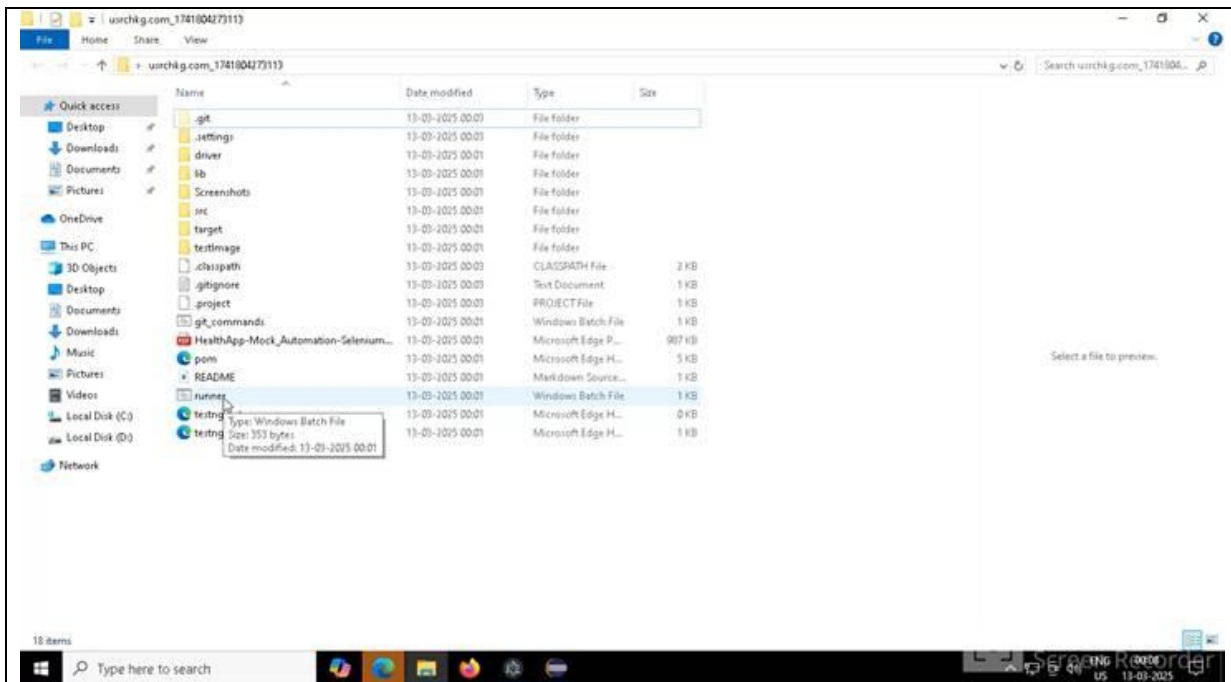
VERIFICATION MODULE
(JAVA & CUCUMBER)

Table of Contents

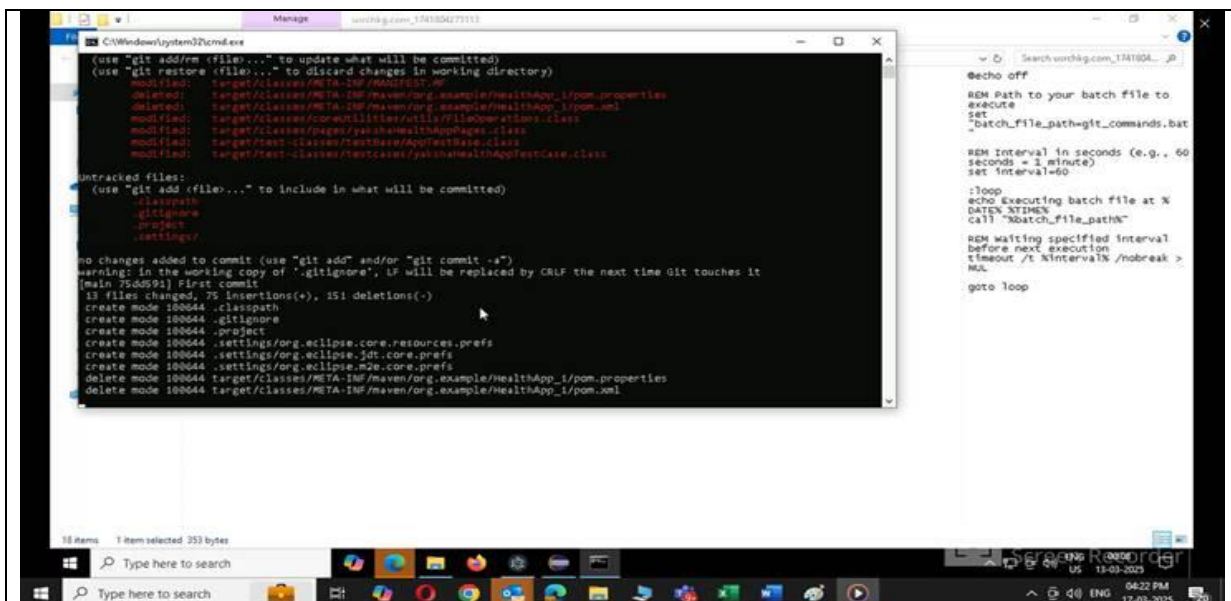
<i>Pre-Requisite</i>	3
<i>Template Code Structure:.....</i>	6
<i>PROBLEM STATEMENT:.....</i>	7
<i>Key Activities to implement:.....</i>	7
<i>EXECUTION STEPS TO FOLLOW:</i>	9

Pre-Requisite

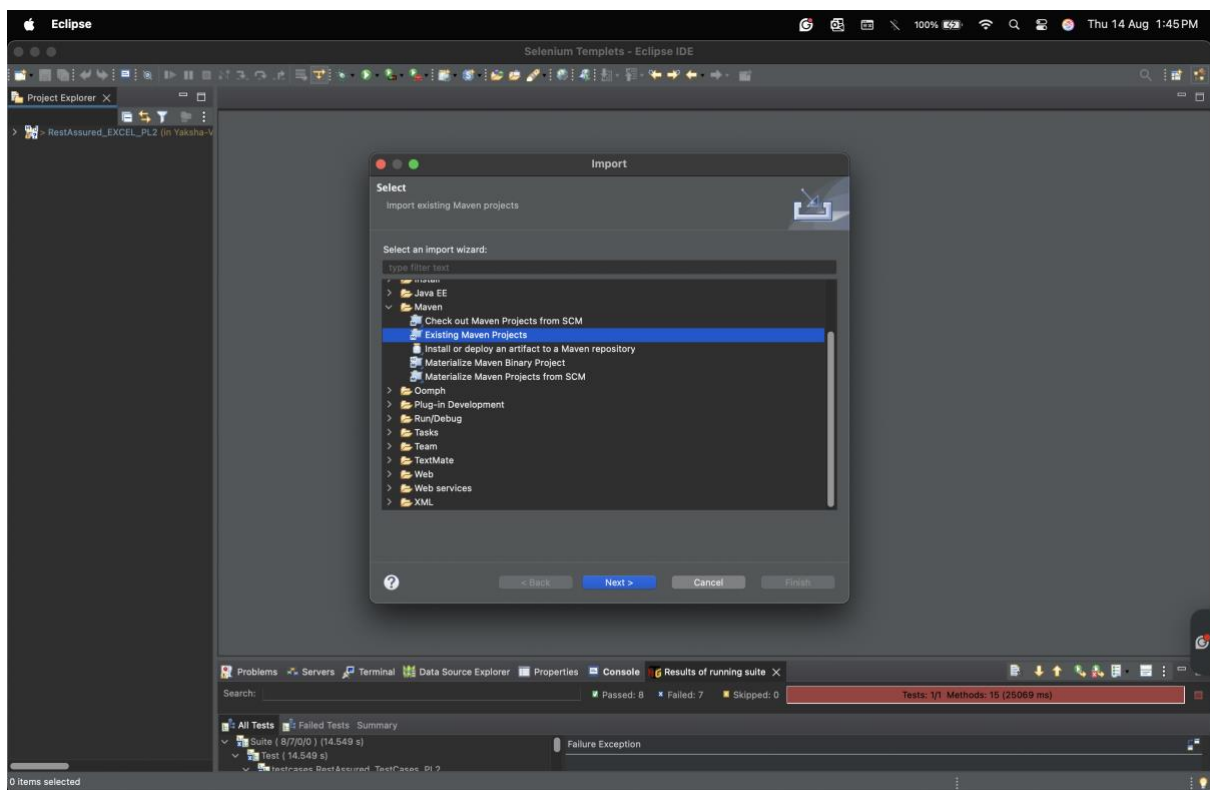
1. Before you start working on your project, **execute the runner file** present in your project folder (Simply by double click) and keep it running. **This is mandatory.**



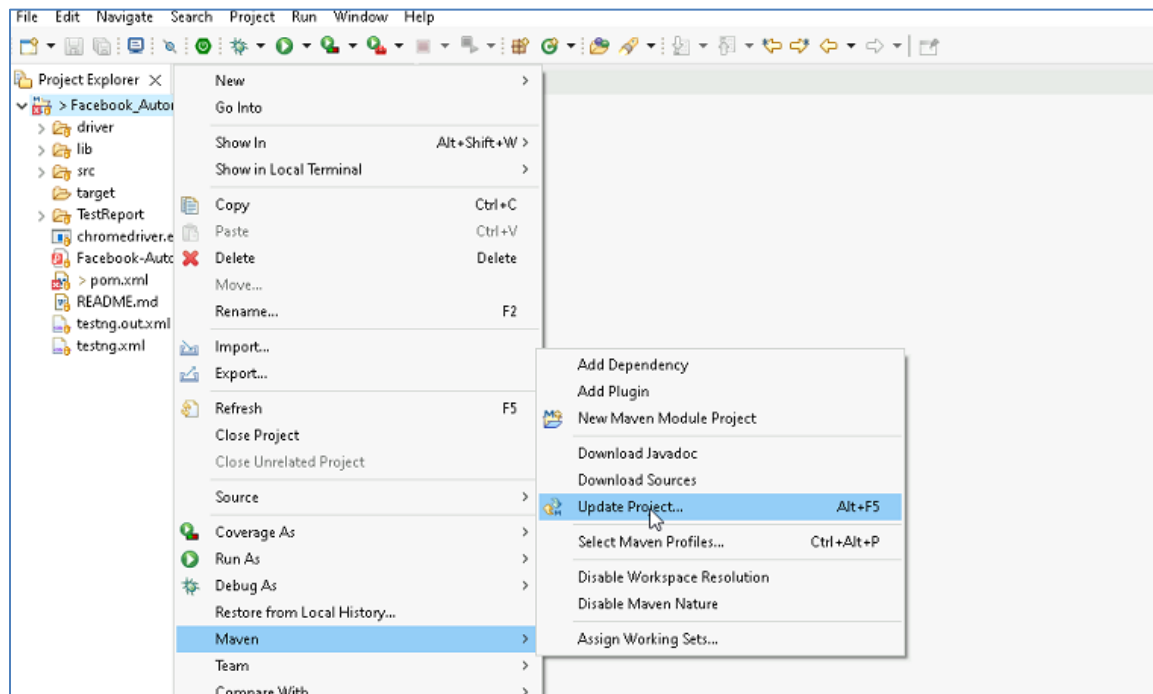
This will launch a command terminal for you where it will keep on pushing your updated code to GIT on regular intervals. Keep that command terminal open at backend and you can continue working on your project.



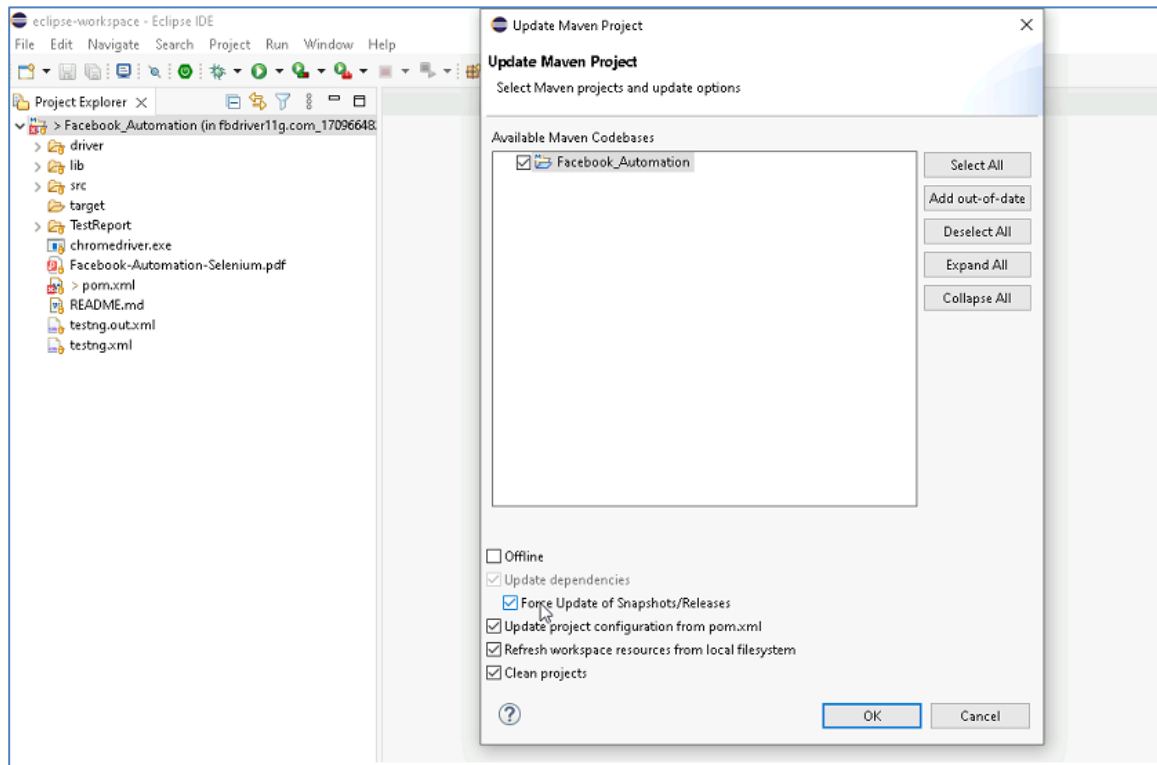
2. Launch the Eclipse IDE(Shortcut icon on VM's Desktop) and import your cloned project to Eclipse as an Existing maven project.



3. As soon as you import the project in Eclipse, update the project using the maven update option as below(if required) . This is to resolve the issue if any Maven dependency is not downloaded properly.
 - a. Right-click on the project: Go to “Maven” and select “Update Project”.



- b. In the Update Maven Project Box Select “Force Update of Snapshots/Releases” and click OK.



Template Code Structure:

Instructions:

1. You only need to work on the packages and files listed in the table below.
2. Ignore all other files and packages.
3. Do **not** make any changes to files outside the listed ones, as this will affect your evaluation.
4. The “**Test**” folder is non-editable. Any attempt to edit and save files there will cause errors and impact your evaluation.

Package	Class/File	Description
src/main/java/coreUtilities/utis/	FileOperations.java	<ul style="list-style-type: none"> The method for reading data as input from an Excel file have already been implemented here. This method is used to fetch the required data from excel including the URL for navigation.
src/main/java/coreUtilities/utis/	CommonEvents.java	<ul style="list-style-type: none"> Create additional supportive common methods here.
/src/main/java/pages	Verification_Page.java	<ul style="list-style-type: none"> Perform all core activities listed under the table “Key Activities to implement” here.(This table is given below.) Refer to the comments in each templated method to understand what’s expected. Define locators and XPathS in this file. Declare any variables or objects needed to share data or status between methods. Do not change the method signatures. If needed.create additional supportive common methods in the CommonEvents class.
/src/test/java/features/	Verification_PL1.feature	<ul style="list-style-type: none"> The file has incomplete scenarios. You must fill them in according to the requirement. Steps should follow Given-When-Then structure. The scenarios should fully validate the requirement.
src/test/java/steps/	VerificationStep.java	<ul style="list-style-type: none"> It’s a step definition file. Each step in Verification_PL1.feature must be linked here.
/src/main/resources/	Config.xlsx	<ul style="list-style-type: none"> URL to navigate to is already defined here
	Login.xlsx	<ul style="list-style-type: none"> Contains valid credentials required for use login
	expected_data.xlsx	<ul style="list-style-type: none"> Contains data to fill in form during TCs execution.
	testng_1.xml	<ul style="list-style-type: none"> Execution needs to kick started from testng_1.xml

PROBLEM STATEMENT:

We need to automate the following activities using Java+Selenium+Cucumber.

Key Activities to implement:

Use the credentials to log in to the “Health App” application.

User Name: “admin”

Password: “pass123”

SI No.	Summary	Action	Expected Result
1	Verify to filter the records by Select active drop down from Requisition Status	<ol style="list-style-type: none">1. Login to the health app2. Scroll and click the “Verification” from left navigation menu.3. Click on "FROM" and select "Jan 2020".3. Click on "TO" and select “Current Date”.4. Click the OK5. Click on drop down6. Click on "Active" drop down option	<p>The 'Req. Status' column should match the option Selected from the dropdown menu.</p> <p>All function implementing this test case must return a Boolean value: True if the user is successfully performs the activity.False otherwise.</p>
2	Verify the presence of view button	<ol style="list-style-type: none">1. Login to the health app2. Scroll and click the “Verification” from left navigation menu.3. Select "All" radio button4. Select "Active" drop down option from req. Status.5. Click on "FROM" and select "Jan 2020"6. Click on "TO" and select "current" Date7. Click on "OK" button.8. Click on "View" button of first record in the table.	<p>Verify the requisition Id should match with the record.</p> <p>All function implementing this test case must return a Boolean value: True if the user is successfully performs the activity.False otherwise.</p>
3	Verify Record Count Matches in Purchase Request	<ol style="list-style-type: none">1. Login to the health app2. Scroll and click the “Verification” from left navigation menu.3. Click on inventory section4. Click on the "Purchase Request" sub-tab.5. Find and click the "All" radio button.	<p>The result count displayed at the bottom of the page should match the total record count.</p> <p>All function implementing this test case must return a Boolean value: True if the user is successfully performs the activity.False otherwise.</p>
4	Verify to scrolling of page	<ol style="list-style-type: none">1. Login to the health app2. Scroll and click the “Verification” from left navigation menu.3. Click on inventory section4. Click on the "Purchase Request" sub-tab.5. Verify "Pending" radio button is visible6. Scroll all the way to the bottom of the page7. Verify "Previous" button is visible8. Scroll all the way to the top of the page9. Verify "Pending" radio button is visible	<p>The user should be able to scroll up and down</p> <p>All function implementing this test case must return a Boolean value: True if the user is successfully performs the activity.False otherwise.</p>
5	Verify the required "Item" field in the Purchase Request	<ol style="list-style-type: none">1. Login to the health app2. Scroll and click the “Verification” from left navigation menu.	<p>Verify error message as "Item is required" is displayed.</p>

	section under the Inventory module.	3. Navigate to "Inventory" module. 1. Navigate to the "Internal" Subsection. 2. Click on "Purchase Request". 3. Click on the "Create Purchase Request" button. 4. Click on the "Request" button.	All function implementing this test case must return a Boolean value: True if the user is successfully performs the activity.False otherwise.
--	-------------------------------------	--	---

NOTE: “Please do not delete any file in the src folder. But you are free to add any other file”.

Expectations:

- 1) Learners should write automation script using Java and selenium to automate all the steps in the above question. In other words, automation script should perform all mentioned steps.
- 2) Learners should not use any tools to create the xpath. They should develop the xpath/cssselector on their own.

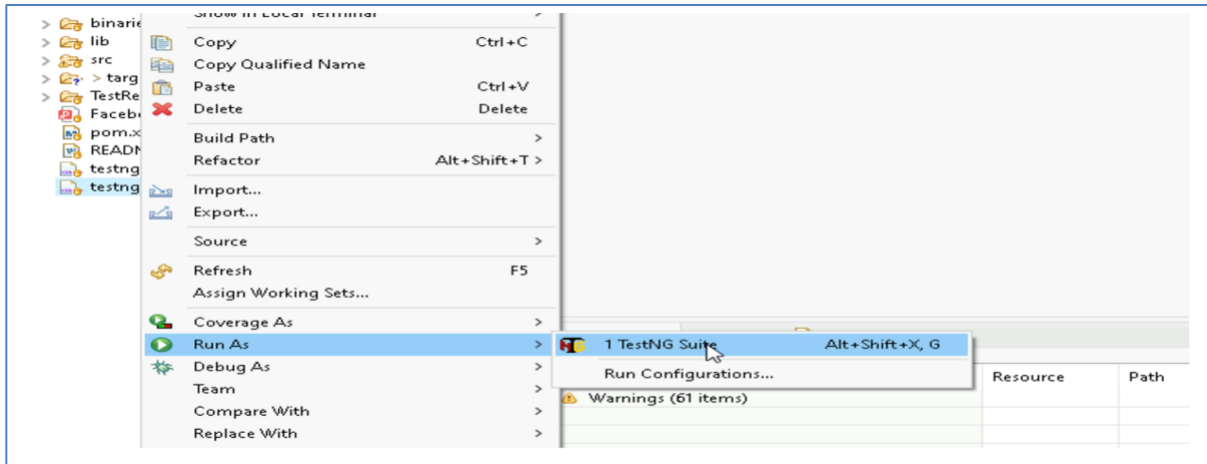
IMPLEMENTATION/FUNCTIONAL REQUIREMENT

1.1 CODE QUALITY/OPTIMIZATIONS

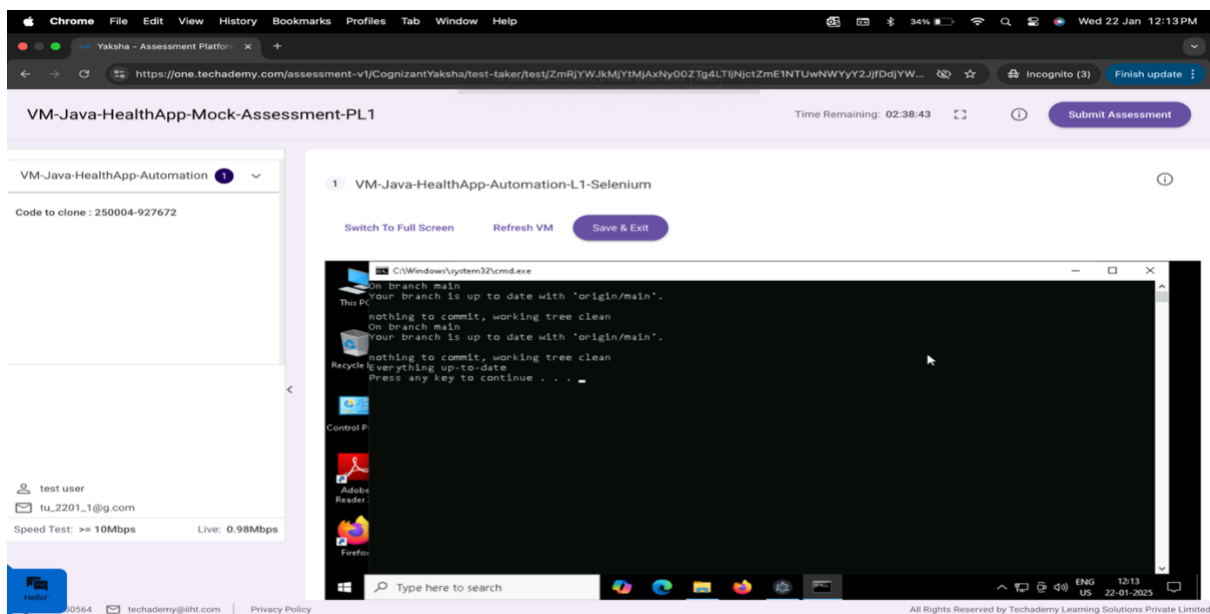
1. Associates should have written clean code that is readable.
 2. Associates need to follow SOLID programming principles.
-

EXECUTION STEPS TO FOLLOW:

1. You are mandatory required to run test cases for applications before final submission. Without this project evaluation will not happen.
2. You can launch test cases any time as follows: Right-click on testng.xml and run as TestNGSuite.



3. To do the final submission of the assessment :
 - a. Press escape to come out of Fullscreen mode.
 - b. Submit the assessment (**Press the button on right hand side , top corner**).



After the successful submission of the assessment, you will get a confirmation message displayed on your screen.

=====

All the Best