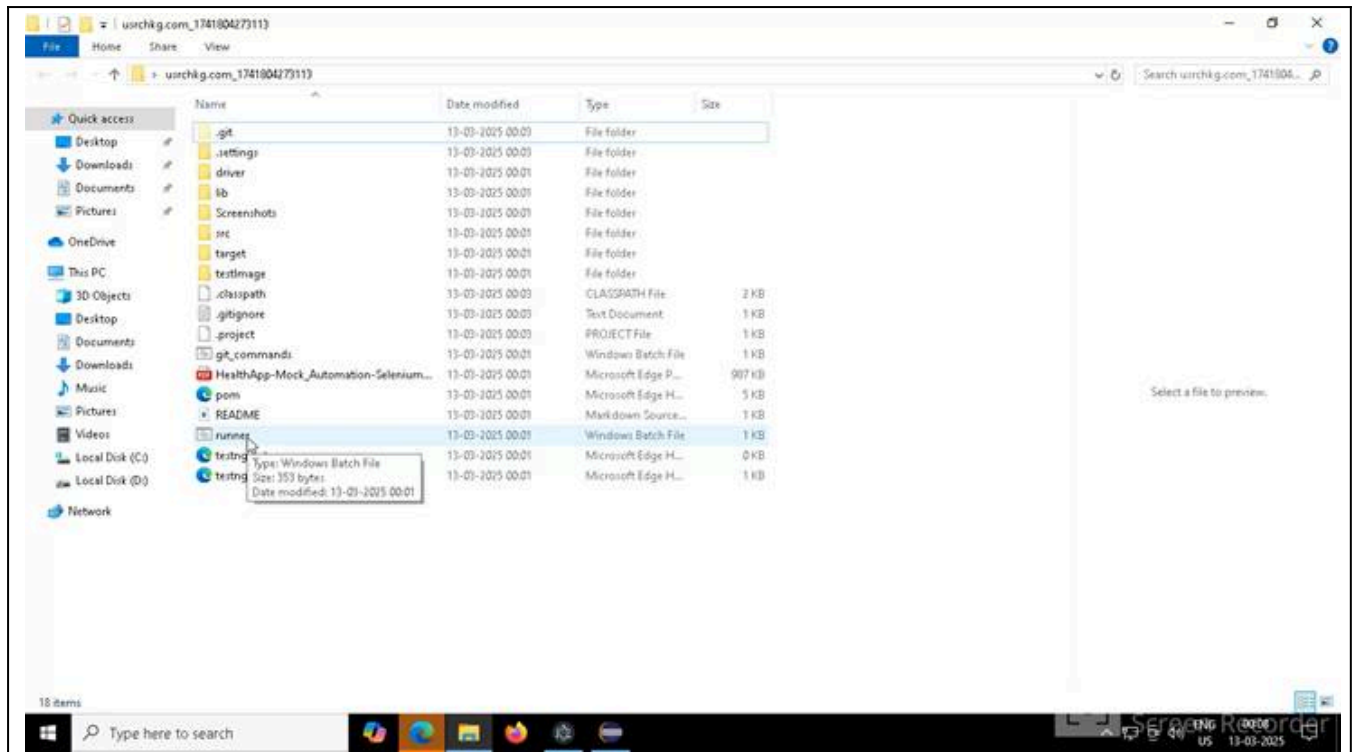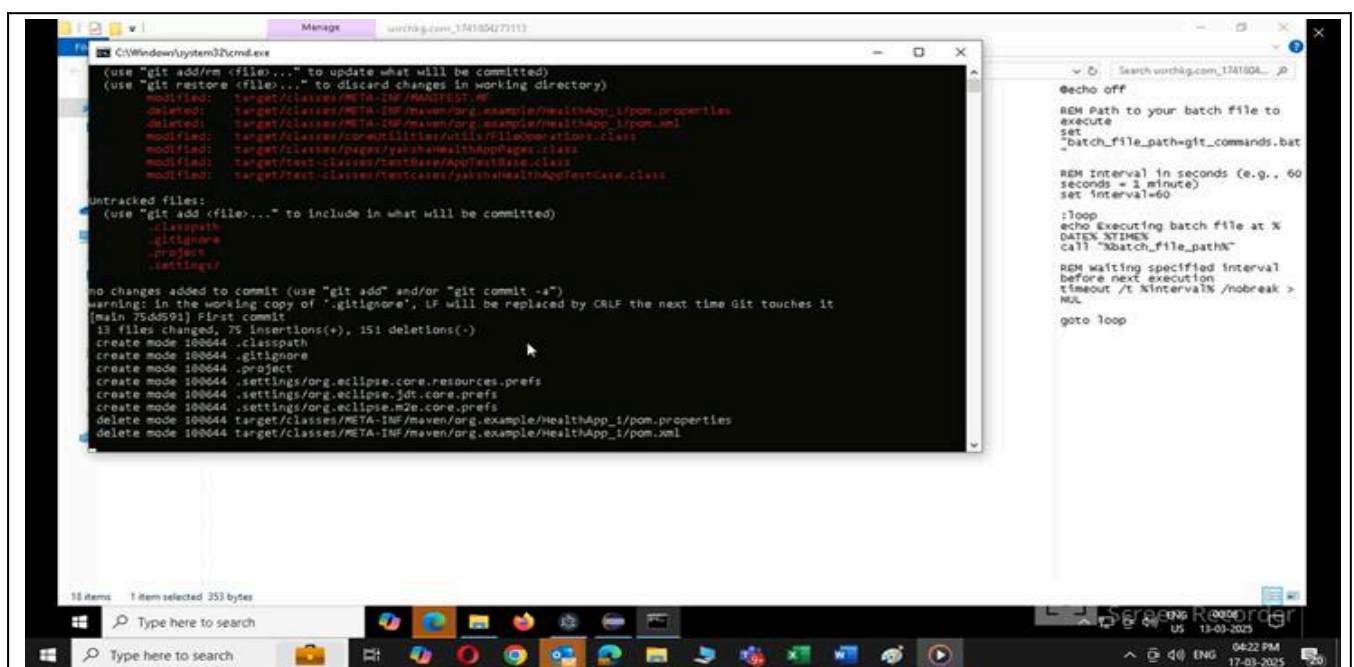# HEALTHAPP AUTOMATION PATIENT MODULE – PL1

Pre-requisite:

Before you start working on your project, execute the runner file present in your project folder (Simply by double click). <mark>This is mandatory</mark>.
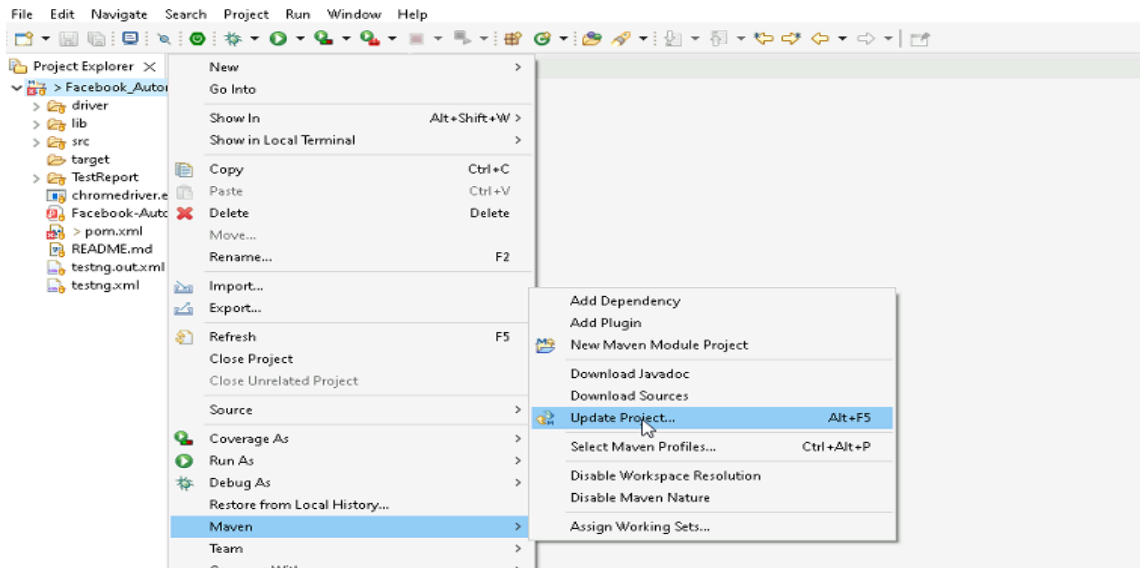

This will launch a command terminal for you where it will keep on pushing your updated code to GIT on regular intervals. Keep that command terminal open at backend and you can continue working on your project.
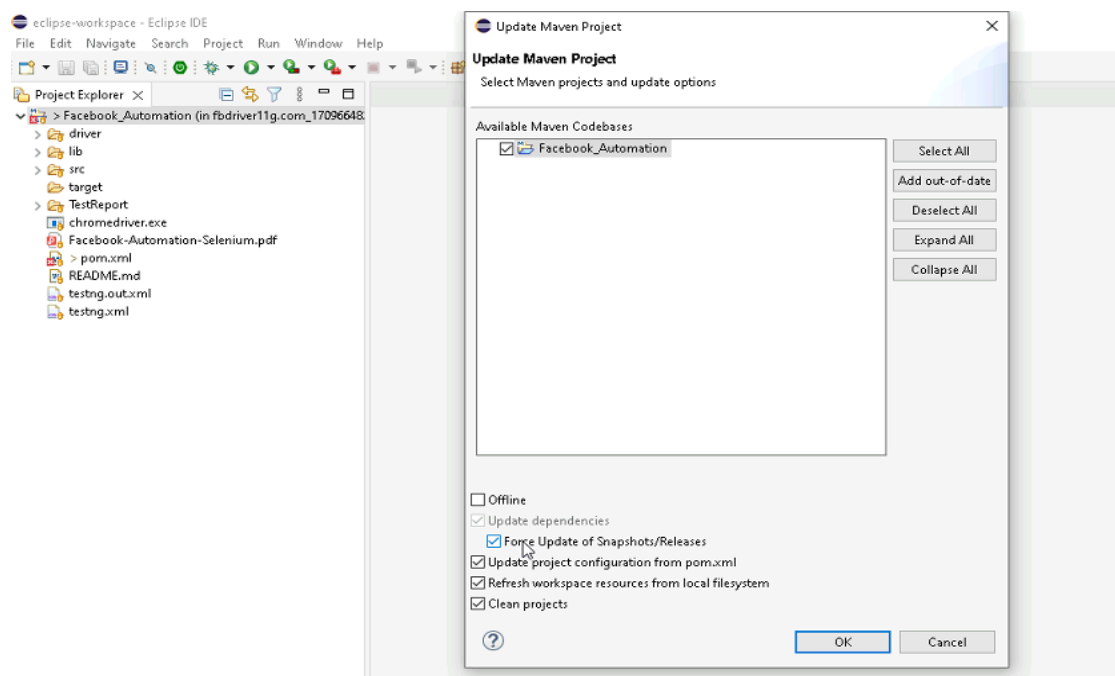
As soon as you import the project in Eclipse, update the project using the maven update option as below. This is to resolve the issue if any Maven dependency is not downloaded properly:

1. Right-click on the project: Go to "Maven" and select "Update Project"

2. In the Update Maven Project Box Select "Force Update of Snapshots/Releases" and click OK

Template Code Structure:

a. Below are the packages and files you will be required to work on.

b. Other Files and packages you can ignore.

c. In other Files and packages do not make any changes. It would affect your evaluation.

d. You are not required to work in the "Test" Folder. Files there are non-editable. Editing those files and trying to save them will throw errors and would affect your evaluation.

Package	Class/File	Description
src/main/java/coreUtilities/utils/	FileOperations.java	**The method for reading data as input from an Excel file has already been implemented here.** **This method is used to fetch the required data from Excel including the URL for navigation.**
/src/main/java/pages	verification_page.java	1. All core activities (mentioned in the list above) are to be performed here. 2. The comments associated with each templated method here describe the expectation. 3. You can define locators and xpath here. 4. Declare any variable/object you need to share data/status between different methods. 5. Do not modify the signature of methods declared here. 6. You can create additional supportive common methods in CommonEvents class.
/src/main/resources/	Config.xlsx	The URL to navigate to is Already defined here
	expected_data.xlsx	Contains data to fill in form\

/src/main/java/coreUtilities/utils	CommonEvents.java	1. Contains all common activities. 2. Certain templated common methods are declared here. 3. You implement them as per your needs. 4. You can add any additional method for common activity here
	Testng.xml	Execution needs to be kick-started from TestNG.xml

PROBLEM STATEMENT

Need to automate the following activities using Selenium + Java.

Key Activities to implement:

Sl No.	Summary	Action	Expected Result
1	* Navigate to the URL. * Retrieve the Title and URL of the Home page. Check if the title & URL matches the expected title.	1. go to the URL: **https://healthapp.yaksha.com/** 2. login as a valid credential **(username: admin, password: pass123)** and click on the "**Sign in**" Button 3. get the title and URL of the Home page, post login	Validate the title and URL of the Home page The title should be : **DanpheHealth** Url should be : **https://healthapp.yaksha.com/Home/Index#/**
2	Ensure the sub-modules "Search Patient" and "Register Patient" are available.	**Preconditions:** The user must be logged into the health app. **Steps:** 1. Click on the expand icon (Toggle key) next to the Patient Module.	Verify that Sub-modules "Search Patient" and "Register Patient" are displayed correctly under the Patient Module.

3	Ensure that the 'Search textbox correctly displays its placeholder text as "Search (Minimum 3 Character)".	**Preconditions**: The user must be logged into the health app and on the 'Search Patient' page within the Patient module.
		Steps:
		  1. Click on the "Search Patient"
		  2. Obtain the placeholder text from the search textbox.
			Verify the placeholder text displayed is "Search (Minimum 3 Character)".
			
			Hint: (Placeholder text is the text value displayed inside the textbox, which will disappear when the user gives any input)

4	Ensure the presence of the "Take a Snapshot" button.	**Preconditions**: The user is logged into the healthApp system and on the "Search Patient" page of the Patient module. **Steps**: 1. Click on the "Register Patient" submodule. 2. Click on the camera icon to start the photo upload process. 3. Click on the "+ New Photo" button.	Verify the presence of the "Take a Snapshot" button.
5	Ensure the error message for the "Phone number" field is **"Primary Phone is required",** while we try to register a patient without filling in any information.	**Preconditions**: The user is logged into the health system and is on the profile picture form of the "Register Patient" submodule. **Steps**: 1. Click on the "Cancel" button to exit the photo upload interface. 2. Click on the "Basic Information" page link to access the form. 3. Without entering any data, directly click the "Register Patient" button.	Verify that the error message for the "Phone number" field is displayed as **"Primary Phone is required"**
6	To ensure that data can be accurately entered into the 'Basic Information' form fields from an Excel source and that these entries are correctly displayed.	**Preconditions**: The user is logged into the healthApp system and is on the 'Basic Information' form of the 'Register Patient' submodule **Steps (Use Expected_Data.xlsx to fetch the required values):** 1. Fill in the 'First Name', 'Middle Name', 'Last Name', 'Age', and 'Phone Number' textbox.	Verify that the value for 'First Name' is displayed as provided in the Expected_Data.xlsx

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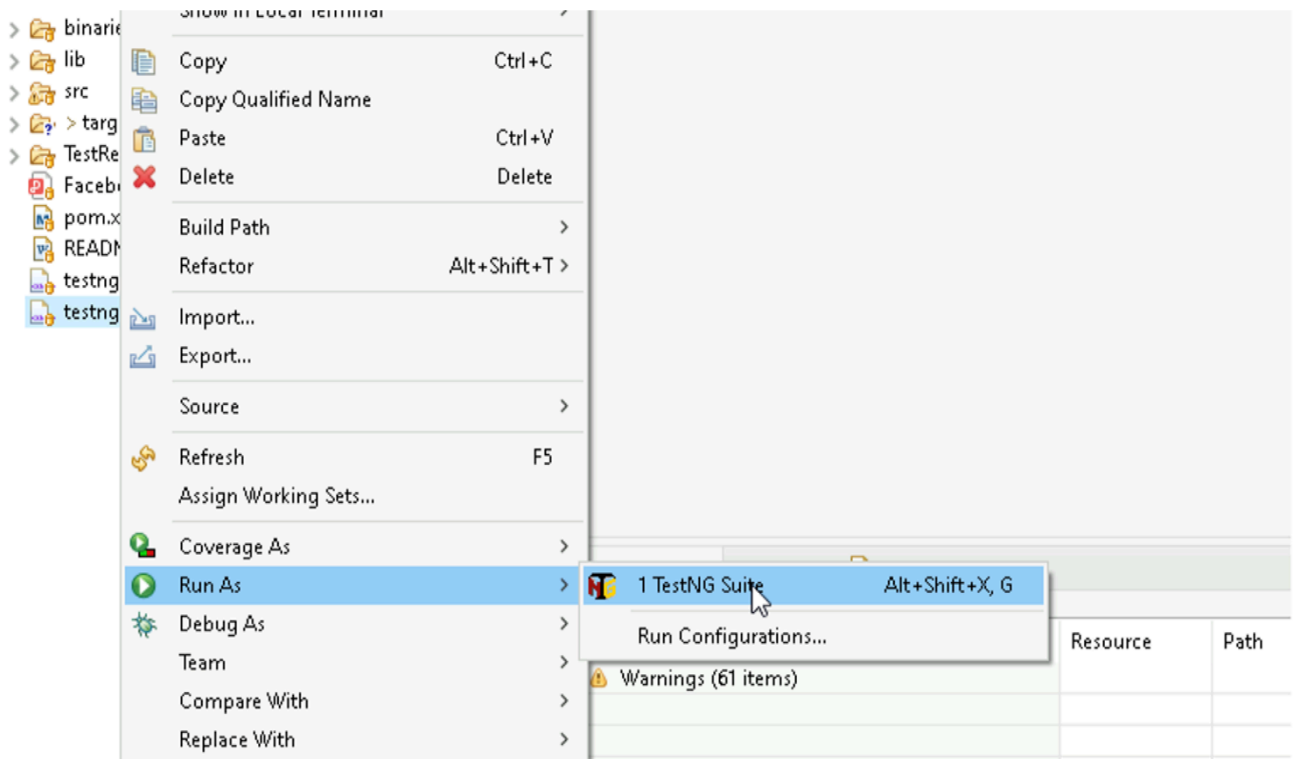2) Learners should not use any tools to create the xpath. They should develop the xpath/cssselector on their own.

IMPLEMENTATION/FUNCTIONAL REQUIREMENT
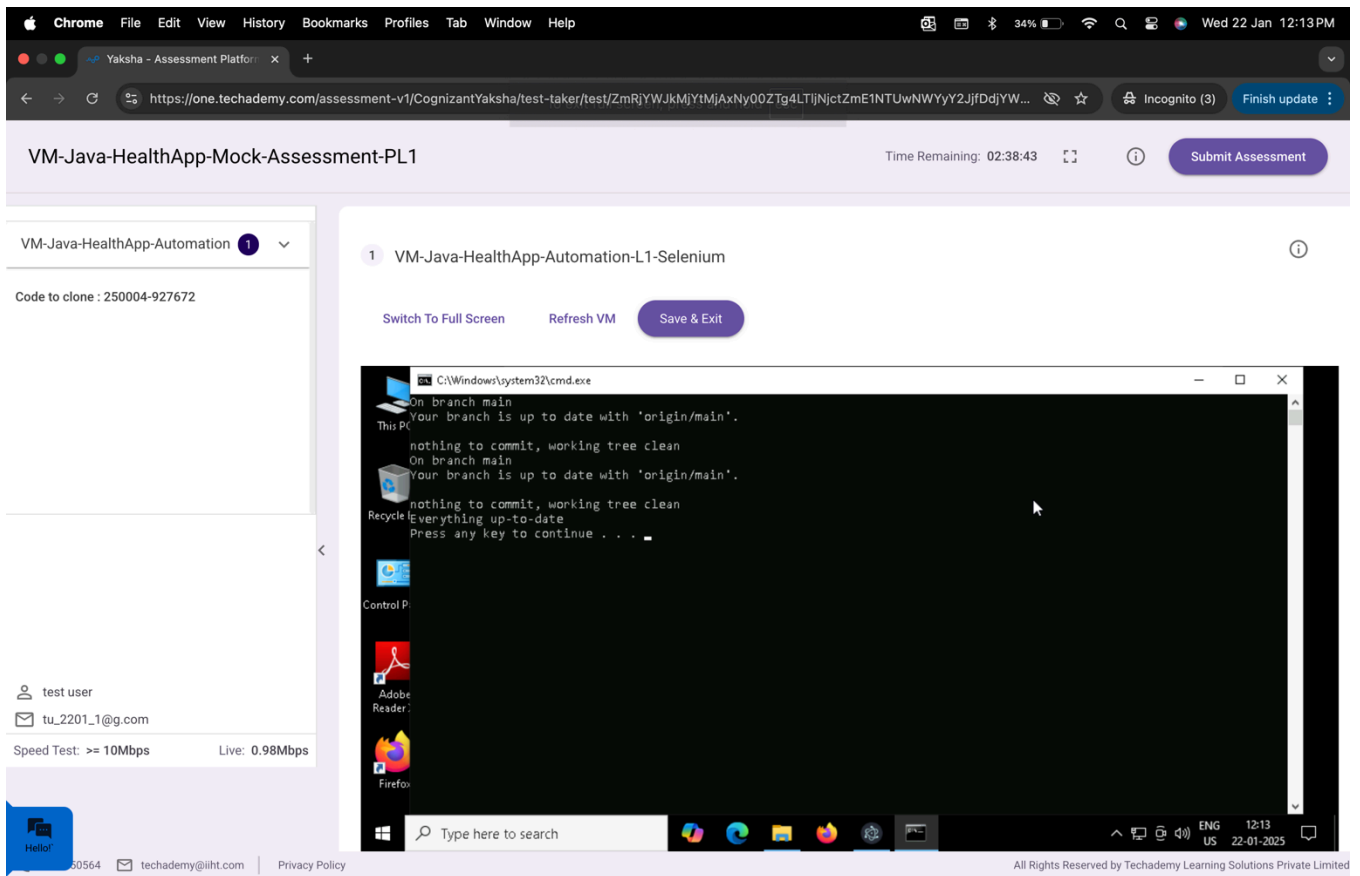
1.1 CODE QUALITY/OPTIMIZATIONS

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 TestNGSuite.**

3. To do the final submission of the assessment :
   a. Press escape to come out of Fullscreen mode.
   b. Submit the assessment.

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

===

All the Best