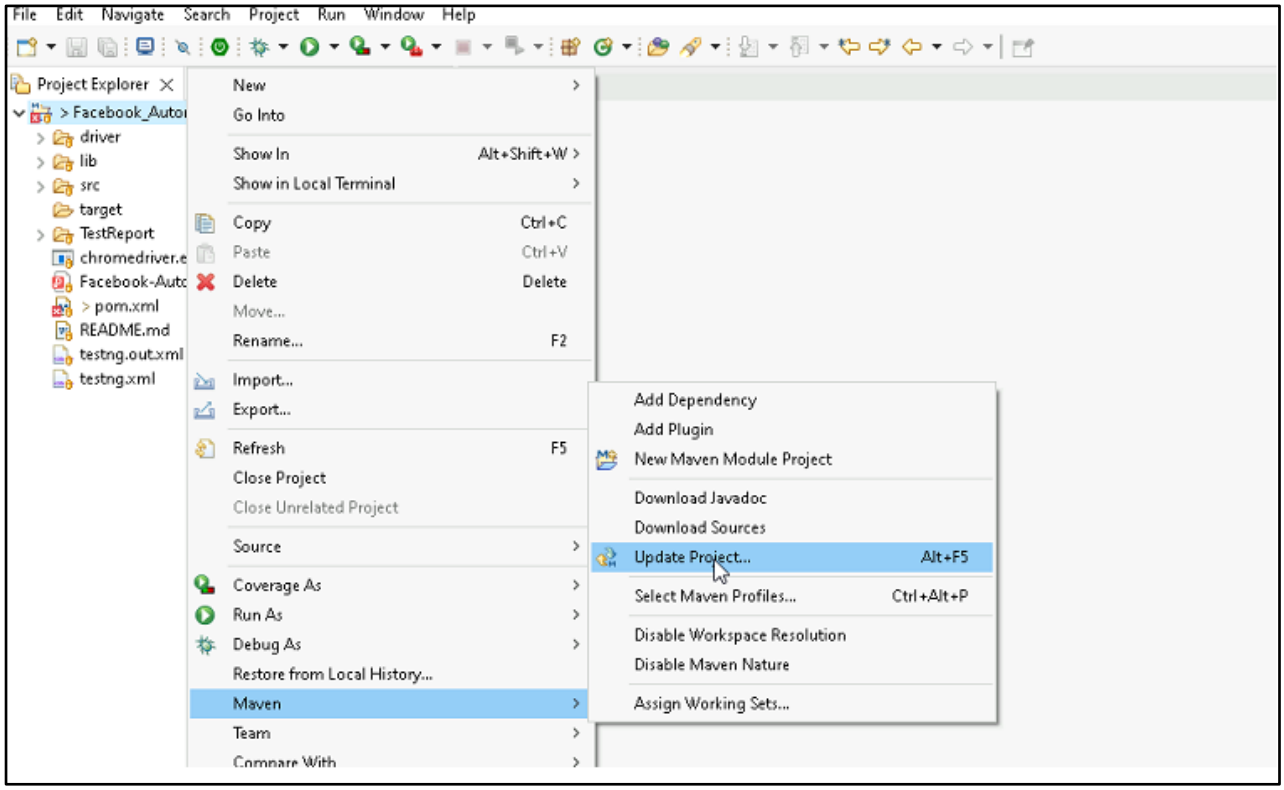
HealthApp Automation SubStore Module-PL1

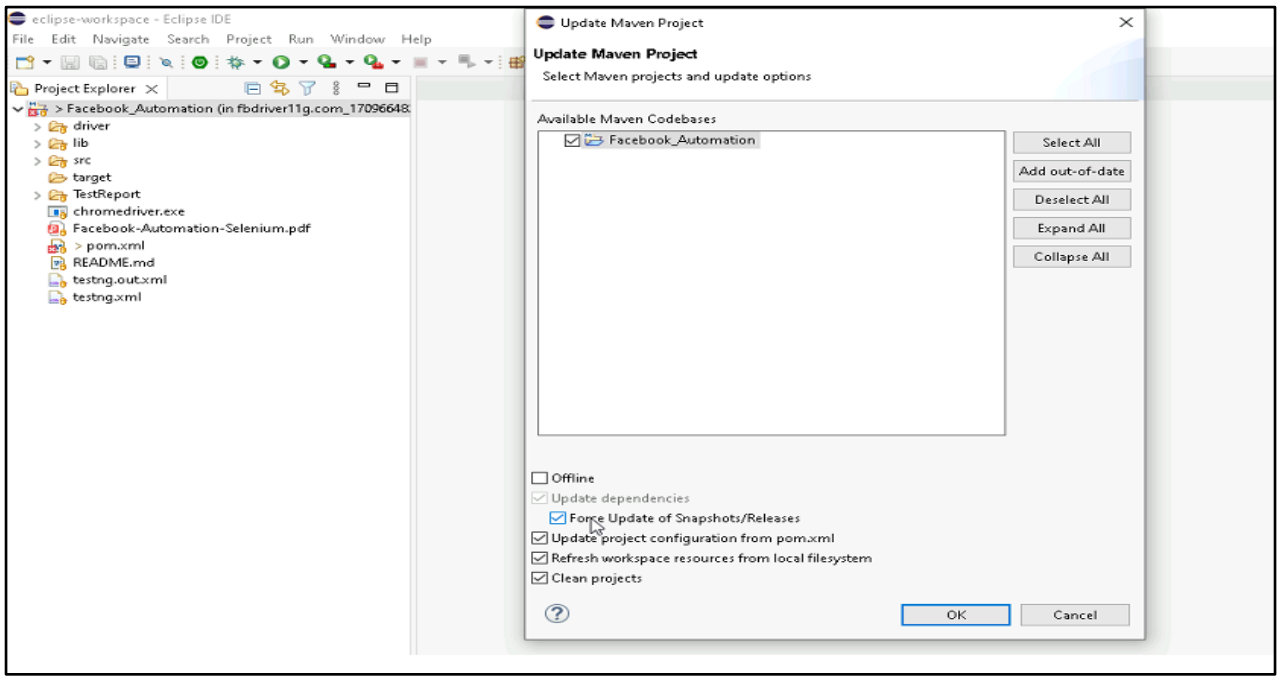
Pre-requisite:

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 not downloaded properly:

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



2. In 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. The files there are non-editable. Editing those files and trying to save them will throw errors and affect your evaluation.

Package	Class/File	Description
/src/main/java/pages	substore_page.java	1. All core activities (mentioned in list above) 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	URL to navigate to. Already URL is 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 method declared here. 3. You implement them as per your need. 4. You can add any additional method for common activity here
	Testng.xml	Execution needs to 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	Verify the SubStore module is present or not	1. go to URL : https://healthapp.yaksha.com/ 2. login as valid credential (username : admin , password : pass123) and click on "Sign in" Button 3. Scroll down menu till SubStore 4. Click on the SubStore	SubStore module should be present
2	To ensure that the "Select Your Substore" heading is displayed and all expected sub-module cards/tiles are also displayed.	Pre-condition: User should be logged in **Steps:** 1. Locate and click on the 'Substore' module link or button in the main navigation menu.	Upon clicking the 'Substore' module, the "Select Your Substore" heading should appear with the correct heading. All specified sub-module cards/tiles should be present and displayed correctly.
3	Ensure that the tooltip text on the substore switch button accurately displays the correct information when hovered over in the "Account" substore.	**Preconditions**:The user must be logged in to the application.The user is already on the "Substore" module page. **Steps**: 1. Click on the “Account” option within the substore module. This should take you to the Account substore section. 2. Move the cursor to hover over the substore switch button. 3. Observe the tooltip that appears when hovering over the substore switch button. 4. Verify that the tooltip text contains the following message: “You are currently in Accounts sub store. To change, you can always click here.”	Verify text on hover contains “You are currently in Accounts sub store. To change, you can always click here.”
4	Ensure that all expected sub-modules are displayed correctly.	**reconditions**: User must be logged into the HealthApp application. User is already on the SubStore module. **Test Steps**: 1. **Select 'Inventory' Sub-Module**: After selecting 'Account', click on the 'Inventory' sub-module within the same modal. 2. **Select 'Pharmacy' Sub-Module**: Finally, click on the 'Pharmacy' sub-module. Verify that clicking it leads to the Pharmacy interface.	All sub-modules should be displayed correctly. Expected Sub modules are : Pharmacy, Inventory
5	To verify that all sub-modules under Inventory module are present and visible in the user interface.	**Preconditions**: The user must be logged into the system. The user should be on the ”Inventory” submodule of "SubStore" module page. **Test Steps**: 1. Observe the sub-modules that appear under this module. 2. Count the number of sub-modules that are visible and take note of their names. 3. Verify each sub-module by checking if it is displayed on the screen. 4. Ensure each sub-module is not only present but also clearly visible.	All section should be displayed correctly. Expected Sub modules are : Stock, Inventory Requisition, Consumption, Reports, Patient Consumption, Return

6	To manually verify that navigation between different submodules within the "Inventory" module updates the URL correctly, reflecting the content of the newly navigated submodule.	**Preconditions**: User must be logged into the application. User must be on the "Inventory" module and its respective sub-module. **Test Steps**: 1. Navigate to the 'Inventory' Submodule. 2. Navigate to 'Stock' Submodule and Verify URL. 3. Navigate to 'Inventory Requisition' Submodule and Verify URL. 4. Move to 'Consumption' Submodule and Verify URL. 5. Proceed to 'Reports' Submodule and Verify URL. 6. Navigate to 'Patient Consumption' Submodule and Verify URL. 7. Go to 'Return' Submodule and Verify URL. 8. Return to 'Stock' Submodule.	Each click should lead to the correct submodule, and the URL should update accordingly to reflect the navigation accurately.
7	To take and save a screenshot of the Inventory section under the Substore module, verifying that the screenshot captures the current state of the user interface and is saved in the "Screenshots" directory.	**Preconditions**: User must be logged into the application. User must be on the "Inventory" module and its respective sub-module. **Test Steps**: 1. Once the Inventory section is visible, use the operating system's screenshot functionality or an in-application feature designed for screenshots, if available. 2. Save the captured image into the designated screenshot folder.	The screenshot should be saved successfully in the specified screenshot folder.
8	To confirm that all specified user interface elements are present and correctly displayed in the "Inventory Requisition" section of the Inventory submodule.	**Preconditions**: User must be logged into the application. User must be on the "Inventory" module and its respective sub-module. **Test Steps**: 1. Within the Inventory submodule, locate and click on the "Inventory Requisition" section or tab. 2. **Verify Buttons**: Check for the presence of the following buttons and ensure they are correctly labeled: First, Previous, Next, Last, Create Requisition, Ok, Print, View, Receive items 3. **Verify Fields**: Locate the search bar and confirm it is visible. 4. **Verify Drop Downs**:Confirm the presence of the following drop-down menus: Filter by Store, Date range, "..." (3 dots). 5. **Verify Radio Buttons**: Check for the presence of radio buttons: All, Pending, Completed, Cancelled, Withdrawn 6. **Verify Date Pickers**: Confirm that there are two date pickers labeled as "From" and "To". 7. **Verify Tooltip**: Hover over any elements with a star figure (usually indicative of importance or more information) to display a tooltip.	All buttons should be present and functional, with correct labeling. The search bar should be accessible for input. Drop-down menus should expand with the correct options listed. Radio buttons should be selectable with clear labels indicating their purpose. Date pickers should allow for date selection without any issues. Tooltips should display appropriate contextual information when hovered over.

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

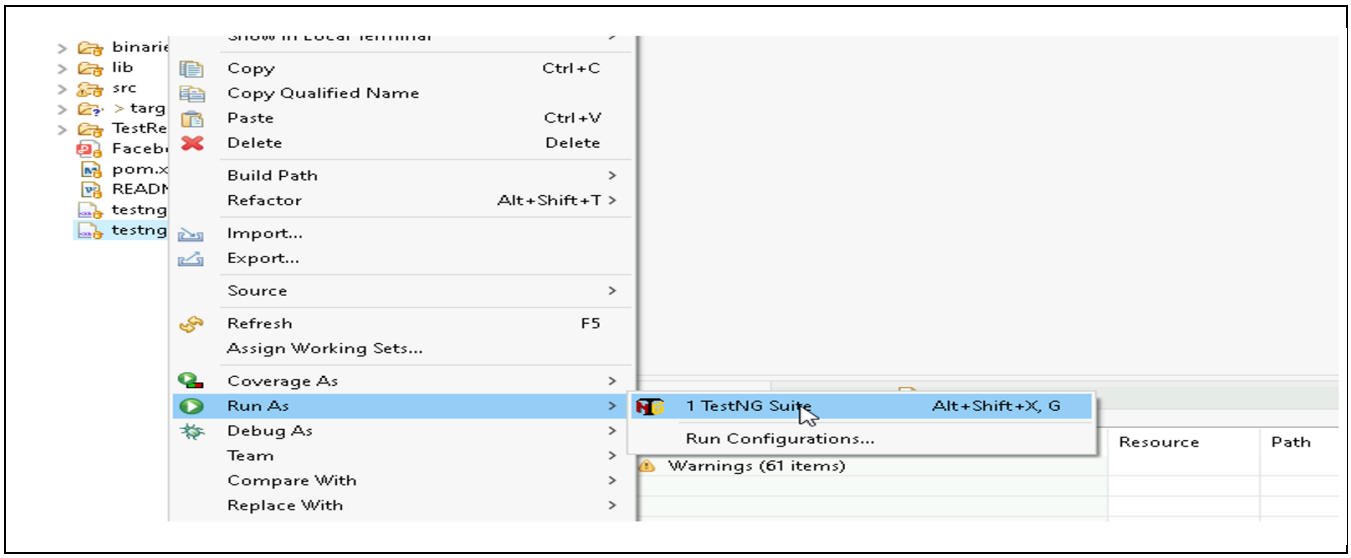
1.1 CODE QUALITY/OPTIMIZATIONS

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

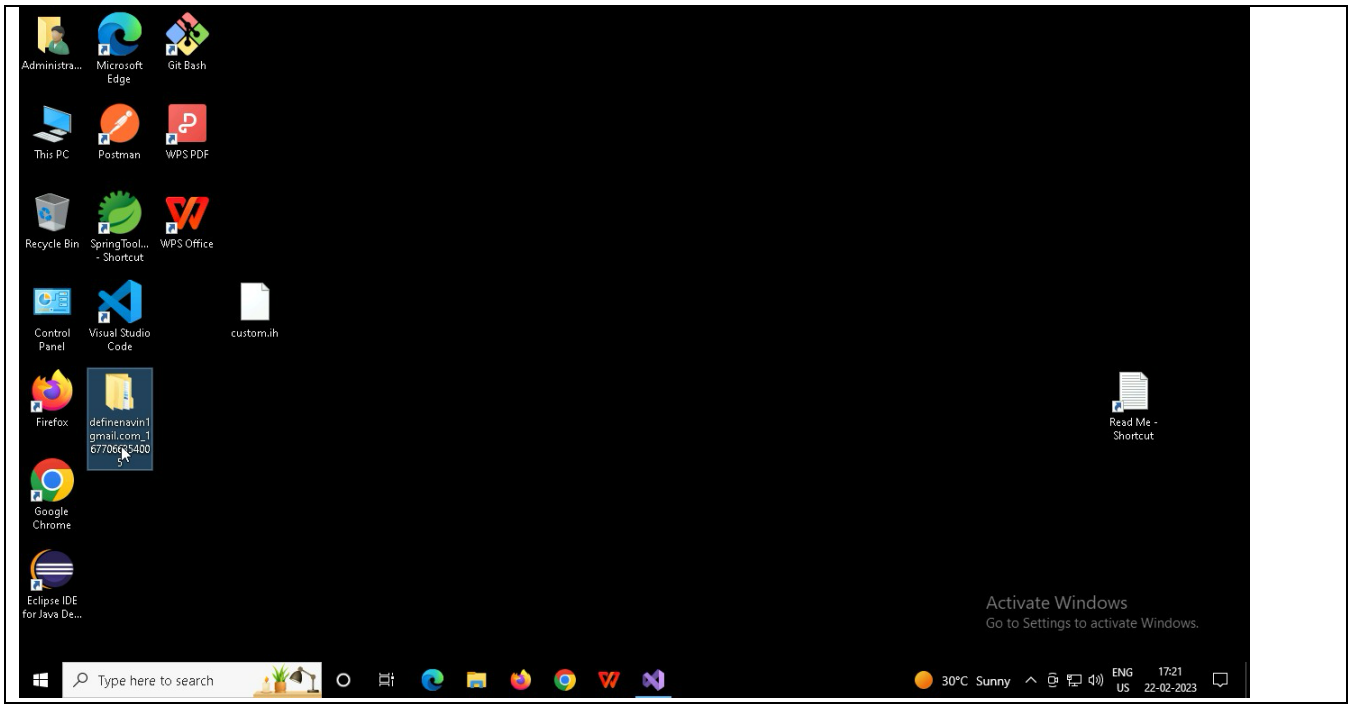
EXECUTION STEPS TO FOLLOW:

You are required to run test cases for applications before final submission, without which project evaluation will not happen.

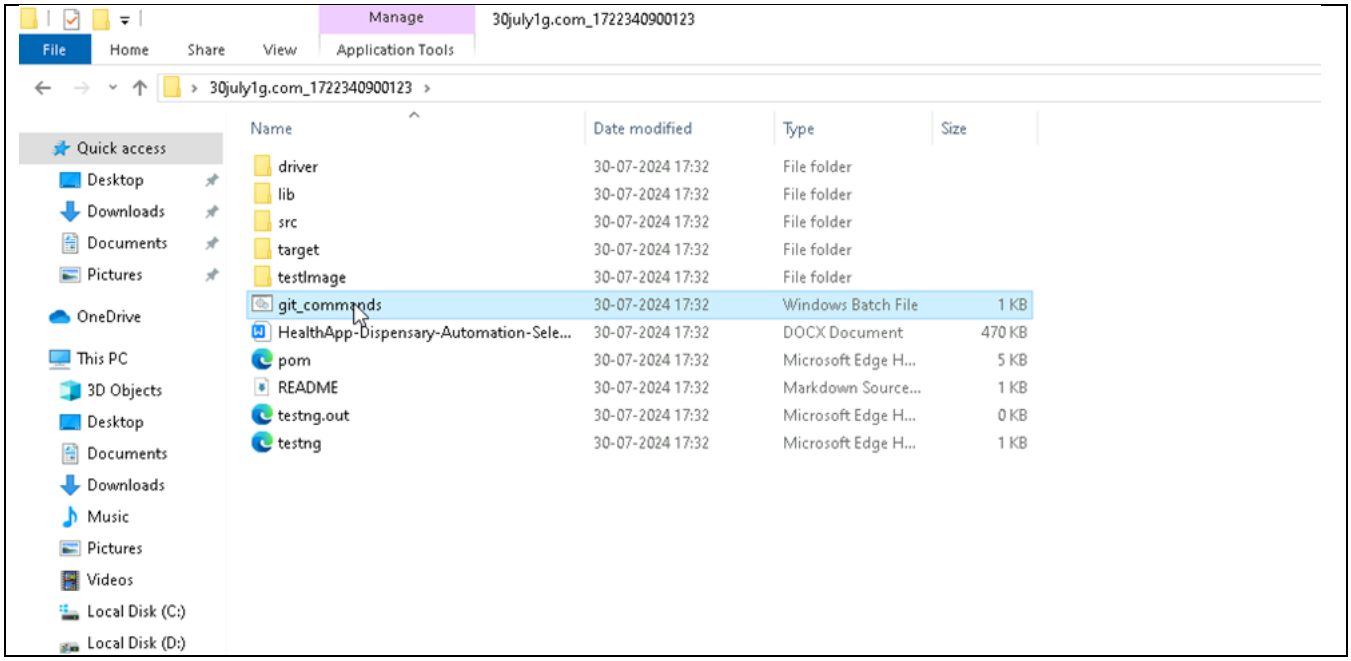
1. **You can launch test cases any time as follows: Right-click on testng.xml and run TestNGSuite**



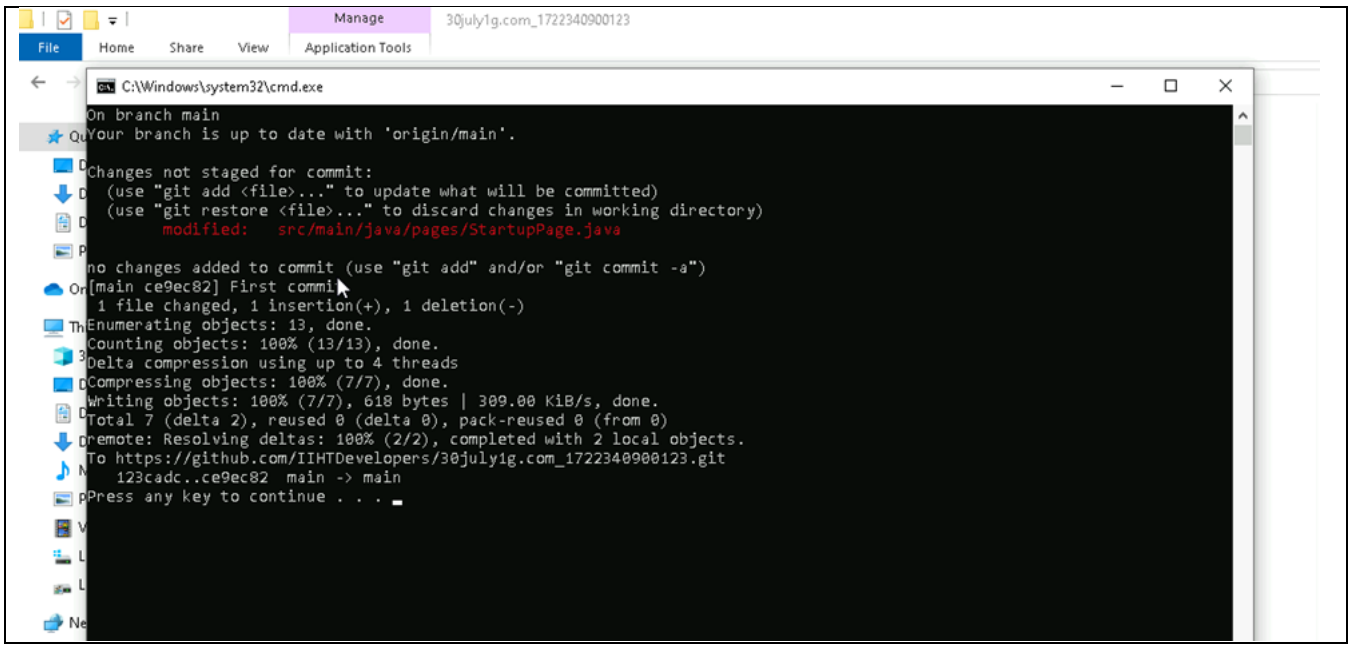
2. **Before final submission, you are also required to push your code to GIT. Following are the steps to follow:**



In your project folder, you will find a batch file named git_commands

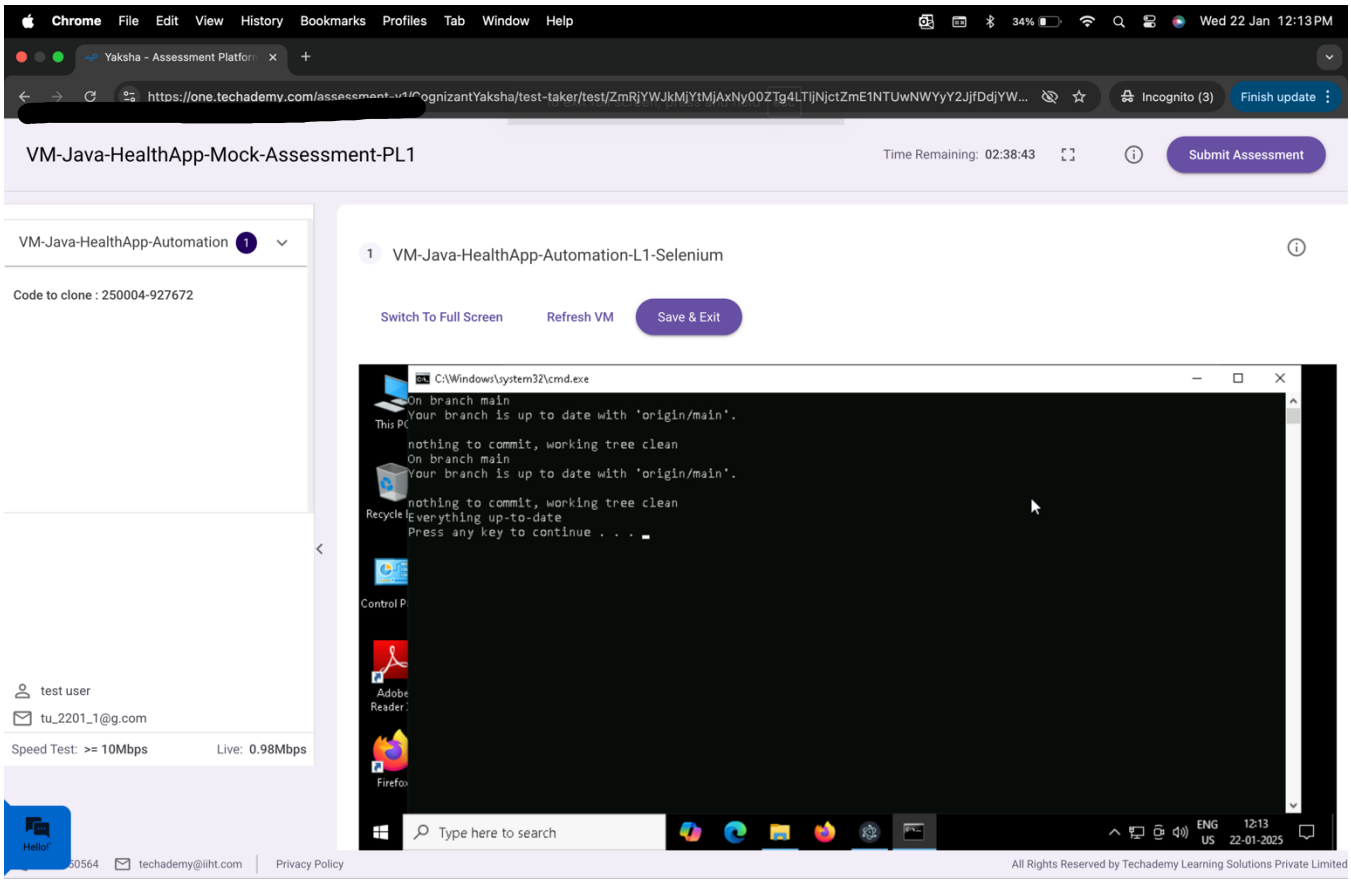


Double-click the batch file to run it. It will run the commands to push your code to GIT.



Once the code is pushed to git, you can go for the final submission of the assessment.

- Press escape to come out of Fullscreen mode.
- Submit the assessment.



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

==========

All the Best