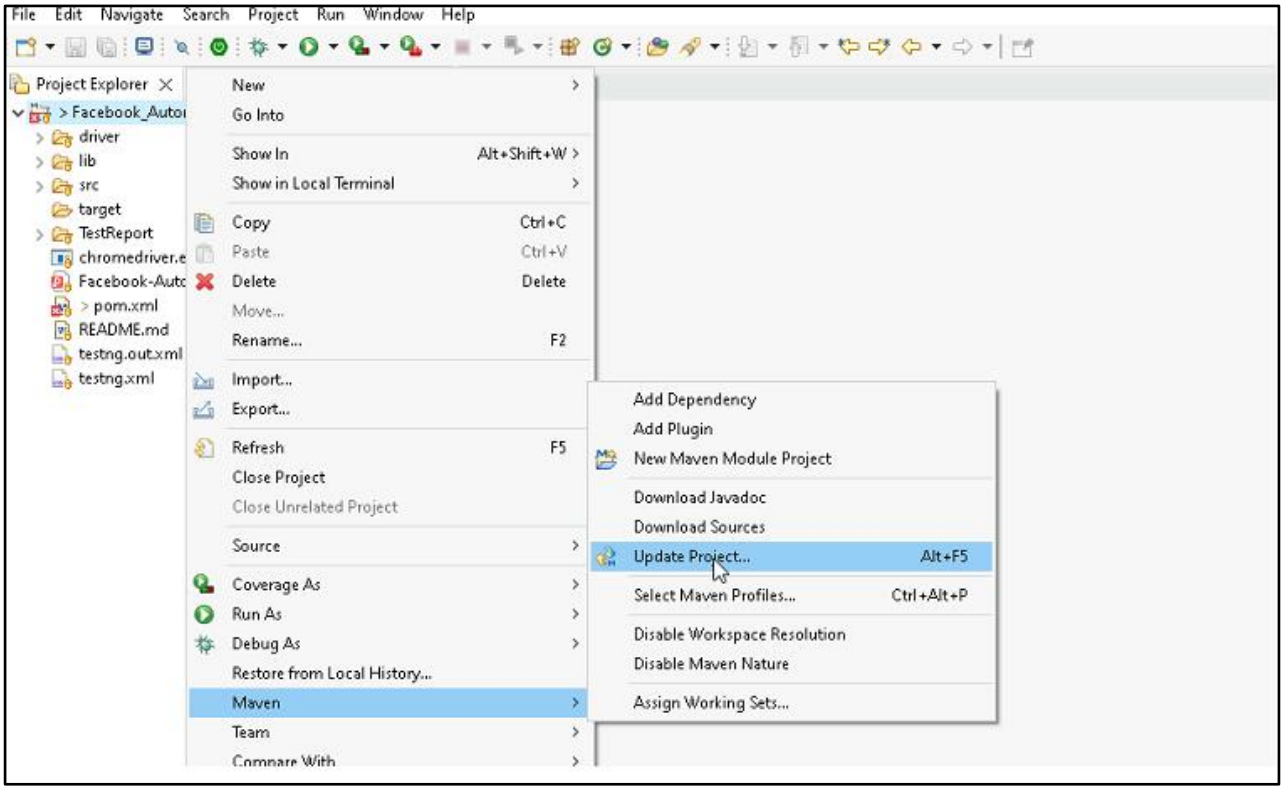
HealthApp Automation Substore Module

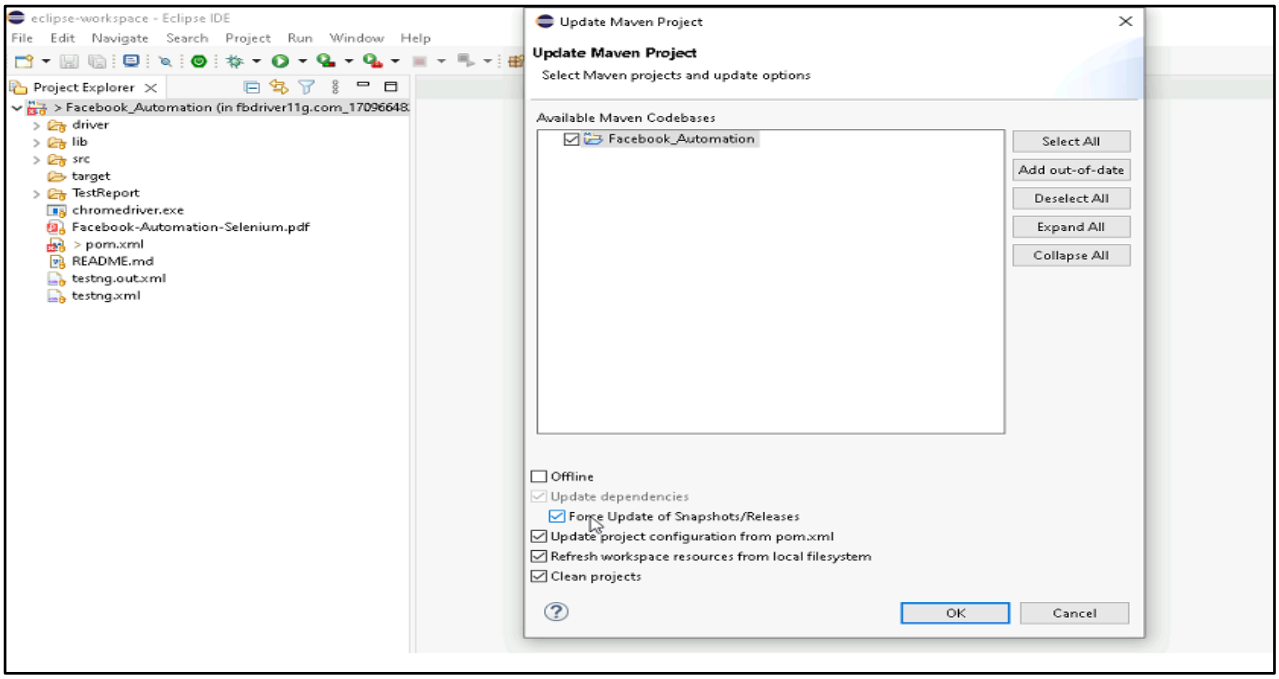
Pre-requisite:

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:

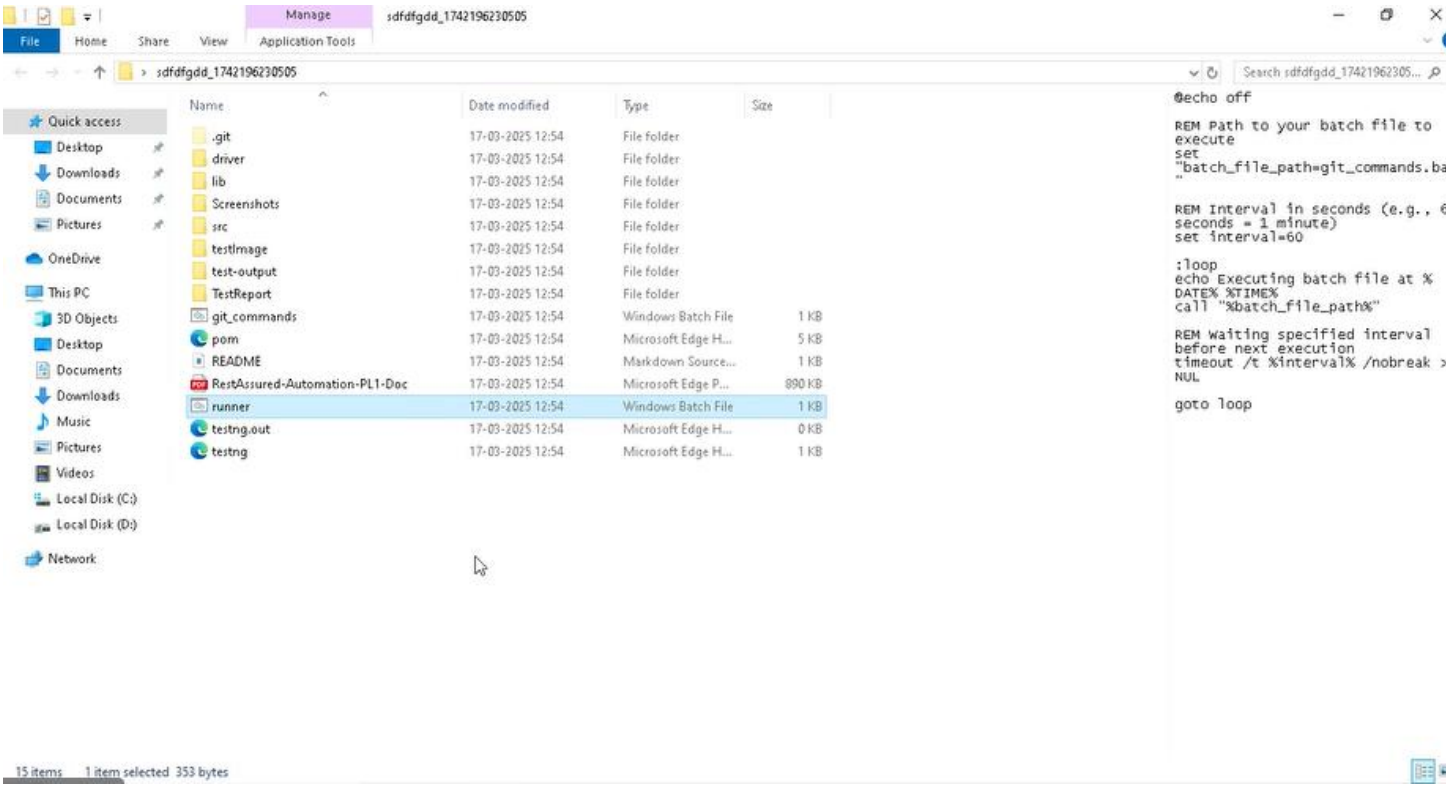
1. Right-click on the project: Go to “Maven” and select “Update Project”

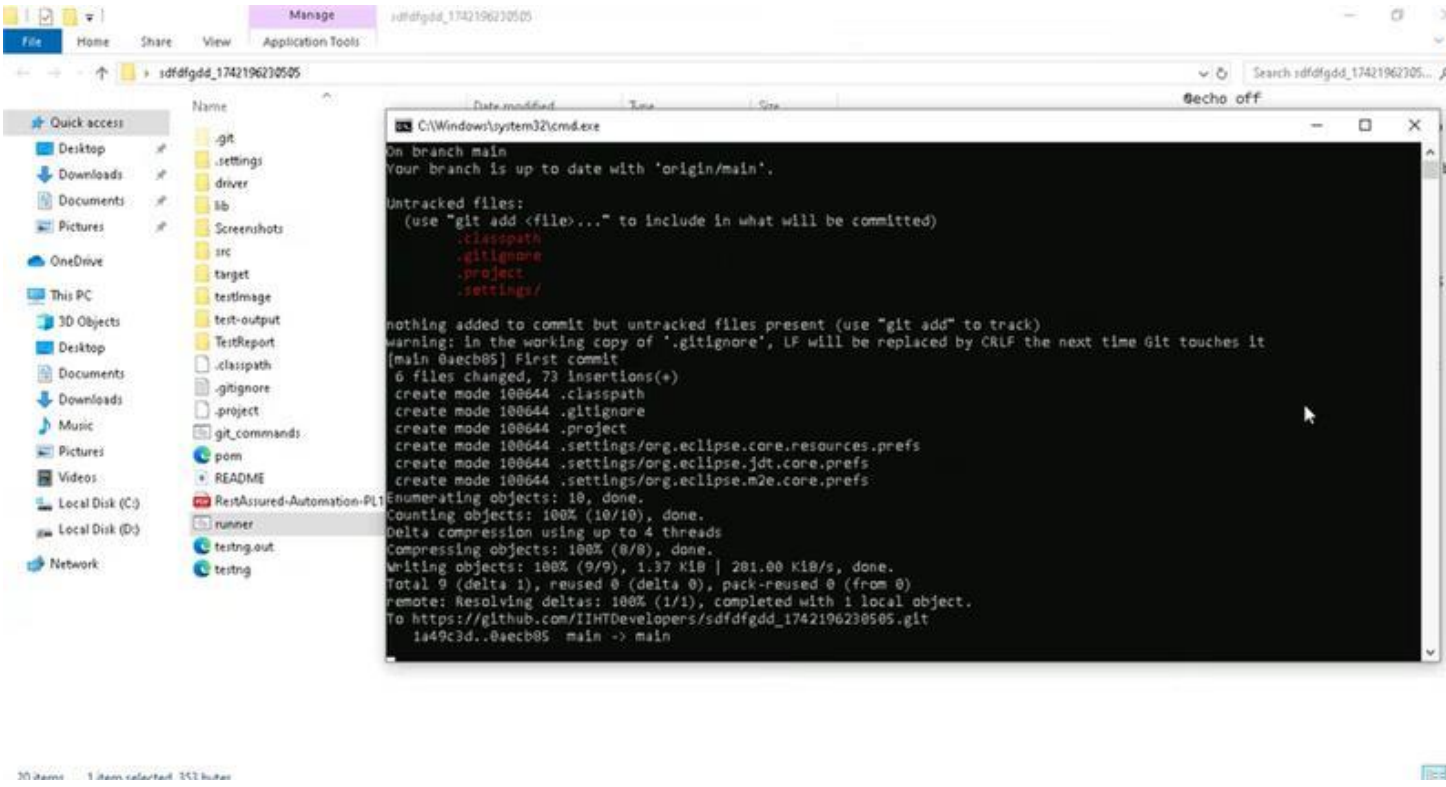


2. In Update Maven Project Box Select “Force Update of Snapshots/Releases” and click OK



** Before starting any implementation. Please execute this “runner” file present in the cloned folder. This will keep pushing the changes at regular intervals.





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/coreUtilities/utils/	FileOperations.java	1. Contains methods to read from excel file. 2. Method is in templated form. 3. **You will be required to implement these methods as very first activity, because even URL to navigate to, is read using these methods.**
/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 "SignIn" Button 3. Scroll down menu till SubStore 4. Click on the SubStore	SubStore module should be present
2	Verify the sub-module buttons present in Select your Substore modal	Pre-condition: User should be logged in 1. Click on the substore module	"Select your Substore" heading should be present on the page. Expected sub-module Card/Tiles must be present.
3	Verify that the "Accounts' sub-module is clickable	Pre-condition: User must be logged in and on the substore module 1. Click on the “Account” substore 2. Hover on substore switch button(Arrow icon) located at top left of page and get tooltip text	Verify text on hover contains "To change, you can always click here."
4	Verify all sub-modules are displayed correctly after Clicking on the "SubStore " Module.	Pre-condition: User should be logged in and it is on SubStore module 1. Login in the healthapp application 2. Click on the Substore 3. "Select your Substore" pop up 4. Click on "Account" sub-modal 5. Click on the "Inventory" sub-module 6. Click on" Pharmacy" sub-module	All sub-modules should be displayed correctly. Expected Sub modules are : Pharmacy, Inventory
5	Verify all section are displayed correctly after Clicking on the "Inventory " Sub-Module.	Pre-condition: User should be logged in and it is on SubStore module 1. Click on the SubStore Module drop-down arrow 2. Click on "Accounts" from the "Select your Subdtore" 3. Click on the "Inventory" 4. Navigate to "Stock" section 5. Click on the all the section of the "inventroy" sub-module 6. Navigate back to "Stock" section	All section should be displayed correctly. Expected Sub modules are : Stock, Inventory Requisition, Consumption, Reports, Patient Consumption, Return
6	Verify to navigate to each sections which are present in the "Inventory" sub-module	Pre-condition: User should be logged in and it is on Inventory sub-module 1. Navigate to the "Inventory" sub-module 2. Click on the "Inventory Requisition" section 3. Click on the "Stock" 4. Click on the "Consumption" 5. Click on the " Reports" 6. Click on the "Patient Consumption" 7. Click on the "Return" 8. Navigate back to the "Inventory Requisition" section	Ensure that it should navigate to each sections of the "Inventory" module

7	Take Screenshot of the current page	Under Substore module > Inventory Section Take the screenshot of the current page	Screenshot shot of the page should be saveed under screenshot folder
8	Verify the presence of Inventory Requisition section in Inventory sub-module with all fields	Pre-condition: User should be logged in and it is on SubStore module 1. Click on the SubStore module drop-down arrow 2. Click on Inventory sub-module 3. Click on Inventory Requisition section	Inventory Requisition section in Inventory sub-module should be present Expected value that are present in Inventory Requisition section are 1. Buttons: First, Previous, Next, Last, Create Requisition, Ok, Print, View. Receive items 2. Fields: search bar 3. Drop down: Filter by Store, Date range, "..."(3 dots) 4. Radio-button: All, Pending, Completed, Cancelled, Withdrawn 5. Date Picker: From, To 6. Tooltip: Star figure
9	Creating and Verifying "Create Requisition" button	Pre-condition: User should be logged in and it is on Inventory sub-module 1. Click on "Inventory Requisition" section 2. Click on "Create Requisition" button 3. Click on "Target Inventory " field and Select "GENERAL-INVENTORY" option 4. Click on "Item Category" drop down and select " Consumables" option 5. Enter "tissue" in ItemName field 6. Enter the "Required Quantity" field 7. Click on "Request" button 8. Click on "Close" icon	The requisition should be successfully created and saved. It should then be visible in the list of Inventory Requisition page with the details entered. after Clicking on the "Request" button this successful message should pop up "success Requisition is Generated and Saved"
10	Retrieve Comments for a Post and validate the response in method: getCommentsForPost(int postId)	1. Call the GET https://jsonplaceholder.typicode.com /posts/{postId}/comments endpoint 2. Pass the postId as a parameter 3. Check if the response body contains an array of comments 4. Validate each comment has a postId, id, name, email, and body properties	Returns a 200 OK status with a list of comments related to the specified post. Response body contains an array of comments with the specified properties
11	Update an existing post and validate the update in method: updatePost(int id, String newTitle, String newBody, int userId)	1. Call the PUT https://jsonplaceholder.typicode.com /posts/{id} endpoint. 2. Provide newTitle, newBody, and userId in the request body. 3. Validate the response contains the updated title, body, and userId fields.	Returns a 200 OK status code with the updated post. Response body contains the updated title, body, userId, and id.

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

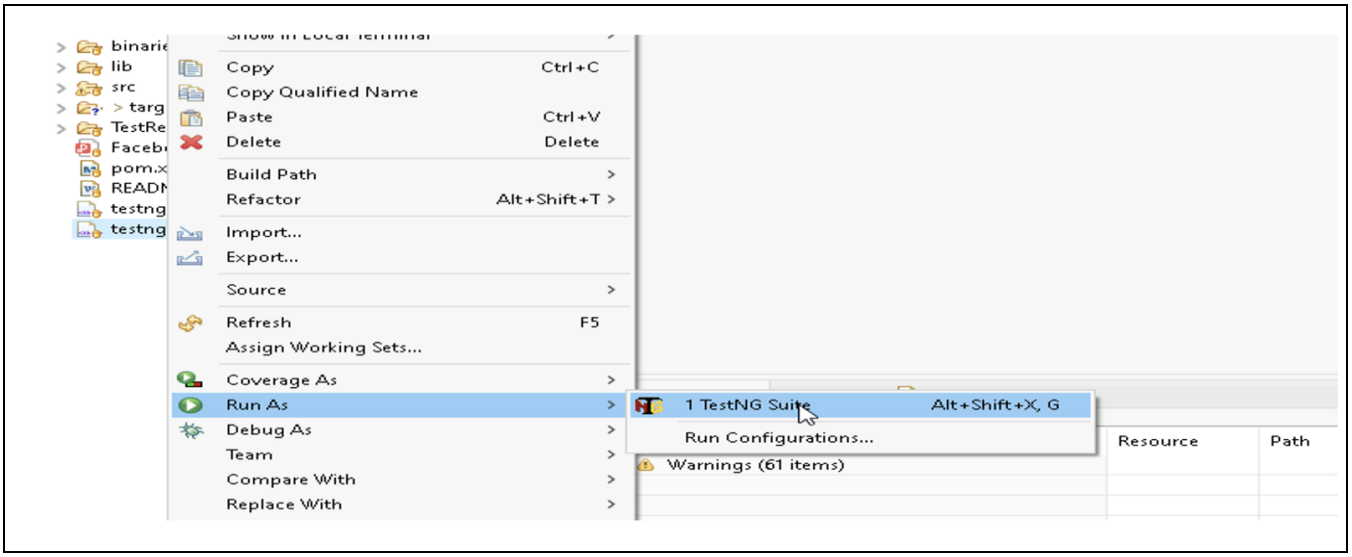
1.1 CODE QUALITY/OPTIMIZATIONS

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

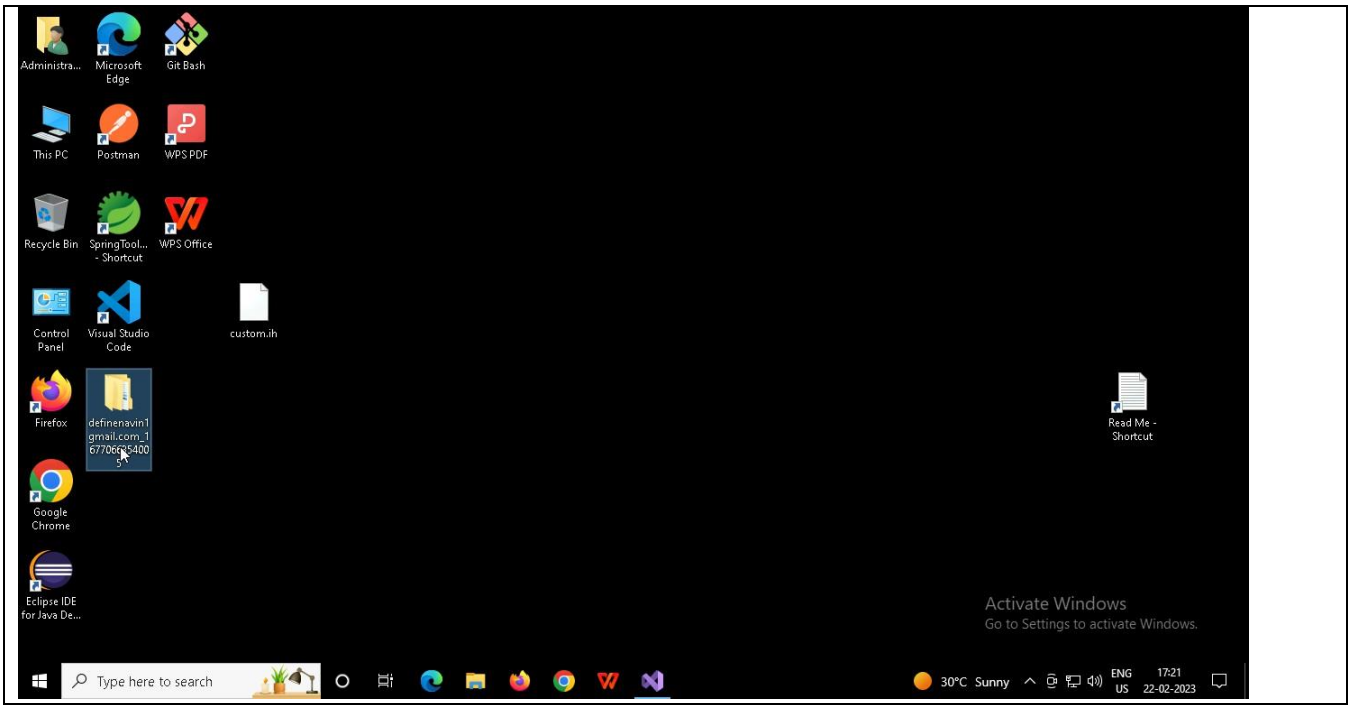
EXECUTION STEPS TO FOLLOW:

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

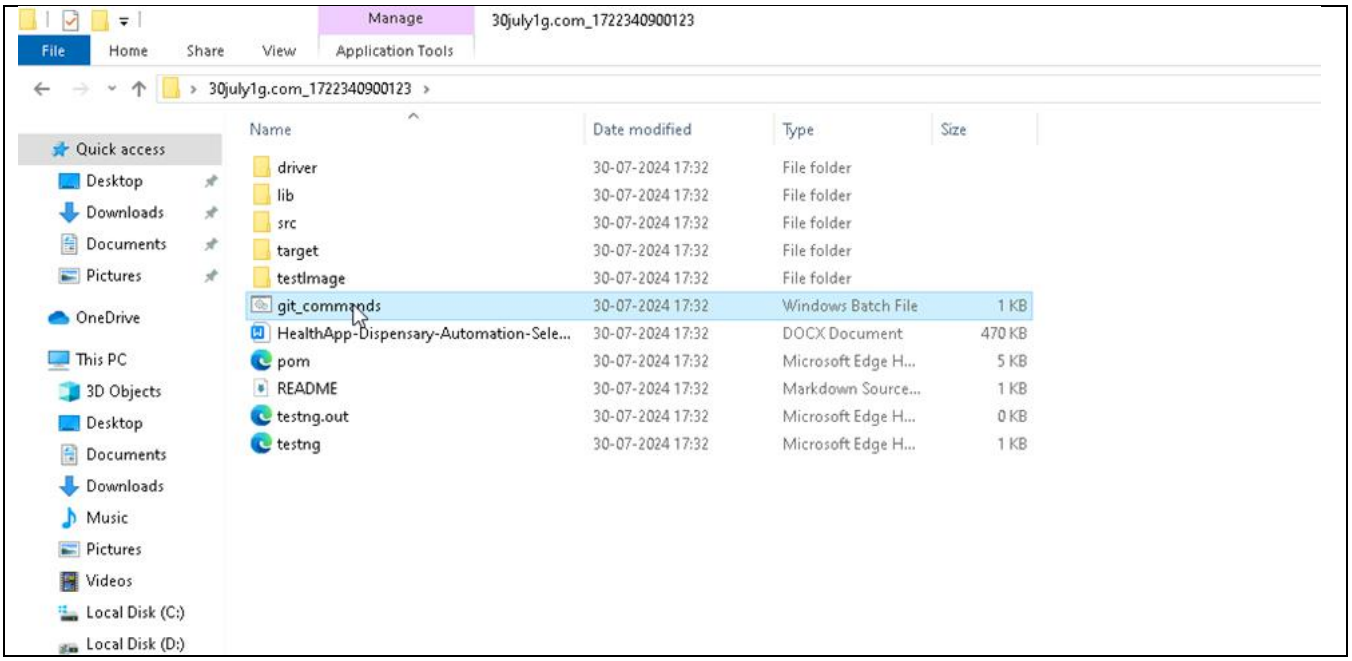
1. **You can launch test cases any time as follows: Right-click on testng.xml and run TestNGSuite**



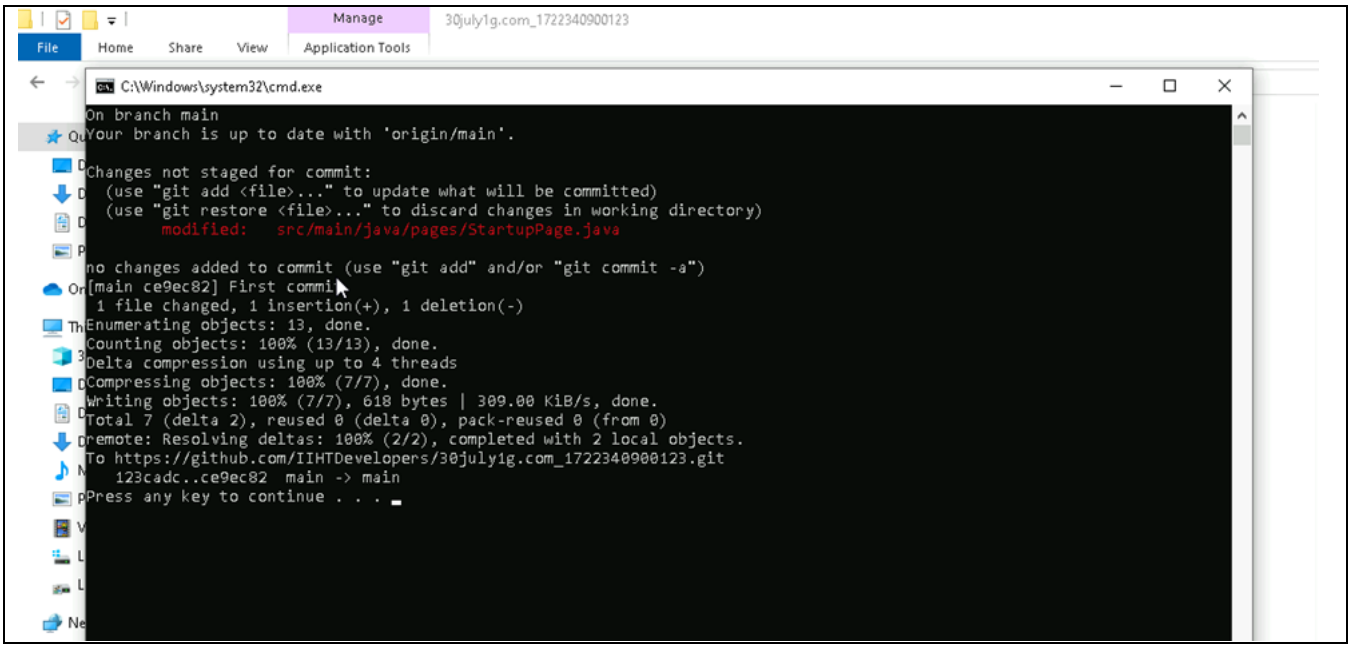
2. **Before final submission, you are also required to push your code to GIT. Following are the steps to follow:**



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.


Manage
30july1g.com_1722340900123
File
Home
Share
View
Application Tools
C:\Windows\system32\cmd.exe
On branch main
Your branch is up to date with 'origin/main'.
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working directory)
modified: src/main/java/pages/StartupPage.java
no changes added to commit (use "git add" and/or "git commit -a")
[main ce9ec82] First commi
1 file changed, 1 insertion(+), 1 deletion(-)
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 618 bytes | 309.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/IIHTDevelopers/30july1g.com_1722340900123.git
123cadc..ce9ec82 main -> main
Press any key to continue . . .

==========

All the Best