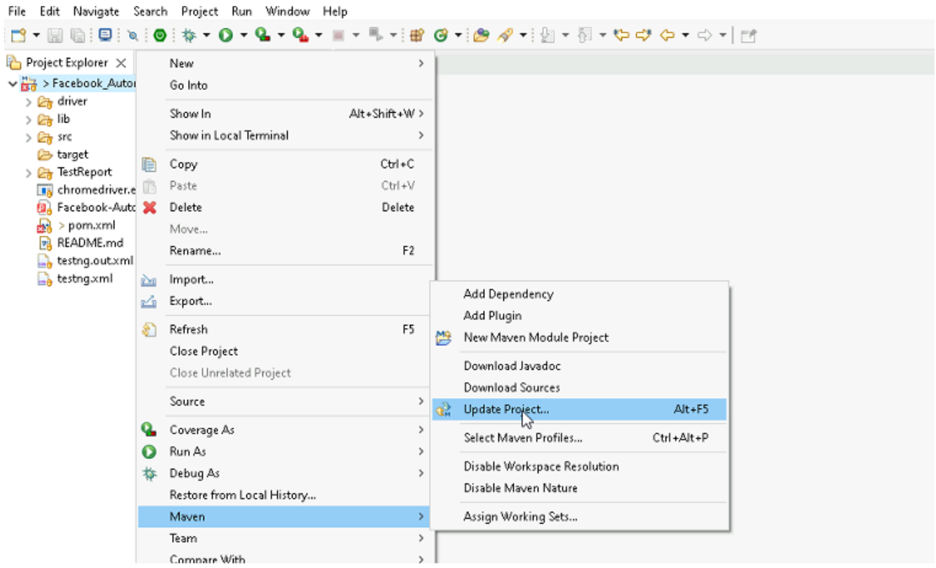
HEALTHAPP AUTOMATION-DISPENSARY MODULE – PL2

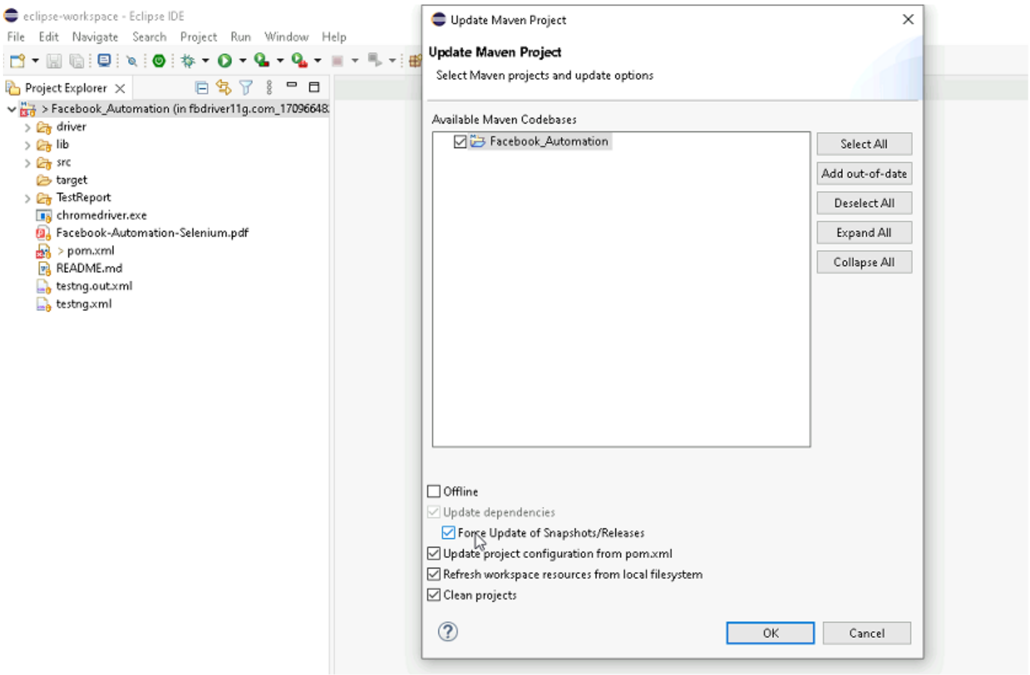
Pre-requisite:

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



Update Maven Project

Update Maven Project

Select Maven projects and update options

Available Maven Codebases

☑ Facebook_Automation

☐ Offline

☑ Update dependencies

☑ Force Update of Snapshots/Releases

☑ Update project configuration from pom.xml

☑ Refresh workspace resources from local filesystem

☑ Clean projects

Template Code Structure:

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

b. Other Files and packages you can ignore.

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

d. You are not required to work in “Test” Folder. Files there are non-editable. Editing those files and trying to save them will throw error and would 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	Dispensary_Pages.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:

S_No.	**Summary**	**Action**	**Expected Result**
1	Verify the title and URL of the current page.	1. Go to URL : https://healthapp.yaksha.com/ 2. Login as valid credential **(username : admin , password : pass123)** and click on "**Sign in**" Button 3. Get the title and URL of the Home page, post login 4. Validate the title and URL of the Home page	Title should be : **DanpheHealth** URL should be : **https://healthapp.yaksha.com/Home/Index#/**
2	Verify Presence of Dispensary Module.	1. Navigate to the Left Navigation Panel 2. Check for the Dispensary Module 3. Verify Presence of Module	dispensary Module should be present
3	Verify Presence of Sub-Modules under Dispensary Module.	1. Locate the "Dispensary" module in the left navigation panel and click the expand icon to view its contents. 2. Look for the following sub-modules within the "Dispensary" module: - Prescription - Sale - Stock - Counter - Reports - Patient Consumption 3. Ensure all listed sub-modules are present.	All mentioned sub-modules should be present under Dispensary Module.

4	Verify the presence of the following elements at the bottom of the "Sale" page: - Print Invoice Button - Discard Button - Invoice History - Credit Limits and Balances	**Precondition:** "**Counter**" sub module of "**Dispensary**" modules should be pre-selected. 1. Click on "Morning Counter" link text 2. It should navigate to the "Sale" page. 3. Scroll to the bottom of the page. 5. Verify the presence of the following elements at the bottom of the "Sale" page: - Print Invoice Button - Discard Button - Invoice History - Credit Limits and Balances	user should be able to scroll the bottom of the "**Sale**" page and able to see the following elements. - Print Invoice Button - Discard Button - Invoice History - Credit Limits and Balances
5	Perform the keyboard operation "Alt+N" to open the "**Add New Patient**" popup form and verify its title.	**Precondition:** "**Sale"** page must be pre-selected 1. Press **"Alt + N"** from the keyboard, 2. Then "A**dd New Patient**" form should popup. 3. Verify the title of the form is "**Add New Patient**" displayed over the top of the form.	**Add new patient** form should be open. Page title must be **"Add New Patient**".
6	Validate the error message shown by "First Name" textfield after clicking the "**Ok"** Button without filling any information in the form.	**Precondition:** Ensure "Add New Patient" Form is Open and ready for inputs. 1. Attempt to submit the form by clicking the "OK" button without entering any information in the form fields. 2. Check for an error message that should appear under the "First Name" textfield. 3. Confirm that the error message is appropriate (e.g., "First Name is required") and clearly visible under the "First Name" textfield.	Error message should be : **First Name is required.**
7	Fill all the text fields which are present inside the "Add New Patient" form. Check each field to ensure the information entered is accurate and correctly formatted.	**Precondition:** Ensure "Add New Patient" Form is Open and ready for inputs. 1. Enter Data in the First Name Field (Input a valid first name into the 'First Name' field.) 2. Enter Data in the Middle Name Field(Input a middle name (if applicable) into the 'Middle Name' field.) 3. Enter Data in the Last Name Field (Input a valid last name into the 'Last Name' field.) 4. Enter Data in the Age Field (Input the patient's age in the 'Age' field. Ensure the age is numerically valid.) 5. Enter Data in the Contact Number Field (Input a valid contact number into the 'Contact Number' field, following the format required (e.g., country code, area code)). This structured format ensures that all necessary personal details are correctly filled into the "Add New Patient" form.	Given text fields value inside the "Add New Patient" form are filled with appropriate information. The entered data should be correctly displayed in First Name, Middle Name, Last Name, Age and Contact Number Field on "**Add New Patient"** form.
8	On the "**New Consumption Entry**'s page, verify the confirmation Message that is "**Are you sure you want to Proceed ?**" .	1. Click on the cross "X" of Add New Patient popup. 2. Click on the "**Patient Consumption**" sub modules of Dispensary module. 3. Click on "**New Consumption**" button, then "New Consumption Entry" form will popup. 3. Directly click on "Save Consumption" button without fill any information. 4. Then a confirmation message will popup (**Message : "Are you sure you want to Proceed ?").** 5. Validate that confirm message. 6. Then click on confirm button. 7. Then close the "New Consumption Entry's page.	After the certain steps defined, Confirmation message displayed should be **"Are you sure you want to Proceed ?".**

9	on the "Stock" page , select "Main store" from "Filter by store" dropdown and ensure that the stock values on the "Stock" page are correctly filtered by the selected store ("Main Store")	1. Navigate to Stock Submodule: Open the "Stock" submodule. 2. Select Main Store: From the "Filter by store" dropdown, choose "Main store" to apply this filter. 3. Confirm Selection: Verify that "Main store" is actively selected in the dropdown to ensure the correct filter is applied. 4. Verify Filtered Stock List: Check the stock list to confirm it is now filtered by "Main store". Specifically, look at the store name of the last record or element in the list to confirm it reads "Main store", indicating the filter is functioning as expected.	After selecting "Main store" from the dropdown, the stock list should correctly display only items associated with "Main store," verifying the filter's effectiveness.
10	On the stock page, Under "**Requisition**" tab clicking the "Create Requisition" button should navigate you to the "Add Requisitions" page.	**Precondition Verification**: Begin with the "Stock" page already open and displayed. 1. **Navigate to Requisitions Tab**: Click on the "Requisitions" tab located within the "Stock" page interface. 2. **Check for Button Presence**: Validate whether the "Create Requisition" button is present under the "Requisitions" tab. It should be visible and accessible. 3. **Click on the "Create Requisition" Button**: Initiate the button to test its function. 4. **Verify Navigation**: Ensure that clicking the button navigates to the "Add Requisition" page. 5. **Confirm Page Name**: Once on the "Add Requisition" page, verify that the page is correctly named "Add Requisition" to confirm you've reached the intended destination.	Clicking the "Create Requisition" button is located under the "Requisitions" tab on the "Stock" page. Verify that clicking it navigates to the "Add Requisitions" page, where the page name should correctly display as "Add Requisitions."
11	On the ""New SSU Patient Registration" under social service module, get the place holder name of "**Address**" textfiled of "**New SSU Patient Registration**" page and verify the place holder name.	**Precondition:** Navigate to Social Service Module 1. Click on the "Social Service" module from the main interface. 2. Click the "Register New SSU Patient" button to open the "New SSU Patient Registration" page. 3. On the "New SSU Patient Registration" page, click on the "Address" textfield to set focus on it. 4. Identify and retrieve the placeholder name of the "Address" text field. 5. Confirm that the placeholder text is as expected. Print this name to the console to ensure it is correctly captured and displayed.	Upon clicking on the "Address" textfield in the "New SSU Patient Registration" page, the placeholder name should be correctly retrieved and displayed, verifying that it matches the expected value. This placeholder name should then be printed to the console as confirmation.
12	Close this "**New SSU Patient Registration**" form page by using JavaScript.	**Precondition**: "**New SSU Patient Registration**" form page must be open. 1. Click on the "**X**" button using JavaScript	"**New SSU Patient Registration**" form page should be closed.
13	Click on the "Appointment" module, navigate to the "New Visit" page via the "New 1" link, hover over the keyboard icon, and verify the tooltip message's accuracy.	**Preconditions:** User must be logged into the system. 1. **Navigate to** the "Appointment" module from the main interface. 2. Click on the "New 1" link, which should redirect you to the "New Visit" page. 3. Move your mouse over to the keyboard	Tooltip message should be : **ALT + N = New Patient**

		icon present on the page. 4. **Verify Tooltip Message**: As you hover over the icon, a tooltip should appear. Verify the message within this tooltip.	
14	On the Dispensary module's "Sale" page, Open the "New Patient page" and select **India** from the **country** dropdown menu and validate India is properly selected or not.	**Preconditions:** User must be logged into the system with the Dispensary module already expanded. 1. Navigate to the "Dispensary" module on the main interface. 2. Click on the "Sales" submodule under the Dispensary module. 3. Press the "Alt+N" keyboard shortcut to activate the "Add New Patient" form. 4. On the "Add New Patient" page, locate and click on the country dropdown. 5. Select "India" from the dropdown options.	After executing the steps, the "Add New Patient" form should be displayed with "India" selected as the country, confirming that the interface responds correctly to user inputs.
15	Verify that a user can take a screenshot of the "Sale" page in the Dispensary module after closing the "Add New Patient" form.	**Preconditions**: User must be logged into the system. "Add New Patient" form is open on the "Sale" page of the Dispensary module. 1. Close the "Add New Patient" form by clicking the designated close button or using the relevant shortcut/command. 2. After the form is closed, ensure the focus is back on the "Sale" page. 3. Take a screenshot of the current state of the "Sale" page. 4. Save the screenshot in the designated "Screenshot" folder.	The "Add New Patient" form should successfully close without any errors. A screenshot of the "Sale" page should be taken and saved correctly in the "Screenshot" folder, capturing the current state of the page as visual confirmation.

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.

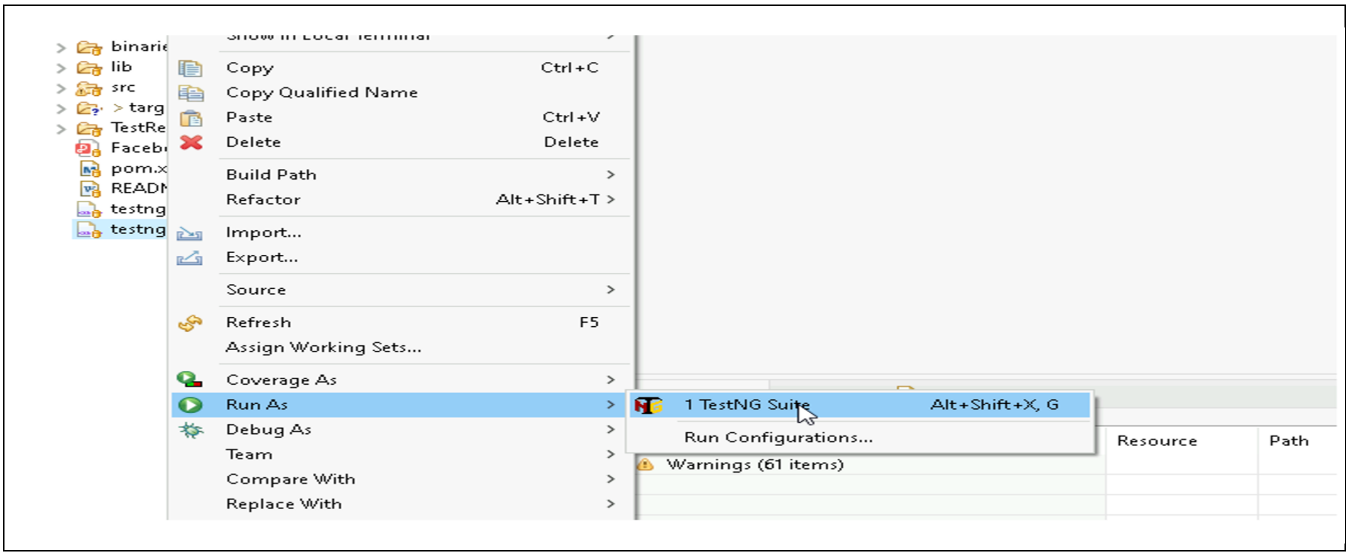
IMPLEMENTATION/FUNCTIONAL REQUIREMENT

1.1 CODE QUALITY/OPTIMIZATIONS

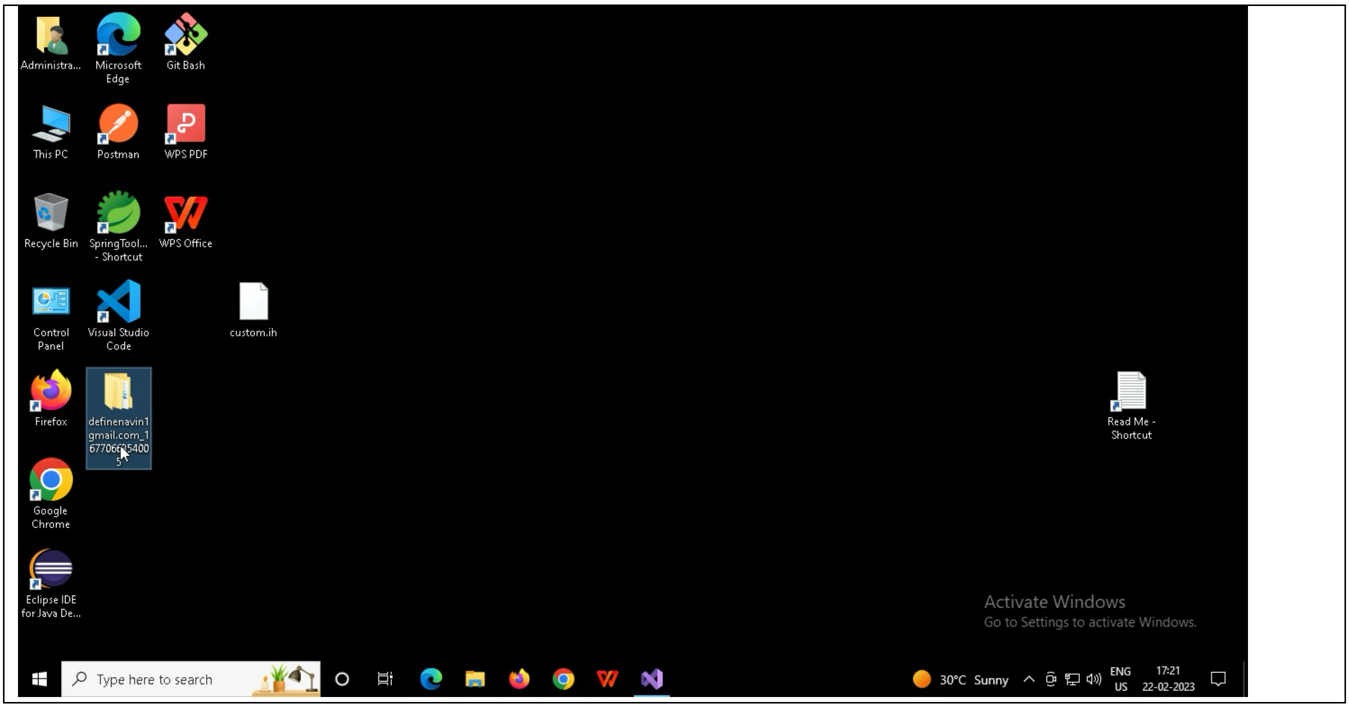
1. Associates should have written clean code that is readable.
2. Associates need to follow SOLID programming principles.

EXECUTION STEPS TO FOLLOW

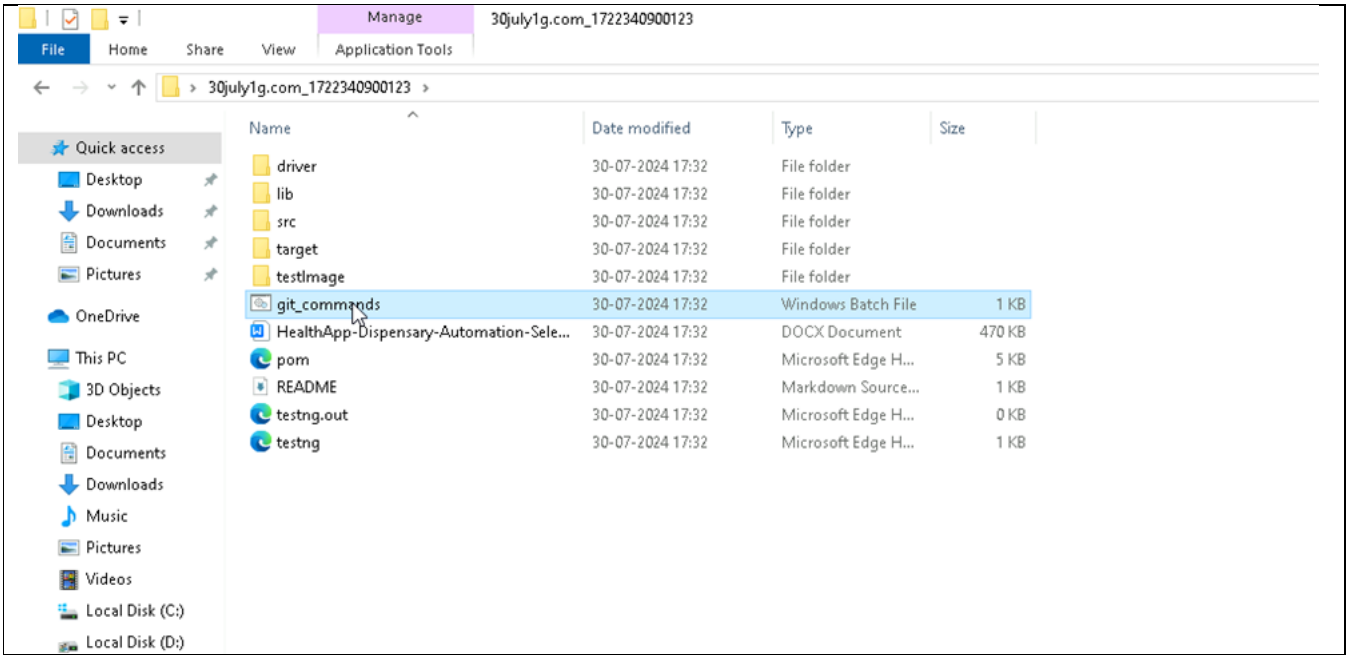
1. **You are mandatory required to run test cases for applications before final submission. Without which project evaluation will not happen.**
2. **You can launch test cases any time as follows: Right click on testng.xml and run TestNGSuite**



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

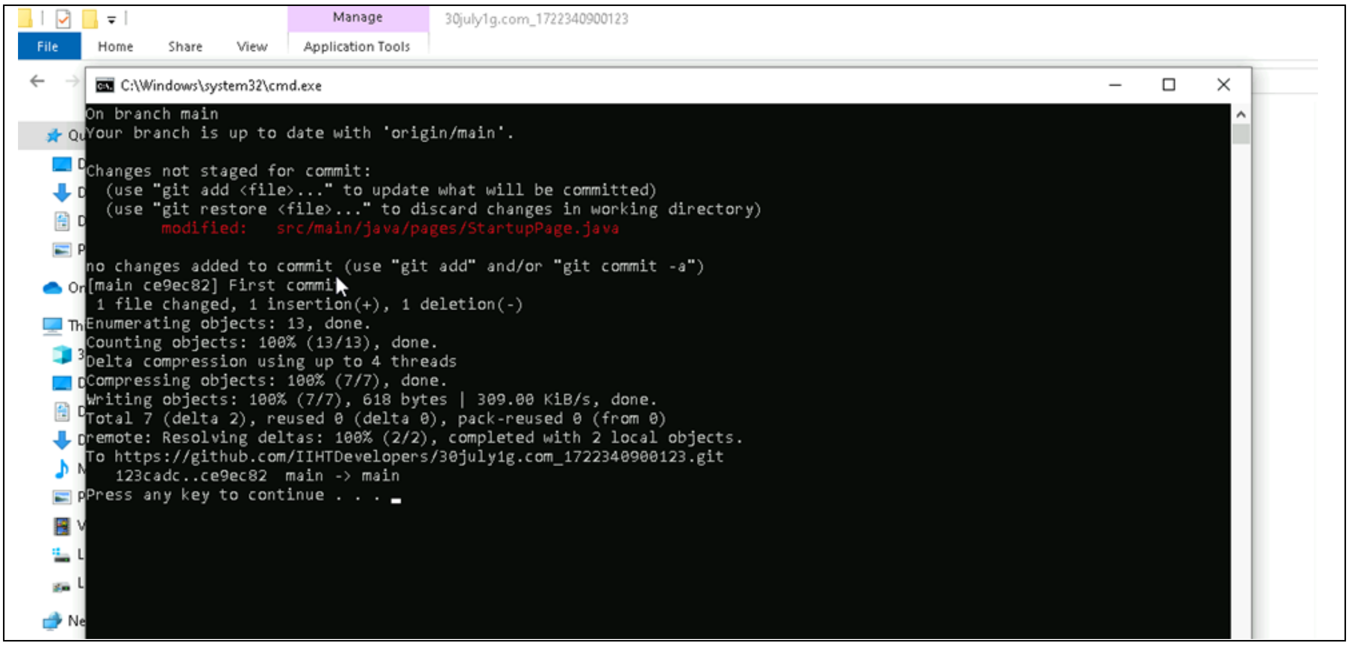


In your project folder, you will find a batch file named git_commands



Name	Date modified	Type	Size
driver	30-07-2024 17:32	File folder	
lib	30-07-2024 17:32	File folder	
src	30-07-2024 17:32	File folder	
target	30-07-2024 17:32	File folder	
testImage	30-07-2024 17:32	File folder	
git_commands	30-07-2024 17:32	Windows Batch File	1 KB
HealthApp-Dispensary-Automation-Sele...	30-07-2024 17:32	DOCX Document	470 KB
pom	30-07-2024 17:32	Microsoft Edge H...	5 KB
README	30-07-2024 17:32	Markdown Source...	1 KB
testng.out	30-07-2024 17:32	Microsoft Edge H...	0 KB
testng	30-07-2024 17:32	Microsoft Edge H...	1 KB

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



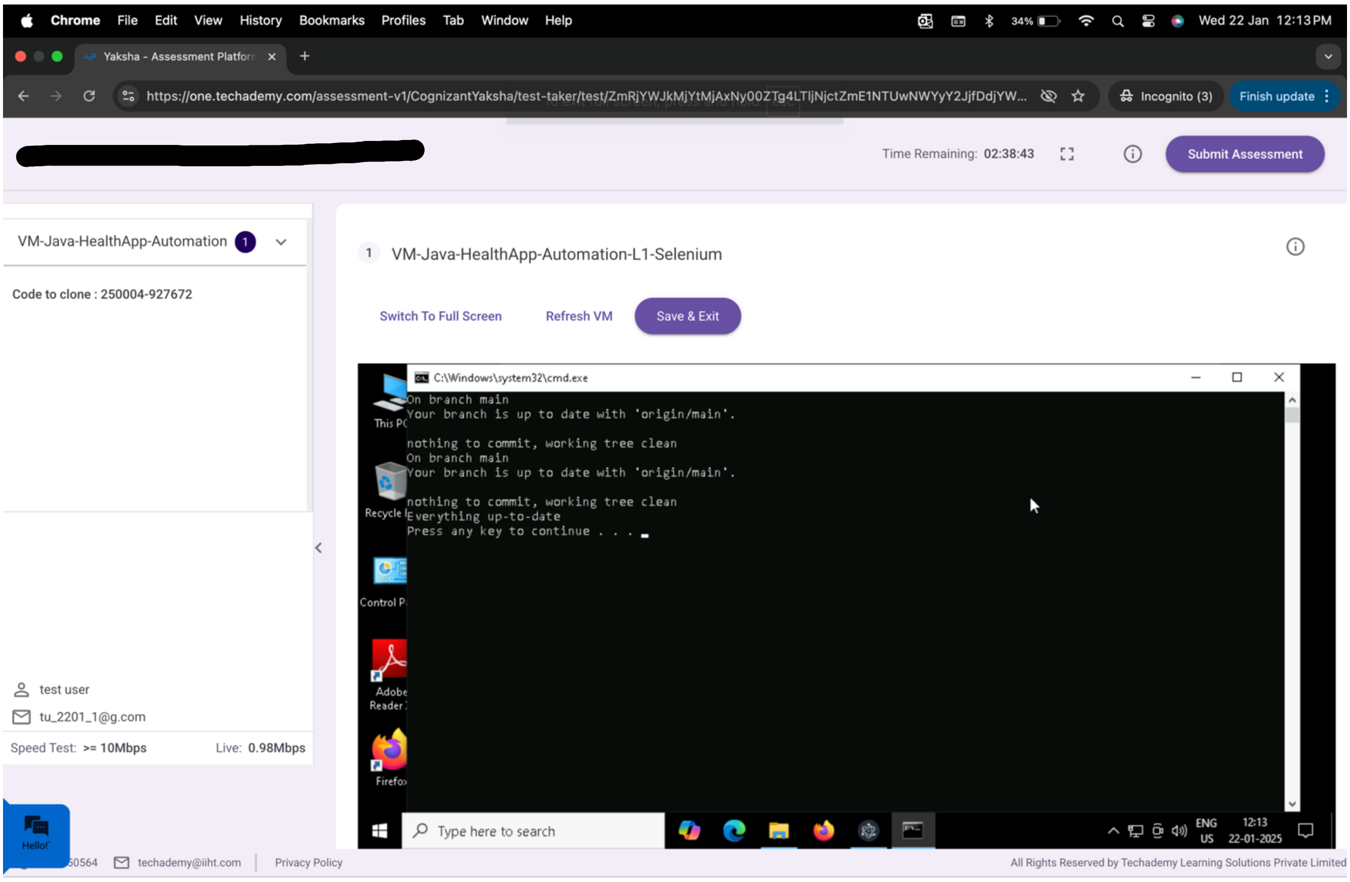
```
On branch main
Your branch is up to date with 'origin/main'.

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 . . .
```

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.



[redacted]

Time Remaining: 02:38:43

1 VM-Java-HealthApp-Automation-L1-Selenium

Code to clone : 250004-927672

test user

tu_2201_1@g.com

Speed Test: >= 10Mbps Live: 0.98Mbps

```
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
Everything up-to-date
Press any key to continue . . .
```

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

==

All the Best