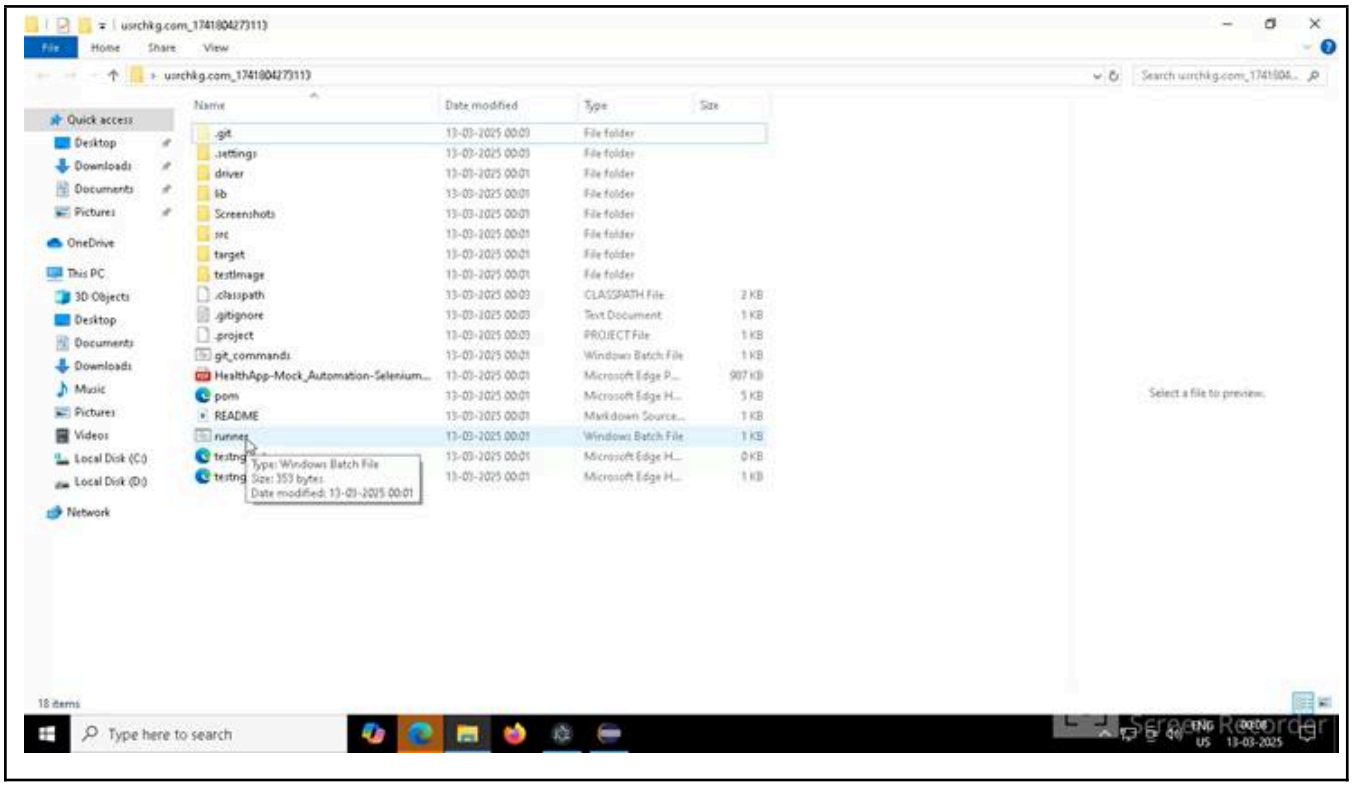
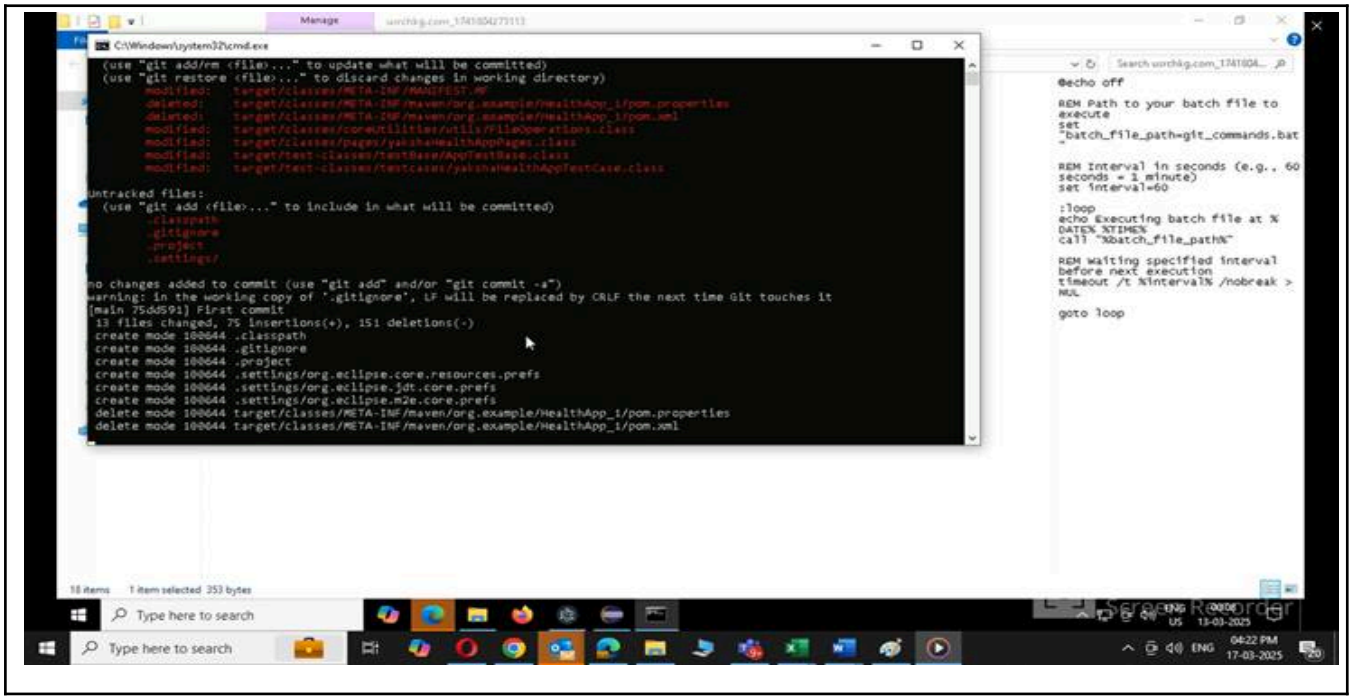
REST-ASSURED
(API-AUTOMATION PROJECT – PL1_8)

Pre-requisite:

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

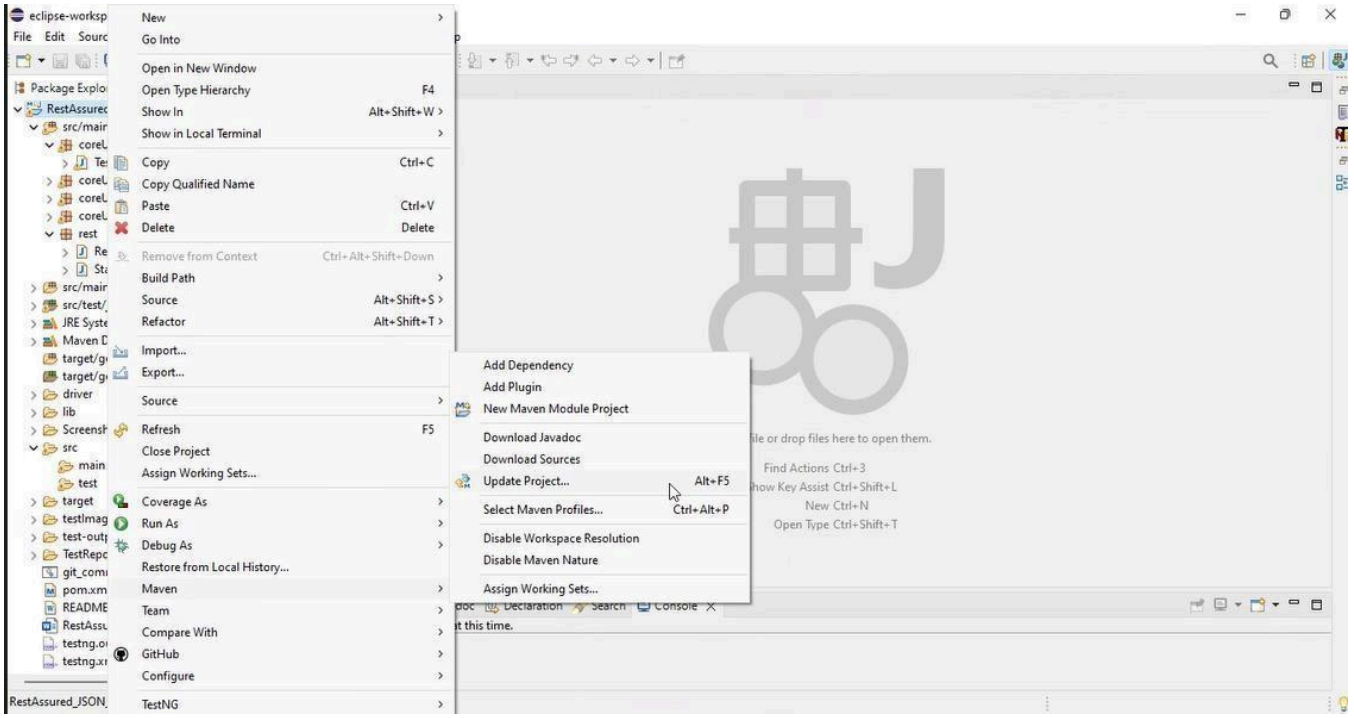


This will launch a command terminal for you where it will keep on pushing your updated code to GIT on regular intervals. Keep that command terminal open at backend and you can continue working on your project.

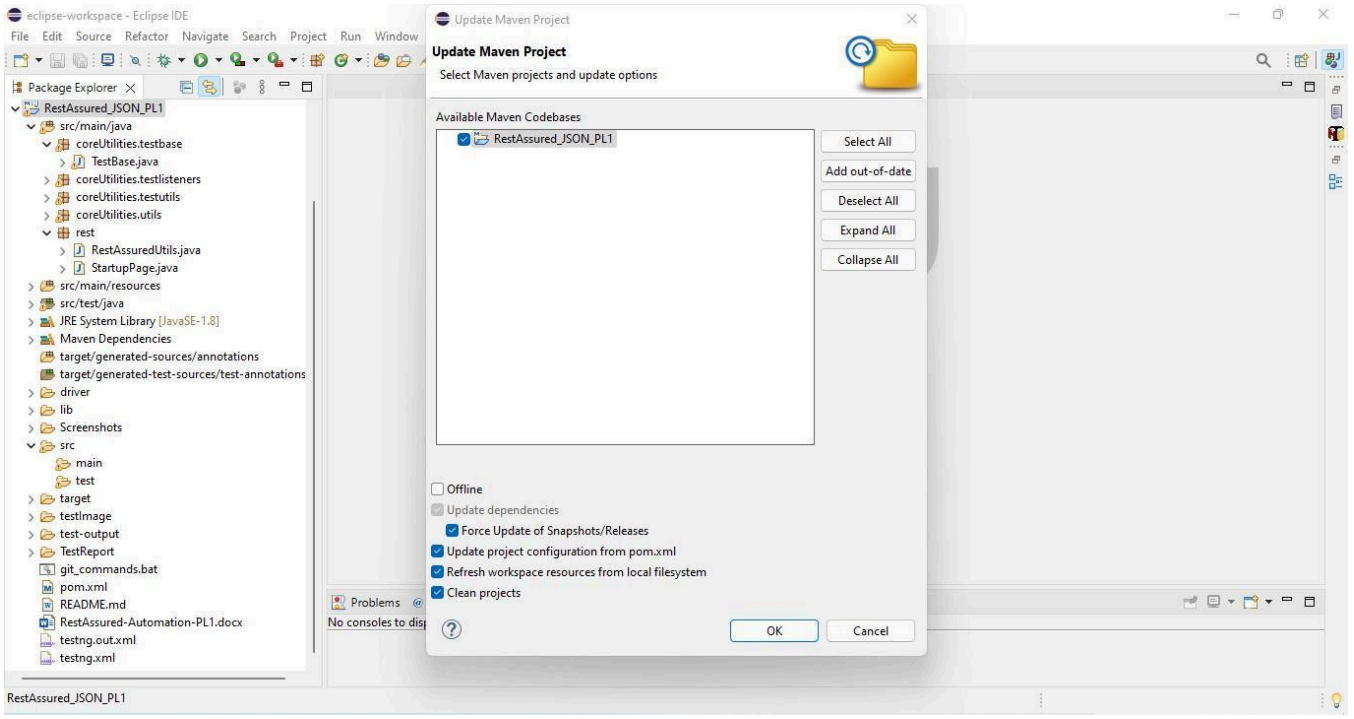


As soon as you import project in eclipse, update the project using maven update option as below. This is to resolve issue if any maven dependency not downloaded properly:

1. Right click on project : Go to “Maven” : Select “Update Project”



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



Template Code Structure:

- Below are the packages and files you will be required to work upon.
- Other Files and packages you can ignore.
- In other Files and packages do not do any changes. It would affect your evaluation.
- You are not required to work in “Test” Folder. Files there are noneditable. 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 for creating post the data should be read from excel.**
/src/main/java/rest	ApiUtil.java	1. All core activities to be performed here. 2. The comments associated with each templated method here describe the expectation. 3. Declare any variable/object you need to share data/status between different methods. 4. Do not modify the signature of methods declared here. 5. You can create additional supportive common methods in CommonEvents class.
/src/main/resources/	TestData.xlsx	Data is present for creating post and put requests
/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 RestAssured.

Key Activities to implement:

Sl No.	Summary	Action	Expected Result
1	Retrieve the Holidays Data in the Method: GetHolidayData(String endpoint, String cookieValue, Map<String, String> body)	1. Create an URL by combining the BASE_URL (already declared) and endpoint (“/web/index.php/api/v2/leave/holidays?fromDate=2025-01-01&toDate=2025-12-31 ”) URL becomes: “opensource-demo.orangehrmlive.com/web/index.php/api/v2/leave/holidays?fromDate=2025-01-01&toDate=2025-12-31” 2. Fetch the Cookie for the session and set the Header Content Type to “application/json”. 3. Trigger a GET call to the specified endpoint. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: ● Response ● statusCode (extracted from Response object) ● Status (extracted from Response object) ● id List<Object> ● name List<Object> ● date List<Object> ● recurring List<Object> ● length List<Object> ● lengthName List<Object> 5.Return the CustomResponse object from the method.	-The method should Returns an object of type CustomResponse containing: ● statusCode=200 ● status=OK. -Implementation should be corrects -Non null and non empty List for: ● id ● name ● date ● recurring ● length ● lengthName -The complete raw response.
2	Retrieve the employees Data in the Method: GetLeaveData(String endpoint, String cookieValue, Map<String, String> body)	1. Create an URL by combining the BASE_URL (already declared) and endpoint "/web/index.php/api/v2/leave/holidays?fromDate=2025-01-01&toDate=2025-12-31") URL becomes: “opensource-demo.orangehrmlive.com"/web/index.php/api/v2/leave/holidays?fromDate=2025-01-01&toDate=2025-12-31" 2. Fetch the Cookie for the session and set the Header Content Type to “application/json”. 3.. Trigger a GET call to the specified endpoint. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object:	-The method should Returns an object of type CustomResponse containing: ● statusCode=200 ● status=OK. -Implementation should be corrects -Non null and non empty List for: ● id ● name ● date

		● Response ● statusCode (extracted from Response object) ● Status (extracted from Response object) ● id List<Object> ● name List<Object> ● date List<Object> ● recurring List<Object> ● length List<Object> ● lengthName List<Object> 5. Return the CustomResponse object from the method.	● recurring ● length ● lengthName -The complete raw response.
3	Retrieve the employees count in the Method: GetEmpCount(String endpoint, String cookieValue, Map<String, String> body)	1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/pim/employees/count") URL becomes: "opensource-demo.orangehrmlive.com/web/index.php/api/v2/pim/employees/count" ” 2. Fetch the Cookie for the session and set the Header Content Type to “application/json”. 3.. Trigger a GET call to the specified endpoint. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: ● Response ● statusCode (extracted from Response object) ● Status (extracted from Response object) ● count `int`. 5. Return the CustomResponse object from the method.	-The method should Returns an object of type CustomResponse containing: ● statusCode=200 ● status=OK. -Implementation should be corrected. -Non null value for: ● count. -The complete raw response.
4	retrieve all leave types with no limit: GetLeaveType(String endpoint, String cookieValue, Map<String, String> body)	1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/leave/leave-types?limit=0") URL becomes: “opensource-demo.orangehrmlive.com/web/index.php/api/v2/leave/leave-types?limit=0” 2. Fetch the Cookie for the session and set the Header Content Type to “application/json”. 3.. Trigger a GET call to the specified endpoint. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: ● Response ● statusCode (extracted from Response object) ● Status (extracted from Response object) ● id List<Object> ● name List<Object> ● deleted List<Object> ● situational List<Object> 5. Return the CustomResponse object from the method	-The method should Returns an object of type CustomResponse containing: ● statusCode=200 ● status=OK. -Implementation should be corrected. -Non null and non empty List for: ● id ● name ● deleted ● situational -The complete raw response.

5	Retrieve the Usage Reports: GetUsageReport(String endpoint, String cookieValue, Map<String, String> body)	1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/leave/reports?name=my_leave_entitlements_and_usage") URL becomes: “opensource-demo.orangehrmlive.com/web/index.php/api/v2/leave/reports?name=my_leave_entitlements_and_usage” 2. Fetch the Cookie for the session and set the Header Content Type to “application/json”. 3.. Trigger a GET call to the specified endpoint. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: ● Response ● statusCode (extracted from Response object) ● Status (extracted from Response object) ● name List<Object> ● prop List<Object> ● size List<Object> ● pin List<Object> ● cellProperties List<Object> 5. Return the CustomResponse object from the method	- The method should Returns an object of type CustomResponse containing: ● statusCode= 200 ● status=OK. -Implementation should be corrected. -Non null and non empty List for: ● name ● prop ● size ● pin ● cellProperties object -The complete raw response.
6	Update termination reason by ID using the Method: PutTerminationReason(String endpoint, String cookieValue, String requestBody).	1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/pim/termination-reasons/1") URL becomes: “opensource-demo.orangehrmlive.com/web/index.php/api/v2/pim/termination-reasons/1” 2. Fetch the Cookie for the session and set the Header Content Type to “application/json”. 3.. Trigger a PUT call to the specified endpoint with the request body (passed in argument). 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: ● Response ● statusCode (extracted from Response object) ● Status (extracted from Response object) ● id Object ● name Object 5. Return the CustomResponse object from the method.	- The method should Returns an object of type CustomResponse containing: ● statusCode= 200 ● Status=OK. -Implementation should be corrected. -Non null value for: ● id ● name. -The complete raw response.

7	Delete employee by ID using Method: DeletePim(String deleteEndPoint, String cookieValue, String requestBody)	1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/pim/employees") URL becomes: “opensource-demo.orangehrmlive.com/web/index.php/api/v2/pim/employees” 2. Fetch the Cookie for the session and set the Header Content Type to “application/json”. 3. Trigger a DELETE call to the specified endpoint with passed body in requestBody. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: ● Response ● statusCode (extracted from Response object) ● Status (extracted from Response object). ● data Object (value at index 0 of the data array) 5. Return the CustomResponse object from the met .	- The method should Returns an object of type CustomResponse containing: ● statusCode= ● 200 ● Status=OK. ● data -Implementation should be corrected. -The complete raw response.
8	Generate Employee dynamically using Method: PostPimEmp(String endpoint, String cookieValue, String requestBody)	1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/pim/employees ") URL becomes: “opensource-demo.orangehrmlive.com /web/index.php/api/v2/pim/employees” 2. Fetch the Cookie for the session and set the Header Content Type to “application/json”. 3.. Trigger a POST call to the specified endpoint and pass the body. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: ● Response ● firstName Object ● employeeId Object ● statusCode (extracted from Response object) ● Status (extracted from Response object). 5. Return the CustomResponse object from the met.	-The method should Returns an object of type CustomResponse containing: ● statusCode= 200 ● status=OK. -Implementation should be corrected. -Request Body should have: ● firstName ● employeeId -The complete raw response.

9	Update the Vim employee using the method: PutVimEmp(String endpoint, String cookieValue, String requestBody) .	1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/pim/employees ") URL becomes: "opensource-demo.orangehrmlive.com /web/index.php/api/v2/pim/employees" 2. Fetch the Cookie for the session and set the Header Content Type to "application/json". 3.. Trigger a PUT call to the specified endpoint and pass the body. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: ● Response ● empNumber Object (passed as id) ● lastName Object ● firstName Object (passed as name) ● statusCode (extracted from Response object) ● Status (extracted from Response object). 5. Return the CustomResponse object from the met.	-The method should Returns an object of type CustomResponse containing: ● statusCode= 200 ● status=OK. -Implementation should be corrected. -Response should have: ● firstName = achrrr ● lastName= haji ● employee number=not null -The complete raw response.
10	Delete a specific employee by dynamically retrieving the PIM employee id using the method: DeletePim(String deleteEndPoint, String cookieValue, String requestBody) .	1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/pim/employees ") URL becomes: "opensource-demo.orangehrmlive.com /web/index.php/api/v2/pim/employees" 3.Fetch the Cookie for the session and set the Header Content Type to "application/json". 4. Fetch the id of the employee to delete. 5. Send a DELETE req to the url 6. You must attach the passed body. 7. Create an object of type CustomResponse and initialize it with values extracted from the Response object: ● Response ● statusCode (extracted from Response object) ● Status (extracted from Response object). ● data Object (value at index 0 of the data array)	-The method should Returns an object of type CustomResponse containing: ● statusCode= 200 ● status=OK ● data -Implementation should be corrected. -The complete raw response.

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 REST Assured to automate the API testing for all the provided methods (e.g., GET, POST, PUT, DELETE).** In other words, the automation script should perform all mentioned API interactions, including validation of responses.
2) **Learners should not use any pre-built libraries or tools to validate API responses (e.g., JSON schema validation tools).** They should manually validate the response content (e.g., status codes, response body, etc.) by writing their own logic for assertion.

IMPLEMENTATION/FUNCTIONAL REQUIREMENT

1.1 CODE QUALITY/OPTIMIZATIONS

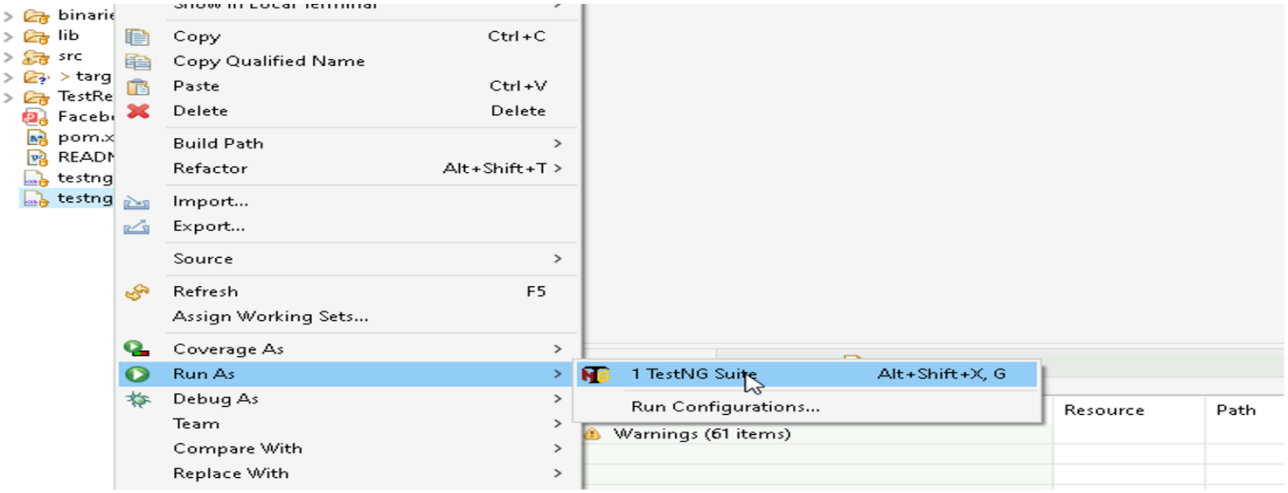
1. Associates should have written clean code that is readable.
2. Associates need to follow SOLID programming principles.

EXECUTION STEPS TO FOLLOW

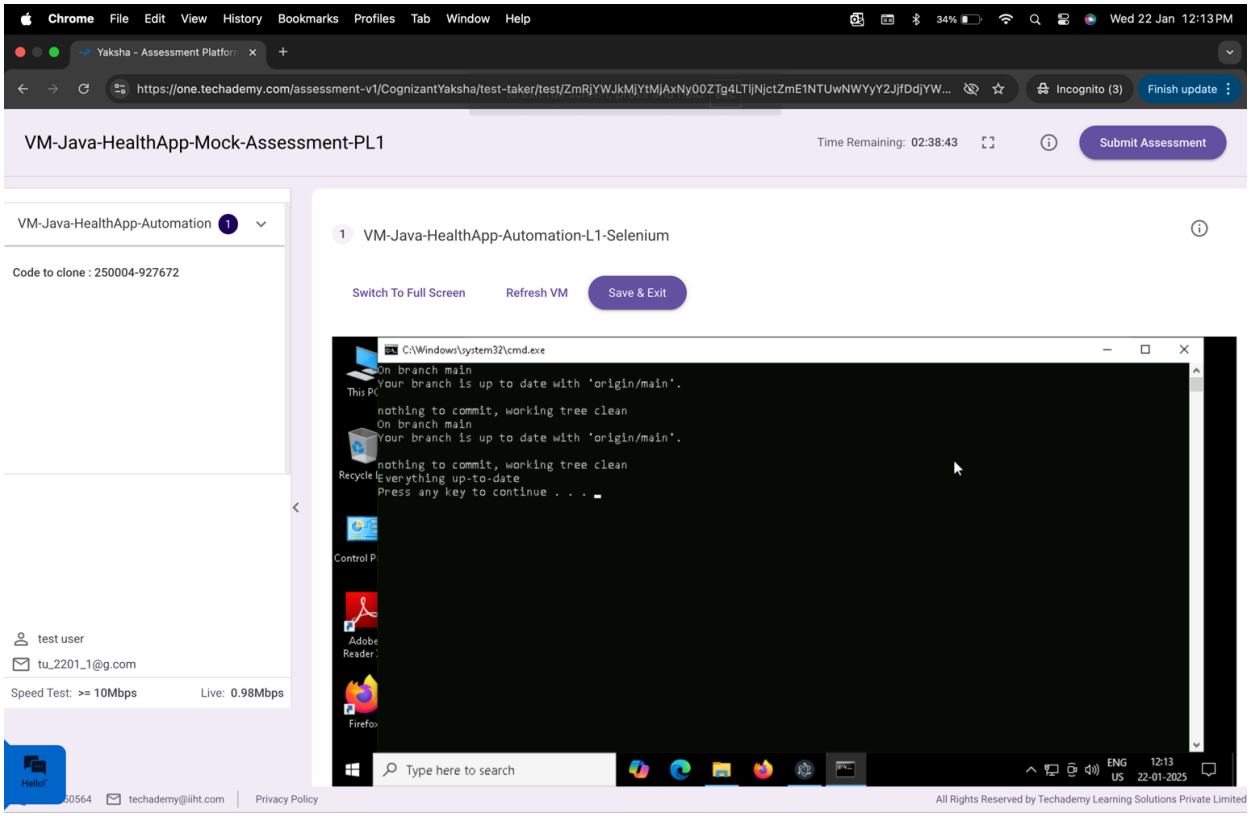
---

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
