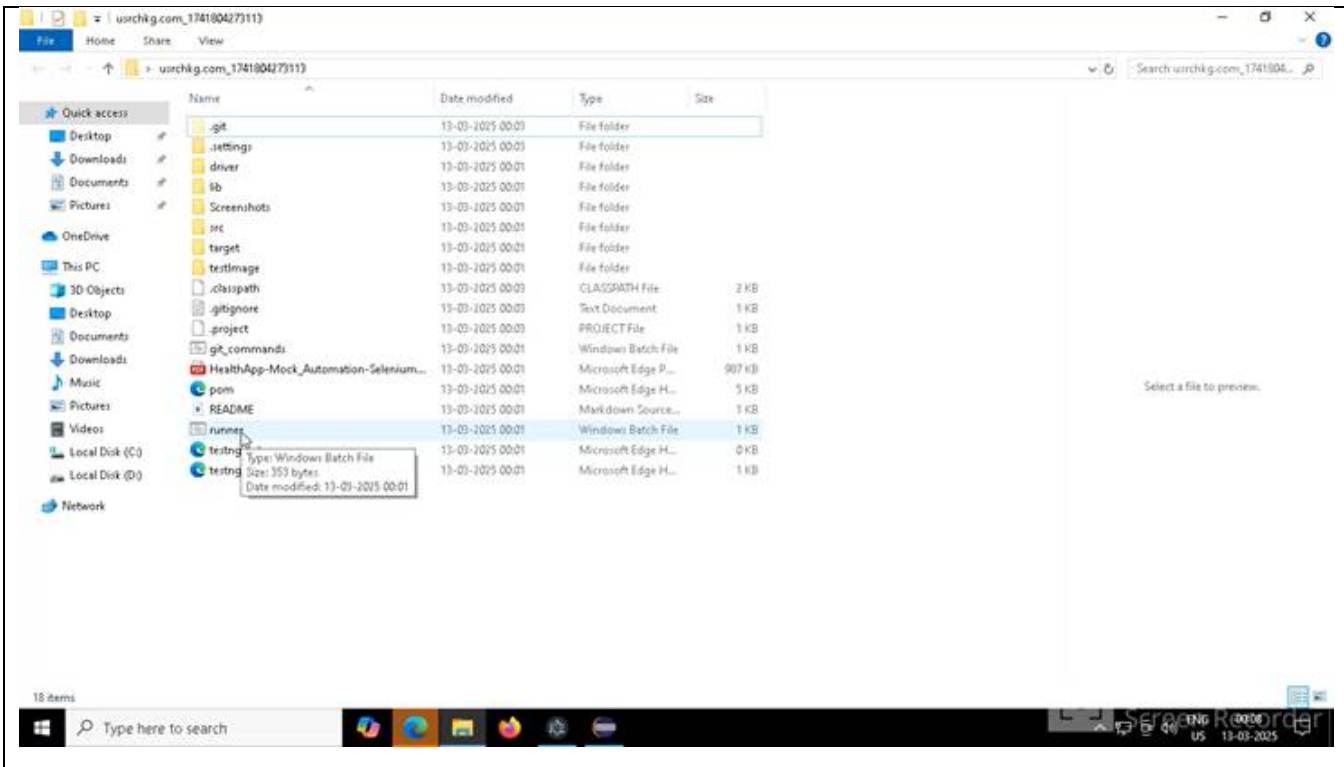


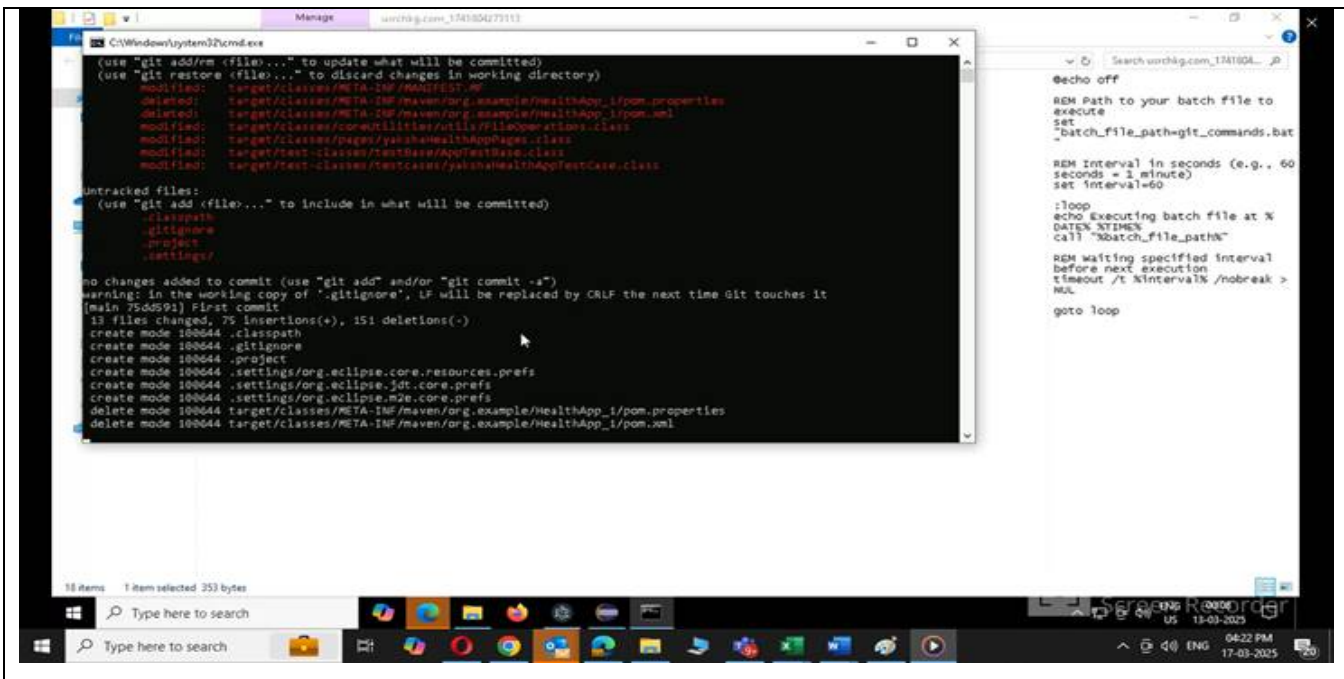
REST-ASSURED
(API - AUTOMATION PROJECT – PL1-1)

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.**

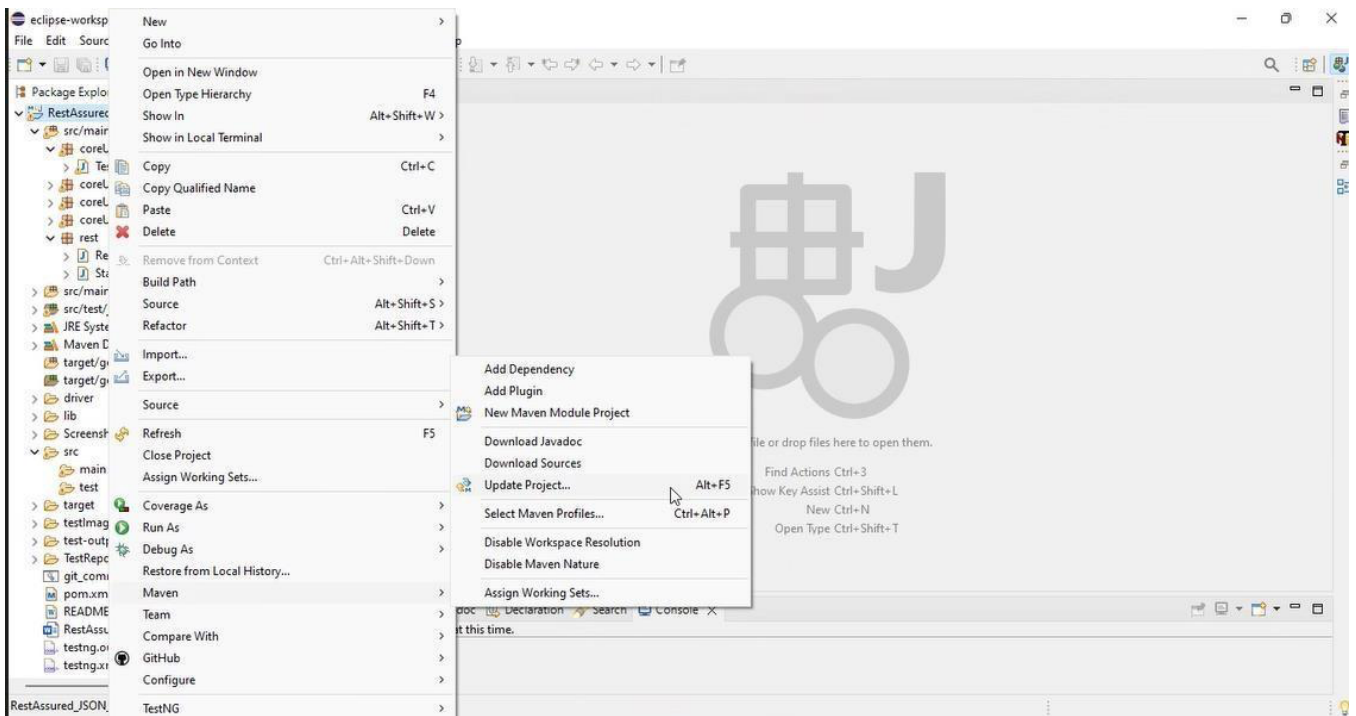


This will launch a command terminal for you where it will keep on pushing your updated code to GIT on regular intervals. Keep that command terminal open at backend and you can continue working on your project.

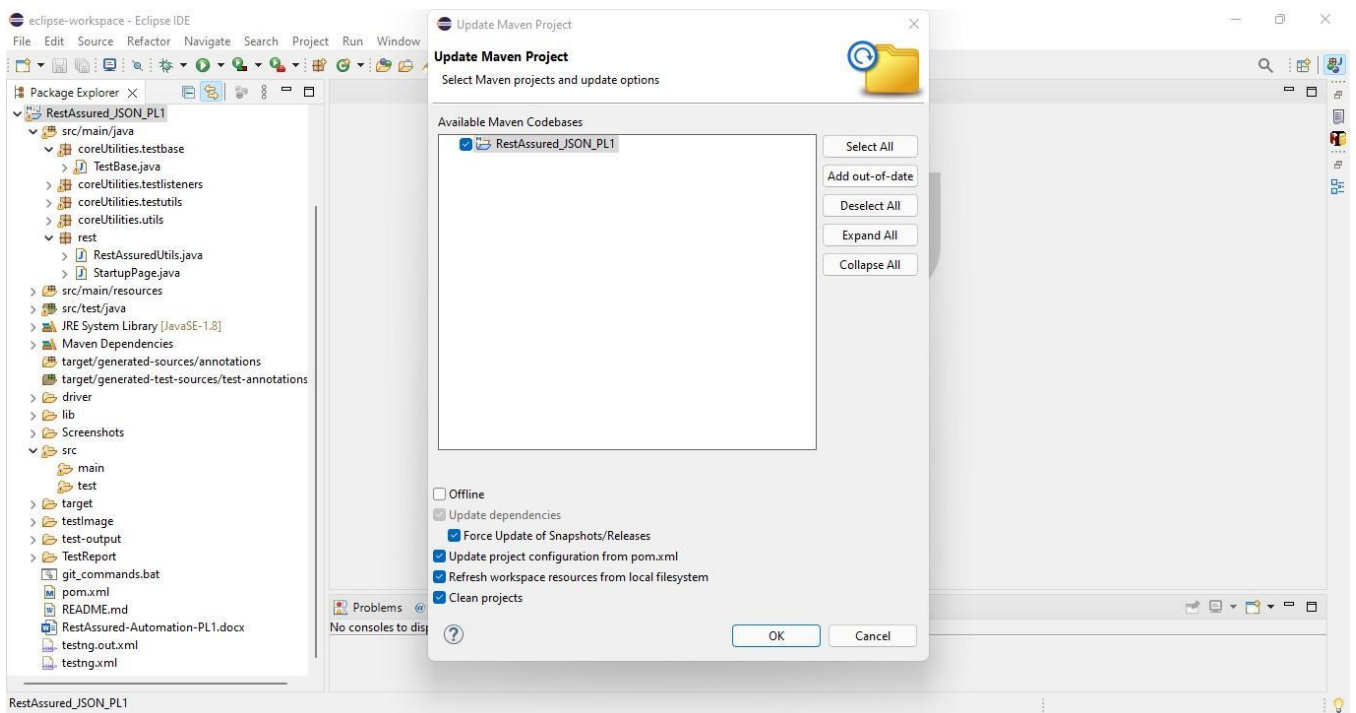


As soon as you import project in eclipse, update the project using maven update option as below. This is to resolve issue if any maven dependency not downloaded properly:

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/utis/	FileOperations.java	<ol style="list-style-type: none"> 1. It contains methods to read data from Excel files. 2. The method is in templated form. 3. You will be required to implement these methods as the first activity, because even the URL to navigate to is read using these methods.
/src/main/java/pages	ApiUtil.java	<ol style="list-style-type: none"> 1. All core activities to be performed here. 2. The comments associated with each templated method here describe the expectation. 3. Declare any 4. 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 7. CommonEvents class.
/src/main/resources/	Config.xlsx	<ol style="list-style-type: none"> 1. Data present to be used in creating a new Post.
	expected_data.xlsx	<ol style="list-style-type: none"> 1. Contains data to fill in the form
/src/main/java/coreUtilities/utis	CommonEvents.java	<ol style="list-style-type: none"> 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 RestAssured.

Key Activities to implement:

#	Summary	Action	Expected Result
1	Perform login with Cookie Authentication using method: GetLogin(String endpoint, String cookieValue, Map<String, String> body)	<ol style="list-style-type: none"> Construct the final URL by combining the BASE_URL and the provided endpoint parameter: <ol style="list-style-type: none"> https://opensource-demo.orangehrmlive.com/web/index.php/auth/login Initialize a RequestSpecification using RestAssured: <ol style="list-style-type: none"> Add a cookie with name "orangehrm" and the value passed as cookieValue. Set the "Content-Type" header to "application/json". Trigger a GET request to the above URL. Create a CustomResponse object with: <ol style="list-style-type: none"> Full Response Extracted statusCode Extracted statusLine Return the CustomResponse object. 	<ol style="list-style-type: none"> Method should return a fully initialized CustomResponse object with: <ol style="list-style-type: none"> statusCode = 200 status = OK (or valid status line like "HTTP/1.1 200 OK") The complete raw Response object
2	Retrieve action summary using session cookie authentication through method: GetEmpActionSummary(String endpoint, String cookieValue, Map<String, String> body)	<ol style="list-style-type: none"> Construct the complete URL by concatenating the BASE_URL and the provided endpoint: <ol style="list-style-type: none"> /web/index.php/api/v2/dashboard/employees/action-summary Create a RequestSpecification using RestAssured: <ol style="list-style-type: none"> Add a session cookie named "orangehrm" with the provided cookieValue. Add a header "Content-Type" with the value "application/json". If a request body is provided (not null), attach it to the request using .body(). Send a GET request to the constructed URL. 	<ol style="list-style-type: none"> The method should return a CustomResponse object with: <ol style="list-style-type: none"> statusCode = 200 status equal to a valid HTTP status line like "HTTP/1.1 200 OK" The complete raw Response object

		<p>5. Create and return a CustomResponse object initialized with:</p> <ol style="list-style-type: none"> The complete Response Extracted statusCode Extracted statusLine languageId, languageName, languageCode lists 	
3	<p>Retrieve dashboard shortcut access permissions using session-based authentication through the method:</p> <p>GetDashboardShortcut(String endpoint, String cookieValue, Map<String, String> body)</p>	<ol style="list-style-type: none"> Construct the final URL by combining the BASE_URL and the provided endpoint: <ul style="list-style-type: none"> https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/dashboard/shortcuts Create a RequestSpecification using RestAssured: <ul style="list-style-type: none"> Add a session cookie named "orangehrm" using the provided cookieValue. Set the request header "Content-Type" to "application/json". If a request body (body) is provided (i.e., not null), attach it using .body() method. Wrap each of these fields into separate List<Object> collections: <ul style="list-style-type: none"> leaveAssignLeave, leaveLeaveList, leaveApplyLeave, leaveMyLeave, timeEmployeeTimesheet, timeMyTimesheet Create a CustomResponse object using: <ul style="list-style-type: none"> The complete Response object Extracted statusCode and statusLine All six field lists Return the populated CustomResponse object. 	<ol style="list-style-type: none"> The method should return a CustomResponse object containing: <ul style="list-style-type: none"> statusCode = 200 A valid status line such as "HTTP/1.1 200 OK" Non-null and non-empty lists for the following keys: <ul style="list-style-type: none"> leave.assign_leave leave.leave_list leave.apply_leave leave.my_leave time.employee_timesheet time.my_timesheet The complete Response object from the API call

4	<p>Retrieve employee leaves information using session-based authentication through method: GetEmpLeaveInfo(String endpoint, String cookieValue, Map<String, String> body)</p>	<ol style="list-style-type: none"> Construct the final URL by combining the BASE_URL and the provided endpoint: <ul style="list-style-type: none"> /web/index.php/api/v2/dashboard/employees/leaves?date={currentDate} Create a RequestSpecification using RestAssured: <ul style="list-style-type: none"> Add a session cookie named "orangehrm" using the provided cookieValue. Set the request header "Content-Type" to "application/json". If a request body (body) is provided (i.e., not null), attach it using .body() method. Create a CustomResponse object using: <ul style="list-style-type: none"> The complete Response object Extracted statusCode and statusLine Return the populated CustomResponse object. 	<ul style="list-style-type: none"> Method must return a valid CustomResponse object containing: <ul style="list-style-type: none"> statusCode = 200 status like "HTTP/1.1 200 OK" Non-null and non-empty lists of leaves The full raw Response object
5	<p>Retrieve employee subunit using session-based cookie authentication via method: GetEmpSubunit(String endpoint, String cookieValue, Map<String, String> body)</p>	<ol style="list-style-type: none"> Construct the final URL by combining the BASE_URL and the provided endpoint: <ul style="list-style-type: none"> /web/index.php/api/v2/dashboard/employees/subunit Create a RequestSpecification using RestAssured: <ul style="list-style-type: none"> Add a session cookie named "orangehrm" using the provided cookieValue. Set the request header "Content-Type" to "application/json". If a request body (body) is provided (i.e., not null), attach it using .body() method. Create a CustomResponse object using: 	<ul style="list-style-type: none"> The method must return a CustomResponse containing: <ul style="list-style-type: none"> statusCode = 200 A valid status line like "HTTP/1.1 200 OK" An Data object with non-null and non-empty lists for: <ul style="list-style-type: none"> id name count The full raw Response object

		<ul style="list-style-type: none"> • The complete Response object • Extracted statusCode and statusLine • Extracted ID, name, and count fields <p>5. Return the populated CustomResponse object.</p>	
6	Update an employee name using session-based authentication via method: PutEmpName(String endpoint, String cookieValue, Object body)	<ol style="list-style-type: none"> 1. Construct the complete URL by appending the endpoint to the base URL: https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/pa y-grades/{id} 2. Initialize a RequestSpecification using RestAssured: <ul style="list-style-type: none"> • Add the session cookie "orangehrm" with the provided cookieValue. • Set the header "Content-Type" to "application/json". 3. If a request body (body) is provided (not null), include it in the request using .body(). 4. Trigger a PUT request to the constructed URL and extract the Response. 5. Wrap all extracted values into individual List<Object> collections. 6. Create and return a CustomResponse object containing: <ul style="list-style-type: none"> • The full Response • The extracted statusCode and statusLine • The employee detail lists (id, name, currency list with details) 	<ul style="list-style-type: none"> • The method should return a CustomResponse object with: <ul style="list-style-type: none"> • statusCode = 200 • A valid status string (e.g., "HTTP/1.1 200 OK") • Non-null and non-empty lists for: <ul style="list-style-type: none"> ■ id ■ name ■ currencies • The complete raw Response object

7	Create employee status using session-based authentication through method: PostEmpStatus(String endpoint, String cookieValue, String body)	<ol style="list-style-type: none">1. Construct the complete URL by appending the endpoint to the base URL: <code>https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/employment-statuses</code>2. Initialize a RequestSpecification using RestAssured:<ul style="list-style-type: none">● Add the session cookie "orangehrm" with the provided <code>cookieValue</code>.● Set the header "Content-Type" to "application/json".3. If a request body (<code>body</code>) is provided (not null), include it in the request using <code>.body()</code>.4. Trigger a POST request to the constructed URL and extract the Response.5. Wrap all extracted values into individual <code>List<Object></code> collections.6. Create and return a CustomResponse object containing:<ul style="list-style-type: none">● The full <code>Response</code>● The extracted <code>statusCode</code> and <code>statusLine</code>● The employee detail lists (<code>id</code>, <code>name</code>)	<ul style="list-style-type: none">● The method should return a valid <code>CustomResponse</code> containing:<ul style="list-style-type: none">● <code>statusCode = 200</code>● A valid HTTP status (e.g., "HTTP/1.1 200 OK")● Non-empty string values for:<ul style="list-style-type: none">■ <code>id</code>■ <code>name</code>● The complete raw <code>Response</code> object

8	<p>Update employee status using session-based authentication through method: PutEmpStatus(String endpoint, String cookieValue, Object body)</p>	<ol style="list-style-type: none"> 1. Retrieve the first available language ID using helper method <code>getfirstlangid()</code>. 2. Construct a request body in JSON format containing the retrieved ID: <pre>{ "ids": [<id>] }</pre> 3. Compose the full URL by appending the endpoint to the base URL: <pre>https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/employment-statuses/{id}</pre> 4. Initialize a RequestSpecification using RestAssured: <ul style="list-style-type: none"> ● Add a session cookie named "orangehrm" with the provided <code>cookieValue</code> ● Set the "Content-Type" header to "application/json" 5. Attach the constructed <code>requestBody</code> to the request. 6. Send a PUT request to the full URL and extract the Response object. 7. Return a CustomResponse object with: <ul style="list-style-type: none"> ● The full <code>Response</code> object ● Extracted <code>statusCode</code> and <code>statusLine</code> ● The employee detail lists (id, name) 	<ul style="list-style-type: none"> ● The method should return a valid <code>CustomResponse</code> containing: <ul style="list-style-type: none"> ● <code>statusCode = 200</code> ● A valid HTTP status (e.g., "HTTP/1.1 200 OK") ● Non-empty string values for: <ul style="list-style-type: none"> ■ <code>id</code> ■ <code>name</code> ● The complete raw <code>Response</code> object
9	<p>Delete employee status using session-based authentication through method: DeleteEmp(String endpoint, String cookieValue, String body)</p>	<ol style="list-style-type: none"> 1. Retrieve the first available language ID using helper method <code>getfirstlangid()</code>. 2. Construct a request body in JSON format containing the retrieved ID: <pre>{ "ids": [<id>] }</pre> 	<ul style="list-style-type: none"> ● The method should return a valid <code>CustomResponse</code> containing: <ul style="list-style-type: none"> ● <code>statusCode = 200</code>

		<p>3. Compose the full URL by appending the endpoint to the base URL:</p> <pre>https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/employment-statuses/{id}</pre> <p>4. Initialize a RequestSpecification using RestAssured:</p> <ul style="list-style-type: none"> • Add a session cookie named "orangehrm" with the provided <code>cookieValue</code> • Set the "Content-Type" header to "application/json" <p>5. Attach the constructed <code>requestBody</code> to the request.</p> <p>6. Send a DELETE request to the full URL and extract the Response object.</p> <p>7. Return a CustomResponse object with:</p> <ul style="list-style-type: none"> • The full <code>Response</code> object • Extracted <code>statusCode</code> and <code>statusLine</code> 	<ul style="list-style-type: none"> • A valid HTTP status (e.g., "HTTP/1.1 200 OK") • The complete raw <code>Response</code> object
10	Create a new OpenID provider entry using a POST request with session-based authentication via method: PostEmpName(String endpoint, String cookieValue, String body)	<p>1. Read OpenID provider data (name,url,clientId,clientSecret) from Excel (PostData10) using <code>FileOperations.readExcelPOI</code>.</p> <p>2. Generate a unique provider name by appending a random suffix to the base name to avoid duplication:</p> <pre>String finalName = "name_" + UUID.randomUUID().toString().substring(0, 5);</pre> <p>3. Construct a JSON <code>requestBody</code> with the following format:</p> <pre>json { "name": "{finalName}" }</pre>	<ul style="list-style-type: none"> • The method should return a valid <code>CustomResponse</code> object containing: <ul style="list-style-type: none"> • <code>statusCode = 200</code> • A valid status string like "HTTP/1.1 200 OK" • Non-null, correctly mapped values for: <ul style="list-style-type: none"> ■ name ■ id • The full <code>Response</code> object

		<p>4. Build the full API endpoint URL:</p> <pre>https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/job-categories</pre> <p>5. Initialize a RequestSpecification using RestAssured:</p> <ul style="list-style-type: none"> • Add the "orangehrm" session cookie • Set "Content-Type" header to "application/json" • Attach the constructed <code>requestBody</code> using <code>.body()</code> <p>6. Trigger a POST request to the endpoint and extract the Response object.</p> <p>7. Return a CustomResponse object with:</p> <ul style="list-style-type: none"> • The full <code>Response</code> • <code>statusCode</code>, <code>statusLine</code> • Extracted fields: <code>id</code>, <code>name</code> 	
--	--	---	--

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

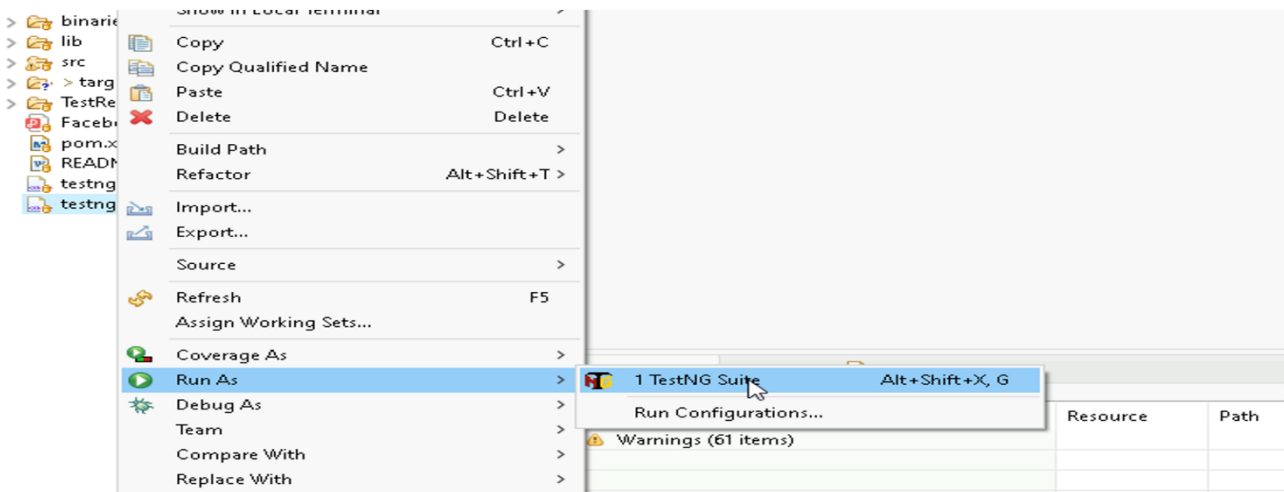
1.1 CODE QUALITY/OPTIMIZATIONS

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

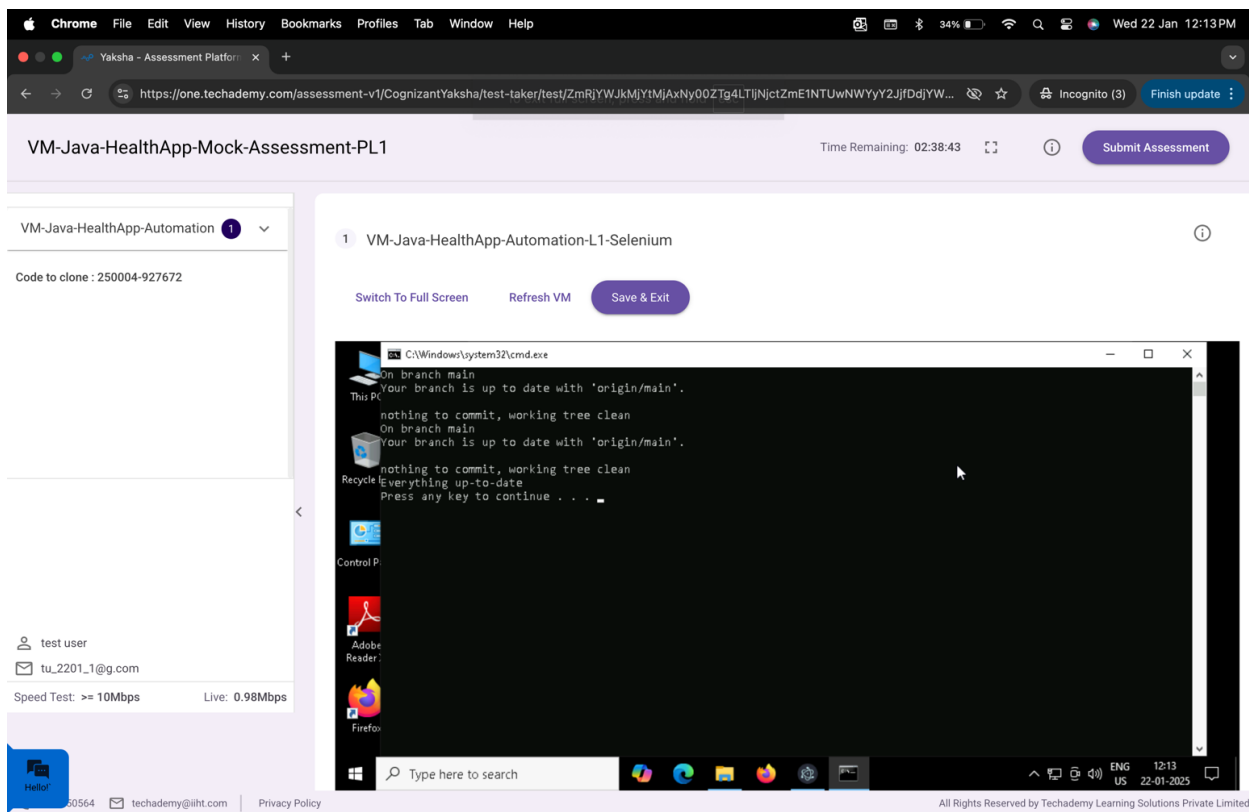
EXECUTION STEPS TO FOLLOW

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
