

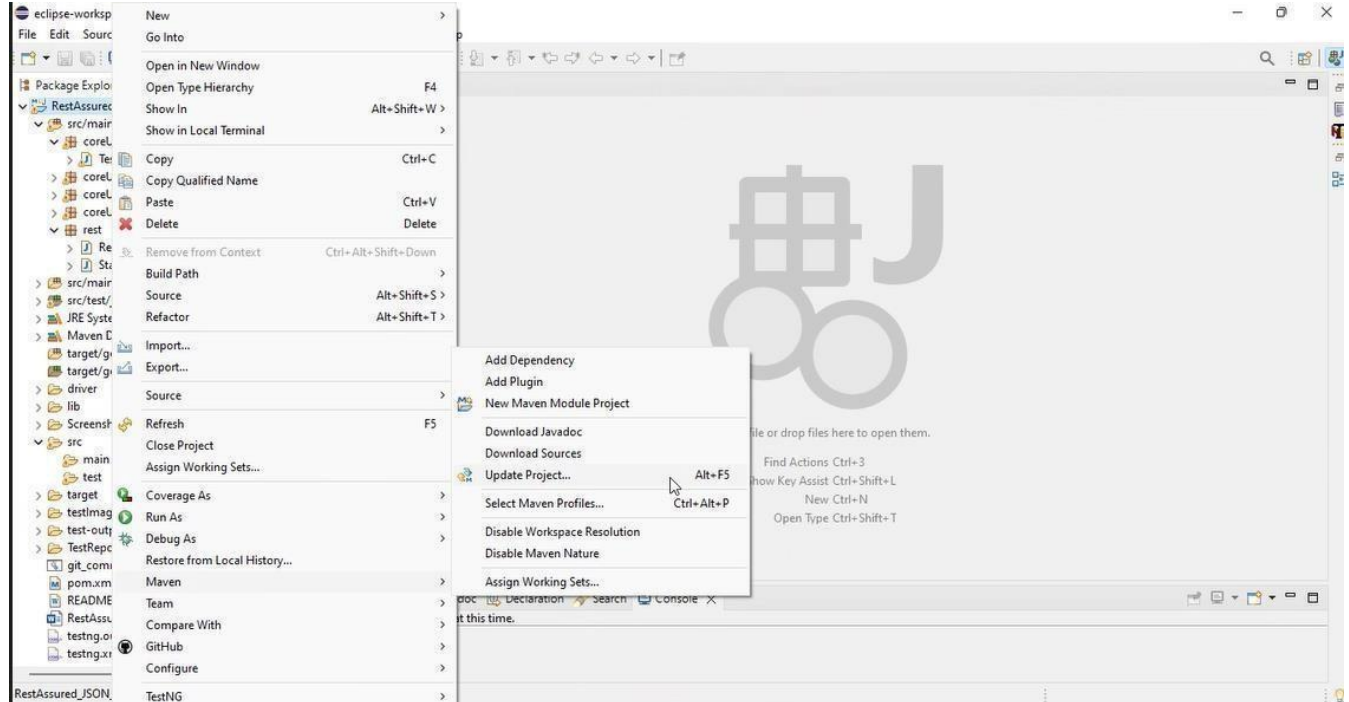
RESTASSURED API AUTOMATION PROJECT



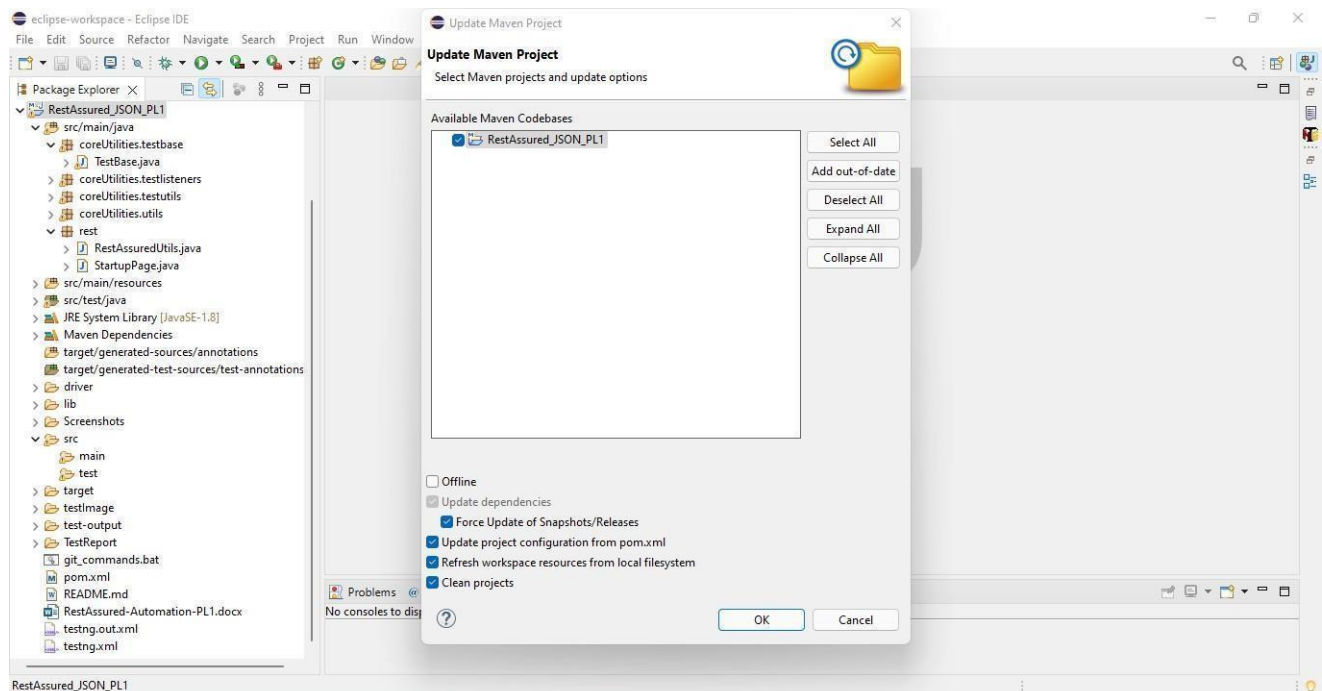
Pre-requisite:

As soon as you import a project in eclipse, update the project using the 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:

- 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 the “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/utis/	FileOperations.java	<ol style="list-style-type: none">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	<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 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/java/rest	AuthUtil.java	<ol style="list-style-type: none">1. Class already defined to read and return bearer token from

		config.properties file.
/src/main/java/rest/	CustomResponse	We have already created custom response object class. You need to use this and use it as return type from ApiUti.java class methods.
/src/main/resources/	Config.xlsx	Data present to be used in Implementing functions.
/src/main/java/coreUtilities/utis	CommonEvents.java	<ol style="list-style-type: none"> 1. Contains all common activities. 2. Certain templated common methods 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:

Below activities need to be implemented in the ApiUtil.java file present in src/main/java/rest package.

Sl No.	Summary	Action	Expected Result
1	Create an Appointment with Authorization in Method: createAppointmentWithAuth(String endpoint, Object body)	<ol style="list-style-type: none">1. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/Appointment/AddAppointment2. Include a bearer token for authentication in authorization header.3. Trigger a POST call to the specified endpoint.4. Create an object of type CustomResponse and initialize it with values extracted from the Response object:<ul style="list-style-type: none">• Response• StatusCode (extracted from Response object)• Status (extracted from Response object)• AppointmentId (extracted from Response object)5. Return the CustomResponse object from the method.	<ul style="list-style-type: none">- Returns an object of type CustomResponse containing statusCode, status, appointmentId and the complete Response object.- StatusCode should be 200.- Status should be OK.- AppointmentId should not be null and should not be empty.
2	Get All Applicable Doctors with Authorization in Method: getAllApplicableDoctorsWithAuth(String endpoint, Object body)	<ol style="list-style-type: none">1. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/Visit/AppointmentApplicableDoctors2. Include a bearer token for authentication in authorization header.	<ul style="list-style-type: none">- Returns an object of type CustomResponse containing statusCode, status, results and the complete Response object.- StatusCode should be 200.- Status should be OK.- Results should not be null and should not be empty.

		<ol style="list-style-type: none"> 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: <ul style="list-style-type: none"> • Response • StatusCode (extracted from Response object) • Status (extracted from Response object) • Results of type List<Map<String, Object>> (extracted from Response object) 5. Return the CustomResponse object from the method. 	
3	<p>Cancels an existing appointment with Authorization in Method:</p> <p>cancelAppointmentWithAuth(String endpoint, Object body)</p>	<ol style="list-style-type: none"> 1. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/Appointment/AppointmentStatus?appointmentId=appointmentId&status=cancelled 2. Include a bearer token for authentication in authorization header. 3. Trigger a PUT call to the specified endpoint. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: <ul style="list-style-type: none"> • Response • StatusCode (extracted from Response object) • Status (extracted from Response object) • ResultMessage (extracted from Response object) 5. Return the CustomResponse object from the method. 	<ul style="list-style-type: none"> - Returns an object of type CustomResponse containing statusCode, status, resultMessage and the complete Response object. - StatusCode should be 200. - Status should be OK. - ResultMessage should not be null and should not be empty.
4	<p>Finds if there is any clashing appointment with Authorization in Method:</p> <p>clashAppointmentWithAuth(String endpoint, Object body)</p>	<ol style="list-style-type: none"> 1. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/Appointment/CheckClashingAppointment?patientId=patientId&r 	<ul style="list-style-type: none"> - Returns an object of type CustomResponse containing statusCode, status, resultMessage and the complete Response object. - StatusCode should be 200.

		<p>equestDate=requestDate&performerId=performerId</p> <ol style="list-style-type: none"> 2. Include a bearer token for authentication in authorization header. 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: <ul style="list-style-type: none"> • Response • StatusCode (extracted from Response object) • Status (extracted from Response object) • ResultMessage (extracted from Response object) 5. Return the CustomResponse object from the method. 	<p>- Status should be OK.</p> <p>- ResultMessage should not be null and should not be empty.</p>
5	<p>Searches for a patient using specified query parameters with Authorization in Method:</p> <p>searchPatientWithAuth(String endpoint, Object body)</p>	<ol style="list-style-type: none"> 1. Create a URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/Patient/SearchRegisteredPatient?search=Test 2. Include a bearer token for authentication in authorization header. 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: <ul style="list-style-type: none"> • Response • StatusCode (extracted from Response object) • Status (extracted from Response object) • Results of type List<Map<String, Object>> (extracted from Response object) 5. Return the CustomResponse object from the method. 	<p>- Returns an object of type CustomResponse containing statusCode, status, results and the complete Response object.</p> <p>- StatusCode should be 200.</p> <p>- Status should be OK.</p> <p>- Results should not be null and should not be empty.</p>

6	<p>Retrieves a list of appointments for a specified performer within a given date range with Authorization in Method:</p> <p>bookingListWithAuthInRange(String endpoint, Object body)</p>	<ol style="list-style-type: none"> 1. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/Appointment/Appointments?FromDate=dateFiveDaysBeforeStr&ToDate=currentDateStr&performerId=performerId&status=new 2. Include a bearer token for authentication in authorization header. 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: <ul style="list-style-type: none"> • Response • StatusCode (extracted from Response object) • Status (extracted from Response object) • Results of type List<Map<String, Object>> (extracted from Response object) 5. Return the CustomResponse object from the method. 	<ul style="list-style-type: none"> - Returns an object of type CustomResponse containing statusCode, status, results and the complete Response object. - StatusCode should be 200. - Status should be OK. - Results should not be null and should not be empty.
7	<p>Retrieves the complete list of stock details from the pharmacy with Authorization in Method:</p> <p>AllStockDetailsWithAuth(String endpoint, Object body)</p>	<ol style="list-style-type: none"> 1. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/PharmacyStock/AllStockDetails 2. Include a bearer token for authentication in authorization header. 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: <ul style="list-style-type: none"> • Response • StatusCode (extracted from Response object) • Status (extracted from Response object) 	<ul style="list-style-type: none"> - Returns an object of type CustomResponse containing statusCode, status, results and the complete Response object. - StatusCode should be 200. - Status should be OK. - Results should not be null and should not be empty.

		<ul style="list-style-type: none"> Results of type List<Map<String, Object>> (extracted from Response object) <p>5. Return the CustomResponse object from the method.</p>	
8	<p>Retrieves details of the main store in the pharmacy settings with Authorization in Method:</p> <p>MainStoreDetailsWithAuth(String endpoint, Object body)</p>	<ol style="list-style-type: none"> 1. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/PharmacySettings/MainStore 2. Include a bearer token for authentication in authorization header. 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: <ul style="list-style-type: none"> • Response • StatusCode (extracted from Response object) • Status (extracted from Response object) • Results of type Map<String, Object> (extracted from Response object) 5. Return the CustomResponse object from the method. 	<p>- Returns an object of type CustomResponse containing statusCode, status, results and the complete Response object.</p> <p>- StatusCode should be 200.</p> <p>- Status should be OK.</p> <p>- Results should not be null and should not be empty.</p>
9	<p>Retrieves a list of pharmacy stores and verifies the details of each store with Authorization in Method:</p> <p>PharmacyStoresWithAuth(String endpoint, Object body)</p>	<ol style="list-style-type: none"> 1. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/Dispensary/PharmacyStores 2. Include a bearer token for authentication in authorization header. 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: <ul style="list-style-type: none"> • Response 	<p>- Returns an object of type CustomResponse containing statusCode, status, results and the complete Response object.</p> <p>- StatusCode should be 200.</p> <p>- Status should be OK.</p> <p>- Results should not be null and should not be empty.</p>

		<ul style="list-style-type: none"> • StatusCode (extracted from Response object) • Status (extracted from Response object) • Results of type List<Map<String, Object>> (extracted from Response object) <p>5. Return the CustomResponse object from the method.</p>	
10	<p>Retrieves and verifies patient consumption details with Authorization in Method:</p> <p>PatientConsumption(String endpoint, Object body)</p>	<ol style="list-style-type: none"> 1. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/PatientConsumption/PatientConsumptions 2. Include a bearer token for authentication in authorization header. 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: <ul style="list-style-type: none"> • Response • StatusCode (extracted from Response object) • Status (extracted from Response object) • Results of type List<Map<String, Object>> (extracted from Response object) 5. Return the CustomResponse object from the method. 	<p>- Returns an object of type CustomResponse containing statusCode, status, results and the complete Response object.</p> <p>- StatusCode should be 200.</p> <p>- Status should be OK.</p> <p>- Results should not be null and should not be empty.</p>
11	<p>Activates a pharmacy counter using counter details with Authorization in Method:</p> <p>ActivatePharmCount(String endpoint, Object body)</p>	<ol style="list-style-type: none"> 1. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/Security/ActivatePharmacyCounter?counterId=counterId&counterName=counterName 2. Include a bearer token for authentication in authorization header. 3. Trigger a PUT call to the specified endpoint. 	<p>- Returns an object of type CustomResponse containing statusCode, status, results and the complete Response object.</p> <p>- StatusCode should be 200.</p> <p>- Status should be OK.</p> <p>- Results should not be null and should not be empty.</p>

		<ol style="list-style-type: none"> Create an object of type CustomResponse and initialize it with values extracted from the Response object: <ul style="list-style-type: none"> Response StatusCode (extracted from Response object) Status (extracted from Response object) Results of type Map<String, Object> (extracted from Response object) Return the CustomResponse object from the method. 	
12	<p>Deactivates a pharmacy counter with Authorization in Method:</p> <p>DeactivatePharmCount(String endpoint, Object body)</p>	<ol style="list-style-type: none"> Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/Security/DeactivatePharmacyCounter Include a bearer token for authentication in authorization header. Trigger a PUT call to the specified endpoint. Create an object of type CustomResponse and initialize it with values extracted from the Response object: <ul style="list-style-type: none"> Response StatusCode (extracted from Response object) Status (extracted from Response object) Results of type Map<String, Object> (extracted from Response object) Return the CustomResponse object from the method. 	<ul style="list-style-type: none"> - Returns an object of type CustomResponse containing statusCode, status, results and the complete Response object. - StatusCode should be 200. - Status should be OK. - Results should not be null and should not be empty.
13	<p>Retrieves and verifies the list of appointment applicable departments with Authorization in Method:</p> <p>AppointApplicDept(String endpoint, Object body)</p>	<ol style="list-style-type: none"> Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/Master/AppointmentApplicableDepartments Include a bearer token for authentication in authorization header. 	<ul style="list-style-type: none"> - Returns an object of type CustomResponse containing statusCode, status, results and the complete Response object. - StatusCode should be 200. - Status should be OK. - Results should not be null

		<ol style="list-style-type: none"> 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: <ul style="list-style-type: none"> • Response • StatusCode (extracted from Response object) • Status (extracted from Response object) • Results of type List<Map<String, Object>> (extracted from Response object) 5. Return the CustomResponse object from the method. 	and should not be empty.
14	Retrieves and verifies the list of currently admitted patients with Authorization in Method: admittedPatientData (String endpoint, Object body)	<ol style="list-style-type: none"> 1. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/Admission/AdmittedPatientsData?admissionStatus=admitted 2. Include a bearer token for authentication in authorization header. 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: <ul style="list-style-type: none"> • Response • StatusCode (extracted from Response object) • Status (extracted from Response object) • Results of type List<Map<String, Object>> (extracted from Response object) 5. Return the CustomResponse object from the method. 	<ul style="list-style-type: none"> - Returns an object of type CustomResponse containing statusCode, status, results and the complete Response object. - StatusCode should be 200. - Status should be OK. - Results should not be null and should not be empty.

15	<p>Adds a new currency using the provided data with Authorization in Method:</p> <p>addCurrencyWithAuth(String endpoint, Object body)</p>	<ol style="list-style-type: none"> 1. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/InventorySettings/Currency 2. Include a bearer token for authentication in authorization header. 3. Trigger a POST call to the specified endpoint. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: <ul style="list-style-type: none"> • Response • StatusCode (extracted from Response object) • Status (extracted from Response object) • Results of type Map<String, Object> (extracted from Response object) 5. Return the CustomResponse object from the method. 	<ul style="list-style-type: none"> - Returns an object of type CustomResponse containing statusCode, status, results and the complete Response object. - StatusCode should be 200. - Status should be OK. - Results should not be null and should not be empty.
16	<p>Finds if a patient with the requested phone number already exists with Authorization in Method:</p> <p>findMatchingPatientWithAuth (String endpoint, Object body)</p>	<ol style="list-style-type: none"> 1. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/Patient/MatchingPatients?FirstName=firstName&LastName=lastName&PhoneNumber=phoneNumber&Age=age&Gender=gender&IsInsurance=isInsurance&IMISCode=imisCode 2. Include a bearer token for authentication in authorization header. 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: <ul style="list-style-type: none"> • Response • StatusCode (extracted from Response object) • Status (extracted from Response object) 	<ul style="list-style-type: none"> - Returns an object of type CustomResponse containing statusCode, status, results and the complete Response object. - StatusCode should be 200. - Status should be OK. - Results should not be null and should not be empty.

		<ul style="list-style-type: none"> Results of type List<Map<String, Object>> (extracted from Response object) <p>5. Return the CustomResponse object from the method.</p>	
17	<p>Lists all registered patients and checks if their patient IDs are unique with Authorization in Method:</p> <p>getRegisteredPatientsWithAuth(String endpoint, Object body)</p>	<ol style="list-style-type: none"> Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/Patient/SearchRegisteredPatient?search= Include a bearer token for authentication in authorization header. Trigger a GET call to the specified endpoint. Create an object of type CustomResponse and initialize it with values extracted from the Response object: <ul style="list-style-type: none"> Response StatusCode (extracted from Response object) Status (extracted from Response object) Results of type List<Map<String, Object>> (extracted from Response object) Return the CustomResponse object from the method. 	<p>- Returns an object of type CustomResponse containing statusCode, status, results and the complete Response object.</p> <p>- StatusCode should be 200.</p> <p>- Status should be OK.</p> <p>- Results should not be null and should not be empty.</p>
18	<p>Retrieves and verifies the list of Billing Counters with Authorization in Method:</p> <p>getBillingCountersWithAuth(String endpoint, Object body)</p>	<ol style="list-style-type: none"> Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final URL becomes as follows: https://healthapp.yaksha.com/api/billing/BillingCounters Include a bearer token for authentication in authorization header. Trigger a GET call to the specified endpoint. Create an object of type CustomResponse and initialize it with values extracted from Response object: <ul style="list-style-type: none"> Response 	<p>- Returns an object of type CustomResponse containing statusCode, status, results and the complete Response object.</p> <p>- StatusCode should be 200.</p> <p>- Status should be OK.</p> <p>- Results should not be null and should not be empty.</p>

		<ul style="list-style-type: none"> • StatusCode (extracted from Response object) • Results of type List<Map<String, Object>> (extracted from Response object. • Status (extracted from Response object) <p>5. Return the CustomResponse object from the method.</p>	
--	--	--	--

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.
-

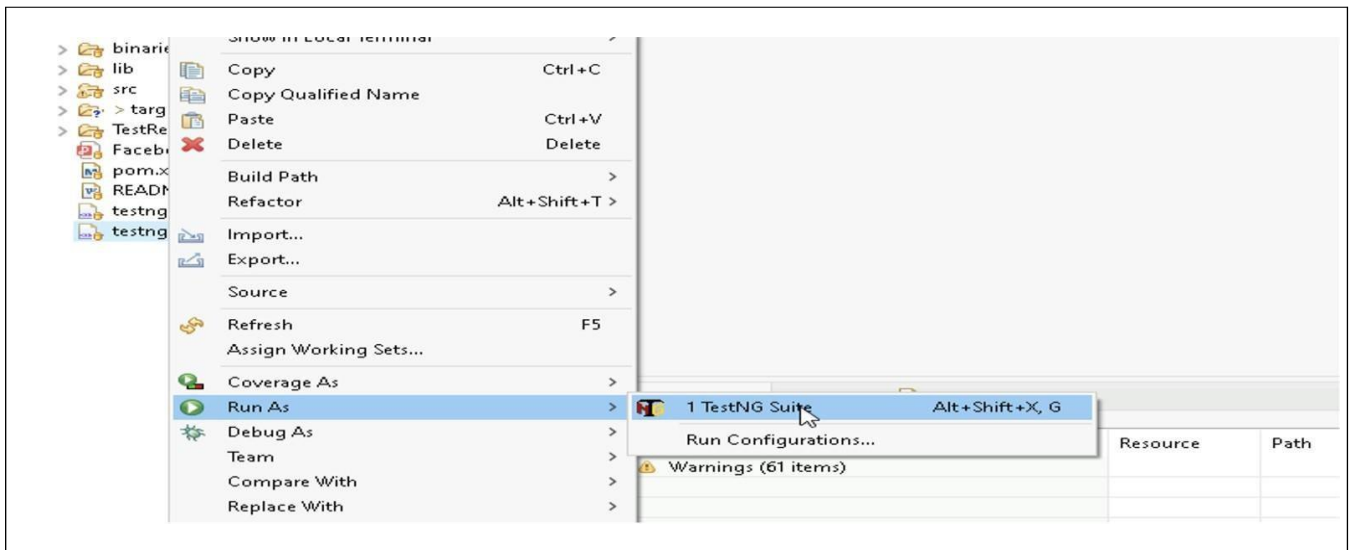
IMPLEMENTATION/FUNCTIONAL REQUIREMENT

1.1 CODE QUALITY/OPTIMIZATIONS

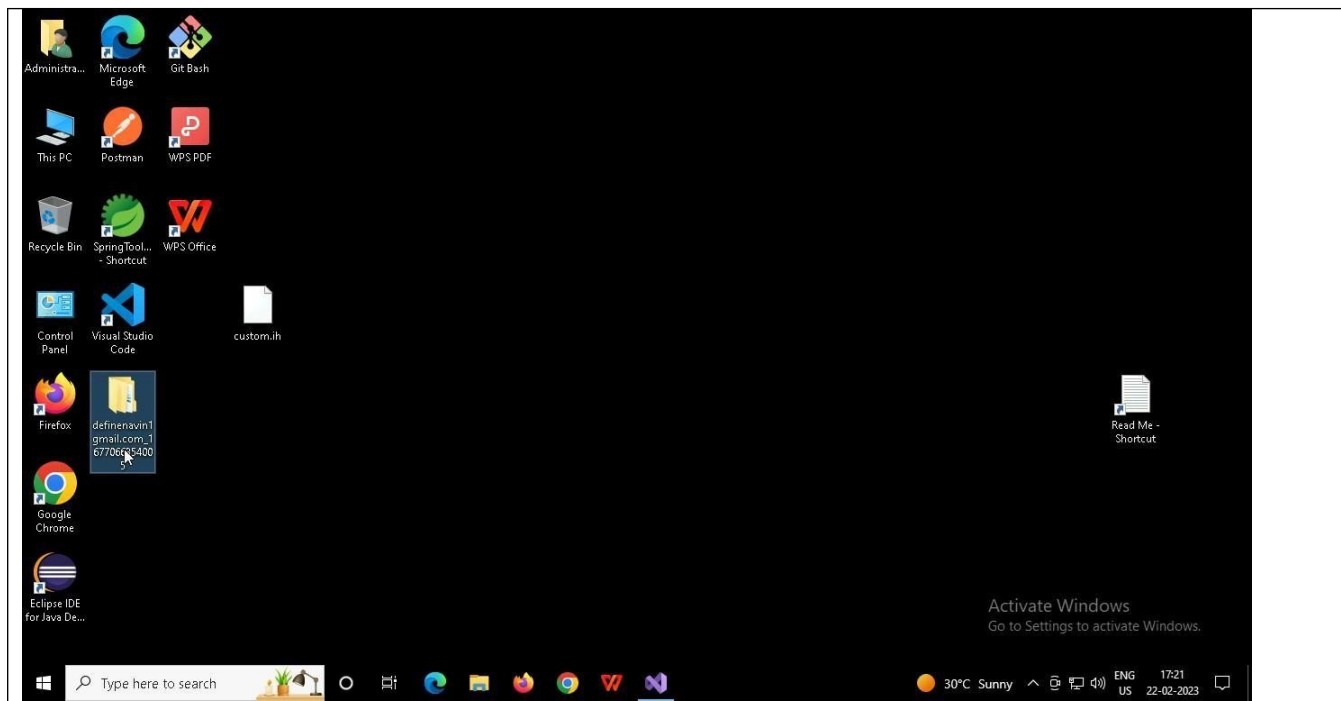
1. Associates should have written clean code that is readable.
2. Associates need to follow SOLID programming principles.

EXECUTION STEPS TO FOLLOW

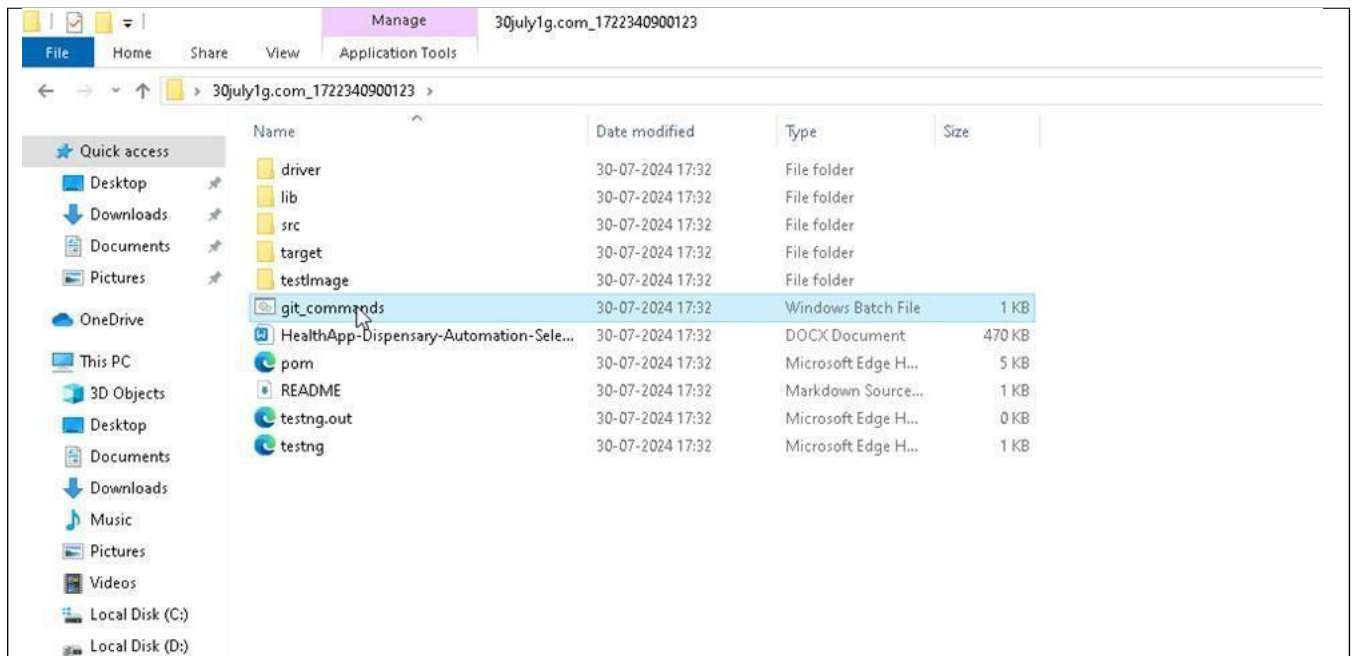
1. **You are mandatorily required to run test cases for applications before final submission. Without which project evaluation will not happen.**
2. **You can launch test cases any time as follows: Right click on testng.xml and run TestNGSuite**



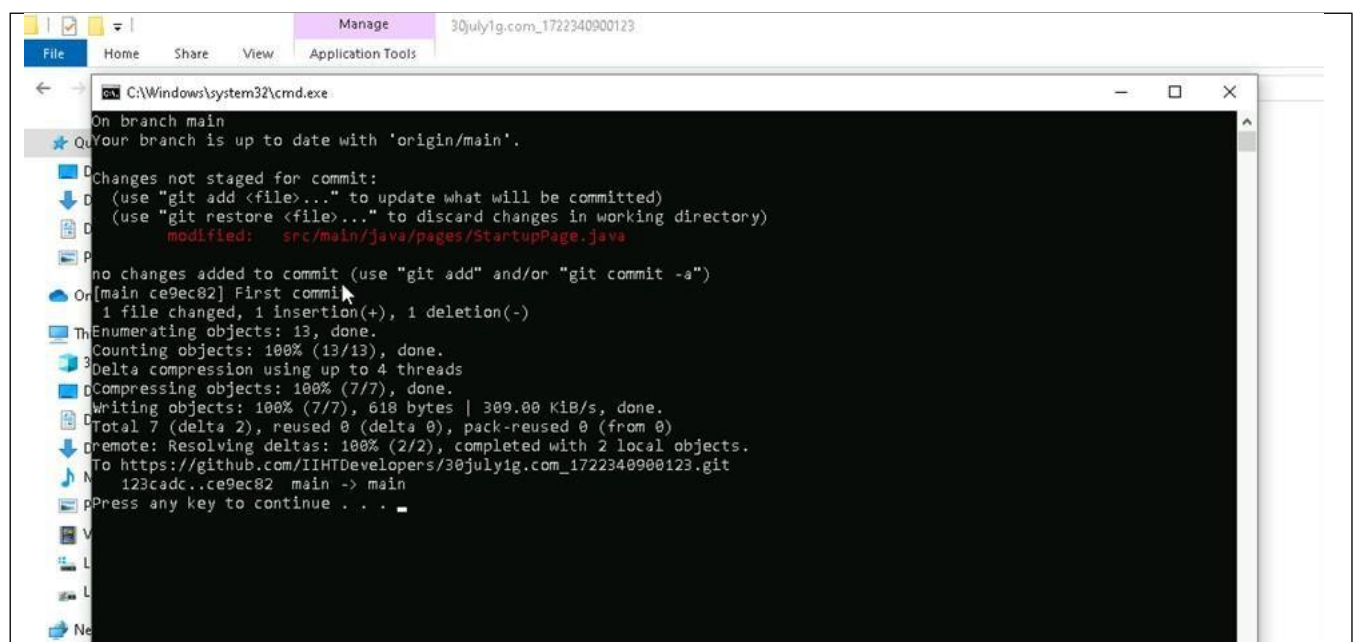
3. Before final submission, you are also required to push your code to GIT. Following are the steps to follow:



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



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



=====

All the Best