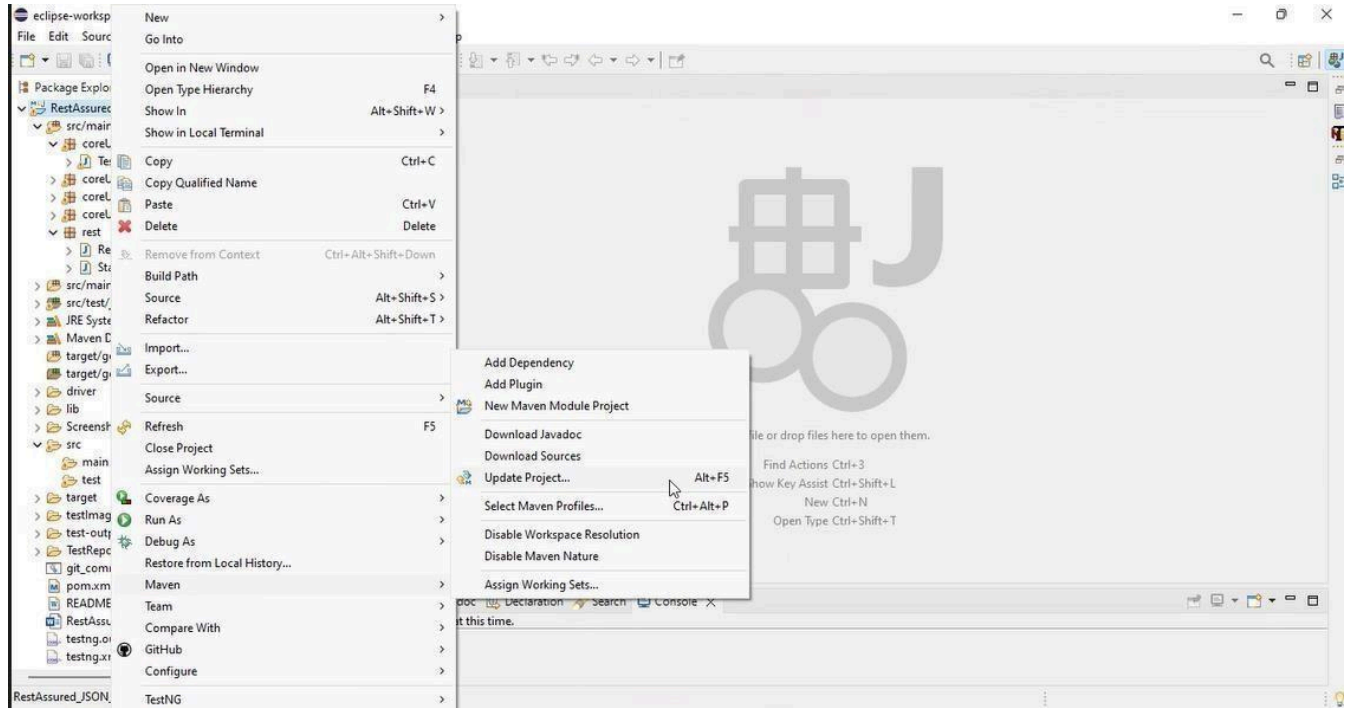# RESTASSURED API AUTOMATION PROJECT

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
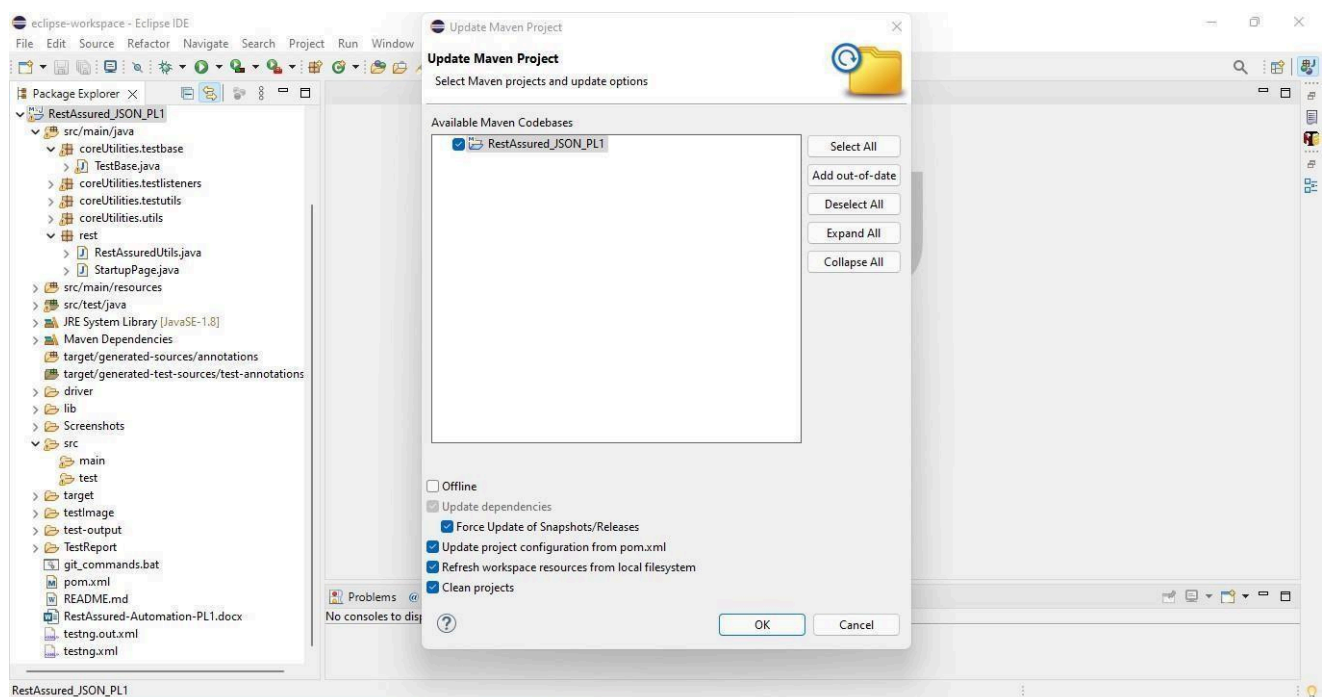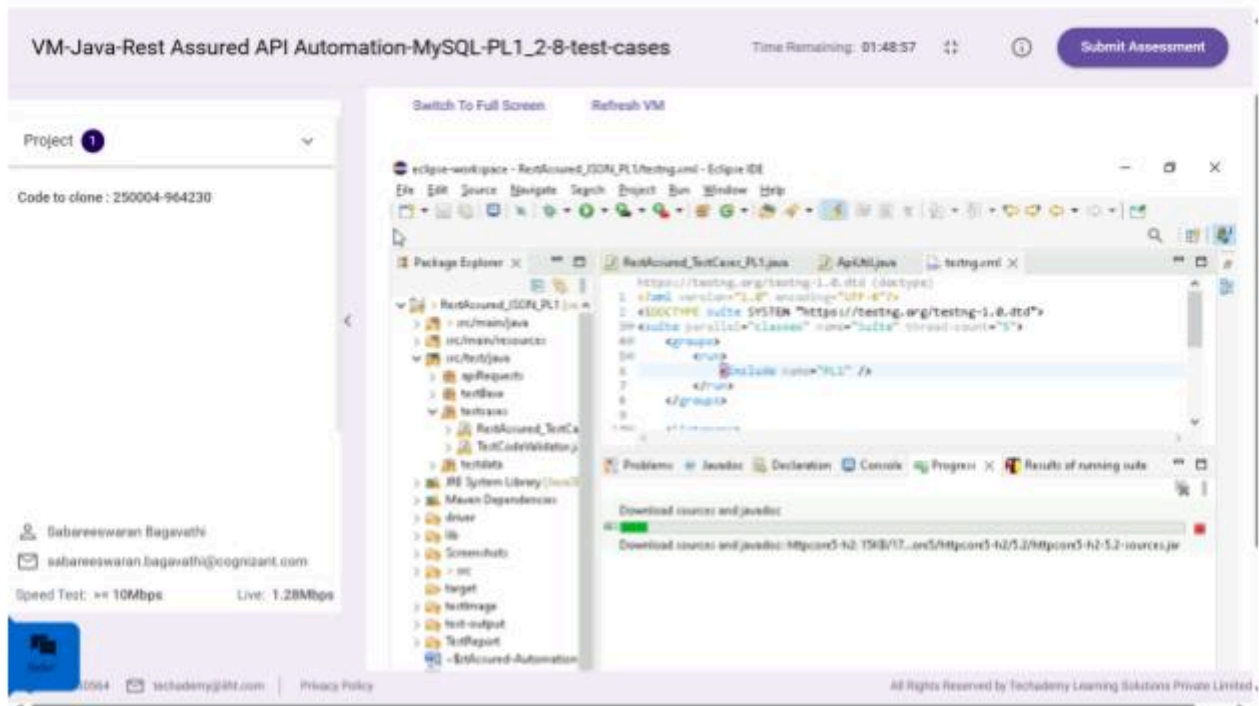
Note: Once all the project dependencies are downloaded you may find certain documentation jar files getting downloaded which may continue for a longer time. This is the known behaviour. This will not block your application development activity. You can ignore it and continue working on your application.

** Before starting any implementation. Please execute this "runner" file present in the cloned folder. This will keep pushing the changes at regular intervals.
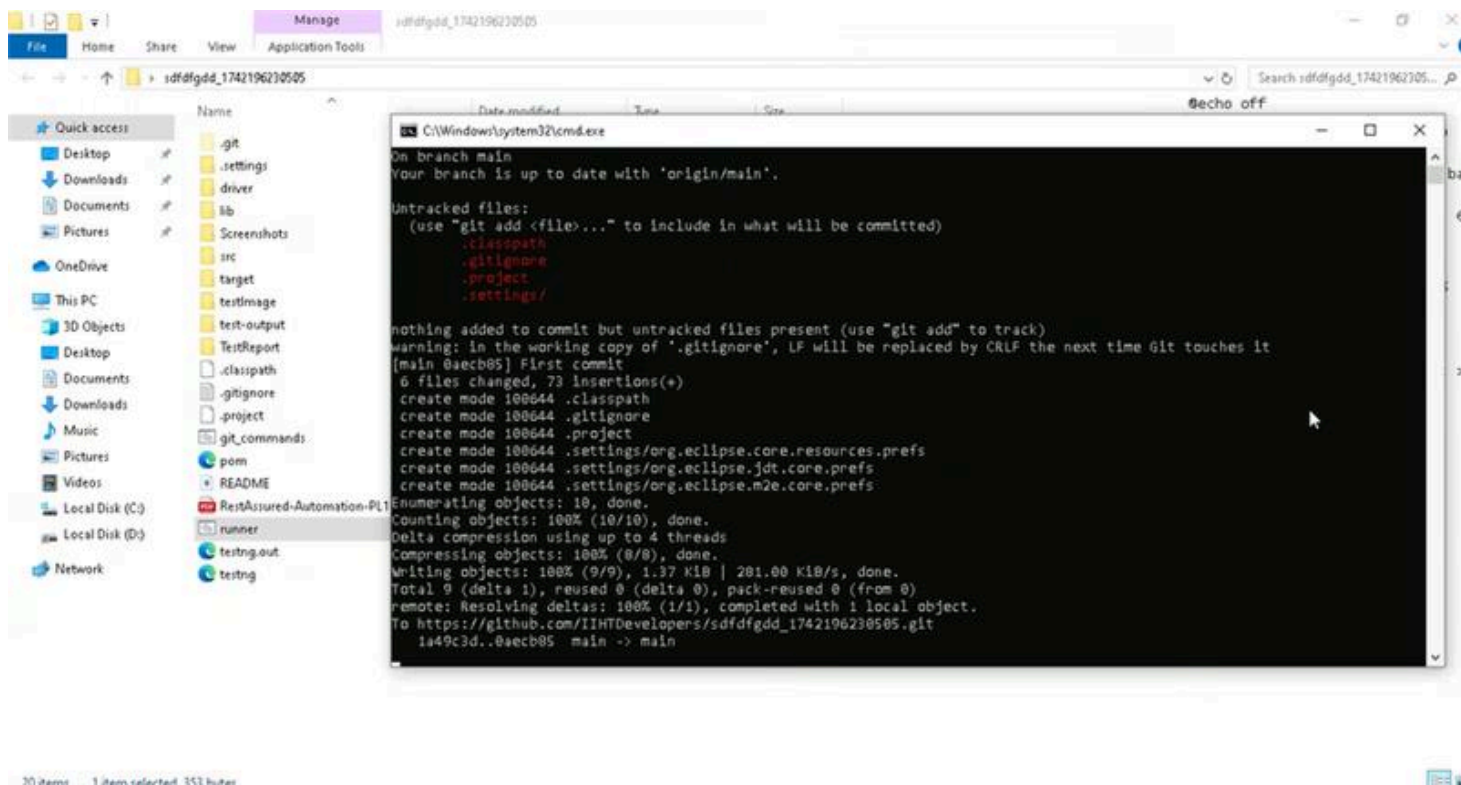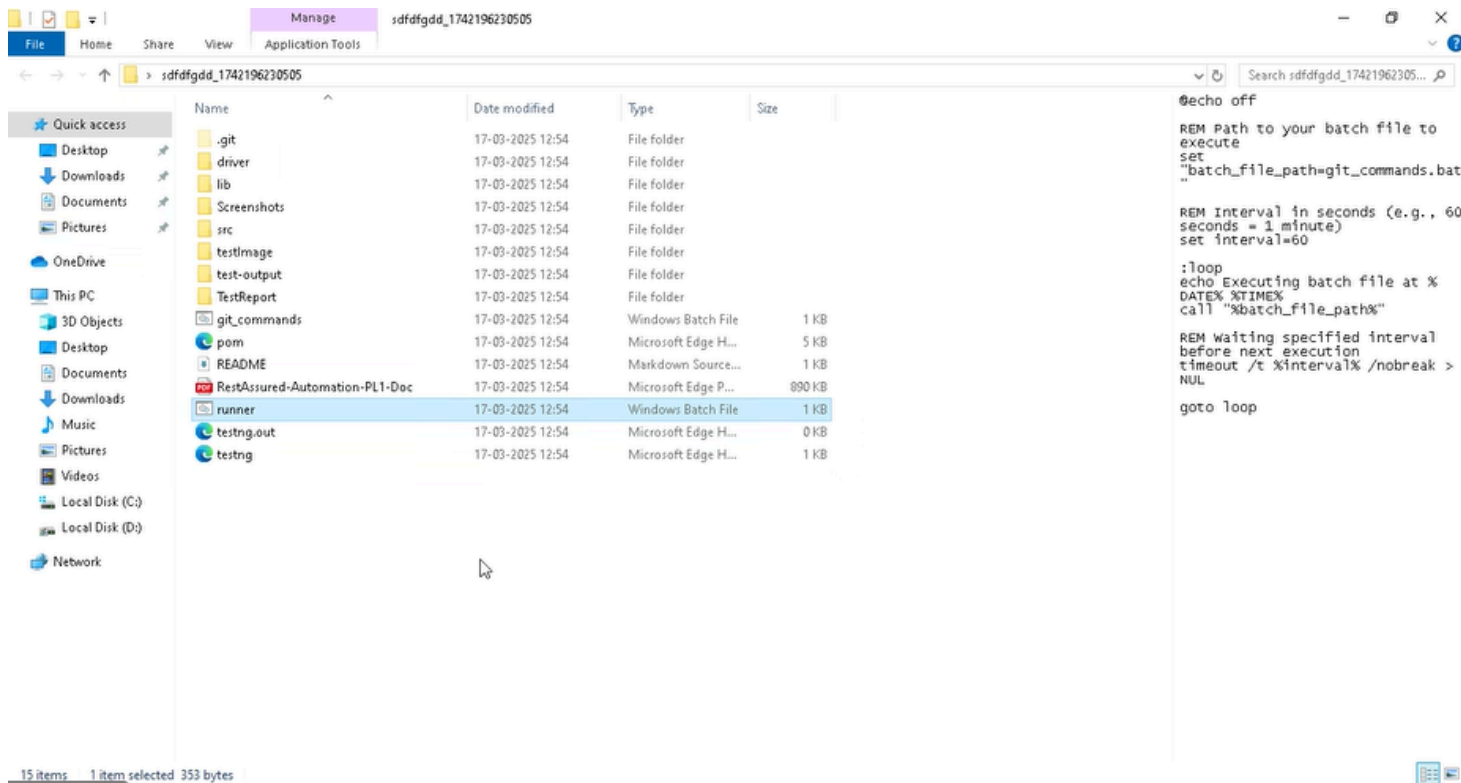
Template Code Structure:

a. Below are the packages and files you will be required to work 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 the very first activity, this will be used to read data 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/java/rest	AuthUtil.java	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 ApiUtil.java class methods.
/src/main/resources/	Config.xlsx	Data present to be used in Implementing functions.
/src/main/java/coreUtilities/utils	CommonEvents.java	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	Retrieve all stocks in Method:
	
	getAllStocks(String endpoint, Object body)
		1. Call the GET on endpoint i.e **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:
		  ● Response
		  ● statusCode (extracted from Response object)
		  ● Status (extracted from Response object)
		  ● Results as List<Map<String, Object>> (extracted from Response object)
		5. Return the CustomResponse object from the method.
			- Returns an object of type CustomResponse containing statusCode, status, Results and the complete Response object.
			
			- StatusCode should be 200.
			
			- Status should be OK.

2	Retrieve all stocks in Method:
	
	getMainStore(String endpoint, Object body)
		1. Call the GET on endpoint i.e **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:
		  ● Response
		  ● statusCode (extracted from Response object)
		  ● Status (extracted from Response object)
		  ● Results as Map<String, Object> (extracted from Response
			- Returns an object of type CustomResponse containing statusCode, status, Results and the complete Response object.
			
			- StatusCode should be 200.
			
			- Status should be OK.

		object)
		5. Return the CustomResponse object from the method.	

3	Retrieve requisition list in Method:
	
	getRequisitionByDateRange(String endpoint, Object body)
		1. Call the GET on endpoint i.e **https://healthapp.yaksha.com/api/ DispensaryRequisition/Dispensary/1?FromDate=" + fromDate + "&ToDate=" + toDate**
		
		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:
		   ● Response
		   ● statusCode (extracted from Response object)
		   ● Status (extracted from Response object)
		   ● Results.requisitionList as List<Map<String, Object>> (extracted from Response object)
		5. Return the CustomResponse object from the method.
			- Returns an object of type CustomResponse containing statusCode, status, Results.requisitionList and the complete Response object.
			
			- StatusCode should be 200.
			
			- Status should be OK.

4	Retrieve patient consumptions in Method:
	
	getPatientConsumptions(String endpoint, Object body)
		1. Call the GET on endpoint i.e **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:
		   ● Response
		   ● statusCode (extracted from Response object)
		   ● Status (extracted from Response object)
		   ● Results as List<Map<String, Object>> (extracted from Response object)
		5. Return the CustomResponse object from the method.
			- Returns an object of type CustomResponse containing statusCode, status, Results and the complete Response object.
			
			- StatusCode should be 200.
			
			- Status should be OK.

| 5 | Retrieve patient consumption info in Method:

getPatientConsumptionInfoByPatientIdAndVisitId(String endpoint, Object body) | 1. Call the GET on endpoint i.e **https://healthapp.yaksha.com/api/PatientConsumption/PatientConsumptionInfo?patientId="+ patientId + "&patientVisitId=" + patientVisitId**

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:
• Response
• statusCode (extracted from Response object)
• Status (extracted from Response object)
• Results.PatientConsumption as Map<String, Object> (extracted from Response object)
5. Return the CustomResponse object from the method. | - Returns an object of type CustomResponse containing statusCode, status, Results.PatientConsumption and the complete Response object.

- StatusCode should be 200.

- Status should be OK. |
| 6 | Search for a Patient with Authorization in Method:

getBillingSchemeBySchemeId(String endpoint, Object body) | 1. Call the GET on endpoint i.e **https://healthapp.yaksha.com/api//PatientConsumption/PharmacyIpBillingScheme?schemeId=" + schemeId**

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:
• Response
• statusCode (extracted from Response object)
• Status (extracted from Response object)
• Results as Map<String, Object> (extracted from Response object)
5. Return the CustomResponse object from the method. | - Returns an object of type CustomResponse containing statusCode, status, Results and the complete Response object.

- StatusCode should be 200.

- Status should be OK. |

| 7 | Search for a Patient with Authorization in Method:

getBillingSummaryByPatientId(String endpoint, Object body) | 1. Call the GET on endpoint i.e **https://healthapp.yaksha.com/api//PharmacySales/PatientBillingSummary?patientId=" + patientId**
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:
   ● Response
   ● statusCode (extracted from Response object)
   ● Status (extracted from Response object)
   ● Results as Map<String, Object> (extracted from Response object)
5. Return the CustomResponse object from the method. | - Returns an object of type CustomResponse containing statusCode, status, Results and the complete Response object.

- StatusCode should be 200.

- Status should be OK. |
| 8 | Search for a Patient with Authorization in Method:

getConsumptionsListOfAPatientById(String endpoint, Object body) | 1. Call the GET on endpoint i.e **https://healthapp.yaksha.com/api//PatientConsumption/ConsumptionsOfPatient?patientId="+patientId+"&patientVisitId="+patientVisitId**
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:
   ● Response
   ● statusCode (extracted from Response object)
   ● Status (extracted from Response object)
   ● Results as List<Map<String, Object>> (extracted from Response object)
5. Return the CustomResponse object from the method. | - Returns an object of type CustomResponse containing statusCode, status, Results and the complete Response object.

- StatusCode should be 200.

- Status should be OK. |

| 9 | Search for a Patient with Authorization in Method:

getReturnConsumptionsList (String endpoint, Object body) | 1. Call the GET on endpoint i.e **https://healthapp.yaksha.com/api/ /PatientConsumption/Returns**

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:
    ● Response
    ● statusCode (extracted from Response object)
    ● Status (extracted from Response object)
    ● Results as List<Map<String, Object>> (extracted from Response object)
5. Return the CustomResponse object from the method. | - Returns an object of type CustomResponse containing statusCode, status, Results and the complete Response object.

- StatusCode should be 200.

- Status should be OK. |
| 10 | Search for a Patient with Authorization in Method:

getDischargedPatients(String endpoint, Object body) | 1. Call the GET on endpoint i.e **https://healthapp.yaksha.com/api/ /Admission/DischargedPatients?admissionStatus=discharged&FromDate=" + fromDate + "&ToDate=" + toDate**

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:
    ● Response
    ● statusCode (extracted from Response object)
    ● Status (extracted from Response object)
    ● Results as List<Map<String, Object>> (extracted from Response object)
5. Return the CustomResponse object from the method. | - Returns an object of type CustomResponse containing statusCode, status, Results and the complete Response object.

- StatusCode should be 200.

- Status should be OK. |

11	Search for a Patient with Authorization in Method:
	
	getFilmTypesInRadiology(String endpoint, Object body)
		1. Call the GET on endpoint i.e **https://healthapp.yaksha.com/api/ /radiology/FilmTypes**
		
		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:
		   ● Response
		   ● statusCode (extracted from Response object)
		   ● Status (extracted from Response object)
		   ● Results as List<Map<String, Object>> (extracted from Response object)
		5. Return the CustomResponse object from the method.
			- Returns an object of type CustomResponse containing statusCode, status, Results and the complete Response object.
			
			- StatusCode should be 200.
			
			- Status should be OK.

12	Search for a Patient with Authorization in Method:
	
	getRequisitionsByOrderStatusAndDateRange(String endpoint, Object body)
		6. Call the GET on endpoint i.e **https://healthapp.yaksha.com/api/ /radiology/Requisitions?reqOrderStatus=" + orderStatus + "&reportOrderStatus=" + reportOrderStatus+"&typeList=" +typeList+"&fromDate="+fromDate + "&toDate=" +toDate**
		
		7. Include a bearer token for authentication in authorization header.
		8. Trigger a GET call to the specified endpoint.
		9. 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)
		   ● Results as List<Map<String, Object>> (extracted from Response object)
		10. Return the CustomResponse object from the method.
			- Returns an object of type CustomResponse containing statusCode, status, Results and the complete Response object.
			
			- StatusCode should be 200.
			
			- Status should be OK.

13	Search for a Patient with Authorization in Method:
	
	getImagingReportsWithStatusAndDateRange(String endpoint, Object body)
		1. Call the GET on endpoint i.e
		https://healthapp.yaksha.com/api//radiology/ImagingReports?reportOrderStatus=" + reportOrderStatus + "&fromDate=" + fromDate+"&toDate="+toDate+ "&typeList=" + typeList
		
		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:
		  ● Response
		  ● statusCode (extracted from Response object)
		  ● Status (extracted from Response object)
		  ● Results as List<Map<String, Object>> (extracted from Response object)
		5. Return the CustomResponse object from the method.
			- Returns an object of type CustomResponse containing statusCode, status, Results and the complete Response object.
			
			- StatusCode should be 200.
			
			- Status should be OK.

14	Search for a Patient with Authorization in Method:
	
	getAdmittedPatientData(String endpoint, Object body)
		1. Call the GET on endpoint i.e
		https://healthapp.yaksha.com/api//Billing/InPatientProvisionalItems?patientId=176&patientVisitId=" + visitId + "&module=" + module
		
		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:
		  ● Response
		  ● statusCode (extracted from Response object)
		  ● Status (extracted from Response object)
		  ● Results as Map<String, Object> (extracted from Response object)
		5. Return the CustomResponse object from the method.
			- Returns an object of type CustomResponse containing statusCode, status, Results and the complete Response object.
			
			- StatusCode should be 200.
			
			- Status should be OK.

| 15 | Search for a Patient with Authorization in Method:

getInPatientProvItems(String endpoint, Object body) | 11. Call the GET on endpoint i.e **https://healthapp.yaksha.com/api//Billing/InPatientProvisionalItems?patientId=176&patientVisitId=147&module=radiology**

12. Include a bearer token for authentication in authorization header.
13. Trigger a GET call to the specified endpoint.
14. 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)
• Results.BillItems as List<Map<String, Object>> (extracted from Response object)
15. Return the CustomResponse object from the method. | - Returns an object of type CustomResponse containing statusCode, status, Results.BillItems and the complete Response object.

- StatusCode should be 200.

- Status should be OK. |

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**
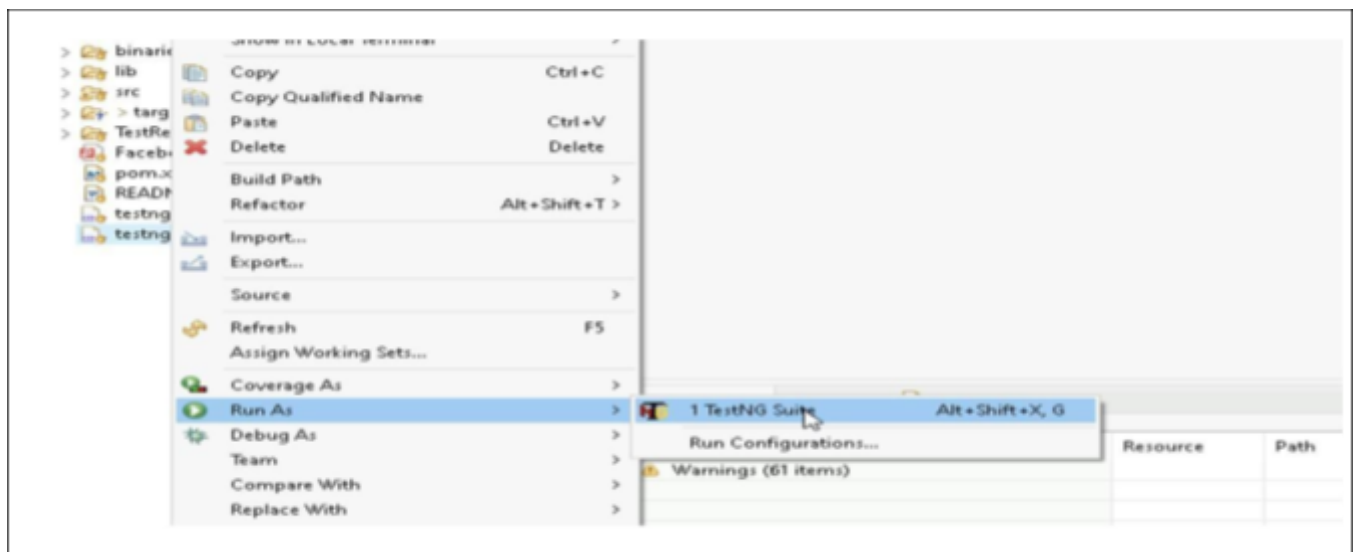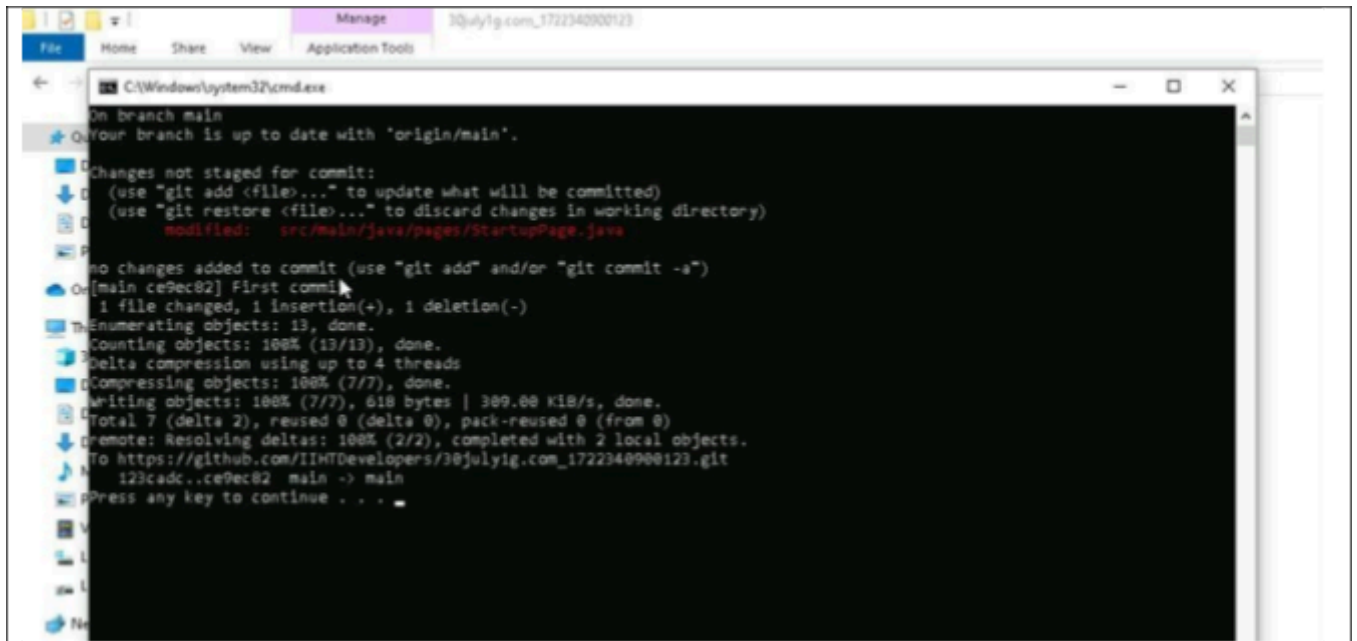

3. **Before final submission, you are also required to push your code to GIT. Following are the steps to follow:**
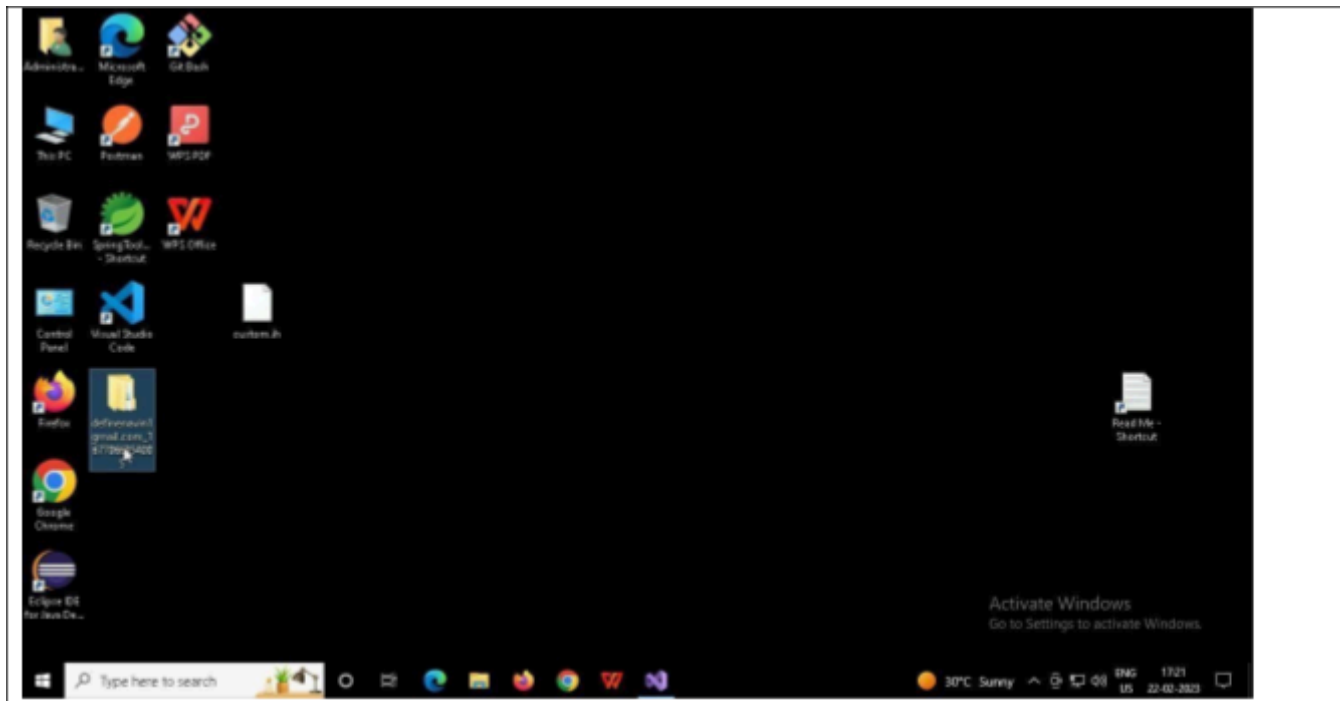
In your project folder, you will find a batch file named git_commands
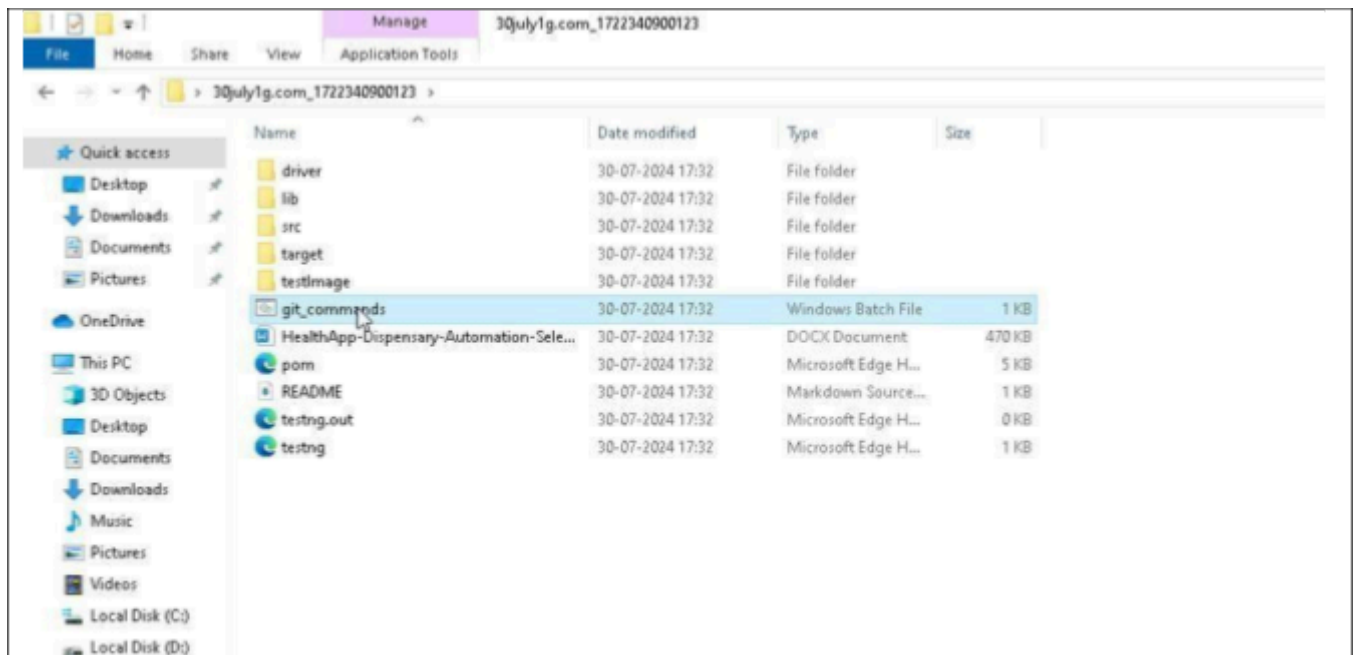
Double-click the batch file to run it. It will run the commands to push your code to GIT.

```
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 commit
 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