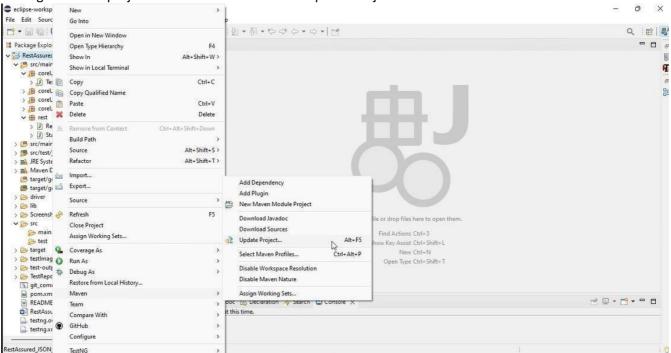
RESTASSURED API AUTOMATION PROJECT

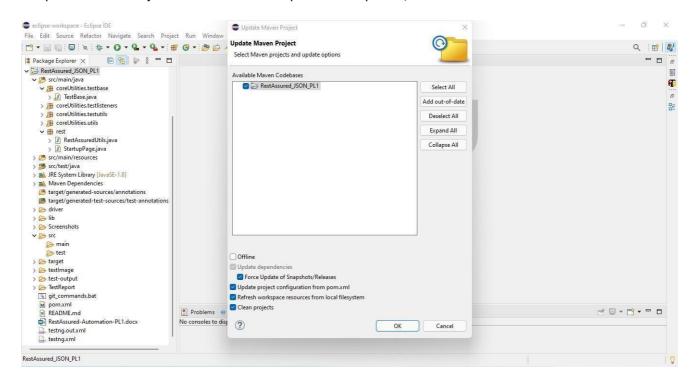
Pre-requisite:

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/utils/	File Operations. java	 Contains methods to read from excel file.
		 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
	A 1111111	excel.
/src/main/java/rest	ApiUtil.java	 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	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/utils	Common Events. java	 Contains all common activities. Certain templated common methods declared here. 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.

SI	Summary	Action	Expected Result
No.			
No. 1	Create an Appointment with Authorization in Method: createAppointmentWithAuth(String endpoint, Object body)	 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/AddAppointment/addAppointment/addAppointment/addAppointment/addAppointment/addAppointment/addAppointment/addAppointment/addAppointment/addAppointment/addAppointment/authentication in authorization header. Trigger a POST call to the specified endpoint. Create an object of type CustomResponse and initialize it with values extracted from the Response object: Response StatusCode (extracted from Response object) AppointmentId (extracted from Response object) Return the CustomResponse object from the method. 	 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: getAllApplicableDoctorsWith Auth(String endpoint, Object body)	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/AppointmentApplicable Doctors 2. Include a bearer token for authentication in authorization header.	 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.

		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 of type List<map<string, object="">></map<string,> 	
		(extracted from Response	
		object)	
		5. Return the CustomResponse object	
		from the method.	
3	Cancels an existing	1. Create an URL by combining the	- Returns an object of type
	appointment with	BASE_URL (already declared) and	CustomResponse
	Authorization in Method:	path parameter (provided as an	containing statusCode,
		argument in method). The final	status, resultMessage and
	cancelAppointmentWithAuth(URL becomes as follows:	the complete Response
	String endpoint, Object body)	https://healthapp.yaksha.com/a	object.
		pi/Appointment/AppointmentSta tus?appointmentId=	- StatusCode should be 200.
		appointmentId&status=cancelled	- Statuscode silodid be 200.
		2. Include a bearer token for	- Status should be OK.
		authentication in authorization	
		header.	- ResultMessage should not
		3. Trigger a PUT call to the specified	be null and should not be
		endpoint.	empty.
		4. Create an object of type	
		CustomResponse and initialize it	
		with values extracted from the	
		Response object:	
		ResponseStatusCode (extracted from	
		Response object)	
		Status (extracted from	
		•	
		Response object)	
		Response object) • ResultMessage (extracted	
		 ResultMessage (extracted from Response object) 	
		 ResultMessage (extracted from Response object) Return the CustomResponse object 	
		 ResultMessage (extracted from Response object) Return the CustomResponse object from the method. 	
4	Finds if there is any clashing	 ResultMessage (extracted from Response object) Return the CustomResponse object from the method. Create an URL by combining the 	- Returns an object of type
4	Finds if there is any clashing appointment with	 ResultMessage (extracted from Response object) Return the CustomResponse object from the method. Create an URL by combining the BASE_URL (already declared) and 	CustomResponse
4	Finds if there is any clashing	 ResultMessage (extracted from Response object) Return the CustomResponse object from the method. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an 	CustomResponse containing statusCode,
4	Finds if there is any clashing appointment with Authorization in Method:	 ResultMessage (extracted from Response object) Return the CustomResponse object from the method. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final 	CustomResponse containing statusCode, status, resultMessage and
4	Finds if there is any clashing appointment with Authorization in Method: clashAppointmentWithAuth(St	 ResultMessage (extracted from Response object) Return the CustomResponse object from the method. 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: 	CustomResponse containing statusCode, status, resultMessage and the complete Response
4	Finds if there is any clashing appointment with Authorization in Method:	 ResultMessage (extracted from Response object) Return the CustomResponse object from the method. Create an URL by combining the BASE_URL (already declared) and path parameter (provided as an argument in method). The final 	CustomResponse containing statusCode, status, resultMessage and

	T		7
		equestDate=requestDate&perfor merId=performerId 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) • ResultMessage (extracted from Response object) 5. Return the CustomResponse object from the method.	- Status should be OK. - ResultMessage should not be null and should not be empty.
5	Searches for a patient using specified query parameters with Authorization in Method: searchPatientWithAuth(String endpoint, Object body)	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/a pi/Patient/SearchRegisteredPatie nt?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: Response StatusCode (extracted from Response object) Status (extracted from Response object) Results of type List <map<string, object="">> (extracted from Response object) S. Return the CustomResponse object from the method.</map<string,>	- 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.

pe
de,
he
t.
_
00.
ull
pe
þe
de,
he
t.
00.
ull

		 Results of type List<map<string, object="">></map<string,>	
8	Retrieves details of the main store in the pharmacy settings with Authorization in Method: MainStoreDetailsWithAuth(String endpoint, Object body)	 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 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: Response StatusCode (extracted from Response object) Status (extracted from Response object) Results of type Map<string, object=""> (extracted from Response object)</string,> 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. Results should not be null and should not be empty.
9	Retrieves a list of pharmacy stores and verifies the details of each store with Authorization in Method: PharmacyStoresWithAuth(String endpoint, Object body)	 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 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: Response 	 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.

		StatusCode (extracted from	
		Response object)	
		Status (extracted from	
		Response object)	
		Results of type	
		List <map<string, object="">></map<string,>	
		(extracted from Response	
		object)	
		5. Return the CustomResponse object	
		from the method.	
10	Retrieves and verifies patient	1. Create an URL by combining the	- Returns an object of type
10	consumption details with	BASE_URL (already declared) and	CustomResponse
	Authorization in Method:	path parameter (provided as an	containing statusCode,
	rationzation in witchoa.	argument in method). The final	status, results and the
	PatientConsumption(String	URL becomes as follows:	complete Response object.
	endpoint, Object body)	https://healthapp.yaksha.com/a	
		pi/PatientConsumption/PatientC	- StatusCode should be 200.
		onsumptions	
		2. Include a bearer token for	- Status should be OK.
		authentication in authorization	
		header.	- Results should not be null
		3. Trigger a GET call to the specified	and should not be empty.
		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 of type 	
		List <map<string, object="">></map<string,>	
		(extracted from Response	
		object)	
		5. Return the CustomResponse object	
		from the method.	
11	Activates a pharmacy counter	1. Create an URL by combining the	- Returns an object of type
	using counter details with	BASE_URL (already declared) and	CustomResponse
	Authorization in Method:	path parameter (provided as an	containing statusCode,
		argument in method). The final	status, results and the
	ActivatePharmCount(String	URL becomes as follows:	complete Response object.
	endpoint, Object body)	https://healthapp.yaksha.com/a	Chatus Cada abasidaba 200
		pi/Security/ActivatePharmacyCou	- StatusCode should be 200.
		nter?counterId=counterId&count	Status should be OK
		erName=counterName	- Status should be OK.
		2. Include a bearer token for	Posulte chould not be mill
		authentication in authorization	- Results should not be null
		header.	and should not be empty.
		3. Trigger a PUT call to the specified	
		endpoint.	

12		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 of type Map <string, object=""> (extracted from Response object) Response object) Response object Status (extracted from Response object) Response object) Response object</string,>	Dotume an object of twee
12	Deactivates a pharmacy counter with Authorization in Method: DeactivatePharmCount(String endpoint, Object body)	 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: Response StatusCode (extracted from Response object) Status (extracted from Response object) Results of type Map<string, object=""> (extracted from Response object)</string,> 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. Results should not be null and should not be empty.
13	Retrieves and verifies the list of appointment applicable departments with Authorization in Method: AppointApplicDept(String endpoint, Object body)	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/Master/AppointmentApplicableDepartments 2. Include a bearer token for authentication in authorization header.	 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

		3.	Trigger a GET call to the specified endpoint.	and should not be empty.
		١,	·	
		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 of type	
			List <map<string, object="">></map<string,>	
			(extracted from Response	
			•	
		_	object)	
		5.	Return the CustomResponse	
		+	ect from the method.	
14	Retrieves and verifies the list	1. (Create an URL by combining the	- Returns an object of type
	of currently admitted patients		BASE_URL (already declared) and	CustomResponse
	with Authorization in Method:		path parameter (provided as an	containing statusCode,
			argument in method). The final	status, results and the
	admittedPatientData		URL becomes as follows:	complete Response object.
	(String endpoint, Object body)		https://healthapp.yaksha.com/a	
			pi/Admission/AdmittedPatientsD	- StatusCode should be 200.
			ata?admissionStatus=admitted	
		2.	Include a bearer token for	- Status should be OK.
			authentication in authorization	
			header.	- Results should not be null
		3.	Trigger a GET call to the specified	and should not be empty.
			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 of type	
			List <map<string, object="">></map<string,>	
			(extracted from Response	
			object)	
		5.	Return the CustomResponse	
		obje	ect from the method.	

15	Adds a new currency using the	1. Create an URL by combining the	- Returns an object of type
	provided data with	BASE_URL (already declared) and	CustomResponse
	Authorization in Method:	path parameter (provided as an	containing statusCode,
		argument in method). The final	status, restults and the
	addCurrencyWithAuth(String	URL becomes as follows:	complete Response object.
	endpoint, Object body)	https://healthapp.yaksha.com/a	
		pi/InventorySettings/Currency	- StatusCode should be 200.
		2. Include a bearer token for	
		authentication in authorization	- Status should be OK.
		header.	5 1 1 1 1
		3. Trigger a POST call to the specified	- Results should not be null
		endpoint.	and should not be empty.
		 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 of type Map<string,< li=""> </string,<>	
		Object> (extracted from	
		Response object)	
		5. Return the CustomResponse object	
		from the method.	
16	Finds if a patient with the	1. Create an URL by combining the	- Returns an object of type
	requested phone number	BASE_URL (already declared) and	CustomResponse
	already exists with	path parameter (provided as an	containing statusCode,
	Authorization in Method:	argument in method). The final URL becomes as follows:	status, results and the complete Response object.
	findMatchingPatientWithAuth	https://healthapp.yaksha.com/a	complete Response object.
		iittps.//iieaitiiapp.yaksiia.coiii/a	
ĺ	_	ni/Patient/MatchingPatients?First	- StatusCode should be 200
	(String endpoint, Object body)	pi/Patient/MatchingPatients?First Name=firstName&LastName=last	- StatusCode should be 200.
	_	Name=firstName&LastName=last	
	_	• • • •	StatusCode should be 200.Status should be OK.
	_	Name=firstName&LastName=last Name&PhoneNumber=phoneNum	
	_	Name=firstName&LastName=last Name&PhoneNumber=phoneNum ber&Age=age&Gender=gender&Is	- Status should be OK.
	_	Name=firstName&LastName=last Name&PhoneNumber=phoneNum ber&Age=age&Gender=gender&Is Insurance=isInsurance&IMISCode =imisCode 2. Include a bearer token for	Status should be OK.Results should not be null
	_	Name=firstName&LastName=last Name&PhoneNumber=phoneNum ber&Age=age&Gender=gender&Is Insurance=isInsurance&IMISCode =imisCode 2. Include a bearer token for authentication in authorization	Status should be OK.Results should not be null
	_	Name=firstName&LastName=last Name&PhoneNumber=phoneNum ber&Age=age&Gender=gender&Is Insurance=isInsurance&IMISCode =imisCode 2. Include a bearer token for authentication in authorization header.	Status should be OK.Results should not be null
	_	Name=firstName&LastName=last Name&PhoneNumber=phoneNum ber&Age=age&Gender=gender&Is Insurance=isInsurance&IMISCode =imisCode 2. Include a bearer token for authentication in authorization header. 3. Trigger a GET call to the specified	Status should be OK.Results should not be null
	_	Name=firstName&LastName=last Name&PhoneNumber=phoneNum ber&Age=age&Gender=gender&Is Insurance=isInsurance&IMISCode =imisCode 2. Include a bearer token for authentication in authorization header. 3. Trigger a GET call to the specified endpoint.	Status should be OK.Results should not be null
	_	Name=firstName&LastName=last Name&PhoneNumber=phoneNum ber&Age=age&Gender=gender&Is Insurance=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	Status should be OK.Results should not be null
	_	Name=firstName&LastName=last Name&PhoneNumber=phoneNum ber&Age=age&Gender=gender&Is Insurance=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	Status should be OK.Results should not be null
	_	Name=firstName&LastName=last Name&PhoneNumber=phoneNum ber&Age=age&Gender=gender&Is Insurance=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	Status should be OK.Results should not be null
	_	Name=firstName&LastName=last Name&PhoneNumber=phoneNum ber&Age=age&Gender=gender&Is Insurance=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:	Status should be OK.Results should not be null
	_	Name=firstName&LastName=last Name&PhoneNumber=phoneNum ber&Age=age&Gender=gender&Is Insurance=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: • Response	Status should be OK.Results should not be null
	_	Name=firstName&LastName=last Name&PhoneNumber=phoneNum ber&Age=age&Gender=gender&Is Insurance=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: • Response • StatusCode (extracted from	Status should be OK.Results should not be null
	_	Name=firstName&LastName=last Name&PhoneNumber=phoneNum ber&Age=age&Gender=gender&Is Insurance=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: • Response • StatusCode (extracted from Response object)	Status should be OK.Results should not be null
	_	Name=firstName&LastName=last Name&PhoneNumber=phoneNum ber&Age=age&Gender=gender&Is Insurance=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: • Response • StatusCode (extracted from	Status should be OK.Results should not be null

		 Results of type List<map<string, object="">></map<string,>	
17	Lists all registered patients and checks if their patient IDs are unique with Authorization in Method: getRegisteredPatientsWithAut h(String endpoint, Object body)	 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/a pi/Patient/SearchRegisteredPatie nt?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: Response StatusCode (extracted from Response object) Results of type List<map<string, object="">> (extracted from Response object)</map<string,> 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. - Results should not be null and should not be empty.
18	Retrieves and verifies the list of Billing Counters with Authorization in Method: getBillingCountersWithAuth(String endpoint, Object body)	 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: Response 	 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.

 StatusCode (extracted from Response object) Results of type List<map<string, Object>> (extracted from Response object.</map<string, Status (extracted from Response object) 	
5. Return the CustomResponse object from the method.	

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.

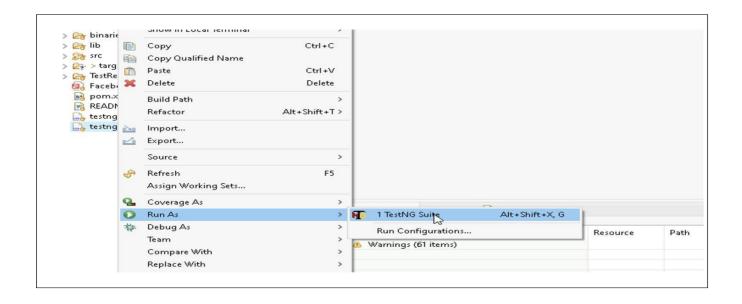
IMPLEMENTATION/FUNCTIONAL REQUIREMENT

1.1 CODE QUALITY/OPTIMIZATIONS

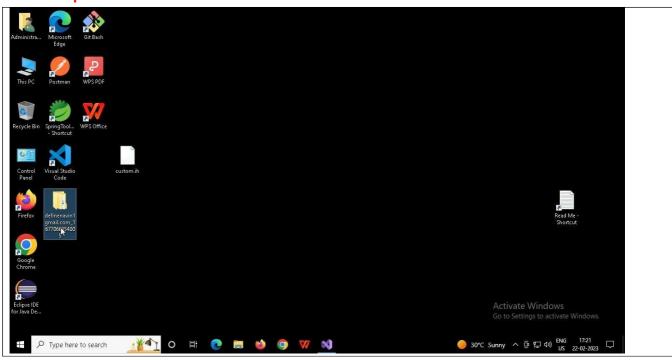
- 1. Associates should have written clean code that is readable.
- 2. Associates need to follow SOLID programming principles.

EXECUTION STEPS TO FOLLOW

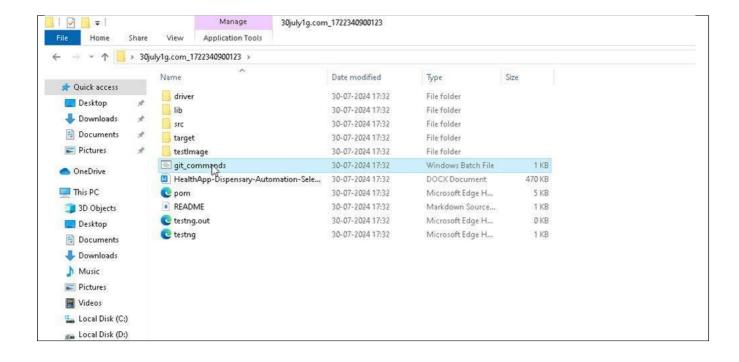
- 1. You are mandatorily required to run test cases for applications before final submission. Without which project evaluation will not happen.
- 2. You can launch test cases any time as follows: Right click on testng.xml and run TestNGSuite



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.

```
| | | = |
                                                                                                      Manage
                                                                                                                                             30july1g.com_1722340900123
                      Home
                                              Share
                                                                    View
                                                                                            Application Tools
                       C:\Windows\system32\cmd.exe
                                                                                                                                                                                                                                                                                                                                                                                                * Ou 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)
        no changes added to commit (use "git add" and/or "git commit -a")

or [main ce9ec82] First commit

1 file changed, 1 insertion(+), 1 deletion(-)

The Enumerating objects: 13, done.

Counting objects: 180% (13/13), done.

30 elta compression using up to 4 threads

Compressing objects: 180% (7/7), done.

Writing objects: 180% (7/7), 618 bytes | 389.80 KiB/s, done.

Total 7 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)

remote: Resolving deltas: 180% (2/2), completed with 2 local objects.

To https://github.com/IIHTDevelopers/30july1g.com_1722348988123.git

123cadc..ce9ec82 main -> main

press any key to continue . . .
        1
        7
        in Ne
```

All the Best