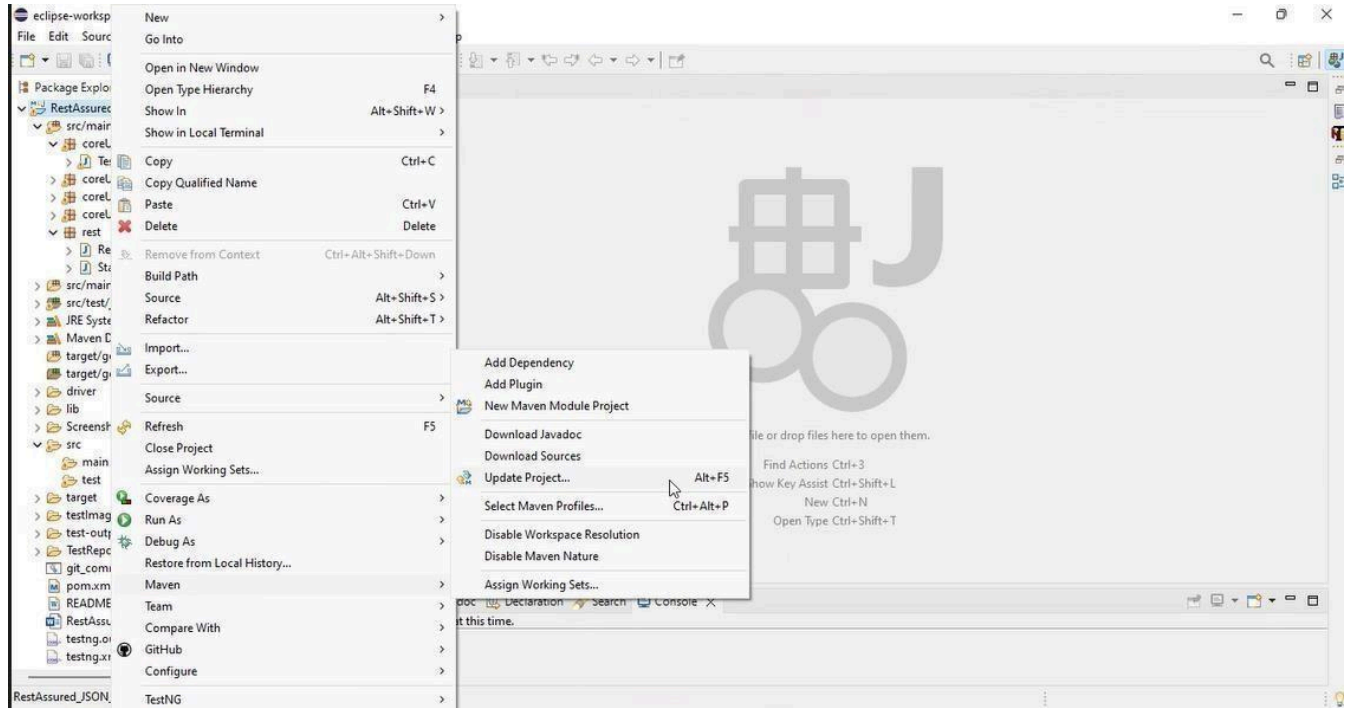


REST ASSURED API AUTOMATION PROJECT

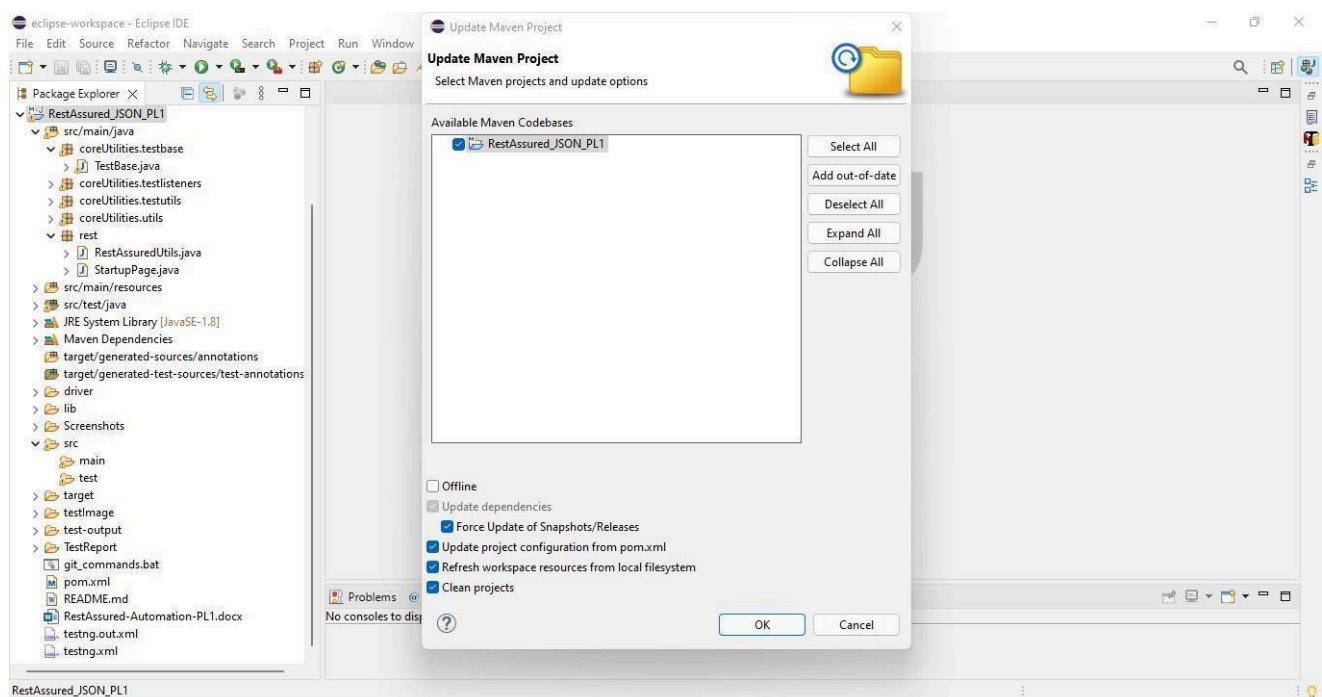
Pre-requisite:

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 “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/utills/	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/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.

SI No.	Summary	Action	Expected Result
1	Retrieve all departments in Method: <code>getAllDepartments(String endpoint, Object body)</code>	<ol style="list-style-type: none">1. Call the GET on endpoint i.e <code>https://healthapp.yaksha.com/api/AssetReports/GetAllDepartments</code>2. Include an authorization header for the request.3. Validate the response contains:<ul style="list-style-type: none">- DepartmentId- DepartmentName4. Ensure all these fields are not null.5. Validate the request indicates successful retrieval of departments.6. Validate that:<ol style="list-style-type: none">a. The response status code is 200 (OK).b. The Status field in the response is "OK".	Returns a 200 OK status with the list of all departments in the "Results" field, containing non-null DepartmentId and DepartmentName.
2	Retrieve all items in Method: <code>getAllItems(String endpoint, Object body)</code>	<ol style="list-style-type: none">1. Call the GET on endpoint i.e <code>https://healthapp.yaksha.com/api/AssetReports/GetAllItems</code>2. Include an authorization header for the request.3. Validate the response contains:<ul style="list-style-type: none">- ItemId- ItemName4. Ensure all these fields are not null.5. Validate the request indicates successful retrieval of items.6. Validate that:<ol style="list-style-type: none">a. The response status code is 200 (OK).b. The Status field in the response is "OK".	Returns a 200 OK status with the list of all items in the "Results" field, containing non-null ItemId and ItemName.

3	<p>Retrieve incentive summary report in Method:</p> <p><code>getIncentiveSummaryReport(String endpoint, Object body)</code></p>	<ol style="list-style-type: none"> 1. Call the GET on endpoint i.e <code>https://healthapp.yaksha.com/BillingReports/INCTV_DocterSummary?FromDate="+fromDate+"&ToDate="+toDate+"&IsRefferalOnly="+isRefferal</code> 2. Add an authorization header to the request. 3. Validate the response contains: <ul style="list-style-type: none"> - JsonData.PrescriberName - JsonData.PrescriberId - JsonData.DocTotalAmount - JsonData.TDSAmount - JsonData.NetPayableAmount 4. Ensure all these fields are not null. 5. Validate the request indicates successful retrieval of the incentive summary report. 6. Validate that: <ol style="list-style-type: none"> a. The response status code is 200 (OK). b. The Status field in the response is "OK". 	<p>Returns a 200 OK status with the incentive summary report in the "Results" field.</p>
4	<p>Retrieve incentive referral summary report in Method:</p> <p><code>getIncReffSummReport(String endpoint, Object body)</code></p>	<ol style="list-style-type: none"> 1. Call the GET on endpoint i.e <code>https://healthapp.yaksha.com/BillingReports/INCTV_DocterSummary?FromDate="+IncFromDate+"&ToDate="+IncToDate+"&IsRefferalOnly="+isRefferal</code> 2. Add the required authorization header. 3. Validate the response contains: <ul style="list-style-type: none"> - JsonData.PrescriberName - JsonData.PrescriberId - JsonData.DocTotalAmount - JsonData.TDSAmount - JsonData.NetPayableAmount 4. Ensure these fields are not null. 5. Validate the request indicates successful retrieval of the incentive referral summary report. 6. Validate that: <ol style="list-style-type: none"> a. The response status code is 200 (OK). b. The Status field in the response is "OK". 	<p>Returns a 200 OK status with the incentive referral summary report in the "Results" field.</p>

5	<p>Retrieve hospital income incentive report in Method:</p> <p><code>getHospIncIncReport(String endpoint, Object body)</code></p>	<ol style="list-style-type: none"> 1. Call the GET on endpoint i.e <code>https://healthapp.yaksha.com/Reporting/HospitalIncomeIncentiveReport?FromDate="+IncFromDate+"&ToDate="+IncToDate+"&ServiceDepartments="+ServiceDepartments</code> 2. Add the required authorization header. 3. Validate the response contains: <ul style="list-style-type: none"> - ServiceDepartmentId - ServiceDepartmentName - NetSales - ReferralCommission - GrossIncome - OtherIncentive - HospitalNetIncome 4. Ensure these fields are not null. 5. Validate the request indicates successful retrieval of the hospital income incentive report. 6. Validate that: <ol style="list-style-type: none"> a. The response status code is 200 (OK). b. The Status field in the response is "OK". 	<p>Returns a 200 OK status with the hospital income incentive report in the "Results" field.</p>
6	<p>Retrieve employee bill items in Method:</p> <p><code>getEmpBillItem(String endpoint, Object body)</code></p>	<ol style="list-style-type: none"> 1. Call the GET on endpoint i.e <code>https://healthapp.yaksha.com/api/Incentive/EmployeeBillItems?employeeId="+EmployeeId</code> 2. Add the necessary authorization header. 3. Validate the response contains: <ul style="list-style-type: none"> - EmployeeIncentiveInfol - EmployeeId - FullName - TDSPercent - EmpTDSPercent - IsActive - EmployeeBillItemsMap 4. Ensure these fields are not null. 5. Validate the request indicates successful retrieval of employee bill items. 6. Validate that: <ol style="list-style-type: none"> a. The response status code is 200 (OK). b. The Status field in the 	<p>Returns a 200 OK status with the employee bill items in the "Results" field.</p>

		response is "OK".	
7	<p>Retrieve inventory fiscal years in Method:</p> <p>getInvntryFiscalYrs(String endpoint, Object body)</p>	<ol style="list-style-type: none"> 1. Call the GET on endpoint i.e https://healthapp.yaksha.com/api/Inventory/InventoryFiscalYears 2. Add the required authorization header. 3. Validate the response contains: <ul style="list-style-type: none"> - FiscalYearId - FiscalYearName - StartDate - EndDate - IsActive 4. Ensure these fields are not null. 5. Validate the request indicates successful retrieval of inventory fiscal years. 6. Validate the response to ensure: <ol style="list-style-type: none"> a. The status code is 200 (OK). b. The Status field is "OK". 	<p>Returns a 200 OK status with the inventory fiscal years in the "Results" field.</p>
8	<p>Activate inventory in Method:</p> <p>getActInventory(String endpoint, Object body)</p>	<ol style="list-style-type: none"> 1. Call the GET on endpoint i.e https://healthapp.yaksha.com/api/ActivateInventory/ 2. Validate the response contains: <ul style="list-style-type: none"> - StoreId - Name - StoreDescription 3. Ensure these fields are not null. 4. Validate the request indicates successful activation of inventory. 5. Validate the response to ensure: <ol style="list-style-type: none"> a. The Results field contains a StatusCode of "200". b. The Status field is "OK". 	<p>Returns a 200 OK status with the list of activated inventory in the "Results" field, containing non-null fields for StoreId, Name, and StoreDescription.</p>

9	<p>Retrieve inventory subcategories in Method:</p> <p><code>getInvSubCat(String URL, Object body)</code></p>	<ol style="list-style-type: none"> 1. Call the GET on endpoint i.e <code>https://healthapp.yaksha.com/api/Inventory/SubCategories</code> 2. Add the required authorization header. 3. Validate the response contains: <ul style="list-style-type: none"> - SubCategoryId - SubCategoryName 4. Ensure these fields are not null. 5. Validate the request indicates successful retrieval of inventory subcategories. 6. Validate the response to ensure: <ol style="list-style-type: none"> a. The status code is 200 (OK). b. The Status field is "OK". 	<p>Returns a 200 OK status with the inventory subcategories in the "Results" field.</p>
10	<p>Retrieve available item quantity by store ID in Method:</p> <p><code>getAvlQtyByStoreId(String URL, Object body)</code></p>	<ol style="list-style-type: none"> 1. Call the GET on endpoint i.e <code>https://healthapp.yaksha.com/api/Inventory/AvailableQuantityByItemIdAndStoreId?itemId="+reqItemId+"&storeId="+reqStoreId</code> 2. Include required query parameters (<code>itemId</code> and <code>storeId</code>). 3. Add the necessary authorization header. 4. Validate the response contains: <ul style="list-style-type: none"> - ItemId - AvailableQuantity - StoreId 5. Ensure these fields are not null. 6. Validate the request indicates successful retrieval of available item quantity. 7. Validate the response to ensure: <ol style="list-style-type: none"> a. The status code is 200 (OK). b. The Status field is "OK". 	<p>Returns a 200 OK status with the available item quantity details in the "Results" field.</p>

11	<p>Retrieve requisition items by ID in Method:</p> <p><code>getReqItemsById(String URL, Object body)</code></p>	<ol style="list-style-type: none"> 1. Call the GET on endpoint i.e <code>https://healthapp.yaksha.com/api/Inventory/RequisitionItemsForView?requisitionId="+requisitionId</code> 2. Include required query parameter (<code>requisitionId</code>). 3. Ensure an authorization header is included. 4. Validate the response contains the profile details: <ul style="list-style-type: none"> - Requisition.RequisitionNo - Requisition.CreatedByName - Requisition.RequisitionStatus 5. Validate that the <code>RequisitionItems</code> list is not null and contains: <ul style="list-style-type: none"> - ItemName - Code - PendingQuantity - RequisitionItemStatus 6. Validate the request indicates successful retrieval of requisition items. 7. Validate the response to ensure: <ol style="list-style-type: none"> a. The status code is 200 (OK). b. The Status field is "OK". 	<p>Returns a 200 OK status with requisition items in the "Results" field.</p>
12	<p>Track requisition by ID in Method:</p> <p><code>trackReqItemById(String URL, Object body)</code></p>	<ol style="list-style-type: none"> 1. Call the GET on the endpoint i.e., <code>https://healthapp.yaksha.com/api/Inventory/TrackRequisition?requisitionId="+trackByRequisitionId</code> 2. Include required query parameters (<code>requisitionId</code>). 3. Include the authorization header. 4. Validate the response contains: <ul style="list-style-type: none"> - CreatedBy - Status - Dispatchers 5. Validate the <code>Dispatchers</code> list is not null and contains: <ul style="list-style-type: none"> - DispatchId - Name 6. Validate the request indicates successful tracking of the requisition. 7. Validate the response to ensure: <ol style="list-style-type: none"> a. The status code is 200 (OK). b. The Status field is "OK". 	<p>Returns a 200 OK status with requisition tracking details in the "Results" field.</p>

13	<p>Retrieve inventory items by store ID in Method:</p> <p><code>getInvItem(String URL, Object body)</code></p>	<ol style="list-style-type: none"> 1. Call the GET on endpoint i.e <code>https://healthapp.yaksha.com/api/WardSupply/GetInventoryItemsByStoreId/7</code> 2. Ensure an authorization header is included. 3. Validate the response contains: <ul style="list-style-type: none"> - ItemId - StockId - ItemName - AvailableQuantity - Code - ItemType 4. Ensure these fields are not null. 5. Validate the request indicates successful retrieval of inventory items by store ID. 6. Validate the response to ensure: <ol style="list-style-type: none"> a. The status code is 200 (OK). b. The Status field is "OK". 	<p>Returns a 200 OK status with inventory items in the "Results" field.</p>
14	<p>Retrieve most sold medicines in Method:</p> <p><code>getMostSoldMed(String URL, Object body)</code></p>	<ol style="list-style-type: none"> 1. Call the GET on endpoint i.e <code>https://healthapp.yaksha.com/PharmacyDashboard/GetPharmacyDashboardMostSoldMedicine?FromDate="+FromDate+"&ToDate="+IncToDate</code> 2. Include required query parameters (FromDate and ToDate). 3. Add the required authorization header. 4. Validate the response contains: <ul style="list-style-type: none"> - ItemName - SoldQuantity 5. Ensure these fields are not null. 6. Validate the request indicates successful retrieval of most sold medicines. 7. Validate the response to ensure: <ol style="list-style-type: none"> a. The status code is 200 (OK). b. The Status field is "OK". 	<p>Returns a 200 OK status with the list of most sold medicines in the "Results" field, containing non-null fields for ItemName and SoldQuantity.</p>

15	<p>Retrieve substore-wise dispatch values in Method:</p> <p><code>getSubDisp(String URL, Object body)</code></p>	<ol style="list-style-type: none"> 1. Call the GET on endpoint i.e <code>https://healthapp.yaksha.com/PharmacyDashboard/GetPharmacyDashboardSubstoreWiseDispatchValue?FromDate="+FromDate+"&ToDate="+IncToDate</code> 2. Ensure an authorization header is included. 3. Validate the response contains: <ul style="list-style-type: none"> - Name - TotalDispatchValue 4. Ensure these fields are not null. 5. Validate the request indicates successful retrieval of substore-wise dispatch values. 6. Validate the response to ensure: <ul style="list-style-type: none"> a. The status code is 200 (OK). b. The Status field is "OK". 	<p>Returns a 200 OK status with substore-wise dispatch values in the "Results" field, containing non-null fields for Name and TotalDispatchValue.</p>
16	<p>Retrieve active suppliers in Method:</p> <p><code>getActSupp(String URL, Object body)</code></p>	<ol style="list-style-type: none"> 1. Call the GET on endpoint i.e <code>https://healthapp.yaksha.com/api/PharmacySettings/ActiveSuppliers</code> 2. Ensure an authorization header is included. 3. Validate the response contains: <ul style="list-style-type: none"> - SupplierId - SupplierName 4. Ensure these fields are not null. 5. Validate the request indicates successful retrieval of active suppliers. 6. Validate the response to ensure: <ul style="list-style-type: none"> a. The status code is 200 (OK). b. The Status field is "OK". 	<p>Returns a 200 OK status with the list of active suppliers in the "Results" field, containing non-null fields for SupplierId and SupplierName.</p>

17	Retrieve units of measurement in Method: getMeasureUnits(String URL, Object body)	<ol style="list-style-type: none"> 1. Call the GET on endpoint i.e https://healthapp.yaksha.com/api/PharmacySettings/UnitOfMeasurements 2. Ensure an authorization header is included. 3. Validate the response contains: - UOMId - UOMName 4. Ensure these fields are not null. 5. Validate the request indicates successful retrieval of units of measurement. 6. Validate the response to ensure: <ol style="list-style-type: none"> a. The status code is 200 (OK). b. The Status field is "OK". 	Returns a 200 OK status with the list of units of measurement in the "Results" field, containing non-null fields for UOMId and UOMName.
18	Retrieve sales categories in Method: getSalesCat(String URL, Object body)	<ol style="list-style-type: none"> 1. Call the GET on endpoint i.e https://healthapp.yaksha.com/api/PharmacySettings/SalesCategories 2. Ensure an authorization header is included. 3. Validate the response contains: - SalesCategoryId - Name 4. Ensure these fields are not null. 5. Validate the request indicates successful retrieval of sales categories. 6. Validate the response to ensure: <ol style="list-style-type: none"> a. The status code is 200 (OK). b. The Status field is "OK". 	Returns a 200 OK status with the list of sales categories in the "Results" field, containing non-null fields for SalesCategoryId and Name.

NOTE: If these are need to be added in the rows then we need to add on all

Assert.assertNotNull(results, "Results field should not be null.");

Assert.assertFalse(results.isEmpty(), "Results list should not be empty.");

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

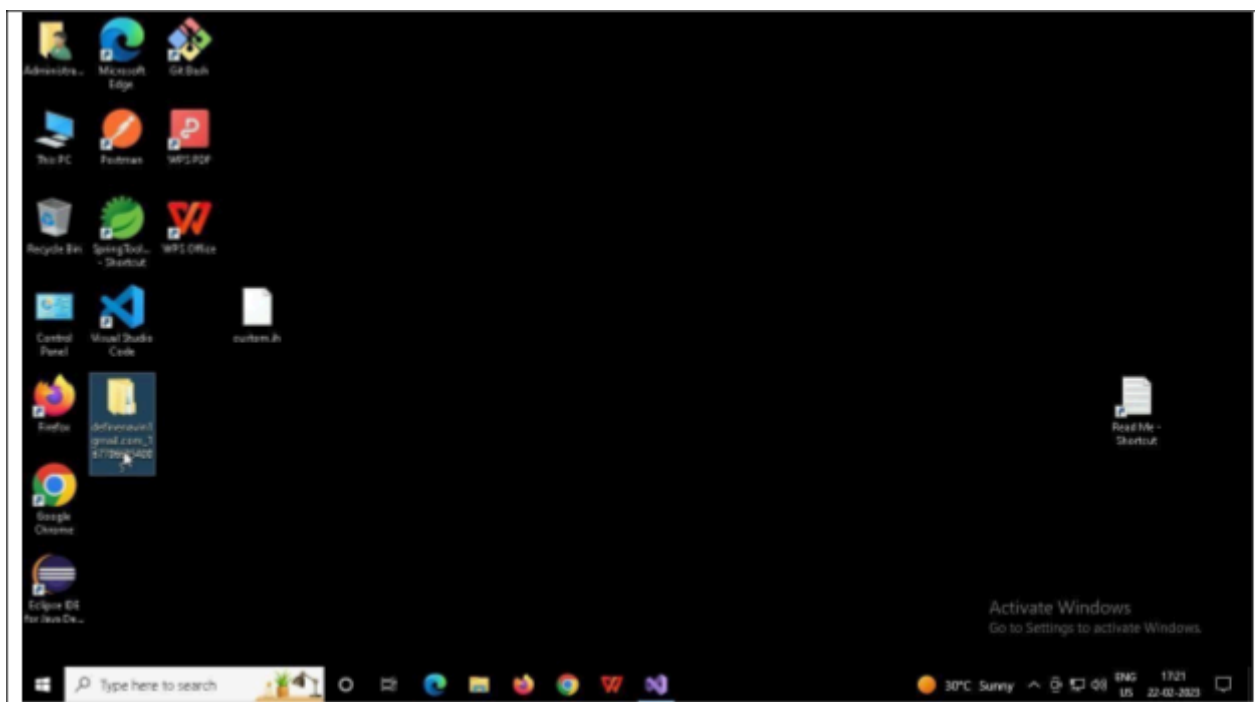
1. Associates should have written clean code that is readable.
2. Associates need to follow SOLID programming principles.

EXECUTION STEPS TO FOLLOW

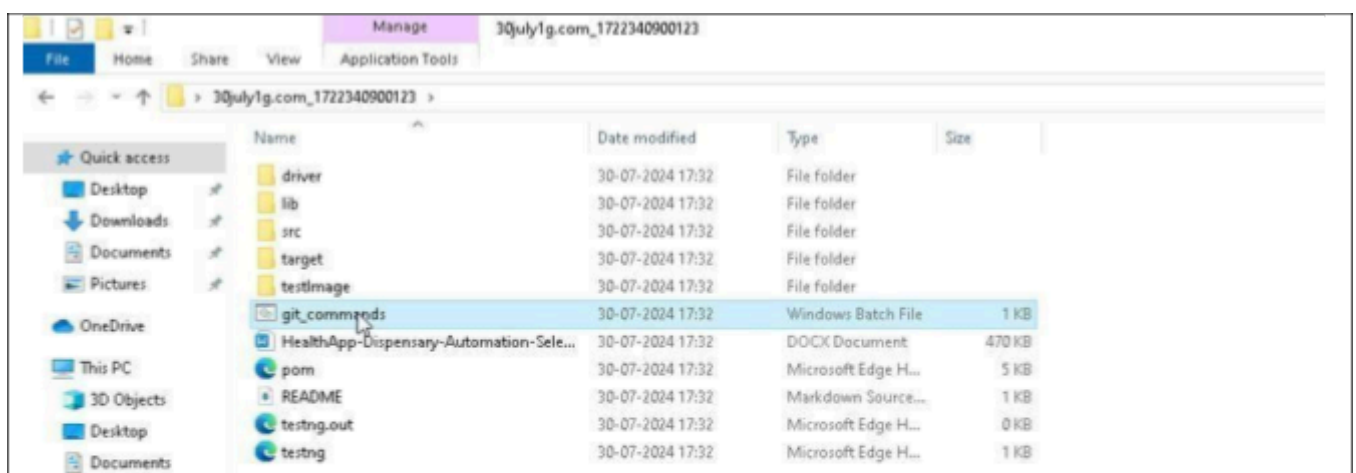
1. **You are mandatorily required to run test cases for applications before final submission. Without which project evaluation will not happen.**
2. **You can launch test cases any time as follows: Right click on testng.xml and run TestNGSuite**



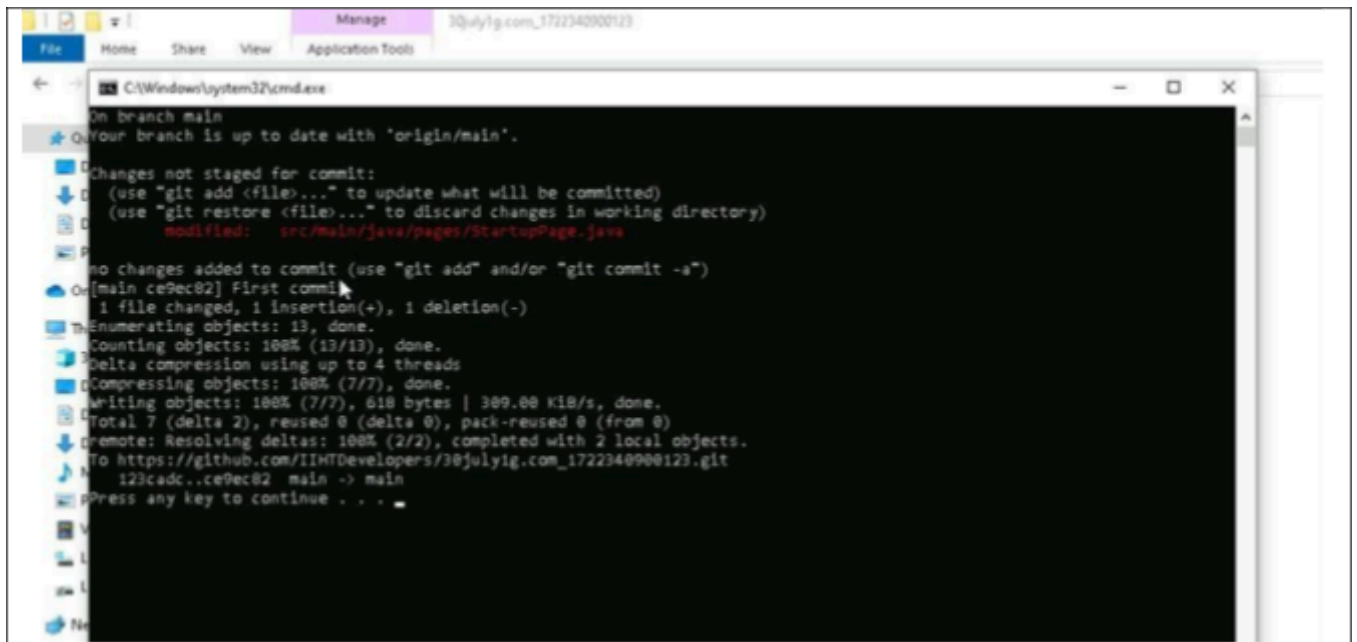
3. Before final submission, you are also required to push your code to GIT. Following are the steps to follow:



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.



```
C:\Windows\system32\cmd.exe
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 -s")
[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/30julyig.com_1722340900123.git
   123cadc..ce9ec82  main -> main
Press any key to continue . . .
```

=====

All the Best