Exercise 1 :

Consider the following phones :

Galaxy S25 Ultra   Galaxy S25 | S25+   Galaxy Z Fold6   Galaxy Z Flip6   Galaxy S24 FE

Find the prices in the Samsung web site and for each size of the phone (128gb,256gb)

Then go to amazon and find if the price matches or not

Take a screen shot in Samsung and amazon websites

IF the price does not match – log it

Find the star rating for each of the model in amazon

Find how may 5 star review and 4 star reviews .. screen shot of the same

The tests have to be data driven and with best practices

If you are able to implement parallel, a bonus point for the same

 Reporting should be good and have screen shots for each step

Guidelines - Expected to develop a testng framework with datadriven approach along with page object model where test data to be read the test data from external source ex: excel, csv etc, followed by best practices & scalable frameworks along with assertions and logging for each test step and detailed reporting in allure with screenshots attached. Note: No hardcoding is allowed while reading the test data or in the methods for output generation

Exercise 2 :

Set up google auth token using the following

https://explorer.apis.google.com/credentials_faq.html

once done

create a new calendar

refer https://developers.google.com/calendar/api/v3/reference/calendars/insert

Store the response

List the calendars And verify the created calendar is part of the list
(refer: https://developers.google.com/calendar/api/v3/reference/calendarList/list)

Create an event in the same calendar (refer:
https://developers.google.com/calendar/api/v3/reference/events/insert)

Verify Event Is created (use Get call)

Update the event (https://developers.google.com/calendar/api/v3/reference/events/patch) with more
attendees or change the time
Or https://developers.google.com/calendar/api/v3/reference/events/update

List the events in the calendar and verify that the event is created

Update the name of the calendar to new calendar

Verify the events don't change

Make the calendar sharable https://developers.google.com/calendar/api/v3/reference/acl/insert

Verify its shared

Create a new event and verify the event is shared

Delete the calendar

Verify its deleted

Use any framework of your choice..the methods created have to be reusable

Additional marks if it is following best practices

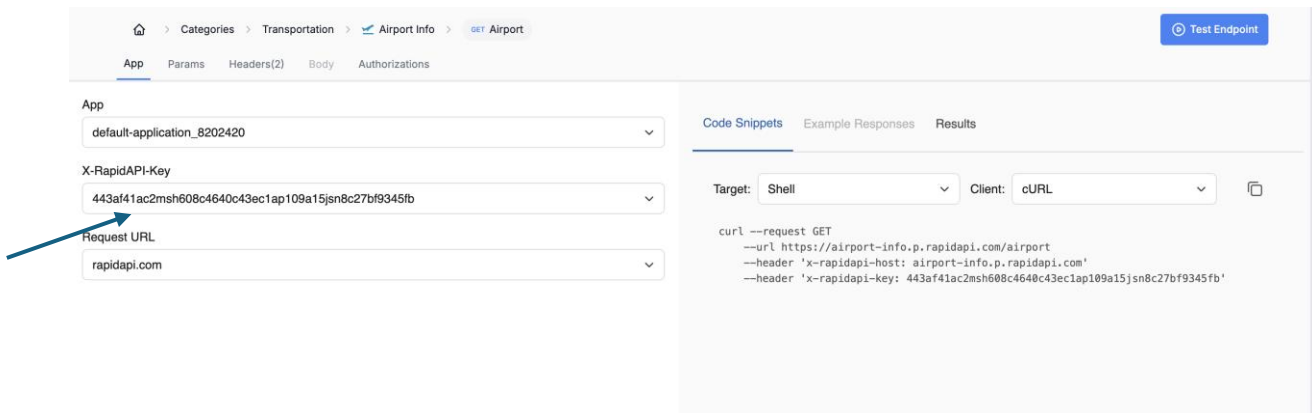Reporting should have request and response for each action

If failed, assertions need to be present

Expected to develop a testng framework with datadriven approach along with page object model where test data to be read the test data from external source ex: excel, csv etc, followed by best practices & scalable frameworks along with assertions and logging for each test step and detailed reporting in allure with screenshots attached. Note: No hardcoding is allowed while reading the test data or in the methods for output generation

Exercise 3:

Goto the following url https://rapidapi.com/Active-api/api/airport-info

Go to play ground and get an api key

For the following url https://airport-info.p.rapidapi.com/airport?iata=HYD

For the following airports, get the airport info

Hyd,blr,del,BOM,MAA,GOI, AMD, COK, PNQ,LKO,CCU

Get the info and also print the name of the airport ,latitude and longitude

Get the icao code and verify

https://airport-info.p.rapidapi.com/airport?icao=VOHS

Gives the same out as that with the iata code

Take the pincode and do a google search for the name of airport, verify pin code matches

Get the airport website and verify that it is correct by going to the website, take a screen shot of the same

Add reporting as needed , each step should have proper logging and attachments

Failures should have proper screen shot and appropriate message

Expected to develop a testng framework with datadriven approach along with page object model where test data to be read the test data from external source ex: excel, csv etc, followed by best practices & scalable frameworks along with assertions and logging for each test step and detailed reporting in allure with screenshots attached. Note: No hardcoding is allowed while reading the test data or in the methods for output generation