Phase-3 End Project WriteUp:

GitHub URL: [Kirans12345/Phase3EndProjectSL (github.com)]

Postman Assignment:

1. To automate functionalities using https://petstore.swagger.io/ REST API services
2. Tools Used

    - **PostMan**
    - **Java 1.8+**
    - **Maven**
    - **Rest Assured Maven dependency version 4.5.1**
    - **TestNG Maven dependency version 7.1.0**
    - **Hamcrest Maven dependency**
    - **Newman for Postman**
    - **JMeter**

3. **Postman Assignment_001:**
    - Inside a postman created a get,post,delete methods
    - URL: https://petstore.swagger.io/v2/pet
    - Created a CSV File in the folder for petID and petName
    - Validated petID
    - Deleted petID using the url
4. **Postman Assignment_002:**
    - Created a PUT call having service URL = https://petstore.swagger.io/v2/pet
    - Created a global variable of this URL where **testURL** is the variable name
    - Created 3 test Environments as DEV, QA, PROD
    - Validated id = 20021 in response
    - Validated response = 200
    - Validated status value, if it is changing as per environment in JSON response
5. **Postman Assignment_003:**
    - Created a GET call with URL: https://petstore.swagger.io/v2/user/Uname001
    - Created a global variable for Uname001
    - Validated response as 200 in Postman
    - Validated username = Uname001 in JSON response
    - Validated email = Positive@Attitude.com in JSON response
    - Validated userStatus = 1 in JSON response
6. **Postman Assignment_004:**
    - Created a GET call with URL: https://petstore.swagger.io/v2/pet/findByStatus

- Created a Postman params as status = available and if after hitting the URL, response status = 200, then validated for all pet details that their response status = available
- Repeated the same for sold
- Repeated the same for pending

7. **Postman Assignment_005:**
    - Created a GET call with URL: https://petstore.swagger.io/v2/user/logout
    - Validated response as 200 in Postman
    - Validated code = 200 in response
    - Validated message = OK in response

REST Assured Assignment:

REST Assured Assignment001:

1. Open Eclipse, Created a Maven project
2. Added Maven dependency for TestNG and REST Assured
3. Created a TestNG test
4. **POST CALL**
    - URL: https://petstore.swagger.io/v2/pet
    - Created a class and method inside it for POST request using restAssured.
    - PetID is parameterized
    - validated response code
    - Validated that PetID from response code should be same as request PetID
5. **GET CALL**
    - URL: https://petstore.swagger.io/v2/pet/2003
    - Inside the same class created method inside it for GET request using restAssured.
    - petID is parameterised
    - Validated response code as 200
    - Validated that response JSON body contains keys status and id
    - Validated status value is available using restAssured
6. **DELETE CALL**
    - Used Hamcrest assertion
    - Created a method to delete the user using the url

REST Assured Assignment003:

1. Created a PUT call having service URL = https://petstore.swagger.io/v2/pet
2. Create three test environments as DEV, QA, PROD
3. validation is done using Hamcrest.
4. Validated response as 200
5. Validated id = 20021 in response

REST Assured Assignment003:

1. Created a TestNG test and implemented the scenario
2. Created a GET call
3. URL: https://petstore.swagger.io/v2/user/Uname001
4. validation is be done using Hamcrest
5. Validated username = Uname001 in JSON response
6. Validated email = Positive@Attitude.com in JSON response
7. Validated userStatus = 1 in JSON response

REST Assured Assignment004:

1. Created a TestNG test and implemented the below scenario
2. Created a class to implement this
3. All assertion and validation is done using Hamcrest
4. Validated code = 200 in response
5. Validated response message contains text 'logged in user session'

REST Assured Assignment005, 6:

1. Created 3 separate classes to implement the above 3 scenarios
2. Used a provided url to complete this
3. and using the steps provided validated all conditions using hamcrest

Jmeter Assignment:

1. Opened JMeter
2. And then created a testplan and added a threadgroup
3. Validated **https://httpbin.org/basic-auth/user/passwd** link using **HTTP Authentication manager**
4. Used username as user, and the password as passwd
5. Validate response is a JSON file using JSON assertion
6. Created one more threadgroup for validating simplilearn xpath using HTTP sampler
7. Increased the thread count for lead testing

Finally completed the project.