Automation Task

1. Eclipse Workspace .
2. Selenium .
3. Development Language : Java .
4. Design Pattern : Page object Model (POM).
5. Project Type: Maven Project .
6. Java Language .
7. Webdriver manager .
8. Data Driven using Json (json-simple) .
9. Allure For Reporting .
10. Testng

----------------------------------------------------------------------------------------------------

Structure for my Framework :

1.I created Maven Project

2.I added all needed Dependencies (Including Data Driven Using Json )

3.I started adding **Base Class** to be base for all Classes

4.I started creating all Cases (Test Classes/ Page Classes) for each needed View (EX: Home ,signin, profile …)

5.In Page classes I started to put the locators for the elements in the Page classes and make the Scenarios in Methods inside the pages classes

6.In Test Classes I Started to call the created methods in page classes

And run the Called cases using Testng .

7.After finishing the project with Hard code data , I started to change the framework to Data Driven .

8.I created Data package and added the following on it:

a) JsonData Reader class

b)UserData.json

I added the Data on UserData.json file, and called this data in JsonData Reader class, and applied it in all Cases

9.Added allure dependency to create allure report

10.I Used Constructors to Identefy the drivers in page Classes

11.I used Inheritance in test classes when I used exceed to identify data from Base class (Start Url ,Drivers,…..)

12.I Used Data Abstract when I called methods from Page classes to Test Classes

13.With every Test case there`s assertion to ensure that Every test cases meet the Expected Result

Rest Assured Task

1. Eclipse Workspace .
2. Rest Assured .
3. Development Language : Java .
4. Design Pattern : Page object Model (POM).
5. Testng
6. Json Dependencies

----------------------------------------------------------------------------------------------------

Structure for my Framework :

1.I created Maven Project .

2.I added all needed Dependencies (Rest assured, Jason,…) .

3.I Divided the 4 methods (Get,Put,Post,Delete) in 4 classes .

4.I started with **Get** class ,

I Opened the Swagger link provided in Task ,

And choosed one of the Get method existed in the pet Section,

The I took the Request Url and used in my Method and asserted the method with Status code .

5.I then started with **Post** class, but faced the issue that most of the videos was using Jsonobject then Use put ,but I couldn`t use it because type of data in the API , so after loooooong search I found that we can use Data in the method as **String ,**this helped me to add the Data normally and asserted it with status code too .

6.Then I started in put Class and used string as previous class and asserted it with status code too .

7.Then I started the Delete class and asserted it with status code 404 as I used Pet ID not exist .