**Problem Statement 1: API Testing Using Postman**

**Note:**

**• Create the Collection.**

**• Create below requests on the created Collections.**

**• Export the collection and share it as a Solution.**

**1. Handling of Get Request:**

• Hit a GET call with URL: <https://jsonplaceholder.typicode.com/todos/3>

• Add a test to

o Validate response as 200 in Postman.

o Validate ID number as 3 in response.

o Validate response title is ‘fugiat veniam minus’. o Validate completed is false in response.

**1. Create an Environment in Postman**

1. **Open Postman.**
2. **Click on the "Environments" tab** on the left sidebar or click the gear icon on the top right and select "Manage Environments."
3. **Click on "Add"** to create a new environment.
4. **Name your environment** as AssignmentData.
5. **Add variables** to the environment:
   * **Variable Name**: baseUrl1
   * **Initial Value**: https://jsonplaceholder.typicode.com
   * **Current Value**: <https://jsonplaceholder.typicode.com>
   * **Variable Name**: endPoint1
   * **Initial Value**: /todos/3
   * **Current Value**: /todos/3
6. **Click "Add"** to save the environment.

A screenshot of a computer

Description automatically generated

**2. Create a New Request in Postman**

1. **Click on the "Collections" tab** on the left sidebar.
2. **Click on "New Collection"** to create a new collection.
3. **Name your collection** Assignment
4. **Click "Add Request"** to create a new request within this collection.

**3. Configure the GET Request**

1. **Name your request**, I gave it Question 1
2. **Select the method** as GET from the dropdown menu.

**4. Change the Environment in Collections**

1. **From the Right drop down corner change the environment to your current environment that you just created.**
2. **Enter the request URL** using the environment variable: {{baseURL1}}{{endPoint1}}
3. Click on send

A screenshot of a computer

Description automatically generated

**4. Add Tests to Validate the Response**

1. **Click on the "Scripts" tab** in the window.
2. **Click on Postbot** and then click on **Add test to the request from right bottom**  to generate the script

**5. Run the Request**

1. **Click the "Send" button** to execute the GET request.
2. **Review the response** in the response section below the request window.
3. **Check the "Test Results" tab** to ensure all your tests pass.

A screenshot of a computer

Description automatically generated



**2. Handling of Get Request with Global Variable:**

• Hit a GET call with URL: <https://reqres.in/api/users?page=2>

• For page=2 in URL, use Pre-request Scripts in Postman to create a Global variable for page number as 2 and pass this page number to URL during runtime

• Add a test to

o Validate response as 200 in Postman

o Validate page number as 2 in response

o Validate URL under support is https://reqres.in/#support-heading in JSON response

**3. Handling of Put Request:**

• Hit a PUT call having service URL = <https://reqres.in/api/users/2>

• PUT Call request JSON body:

{

"name": "morpheus",

"job": "JAVA Development"

}

• Add a test to

o Validate the response with 200 OK.

o Validate the Response time less than 500ms

**5. Handling of DELETE Request:**

• Hit a DELETE call having service URL = <https://jsonplaceholder.typicode.com/comments/101>

• Add a test to validate the response with 200 OK.

**6. Handling of GET Request:**

• Hit the GET Request <https://webapps.tekstac.com/MovieAPI/UserService/viewUserList>

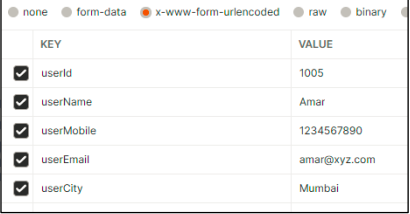
• Request Method: GET.

• Add a test to validate the Content-Type of the Response.

**7. Handling of POST Request:**

• Hit the POST Request <https://webapps.tekstac.com/MovieAPI/UserService/addUser>

• Pass the below mentioned data along with the request body.



• Add a test to validate if the response body contains the string "Mumbai"

**8. Handling of DELETE Request:**

• Method: DELETE

• URL: <https://webapps.tekstac.com/MovieAPI/CinemaHallService/deleteCinemaHallById/3101>

• Add the test to check if cinemaHallId 3101 is Not included in the response