Candidate Name:MD ATIQUL ISLAM.

Email- atik.cmttiu1001@gmail.com

Qustions of the Assessment:

#API testing:

1. go to (https://reqres.in/) and take (CRUD) 5 APIs and automate this API by Karate/Cypress/Robot Framework.

2. go to (https://petstore.swagger.io/#/) and take (CRUD) APIs and validate this API with Postman, write 3 tests by using dynamic Assertion. Generate Newman report.

Answer.1: I can provide you with a general outline of how you can automate CRUD operations on APIs using Karate, Cypress, and Robot Framework and I can guide you on how to write such automation scripts based on your API endpoints.

1. *Karate Framework*:

   - Create feature files, one for each CRUD operation (Create, Read, Update, Delete).

   - Use Karate's DSL to define HTTP requests, such as POST, GET, PUT, and DELETE.

   - Write assertions to validate the responses.

   - Here's a simplified example of creating a user:

```
cucumber

Feature: Create User


Scenario: Create a new user

  Given url 'https://reqres.in/api/users'

  And request { "name": "John", "job": "Engineer" }

  When method POST

  Then status 201
```

```
    And match response.id == 7
```

2. *Cypress*:

   - Create Cypress test files for each CRUD operation.

   - Use Cypress commands to make HTTP requests (e.g., `cy.request()`).

   - Use assertions to validate the responses.

   - Here's a simplified example of creating a user:

```
javascript
describe('Create User', () => {
    it('Creates a new user', () => {
        cy.request('POST', 'https://reqres.in/api/users', {
            name: 'John',
            job: 'Engineer'
        }).then((response) => {
            expect(response.status).to.equal(201);
            expect(response.body.id).to.equal(7);
        });
    });
});
```

3. *Robot Framework*:

   - Create Robot Framework test cases for each CRUD operation.

   - Use the "RequestsLibrary" or a custom library for HTTP requests.

- Define keywords to handle assertions.

- Here's a simplified example of creating a user:

```
robot

*** Settings ***

Library                    RequestsLibrary


*** Test Cases ***

Create User

        ${headers}=       Create Dictionary       Content-Type=application/json

        ${data}=             Create Dictionary       name=John       job=Engineer

        ${response}=      Post Request       https://reqres.in/api/users       headers=${headers}
json=${data}

        Should Be Equal As Strings       ${response.status_code}       201

        ${json_response}=      To Json       ${response.text}

        Dictionary Should Contain Value       ${json_response}       7
```

Answer.2: Here are the general steps:

1. *Install Postman:* If you haven't already, download and install Postman from the official website.

2. *Open Postman:* Launch the Postman application.

3. *Import API Collection:* In Postman, you can import an API collection to work with. If you have the Swagger Petstore API collection, you can import it into Postman. To do this, click on "Import" in Postman, and select the Swagger Petstore API collection file.

4. *Create Dynamic Assertions:* To create dynamic assertions, you'll typically use environment variables and test scripts in Postman.

- Set up environment variables to hold dynamic data like IDs or values you want to validate.
- In your API request, use these environment variables to capture the response data.
- Write test scripts that check if the response data matches your expectations, using environment variables as placeholders for dynamic values.

Here's a simplified example:

javascript

```
// Set the environment variable 'expectedName' with the expected name
pm.environment.set('expectedName', 'Fluffy');

// Get the actual name from the API response
var actualName = pm.response.json().name;

// Check if the actual name matches the expected name
pm.test('Name should match', function() {
    pm.expect(actualName).to.eql(pm.environment.get('expectedName'));
});
```

5. *Run Tests:* Execute the API requests and tests within Postman to validate the CRUD operations.

6. *Generate Newman Report and Codes:*

- To generate a Newman report, you'll need to install Newman, which is a command-line tool for Postman. You can install it using npm (Node Package Manager) if you haven't already:

```
npm install -g newman
```

- Run Newman with the collection and environment:

```
newman run YourCollectionFile.json -e YourEnvironmentFile.json -r cli,html
```

Replace `YourCollectionFile.json` and `YourEnvironmentFile.json` with your actual file names.

- This command will run your collection with Newman, generate a command-line report, and an HTML report.

- To view the HTML report, open it in a web browser.