**API Test Cases:**

1. **Create Operation:**

**Endpoint**: <https://dummyapi.io/data/v1/user/>create

**Method**: POST

**Header**-   
Key- app-id  
Value- 64d28dca50c5bd37ea9c9c96

**Description**: Create a new entity using POST request.

**Test Steps:**

1. Send a POST request to the `/create` endpoint with valid payload data.

2. Verify that the response status code is 201 (Created).

3. Verify that the response body contains the created entity's details.

4. Verify that the entity is added to the database.

**Test Data-**

Payload- **Body-** {

    "firstName": "Shiv",

    "lastName": "Kumar",

    "gender": "male",

    "email": "test14@example.com"

}

2. **Read Operation:**

**Endpoint**: <https://dummyapi.io/data/v1/user/>

**Method**: GET

**Header**-   
Key- app-id  
Value- 64d28dca50c5bd37ea9c9c96

**Description**: Retrieve the list of all Users

**Test Steps**:

1. Send a GET request to `/user` endpoint.

2. Verify that the response status code is 200 (OK).

3. Verify that the response body contains list of all Users

3. **Update Operation:**

**Endpoint**: https://dummyapi.io/data/v1/user/60d0fe4f5311236168a109ca

**Method**: PUT

**Header**-   
Key- app-id  
Value- 64d28dca50c5bd37ea9c9c96

**Description**: Update user details using PUT request.

**Test Steps:**

1. Choose a valid `userId`.

2. Send a GET request to `/users/{userId}` endpoint to get the original user details.

3. Modify the user details (e.g., change name or email) in the payload.

4. Send a PUT request to `/users/{userId}` endpoint with the modified payload.

5. Verify that the response status code is 200 (OK).

6. Send a GET request to `/users/{userId}` to verify that the user details are updated.

**Test Data/ Payload:**

{

  "title": "Mr"

}

4. **Patch Operation:**

**Endpoint**: `/posts/{postId}`

**Method**: PATCH

**Header**-   
Key- app-id  
Value- 64d28dca50c5bd37ea9c9c96

**Description**: Update specific attributes of a post using PATCH request.

**Test Steps:**

1. Choose a valid `postId`.

2. Send a GET request to `/posts/{postId}` endpoint to get the original post details.

3. Modify specific attributes (e.g., update the title) in the payload.

4. Send a PATCH request to `/posts/{postId}` endpoint with the modified payload.

5. Verify that the response status code is 200 (OK).

6. Send a GET request to `/posts/{postId}` to verify that the specific attributes are updated.

5. **Delete Operation:**

**Endpoint**: <https://dummyapi.io/data/v1/user/64d3edbc23863cd1451ba97d>

**Method**: DELETE

**Header**-   
Key- app-id  
Value- 64d28dca50c5bd37ea9c9c96

**Description**: Delete a specific post using DELETE request.

**Test Steps:**

1. Choose a valid `postId`.

2. Send a DELETE request to `/user/{postId}` endpoint.

3. Verify that the response status code is 200 (Ok).

4. Attempt to retrieve the deleted post using a GET request and verify that the response status code is 404 (Not Found).