**API Testing Scripts and Defects**

package api.tests;

import io.restassured.RestAssured;

import io.restassured.response.Response;

import org.testng.Assert;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

import static io.restassured.RestAssured.\*;

import static io.restassured.matcher.RestAssuredMatchers.\*;

import static org.hamcrest.Matchers.\*;

public class APITests {

@BeforeClass

public void setup() {

RestAssured.baseURI = "https://api.crowdfunding.com";

}

@Test(priority = 1)

public void adminLoginTest() {

Response response =

given().contentType("application/json")

.body("{\"username\": \"admin\", \"password\": \"admin123\"}")

.when().post("/admin/login")

.then().statusCode(200)

.extract().response();

String token = response.jsonPath().getString("token");

Assert.assertNotNull(token, "Token should not be null");

}

@Test(priority = 2)

public void getActiveProjects() {

given().contentType("application/json")

.when().get("/projects/active")

.then().statusCode(200)

.body("size()", greaterThan(0));

}

@Test(priority = 3)

public void fundProject() {

given().contentType("application/json")

.body("{\"projectId\": 101, \"amount\": 5000}")

.when().post("/fund")

.then().statusCode(201)

.body("message", equalTo("Funding successful"));

}

}

**Defect Report - Crowdfunding API Testing**

**1. Defect ID:** CF-001  
**Title:** Admin Login Returns Invalid Token  
**Severity:** Critical  
**Description:** The admin login API returns a token, but it is invalid when used for authentication in subsequent requests.  
**Steps to Reproduce:**

1. Send a POST request to /admin/login with valid credentials.
2. Extract the token from the response.
3. Use the token in the Authorization header for any admin-related API request.
4. Observe that the API returns an "Unauthorized" error.  
   **Expected Result:** The token should be valid for authentication.  
   **Actual Result:** The token is invalid.  
   **Status:** Open  
   **Assigned To:** Backend Team

**2. Defect ID:** CF-002  
**Title:** Fund Project API Allows Negative Amount  
**Severity:** High  
**Description:** The fund project API does not validate the amount field, allowing negative values to be entered.  
**Steps to Reproduce:**

1. Send a POST request to /fund with {"projectId": 101, "amount": -5000}.
2. Observe that the response returns a success message instead of an error.  
   **Expected Result:** The API should return an error for negative amounts.  
   **Actual Result:** The API processes the request successfully.  
   **Status:** Open  
   **Assigned To:** Backend Team

**3. Defect ID:** CF-003  
**Title:** Get Active Projects API Returns Empty List  
**Severity:** Medium  
**Description:** The /projects/active API intermittently returns an empty list, even when active projects exist.  
**Steps to Reproduce:**

1. Send a GET request to /projects/active.
2. Observe that the response contains an empty array.  
   **Expected Result:** The API should return a list of active projects.  
   **Actual Result:** The response contains an empty array.  
   **Status:** Open  
   **Assigned To:** Backend Team

**4. Defect ID:** CF-004  
**Title:** Project Upload API Accepts Empty Fields  
**Severity:** High  
**Description:** The /uploadProject API allows project uploads with missing required fields like title and description.  
**Steps to Reproduce:**

1. Send a POST request to /uploadProject with an empty title or description.
2. Observe that the API returns a success response instead of an error.  
   **Expected Result:** The API should return an error for missing required fields.  
   **Actual Result:** The API processes the request successfully.  
   **Status:** Open  
   **Assigned To:** Backend Team

**5. Defect ID:** CF-005  
**Title:** User Registration API Does Not Validate Email Format  
**Severity:** Medium  
**Description:** The /register API allows users to register with an invalid email format.  
**Steps to Reproduce:**

1. Send a POST request to /register with an invalid email format (e.g., user@com).
2. Observe that the API returns a success message instead of an error.  
   **Expected Result:** The API should validate email formats and reject invalid ones.  
   **Actual Result:** The API allows invalid emails.  
   **Status:** Open  
   **Assigned To:** Backend Team

**6. Defect ID:** CF-006  
**Title:** Fund Provider Cannot View Funded Projects  
**Severity:** High  
**Description:** The /fundedProjects API returns an empty list for fund providers even if they have funded projects.  
**Steps to Reproduce:**

1. Send a GET request to /fundedProjects as a fund provider.
2. Observe that the response contains an empty array.  
   **Expected Result:** The API should return a list of funded projects.  
   **Actual Result:** The API returns an empty array.  
   **Status:** Open  
   **Assigned To:** Backend Team