API Testing Mini Project Using Postman

Project Overview

Developed a comprehensive API testing project using Postman to demonstrate proficiency in RESTful API testing, collection management, and quality assurance practices. Executed complete user flow testing using Postman Collection Runner.

Technologies Used

- **Tool**: Postman (Collection Runner)
- **API**: ReqRes.in (REST API)
- **Testing Approach**: Sequential API testing with manual test configuration
- **HTTP Methods**: GET, POST, PUT, PATCH, DELETE
- **Setup**: Manual environment and collection configuration

Project Objectives

- Implement end-to-end user flow automation testing
- Validate API responses and status codes
- Extract and manage authentication tokens dynamically
- Perform CRUD (Create, Read, Update, Delete) operations testing
- Ensure data integrity across multiple API requests

Implementation Details

Environment Setup

- Created custom environment in Postman for variable management
- Configured dynamic token storage for authenticated requests
- Established reusable environment variables for scalable testing

Test Collection: "User Flow Automation"

Designed and executed 6 API test requests using Postman Collection Runner:

1. Login & Token Extraction (POST)

- **Endpoint**: /api/login
- **Purpose**: Authenticate user and extract bearer token
- **Validations**:
    - Status code verification (200 OK)
    - JSON response structure validation
    - Token property existence check
- **Response Time**: 285ms

2. Create User (POST)

- **Endpoint**: /api/users
- **Purpose**: Create new user record
- **Validation**: Status code 201 (Created)
- **Response Time**: 290ms

3. Update User (PUT)

- **Endpoint**: /api/users/240
- **Purpose**: Complete user record update
- **Validation**: Status code 200 (OK)

- **Response Time**: 313ms

4. Add Field to User (PATCH)

- **Endpoint**: /api/users/2/240
- **Purpose**: Partial update - add new field to existing user
- **Validation**: Status code 200 (OK)
- **Response Time**: 301ms

5. Retrieve User (GET)

- **Endpoint**: /api/users/2
- **Purpose**: Fetch user details
- **Validation**: Status code 200 (OK)
- **Response Time**: 108ms

6. Delete User (DELETE)

- **Endpoint**: /api/users/240
- **Purpose**: Remove user record
- **Validation**: Status code 204 (No Content)
- **Response Time**: 313ms

Test Results & Metrics

Execution Summary

- **Total Tests**: 8 assertions
- **Tests Passed**: 8 (100% success rate)
- **Tests Failed**: 0
- **Total Execution Time**: 1,610ms (1.61 seconds)
- **Collection Runs**: 1

Key Achievements

- ✅ Achieved 100% test pass rate
- ✅ Successfully automated complete user lifecycle testing
- ✅ Implemented dynamic token management
- ✅ Validated all HTTP methods (GET, POST, PUT, PATCH, DELETE)
- ✅ Ensured proper status code handling for each request type
- ✅ Maintained optimal API response times (average 268ms)

Technical Skills Demonstrated

- **API Testing**: REST API testing, HTTP methods understanding, status code validation
- **Postman Proficiency**: Collections, environments, variables, test scripts, Collection Runner
- **Manual Testing**: Request configuration, test case design, response validation
- **Data Management**: Environment variables setup, dynamic token extraction
- **Quality Assurance**: Test assertions, response validation, JSON structure verification
- **Documentation**: Test organization and result analysis

Learning Outcomes

- Gained hands-on experience with RESTful API testing
- Understood authentication workflows and token management
- Learned to design comprehensive test scenarios for CRUD operations
- Developed skills in automated testing and test result analysis
- Applied best practices for API test organization and execution

Future Enhancements

- Implement data-driven testing using CSV/JSON files
- Add negative test scenarios for error handling validation

- Integrate with CI/CD pipeline using Newman
- Create custom test reports with detailed analytics
- Expand test coverage with edge cases and boundary testing

Project Date: November 2025
Status: Completed Successfully