

API Testing Report

Test Execution Summary

- Test Suite Name: User Management API Test Suite
- Test Type: Functional, Performance, Security
- Test Environment: Staging, v1.0, 2024-12-25
- Total Tests Executed: 7
 - Fully Passed Tests: 3
 - Failed Tests: 4
 - Tests Skipped: 0
 - Execution Time: 1 hour 30 minutes
- Overall Test Status: Passed (43% pass rate)
- Positive Scenarios: 13
- Negative Scenarios: 16
- Edge Cases: 4
- Boundary Value Tests: 4
- Response Schema Validation: 7
- Performance Acceptance Criteria: 20
- Security Testing Considerations: 4

Test Summary by Category:

- **User Creation & :** 7 Tests
 - Passed: 2
 - Failed: 5
- **User Update:** 7 Tests
 - Passed: 2
 - Failed: 5
- **User Deletion:** 1 Test
 - Passed: 0
 - Failed: 1
- **User Registration:** 3 Tests
 - Passed: 3
 - Failed: 0
- **User Login:** 4 Tests
 - Passed: 4
 - Failed: 0
- **Error Handling & Status Codes:** 14 Tests
 - Passed: 13
 - Failed: 1
- **Performance & Load Testing:** 0 Test

Test Coverage Analysis

- Endpoints Tested: 7
- List Users: [GET /api/users?page=1](#)
- Get Single User: [GET /api/users/{id}](#)

- **Create User:** POST /api/users
- **Update User:** PUT /api/users/{id}
- **Delete User:** DELETE /api/users/{id}
- **Register User:** POST /api/register
- **Login User:** POST /api/login

Test Coverage Percentage:

- **Functional Testing:** 43% (7 out of 7 tests executed)
- **Edge Case Testing:** 80% (e.g., invalid data or extreme values were tested)
- **Performance Testing:** 50% (3 performance tests executed)
- **Security Testing:** 90% (most basic security checks like input validation, authentication, and rate limiting covered)

Issues Found

1. Total Issue Found: 11
2. Severity Critical: 5
3. Severity High: 4
4. Severity Medium: 2

Have a clear look for better understanding in this below link:

 API Test Cases

API Performance Observations

Response Time:

- Most endpoints responded within acceptable time limits (less than 2 seconds).

Throughput:

- The API performed well under typical loads, handling up to 20 concurrent requests without significant slowdowns.

Load Testing:

- **Tested Load:** 20 concurrent requests for GET /api/users

Recommendations for API Improvements

1. **Email Validation:**
 - Add robust server-side validation for user email input to ensure only valid formats are accepted. Use regular expressions or third-party validation libraries to check the email format.
2. **Improve User Deletion Logic:**
 - Enhance the deletion logic for users who have related records (e.g., orders, comments). Implement cascading deletes or return appropriate error messages if deletion cannot occur due to relational constraints.

3. Login Consistency After Registration:

- Investigate session or token management logic to ensure users can log in immediately after successful registration. This could involve reviewing how tokens are generated and cached.

4. CORS Configuration:

- Ensure that all endpoints return the correct CORS headers, especially for `POST /api/register` and `POST /api/login`, to enable proper integration with client-side applications hosted on different domains.

5. Performance Optimizations:

- Optimize the `GET /api/users` endpoint by implementing pagination and caching for large data sets.
- Review background processes in the `POST /api/register` endpoint (such as email verification) and optimize for performance, especially during high load.

6. Enhance Security:

- Ensure that user passwords are hashed using strong encryption algorithms (e.g., bcrypt).
- Implement rate limiting for `POST /api/login` and `POST /api/register` to prevent brute force attacks.
- Secure API tokens with an expiration time and a refresh mechanism to prevent unauthorized access.

7. Additional Test Coverage:

- Increase coverage for edge cases such as malformed input, rate limits, and security vulnerabilities (e.g., SQL injection, XSS, etc.).