**Assignment no 04**

**Add Web API Test Automation and Performance Test Automation**

**Functional Requirements (FR)**

1. **FR1**: The API shall return a status code of 200 OK for successful GET, POST, PUT, and DELETE requests.
2. **FR2**: The API shall validate input data and return appropriate error codes, e.g., 400 Bad Request for invalid data and 404 Not Found for non-existent resources.
3. **FR3**: The API shall support user authentication using token-based authentication (e.g., JWT).
4. **FR4**: The API shall support pagination for endpoints returning large datasets.
5. **FR5**: The API shall allow creating, updating, retrieving, and deleting resources.
6. **FR6**: The API shall provide search and filter functionality for querying resources.
7. **FR7**: The API shall log all incoming requests and corresponding responses for debugging purposes.

**Non-Functional Requirements (NFR)**

**Performance Requirements**

1. **NFR1**: The API shall handle a minimum of **500 concurrent users** with a response time of **less than 200ms** for 95% of requests.
2. **NFR2**: The API shall remain operational under stress conditions, supporting **up to 1000 requests per second** without crashes.
3. **NFR3**: The API shall have an average uptime of **99.9%** over a month.
4. **NFR4**: The API shall support **rate limiting** to prevent abuse by excessive requests from a single client.

**Security Requirements**

1. **NFR5**: The API shall implement HTTPS for secure communication.
2. **NFR6**: The API shall validate all input to prevent SQL injection, XSS, and other vulnerabilities.
3. **NFR7**: The API shall support role-based access control (RBAC) for securing endpoints based on user roles.
4. **NFR8**: The API shall return **401 Unauthorized** for requests without valid tokens and **403 Forbidden** for unauthorized access to resources.
5. **NFR9**: The API shall log all authentication and authorization attempts for auditing.

**Scalability and Maintainability**

1. **NFR10**: The API shall be scalable to support an increase in traffic by at least **200%** without requiring code changes.
2. **NFR11**: The API shall be modular and adhere to RESTful principles to simplify future maintenance and updates.

**Usability**

1. **NFR12**: The API documentation shall be available in Swagger or similar tools for ease of understanding by developers.
2. **NFR13**: All error messages returned by the API shall be user-friendly and provide clear guidance for troubleshooting.

**Test Plan Document**

**Objective**

To ensure the APIs are tested thoroughly for functionality, performance, security, and reliability.

**Scope**

* Functional Testing
* Performance Testing
* Security Testing
* Integration Testing
* Regression Testing

**Test Strategies**

1. **Functional Testing:**
   * Tools: Postman, Rest Assured
   * Test cases for all CRUD operations, edge cases, and error scenarios.
2. **Performance Testing:**
   * Tool: K6
   * Scenarios:
     + Load Test: Simulate 50-100 concurrent users.
     + Stress Test: Increase load until failure and measure limits.
     + Endurance Test: Simulate continuous load over 24 hours.
3. **Security Testing:**
   * Tools: OWASP ZAP, Burp Suite
   * Scenarios:
     + Test for vulnerabilities like SQL injection, XSS, and CSRF.
     + Verify authentication and authorization.
4. **Integration Testing:**
   * Verify data flow between APIs and dependent systems.
5. **Regression Testing:**
   * Ensure no new bugs are introduced when changes are made.

**Test Automation**

* Use Maven for managing dependencies.
* Integrate tests into a CI/CD pipeline using Jenkins or GitHub Actions.

**Deliverables**

* Test Case Document: Contains detailed test cases with expected results.
* Performance Test Reports: Includes metrics like response time, throughput, and error rates.
* Security Test Reports: Details vulnerabilities identified and mitigation steps.
* Automation Scripts: For functional and performance tests.