Testable Non Functional Requirements For Generic Web API

# Performance Requirements:

## 1. Response Time (ISO 25010: Time Behavior)

* 90% of API requests should have a response time of 300 milliseconds or less.
* 99% of API requests should have a response time of 500 milliseconds or less.

## 2. Throughput (ISO 25010: Efficiency)

* The web API must handle a minimum of 1000 requests per second per core.
* Peak throughput must support bursts of up to 5000 requests per second.

## 3. Scalability (ISO 25010: Capacity

* The web API should horizontally scale to accommodate a 50% increase in concurrent users within 30 minutes.
* Evaluate the system's ability to handle a 20% increase in data volume.

# Reliability Requirements:

## 1. Availability (ISO 25010: Availability

* The web API will be available 99.99% of the time per day under the maximum load of 10 million concurrent users.
* Planned maintenance should not exceed 1 hour per month.

## 2. Fault Tolerance (ISO 25010: Reliability

* The web API should continue to operate seamlessly even when 20% of services experience temporary failures.
* Implement automated failover mechanisms for critical services.

## 3. Recovery (ISO 25010: Recoverability

* In the event of a failure, the web API shall recover within 5 minutes, restoring full functionality.
* Regularly perform disaster recovery drills to ensure a comprehensive recovery plan.

# Security Requirements:

## 1. Data Encryption (ISO 25010: Security

* + All sensitive data transmitted between the client and the API must be encrypted using TLS 1.2 or later.
  + Validate compliance with the latest encryption standards and vulnerabilities.

## 2. Authentication (ISO 25010: Security

* + The web API must support OAuth 2.0 for user authentication and authorization.
  + Implement multi-factor authentication for administrative access.

## 3. Injection Prevention (ISO 25010: Security

* + The API shall implement input validation to prevent SQL injection and other code injection attacks based on OWASP guidelines.
  + Regularly scan codebase for vulnerabilities using automated tools like OWASP ZAP.

## 4. Access Control (ISO 25010: Security

* + Enforce the principle of least privilege for API access.
  + Implement role-based access control for API resources.

# Developer Guidelines:

## 1. Code Structure (Based on MVC Framework

* + Use a consistent and well-defined project structure following the principles of the chosen MVC framework.
  + Adhere to coding standards and style guidelines, such as PEP 8 for Python or ESLint for JavaScript.

## 2. Documentation (ISO 25010: Documentation

* + The project codebase shall include comprehensive inline documentation using a standard documentation tool (e.g., Javadoc or Doxygen).
  + Maintain a centralized and up-to-date README file explaining project setup, configuration, and usage.

## 3. Code Review (ISO 25010: Maintainability

* + All code changes must undergo a thorough code review process before being merged into the main branch.
  + Conduct periodic code reviews to identify and address technical debt.

## 4. Testing (ISO 25010: Reliability

* + The project must have a suite of automated tests covering critical functionalities, with a code coverage of at least 80%.
  + Include unit tests, integration tests, and end-to-end tests in the testing suite.

## 5. Version Control (ISO 25010: Version Compatibility

* + The project should use a version control system (e.g., Git) with clear branching and tagging conventions.
  + Ensure backward compatibility for API versions and clearly communicate any breaking changes.

## 6. Contribution Guidelines (ISO 25010: Collaboration

* + The project shall have clear contribution guidelines detailing the process for submitting issues, proposing features, and making pull requests.
  + Foster a welcoming and inclusive community for contributors.

## 7. Dependency Management (ISO 25010: Modularity

* + The project must use a package manager (e.g., npm, pip) to manage dependencies and ensure version compatibility.
  + Regularly update dependencies to address security vulnerabilities and take advantage of new features.

## 8. Build and Deployment Automation (ISO 25010: Installability

* + Implement continuous integration and continuous deployment (CI/CD) pipelines to automate the build and deployment process.
  + Ensure reproducibility of builds across different environments.

## 9. Error Handling (ISO 25010: Fault Tolerance

* + Implement robust error handling mechanisms with informative error messages and proper logging.
  + Monitor and analyze error logs to proactively identify and address potential issues.

## 10. Performance Monitoring (ISO 25010: Performance Efficiency

* + Integrate performance monitoring tools to track and analyze key performance metrics.
  + Set up alerts for performance degradation or anomalies.

## 11. Scalability Planning (ISO 25010: Capacity

* + Develop and document a scalability plan that outlines the process for scaling infrastructure in response to increased demand.
  + Regularly test the scalability plan to ensure its effectiveness.

These requirements cover a wide range of aspects to ensure the performance, reliability, security, and maintainability of the web API, along with guidelines for effective development practices.