**Generic Requirements for a Web API:**

Performance Requirements:

1. **Response Time:**
   * The web API should respond to 95% of requests within 500 milliseconds under normal operating conditions.
   * Test: Conduct performance testing to measure and validate response times.
2. **Throughput:**
   * The system should handle a minimum of 1000 requests per second.
   * Test: Perform load testing to ensure the system meets the specified throughput requirement.
3. **Scalability:**
   * The API should be able to scale horizontally to accommodate a 20% increase in user load within 30 minutes.
   * Test: Conduct scalability testing to validate the system's ability to scale.

Reliability Requirements:

1. **Availability:**
   * The web API will be available 99.99% of the time per day under the maximum load of 10 million concurrent users.
   * Test: Measure system availability under load and validate against the defined availability percentage.
2. **Fault Tolerance:**
   * The system should continue to operate without a single point of failure, ensuring uninterrupted service even in the event of hardware or software failures.
   * Test: Conduct fault injection testing to ensure fault tolerance mechanisms are effective.
3. **Recovery Time:**
   * In the event of a failure, the system should recover and resume normal operations within 5 minutes.
   * Test: Simulate failures and measure the time it takes for the system to recover.

Security Requirements:

1. **Data Encryption:**
   * All sensitive data transmitted over the API should be encrypted using industry-standard encryption algorithms (e.g., TLS).
   * Test: Perform security scans to ensure data in transit is properly encrypted.
2. **Authentication:**
   * The API must implement strong user authentication mechanisms, such as OAuth 2.0, and enforce proper authorization controls.
   * Test: Conduct penetration testing to identify and address potential authentication and authorization vulnerabilities.
3. **Injection Prevention:**
   * The API should be protected against common security threats, including SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF), following OWASP guidelines.
   * Test: Use automated tools and manual testing to identify and mitigate injection vulnerabilities.

Developer Guidelines:

1. **Code Standards:**
   * Adhere to a consistent coding style and follow best practices for the programming language.
   * Test: Implement code reviews and automated tools to enforce coding standards.
2. **Documentation:**
   * Provide comprehensive and up-to-date documentation for the codebase, including API documentation for developers.
   * Test: Validate that documentation is complete and accurate; use automated tools to check documentation coverage.
3. **Testing:**
   * All code changes should include unit tests with a minimum code coverage of 80%.
   * Test: Run automated tests and measure code coverage to ensure compliance with testing requirements.
4. **Code Review:**
   * Enforce a code review process to ensure the quality and security of the codebase.
   * Test: Conduct regular code reviews using established guidelines and tools.