**TEST PLAN FOR RESTFUL BOOKER**

**1. Introduction**

**Objective:** This Test Plan aims to ensure the functionality, reliability, security, and performance of the RESTful APIs documented at https://restful-booker.herokuapp.com/apidoc/index.html.

**Scope:** The testing will cover all API endpoints listed in the documentation, including basic functionality, error handling, security, and performance testing.

**2. Test Strategy**

**Testing Types:**

1. Functional Testing: Ensure each API endpoint functions as expected.
2. Security Testing: Validate authorization and authentication mechanisms.
3. Performance Testing: Assess API response times and load handling capabilities.
4. Edge Case Testing: Test with boundary values and unexpected inputs.
5. Error Handling Testing: Verify how APIs handle various error scenarios.

**Testing Tools:**

* Postman for functional testing and initial load testing.
* JMeter for performance testing (if needed for load testing).

**3. Test Environment**

**Base URL**: https://restful-booker.herokuapp.com/

**Tools:**

* Postman (for manual testing and initial automation)
* JMeter (for load testing if applicable)
* Git (for version control of test scripts)
* CI/CD integration (if applicable)

**API Functional Testing**

**1. Authentication:**

- Verify authentication mechanisms (basic authentication or token-based).

- Test login/logout functionality.

**2. Booking Operations:**

- Create a new booking.

- Retrieve booking details by ID.

- Update an existing booking.

- Delete a booking.

- Get booking by parameters (date range, name, etc.).

**3. Health Check:**

- Verify API health status endpoint (/ping or equivalent).

**Security Testing**

**1. Authentication and Authorization:**

- Test with valid credentials.

- Test with invalid credentials.

- Ensure only authorized users can perform restricted operations.

**2. Input Validation:**

- Test with SQL injection attempts.

- Test with XSS injection attempts.

- Test with invalid JSON structures.

**Performance Testing**

**1. Load Testing:**

- Simulate multiple users creating bookings simultaneously.

- Measure API response times under load.

**2. Stress Testing:**

- Increase the load gradually to find breaking points.

- Evaluate how the API handles stress situations.

**Error Handling Testing**

**1. Invalid Requests:**

- Test with missing required parameters.

- Test with incorrect data formats.

- Test with out-of-bound values.

**2. Server Errors**:

- Simulate server errors (500 Internal Server Error).

- Verify error responses are appropriate and informative.

**5. Test Execution**

**Test Preparation**

1. Set up test environment with necessary tools and configurations.

2. Retrieve or generate test data (e.g., bookings, user credentials).

**Test Execution Process**

**1. Functional Testing:**

- Execute each API endpoint according to defined test scenarios.

- Document results and any deviations from expected behaviour.

**2. Security Testing:**

- Verify authentication flows.

- Test input validations for security vulnerabilities.

**3. Performance Testing:**

- Perform load and stress tests using JMeter or equivalent tool.

- Measure and analyse response times, throughput, and resource utilization.

**4. Error Handling Testing:**

- Execute error scenarios outlined in test scenarios.

- Verify error messages and HTTP status codes.

**6. Reporting**

**1. Test Results Documentation:**

- Record test execution results (pass/fail).

- Document any defects found with detailed steps to reproduce.

**2. \*\*Metrics and Analysis\*\*:**

- Provide performance metrics (response times, error rates).

- Summarize overall test coverage and results.

**7. Risks and Contingencies**

**Risks:**

- Dependency on external API availability.

- Limited control over server-side configurations.

**Contingencies:**

- Plan for testing during off-peak hours if load testing impacts production users.

- Have backup test data and scripts ready in case of environment resets.

**8. Conclusion**

**Summary:** This Test Plan outlines comprehensive testing for the RESTful APIs provided at https://restful-booker.herokuapp.com/apidoc/index.html. It covers functional, security, performance, and error handling aspects to ensure robustness and reliability of the APIs.

By following this Test Plan, you can systematically verify the functionality, performance, and security of the RESTful APIs, ensuring they meet expected standards and are ready for production use. Adjust and expand this plan based on specific project requirements and additional testing needs.