**Test Plan – Restful Booker Application**

Created By: Poorani Santhosh

1. **Objective:**

This project goal is to validate the functionalities of hotel booking system. It checks for the functionality, reliability, scalability and performance of the system. The Restful Booker Application API supports all the operations of CRUD (Create, Read, Update and Delete).

1. **Scope:**

The scope of the test plan includes:

**Features:**

* Creating Tokens
* Health Check up
* Get Booking ID’s
* Get Booking
* Create Booking
* Update Full Booking
* Update Partial Booking
* Delete Booking

**Types of Testing:**

* Manual Testing
* Automation Testing
* Performance Testing
* Accessibility Testing

**Environments:**

* **Browser Compatible** : Chrome, Edge, Mozilla Firefox
* **API Testing Tools :** POSTMAN / Swagger UI
* **Operating Systems :** Windows, Mac OS, Linux

**Evaluation Standards:**

* Number of defects found
* Time taken to complete testing
* API response time and reliability

**Team Roles and Responsibilities:**

* **QA Lead :** Oversees the testing process
* **QA Testers :** Execution and reporting of defects
* **Developers :** Fix defects and support QA Team
* **Project Manager:** Manage timelines and resources.

1. **Inclusions:**

* Functional testing of core API endpoints like Token generation and CRUD operations
* Validating of HTTP status codes and response bodies
* Testing with valid and invalid data
* Token – based authentication flow
* Schema Validation for API responses
* Boundary value and edge case testing for booking data.

1. **Exclusions:**

* UI – level testing of web pages (primary focus is on API endpoints).
* Load testing beyond thresholds.

1. **Test Environments:**

* **Operating Systems :** Windows 11, Mac OS, Linux
* **Browsers :** Google Chrome, Mozilla Firefox, Microsoft Edge (API access)
* **Devices :** Desktop and Laptops
* **Network Connectivity:** High Speed Wi-Fi, Wired Ethernet.
* **Hardware / Software Requirements :** 8GB RAM, API testing tools like POSTMAN, JMeter
* Python / Java with Rest Assured for automated API test cases
* **Security Protocols :** Use bearer tokens for authentication
* **Access Permissions:** Limited access for role based permissions for testers and developers.

1. **Defect Reporting Procedure:**

**Tools:** JIRA

**Defect Identification Criteria:**

* API returns wrong data
* Incorrect status codes
* Authentication issues
* Performance thresholds
* Functional failures in CRUD operations

**Defect Reporting Format:**

Every defect should contain the following fields:

* Defect Title
* Environment

**Steps to Reproduce**

* Pre – Conditions
* Request / Response logs
* Priority and Severity
* Attachments if required

Communication for test progress and defect discussions through Email and Daily stand - ups.

1. **Test Strategy:**

**Testing Types:**

* **Functional Testing:** To validate each API endpoint (CRUD) performs as expected with valid inputs.
* **Negative Testing:** To ensure the API gracefully handles invalid inputs.
* **Authentication Testing:** To test the /auth endpoint and validate session – based access control using tokens.
* **Boundary Testing:** To check the system behavior at the boundary values.
* **End – to – End Testing:** To validate complete booking life cycle.
* **Schema Validation:** To confirm the API responses.
* **Performance Testing:** To check the response time and load balancing.

**Test Design Techniques:**

* **Equivalence Partitioning:** Group similar test data from each group to reduce redundancy.
* **Boundary Value Analysis:** Checking the smaller, larger and outside range values for numeric and date fields.
* **Use Case Testing:** Designing test cases based on real world booking flows including authentication and endpoints.
* **Error Guessing:** Missing some data’s and methods to trigger failures.
* **Data Driven Testing :** Use bulk data with varied input data for bulk testing.

**Testing Levels**

* API Testing
* System Integration Testing
* Regression Testing
* Retesting

1. **Test Schedule:**

* Test Planning
* Test Case Writing
* Test Execution and Bug Report
* Test Report

1. **Test Deliverables:**

* Test Plan Document
* Test Cases
* Defect Reports
* Final Test Summary Report

1. **Entry and Exit Criteria:**

**Entry:**

* Requirement Document available
* Test Cases created and signed off
* Build ready

**Exit:**

* All Test Cases executed as per the requirement document
* All bugs are fixed and retested
* Final Test summary report signed off

1. **Tools:**

* JIRA (Defect Reporting)
* POSTMAN (API)

1. **Risks and Mitigations:**

* API endpoint downtime
* Resource availability
* Buffer time
* Limited Testing Time

1. **Approvals:**

Documents to be submitted for client approval are:

* Test Plan
* Test Case
* Test Execution Report
* Final Summary Report