**Test Plan for Restful Booker API**

**Create by – Sonali Dhoke (AI)**

**Index Page**

1. **Objective**
   * Purpose of the Test Plan
2. **Scope**
   * Overview of the testing areas including functional, data validation, error handling, security, integration, and edge case testing
3. **Inclusions**
   * List of API endpoints to be tested and associated test scenarios
4. **Test Environments**
   * URLs, OS/Browser configurations, devices, networks, and security/access details
5. **Defect Reporting Procedure**
   * Tools, roles, and procedures for defect reporting
6. **Test Strategy**
   * Techniques, testing types, and best practices for the test execution
7. **Test Schedule**
   * Timeline of testing activities from planning to reporting
8. **Test Deliverables**
   * Documents and reports to be delivered during and after the testing phase
9. **Entry and Exit Criteria**
   * Conditions for starting and concluding the testing process
10. **Test Execution**
    * Execution phase details, including entry and exit points
11. **Test Closure**
    * Steps for finalizing testing and obtaining approvals
12. **Tools**
    * List of tools used for testing, defect management, and documentation
13. **Risks and Mitigations**
    * Potential risks during testing and their corresponding mitigation strategies
14. **Approvals**
    * Final approval process for test plans, cases, and reports

**Objective**

The goal of this test plan is to ensure the quality, functionality, and reliability of the Restful Booker API hosted at [https://restful-booker.herokuapp.com](https://restful-booker.herokuapp.com/). The API is designed to handle booking requests for a fictional hotel booking system.

**Scope**

1. **Functional Testing:**
   * Validate endpoint behaviour for booking creation, retrieval, update, and deletion.
   * Confirm correctness of authentication mechanisms.
2. **Data Validation Testing:**
   * Test input fields for boundary and invalid values.
   * Verify response content accuracy.
3. **Error Handling Testing:**
   * Ensure proper status codes and error messages.
   * Avoid sensitive information leakage.
4. **Security Testing: NOT DOING ANY SECURITY TESTING**
   * Test access control with tokens.
   * Validate HTTPS usage and input sanitization.
5. **Integration Testing:**
   * Test the interactions between endpoints and booking lifecycle.
6. **Regression and Edge Case Testing:**
   * Validate unchanged functionality after updates.
   * Explore uncommon input scenarios.

**Inclusions**

* **POST /auth** - Token generation (authentication)
* **GET /booking** - Retrieve all booking IDs
* **GET /booking/{id}** - Retrieve a specific booking
* **POST /booking** - Create a new booking
* **PUT /booking/{id}** - Update booking
* **PATCH /booking/{id}** - Partial update
* **DELETE /booking/{id}** - Delete a booking
* **GET /ping** - API health check

All endpoints are tested with both valid and invalid inputs, authentication, and boundary scenarios.

**Test Environments**

* **Environment URLs:**
  + QA: <https://restful-booker.herokuapp.com/apidoc/index.html>
  + Pre-Prod: <https://restful-booker.herokuapp.com/apidoc/index.html>
* **OS/Browsers:**
  + Windows 10 – Chrome, Firefox, Edge
  + macOS – Safari
  + Android – Chrome
  + iOS – Safari
* **Devices:**
  + Desktop, Laptop, Tablets, Smartphones
* **Network:** Wi-Fi and mobile network (3G/4G)
* **Security & Access:**
  + Token-based access
  + Passwords and cookies for protected endpoints

**Defect Reporting Procedure**

* Tools: JIRA
* Roles: Frontend (Devesh), Backend (Sonal), DevOps (Prajeeth)
* Steps:
  + Document issue with screenshots/logs.
  + Assign severity and priority.
  + Update stakeholders regularly.

**Test Strategy**

* Techniques:
  + Equivalence Partitioning, Boundary Value Analysis, Error Guessing
* Testing Types:
  + Smoke, Sanity, Functional, Regression, Usability
* Best Practices:
  + Shift Left, Exploratory, End-to-End Testing

**Test Schedule**

* Test Plan Creation: Day 1
* Test Case Creation: Day 2-3
* Test Execution: Day 4-6
* Summary Reports: Day 7

**Test Deliverables**

* Test Plan Document
* Test Scenarios & Cases
* Execution Reports
* Defect Reports
* Summary Report

**Entry and Exit Criteria**

* **Entry:**
  + Requirements are shared
  + environment is ready Test
* **Exit:**
  + All test cases executed
  + Major defects fixed

**Test Execution**

* **Entry:**
  + Sign-off on test cases
  + Application accessible
* **Exit:**
  + Execution report prepared
  + Defects logged

**Test Closure**

* **Entry:** Test artifacts ready
* **Exit:** Summary shared and approved

**Tools**

* JIRA
* Postman/cURL
* Mind Mapping Tools
* Word, Excel

**Risks and Mitigations**

* **Risk:** API Unavailable
  + **Mitigation:** Work on documentation/test data setup
* **Risk:** Limited Testing Time
  + **Mitigation:** Add resources dynamically
* **Risk:** Token issues
  + **Mitigation:** Regenerate or mock authentication

**Approvals**

* Test Plan
* Test Cases
* Test Summary Reports

Testing will proceed only after formal approvals by the QA lead and stakeholders.