**Test Plan for Restful Booker API**

**1. Objective**

This document outlines the test plan for the Restful Booker API hosted at https://restful-booker.herokuapp.com. The objective is to ensure the quality, functionality, and reliability of the API by verifying its CRUD operations and authentication mechanisms.

**2. Scope**

**Features to be Tested:**

* Create Booking
* Update Booking
* Partial Update Booking
* Delete Booking
* Ping (Health Check)
* Authentication (Create Token)
* Get Booking IDs

**Types of Testing:**

* Manual Testing
* Automated Testing
* Functional Testing
* Performance Testing
* Security Testing
* API Testing

**Environments:**

* Web Browsers: Chrome, Firefox, Edge
* Operating Systems: Windows, macOS, Linux
* API Clients: Postman, cURL
* Network Connectivity: Wi-Fi, wired connections

**Evaluation Criteria:**

* Number of defects found
* API response times
* API reliability across multiple test cycles

**Team Roles and Responsibilities:**

* Test Lead: Oversees testing strategy and execution
* Testers: Execute test cases and report defects
* Developers: Fix reported defects
* DevOps: Maintain test environments

**3. Inclusions**

* **Introduction**: Overview of the test plan, scope, and goals.
* **Test Objectives**: Identify defects, validate API functionality, and ensure reliability.

**4. Exclusions**

* UI Testing (since this is a backend API test plan)
* Mobile App Testing

**5. Test Environments**

* **Operating Systems**: Windows 10, macOS, Linux
* **Browsers**: Google Chrome, Mozilla Firefox, Microsoft Edge
* **Devices**: Desktop computers, laptops
* **Network Connectivity**: Wi-Fi, wired connections
* **Security Protocols**: API token-based authentication
* **Access Permissions**: Testers, developers, stakeholders

**6. Defect Reporting Procedure**

* **Criteria for Identifying Defects**: API failures, unexpected responses, slow response times
* **Steps for Reporting Defects**: Use JIRA for bug tracking, provide detailed reproduction steps
* **Triage and Prioritization**: Assign severity and priority levels
* **Tracking Tools**: JIRA, TestRail
* **Roles and Responsibilities**: Testers log bugs, developers fix issues
* **Metrics**: Number of defects, response time, resolution time

**7. Test Strategy**

**Step 1: Test Case Creation**

* Equivalence Partitioning
* Boundary Value Analysis
* Use Case Testing

**Step 2: Test Execution**

* Smoke Testing
* Functional Testing
* Regression Testing
* API Load Testing
* Security Testing

**Step 3: Best Practices**

* Shift Left Testing: Early testing in development
* Exploratory Testing: Beyond test case execution
* End-to-End Flow Testing: Full user scenarios

**8. Test Schedule**

|  |  |
| --- | --- |
| **Task** | **Duration** |
| Test Plan Creation | 3 days |
| Test Case Creation | 5 days |
| Test Execution | 10 days |
| Defect Fixing & Retesting | 5 days |
| Test Closure & Reporting | 2 days |

**9. Test Deliverables**

* **Entry and Exit Criteria**
  + Entry: Defined test scenarios, API documentation
  + Exit: Test case reports, defect reports

**10. Entry and Exit Criteria**

* **Requirement Analysis**
  + Entry: API documentation available
  + Exit: Clarified requirements
* **Test Execution**
  + Entry: Test cases ready, API stable
  + Exit: Test cases executed, defects reported
* **Test Closure**
  + Entry: Defect reports analyzed
  + Exit: Test summary reports finalized

**11. Tools**

* **API Testing**: Postman, cURL
* **Bug Tracking**: JIRA
* **Test Management**: TestRail
* **Performance Testing**: JMeter

**12. Risks and Mitigations**

|  |  |
| --- | --- |
| **Risk** | **Mitigation Strategy** |
| API Server Downtime | Backup test environment |
| Authentication Issues | Pre-generate test tokens |
| Limited Testing Time | Prioritize high-impact test cases |

**13. Approvals**

* **Documents for Approval**: Test Plan, Test Cases, Defect Reports
* **Stakeholders**: Test Lead, Developers, QA Manager