Test Plan (Restful Booker)

Objective

The primary objective of the API testing for Restful Booker API is to ensure the quality, functionality, and reliability of the Restful Booker API hosted at https://restful-booker.herokuapp.com. This test plan aims to systematically validate the API endpoints, adhering to industry best practices and standards, with the goal of delivering a high-quality, stable, and secure API.

The API, designed to handle booking requests for a fictional hotel booking system, will be subjected to comprehensive testing to identify and address potential issues related to data integrity, error handling, authentication, and overall system responsiveness. Through this testing process, we aim to verify that the Restful Booker API meets its functional requirements and is capable of providing a seamless experience for users interacting with the hotel booking system. The results of this testing will contribute to the overall assurance of the API's performance, ensuring it aligns with the specified requirements and delivers a reliable foundation for the Restful Booker application.

Scope

Scope of Test Plan for Restful Booker API:

1. **Functional Testing:**
   - Verify the correctness and functionality of all API endpoints as per the API documentation.
   - Test various scenarios for booking creation, modification, and cancellation.
   - Validate user authentication and authorization mechanisms for protected endpoints.

2. **Data Validation Testing**:
   - Ensure that the API correctly validates input data, rejecting invalid requests.
   - Test boundary values for input fields to check for any unexpected behavior.
   - Validate the accuracy of data returned in responses.

3. **Error Handling Testing**:
   - Verify that appropriate error codes and messages are returned for invalid requests.
   - Check error responses for sensitive information disclosure.
   - Validate the API's ability to handle unexpected errors gracefully.

4. **Performance Testing**:
   - Assess the API's response time under normal and peak loads to identify potential bottlenecks.
   - Measure the API's throughput and scalability to handle concurrent requests.

It's important to note that the scope of the test plan may evolve during the testing process based on feedback, changing requirements, or discoveries during testing. The scope should be reviewed and adjusted accordingly throughout the testing phase to ensure comprehensive coverage of the Restful Booker API.

Inclusions

Create (POST) Operations:

Test the API's ability to create new bookings using valid input data.
Verify that appropriate error responses are returned for invalid or missing data.
Validate that newly created bookings are stored correctly in the system.

Read (GET) Operations:

Test the API's ability to retrieve booking information by various criteria (e.g., booking ID, date range, guest name).
Verify that the API returns the correct data in response to read requests.
Test for correct handling of non-existent or invalid booking IDs.

Update (PUT) Operations:

Test the API's ability to update existing bookings with valid data.
Verify that the API rejects invalid update requests with appropriate error responses.
Validate that the booking data is correctly modified in the system after updates.

Delete (DELETE) Operations:

Test the API's ability to delete bookings by providing valid booking IDs.
Verify that the API returns appropriate responses after successful deletion.
Validate that the deleted bookings are removed from the system.

Boundary Testing:

Test the API with minimum and maximum allowed values for input fields.
Validate the behavior of the API with values close to the boundaries.

Data Validation:

Test the API's response to various data validation scenarios (e.g., invalid characters, data types, mandatory fields).
Verify that the API handles validation errors appropriately.

Authentication and Authorization:

Test CRUD operations for both authenticated and unauthenticated users.
Verify that only authorized users can perform certain CRUD operations.

Error Handling:

Test the API's response when invalid or malformed requests are made for CRUD operations.
Validate that appropriate error codes and messages are returned.

Performance Testing:

Evaluate the API's response time for CRUD operations under normal and peak loads. Measure the throughput and scalability of the API.

Test Environments

- The **operating systems** and versions that will be used for testing, such as Windows 10, macOS, or Linux.

- The **browsers** and versions that will be tested, such as Google Chrome, Mozilla Firefox, or Microsoft Edge.

- The **devices** that will be used for testing, such as desktop computers, laptops, tablets, and smartphones.

Name	**Env URL**
QA	https://restful-booker.herokuapp.com/apidoc/index.html
Pre-Prod	https://restful-booker.herokuapp.com/apidoc/index.html

Test Strategy

Step 1: Create test scenarios and test cases for the various features in Scope.

The Following test design techniques will be applied in the creating test cases:

- Equivalence Class Partition
- Boundary Value Analysis
- Decision Table Testing
- State Transition Testing
- Use Case Testing

Step 2: Testing procedure for testing request:

- First, we'll conduct smoke testing to see if the various and important functionalities of the application are working.
- We reject the build if the Smoke Testing fails and will wait for the stable build before performing in-depth testing of the application functionalities.
- Once we receive a stable build, which passes Smoke Testing, we perform in-depth testing using the Test Cases created.
- Report the detected defects/bugs in bug tracking tool.

As part of the Testing, we will perform the types of testing below:

- Smoke Testing and Sanity Testing
- Regression Testing and Retesting
- Usability Testing, Functionality & UI Testing
- We repeat Test Cycles until we get the quality product.

Step3: We will follow the best practices to make our testing better:

- **Context Driven Testing** – We will be performing Testing as per the context of the given application.
- **Exploratory Testing** – Using our expertise we will perform Exploratory Testing, apart from the normal execution of the Test cases.
- **End to End Flow Testing** – We will test the end-to-end scenario which involves multiple functionalities to simulate the end user flows.

Test Schedule

Following is the test schedule planned for the project –
Task Time Duration

Task	**Dates**
▪ Creating Test Plan	- 1/20/2024 to 1/23/2024
▪ Test Case Creation	- 1/23/2024 to 1/28/2024
▪ Test Case Execution	- 1/29/2023 to 2/9/2023
▪ Summary Reports Submission Date	- 2/10/2023

Tools

The following are the list of Tools we will be using in this Project:
- JIRA
- TestRail
- Postman
- Snipping Screenshot Tool
- Word and Excel documents

Entry and Exit Criteria

1. Requirement Analysis

Entry Criteria:

- Once the testing team receives the Requirements Documents or details about the Project

Exit Criteria:

- A list of Requirements is explored and understood by the Testing team.
- Doubts and Ambiguities are cleared.

2. Test Execution

Entry Criteria:

- Test Scenarios and Test Cases Documents are signed by the Client.
- Application is ready for Testing.

Exit Criteria:

- Test Case Reports, Defect Reports are ready.

3. Test Closure

Entry Criteria:

- Test Case Reports, Defect Reports are ready.

Exit Criteria:

- Test Summary Report.