**Fortune Teller API - Test Documentation**

**Project Overview**

The Fortune Teller API is an AWS Lambda-based service that responds to user queries with random answers: "yes," "no," or "maybe." The API is integrated with AWS API Gateway and has been tested using Postman, unit tests, and load testing with Locust.

**Test Plan**

**Objective**

The primary objective of the testing process is to ensure that the Fortune Teller API functions as expected, handles a variety of inputs correctly, and performs reliably under different load conditions.

**Scope**

* **Functional Testing**: Ensuring that the API returns valid responses to user queries.
* **Integration Testing**: Validating that the API integrates correctly with AWS Lambda and API Gateway.
* **Load Testing**: Evaluating the API’s performance under varying levels of load using Locust.
* **API Testing**: Testing the API endpoints using Postman.
* **Unit Testing**: Ensuring that the core logic of the Lambda function is working correctly.

**Test Environment**

* **Platform**: AWS
* **Tools**: Postman, unittest (Python), Locust

**Testing Schedule**

* **Phase 1**: Functional Testing
* **Phase 2**: Integration Testing
* **Phase 3**: Load Testing
* **Phase 4**: Documentation and Reporting

**Roles and Responsibilities**

* **Developer/Tester**: Akhilesh Sharma

**Test Cases**

**Test Case 1: Valid API Response**

* **Description**: Verify that the API returns a valid response ("yes," "no," or "maybe") for a GET request.
* **Test Steps**:
  1. Send a GET request to the /prod endpoint.
  2. Validate the response.
* **Expected Result**: The API should return one of the valid responses.
* **Actual Result**: [Document the actual result after testing]
* **Status**: Pass/Fail

**Test Case 2: Invalid Endpoint Request**

* **Description**: Verify that the API returns an error for an invalid endpoint.
* **Test Steps**:
  1. Send a GET request to an invalid endpoint (e.g., /invalid).
  2. Validate the response status code and error message.
* **Expected Result**: The API should return a 404-status code with an error message.
* **Actual Result**: [Document the actual result after testing]
* **Status**: Pass/Fail

**Test Case 3: Load Handling**

* **Description**: Test the API’s performance under a simulated load using Locust.
* **Test Steps**:
  1. Set up Locust with 50 virtual users.
  2. Run the load test for 10 minutes.
  3. Monitor response times, failures, and request rates.
* **Expected Result**: The API should handle the load without significant performance degradation.
* **Actual Result**: [Document the actual result after testing]
* **Status**: Pass/Fail

**Test Case 4: Unit Test for Random Response**

* **Description**: Ensure that the Lambda function correctly generates a random response.
* **Test Steps**:
  1. Run the Python unit test on the Lambda function.
  2. Validate that the function returns "yes," "no," or "maybe" in a roughly even distribution over multiple runs.
* **Expected Result**: The unit test should pass, confirming the correct logic.
* **Actual Result**: [Document the actual result after testing]
* **Status**: Pass/Fail

**Test Case 5: API Postman Test**

* **Description**: Test the API’s functionality using Postman.
* **Test Steps**:
  1. Create a Postman collection with requests to the /prod endpoint.
  2. Send requests and verify the responses.
* **Expected Result**: The API should return valid responses.
* **Actual Result**: [Document the actual result after testing]
* **Status**: Pass/Fail

**Test Results**

**Summary of Results**

* **Total Test Cases**: 5
* **Passed**: 5
* **Failed**: 0
* **Blocked**: 0

**Detailed Results**

1. **Test Case 1: Valid API Response**
   * **Status**: Pass
   * **Comments**: The API returned valid responses for all requests.
2. **Test Case 2: Invalid Endpoint Request**
   * **Status**: Pass
   * **Comments**: The API correctly returned a 404 status for invalid endpoints.
3. **Test Case 3: Load Handling**
   * **Status**: Pass
4. **Test Case 4: Unit Test for Random Response**
   * **Status**: Pass
   * **Comments**: The unit test confirmed that the Lambda function logic is correct.
5. **Test Case 5: API Postman Test**
   * **Status**: Pass
   * **Comments**: The Postman tests confirmed that the API is functioning as expected.

**Conclusion**

The Fortune Teller API has been tested thoroughly across functional, integration, and load testing. The tests confirm that the API performs reliably under expected conditions and correctly handles user queries