API TEST PLAN

1. Test Strategy

The primary objective of API testing for this module is to validate the communication between the invoicing system and external data sources (e.g., Snowflake), as well as ensuring the integrity of the data transformation and invoice generation processes.

2. Test Scope

- **Data Ingestion**: Validate APIs pulling raw data from Snowflake (CSV/JSON formats).
- **Data Transformation**: Ensure business rules and calculations are correctly applied.
- **Invoice Generation**: Validate correct generation of invoices and pricing models.
- **Error Handling**: Ensure proper error logging and alerts are triggered for discrepancies.
- **Audit & Reporting**: Validate API endpoints generating audit trails and reports.

3. Test Objectives

- **Verify API Integration**: Ensure APIs can successfully retrieve and send data from/to Snowflake and external systems.
- **Validate Data Consistency**: Ensure correct cleansing, transformation, and calculations.
- **Test Invoice Generation**: Confirm invoices are generated correctly based on data and business rules.
- **Error Handling and Logging**: Ensure discrepancies and errors are logged and proper alerts are triggered.
- **Performance and Scalability**: Validate API performance under high data loads, especially with large datasets.
- **Security**: Ensure encryption of data in transit and at rest, as well as proper access controls.

4. Test Categories

a. Functional Testing

Data Ingestion API:

- Test with valid CSV/JSON data.
- Test with invalid, incomplete, or corrupted data to ensure proper error handling.
- Validate real-time data ingestion functionality where applicable.

Data Transformation API:

- Test business rules for data cleansing, normalization, and calculation (e.g., net power generated, total consumption).
- Test edge cases for different data inputs.

Invoice Generation API:

- Test invoice generation for various pricing models and custom rate calculations.
- Verify invoices are properly itemized with correct metrics (e.g., power generated, consumption, and market prices).

Audit & Reporting API:

- Test generation of audit trails for data transformations and invoice creation.
- Ensure reports are exportable in PDF and Excel formats.
- Validate API endpoints for summary and detailed report generation.

Error Handling:

- Simulate data discrepancies to ensure errors are logged and manual review alerts are triggered.

b. Non-Functional Testing

Performance Testing:

- Test API response times with varying dataset sizes (small, medium, large).
- Simulate concurrent requests for data ingestion and invoice generation to validate scalability.

Security Testing:

- Test encryption of data in transit and at rest using industry-standard protocols.
- Verify that access controls are in place for sensitive data (role-based access control).

Reliability Testing:

- Test API uptime and ensure critical processes meet the 99.9% uptime requirement.
- Validate system’s ability to handle high-volume data ingestion and processing.

c. Usability Testing

Interface API Testing:

- Ensure APIs are easy to integrate and the responses are structured in a user-friendly way.
- Validate error messages and logs are clear for non-technical users.

5. Test Data

- **Valid Test Data**: Simulate different data sets from Snowflake (CSV/JSON) representing valid power generation, consumption, market price, and PPA terms.
- **Invalid Test Data**: Include missing values, corrupt files, or unsupported formats (e.g., incomplete CSV, malformed JSON).
- **Edge Case Data**: Extreme values for power generation, consumption, and pricing calculations (e.g., very large numbers, zero values).
- **Security Test Data**: Sensitive data like power generation statistics, consumption data, and PPA terms for security validation.

6. Test Environments

- **API Testing Environment**: Ensure the test environment mimics the production environment with external systems (Snowflake, APIs) properly integrated.
- **Load Testing Environment**: Simulate high-volume API calls and concurrent processing scenarios.

7. Test Tools

- **API Testing Tools**: Postman or SoapUI for manual API testing.
- **Automation Tools**: Cypress for automated API tests (specifically for validating endpoint functionality, response data, and performance).
- **Performance Testing Tools**: Apache JMeter for stress and load testing.

8. Test Schedule

- **Phase 1**: Functional API Testing (Data Ingestion, Transformation, Invoice Generation) – Week 1-2.
- **Phase 2**: Error Handling, Audit Trail, and Reporting Testing – Week 2-3.
- **Phase 3**: Performance and Scalability Testing – Week 4.
- **Phase 4**: Security Testing and Final Validation – Week 5.

9. Test Deliverables

- Test Plan (This document).
- Test Cases and Scenarios.
- Test Results and Logs.
- Defect Logs (if applicable).
- Test Summary Report.

10. Risks and Mitigations

- **Risk**: External system unavailability (e.g., Snowflake).
    - **Mitigation**: Use mock data and APIs for testing.
- **Risk**: Data discrepancies or corruption.
    - **Mitigation**: Thorough validation and error handling in test cases.
- **Risk**: System underperformance during high-load scenarios.

    - **Mitigation**: Conduct extensive load and performance testing in a controlled environment.