Test Plan for Invoicing Module

Objective:

To ensure the Invoicing Module meets its functional and non-functional requirements, delivering accurate, reliable, and scalable solutions for PPA invoice generation.

Scope:

- Validate data ingestion from Snowflake and other external systems.
- Ensure accuracy of data transformations and derived metrics.
- Verify compliance with PPA terms in invoice generation.
- Test performance, scalability, and error handling mechanisms.

Test Strategy:

Testing Types:

Functional Testing: Validate individual functionalities such as data ingestion, transformation, and invoice generation.

Integration Testing: Ensure seamless integration between Snowflake, APIs, and the invoicing system.

Performance Testing: Assess the system's ability to handle large datasets and peak loads.

Usability Testing: Confirm that the user interface is intuitive and meets stakeholder expectations.

Security Testing: Verify that sensitive data is protected during transit and at rest.

Test Levels:

Unit Testing: Validate individual components (e.g., data parsing, calculations).

System Testing: Test end-to-end workflows, from data ingestion to invoice generation.

Regression Testing: Ensure existing functionality is unaffected by updates or new features.

Test Environment:

Hardware Requirements:

Cloud infrastructure with scalable storage and compute resources.

Software Requirements:

- Snowflake for data ingestion.
- APIs for third-party integrations.
- Test automation tools such as Cypress and Postman.

Test Cases:

1. **Data Ingestion:**
    - Verify data is correctly ingested from Snowflake.
    - Test scenarios with missing or corrupt data.
2. **Data Transformation:**
    - Validate transformation logic for different input datasets.
    - Test edge cases for metric calculations.
3. **Invoice Generation:**
    - Ensure invoices comply with PPA terms.
    - Test dynamic pricing models and itemized summaries.
4. **Error Handling:**
    - Verify system behavior with incomplete or invalid data.
    - Confirm error messages and logs are generated as expected.
5. **Performance:**
    - Measure time taken to process large datasets.
    - Test concurrent data processing under peak loads.

Defect Management:

- Defects shall be logged and tracked using tools like Jira.
- Defects shall be categorized by severity and priority.

Test Schedule:

Phase	Start Date	End Date
Test Planning	YYYY-MM-DD	YYYY-MM-DD
Test Case Design	YYYY-MM-DD	YYYY-MM-DD
Test Execution	YYYY-MM-DD	YYYY-MM-DD
Defect Resolution	YYYY-MM-DD	YYYY-MM-DD
Test Closure	YYYY-MM-DD	YYYY-MM-DD

Deliverables:

- Test Plan Document.
- Test Cases.
- Test Execution Reports.
- Defect Logs
- Final Test Summary Report.