Project Documentation: IT Asset Maintenance Forecasting

1.0 Project Overview

The objective of this project is to forecast maintenance needs for the organization's IT assets. This analysis will involve analyzing asset distribution, service history, and maintenance schedules to provide data-driven insights.

The ultimate goal is to optimize maintenance logistics, reduce asset downtime, and manage IT budgets more effectively. The project will be executed using **Python** for data analysis and forecasting, **MS SQL** for data storage and querying, and **Tableau** for interactive visualization.

2.0 Technology Stack

- **Data Analysis & Forecasting:** Python (with Pandas, Matplotlib/Seaborn, Scikit-learn)
- **Database:** Microsoft SQL Server (MS SQL)
- **Visualization:** Tableau

3.0 Task Breakdown by Phase

Phase 1: Data Analysis & Forecasting (Python)

This phase focuses on cleaning the data, performing exploratory analysis to understand current asset status, and building a model to predict future maintenance.

Task 1.1: Data Cleaning & Feature Engineering

- **Objective:** To ensure data quality and create relevant time-based features for analysis.
- **Actions:**
    - ○ Load the `IT_ASSESMENT.xlsx - ITAssets.csv` dataset into a Pandas DataFrame.
    - ○ Check for and handle any missing or null values.
    - ○ Convert `PurchaseDate`, `LastServiceDate`, and `NextServiceDue` columns to proper datetime objects.
    - ○ Engineer new features: `AssetAge` (Current Date - PurchaseDate), `DaysSinceLastService` (Current Date - LastServiceDate), and `DaysUntilDue` (NextServiceDue - Current Date).

Task 1.2: Exploratory Data Analysis (EDA)

- **Objective:** To analyze the distribution of asset types and their service history.
- **Actions:**

    - Use Pandas to calculate and plot the distribution of `AssetType` (e.g., Laptop, Printer, Monitor).
    - Analyze the `Status` column (Working, Under Repair) to find the current failure rate, potentially grouped by `AssetType`.
    - Calculate descriptive statistics for `AssetAge` and `DaysSinceLastService` to understand the average lifespan and service cycle.

Task 1.3: Maintenance Schedule Analysis & Visualization

- **Objective:** To identify and visualize assets that are nearing their scheduled maintenance dates.
- **Actions:**
    - Filter the DataFrame to create a list of 'At-Risk' assets (e.g., where `DaysUntilDue` is less than 30 days).
    - Use Seaborn or Matplotlib to create a bar chart showing the count of assets due for maintenance, grouped by `AssetType` and `Location`.

Phase 2: Database Management & Querying (MS SQL)

This phase involves creating a structured database to store the processed asset data and using SQL to perform efficient data retrieval and aggregation.

Task 2.1: Database Schema Creation

- **Objective:** To design and create a robust database structure for storing the asset and maintenance data.
- **Actions:**
    - Write a SQL script (`CREATE TABLE`) to define a table named `ITAssets`.
    - Specify appropriate data types for each column (e.g., `VARCHAR(50)` for AssetID, `DATE` for date columns, `INT` for calculated day counts).

Task 2.2: Data Ingestion

- **Objective:** To populate the database with the cleaned dataset and engineered features from Phase 1.
- **Actions:**
    - Export the cleaned DataFrame from Python to a new CSV/Excel file.
    - Use the `INSERT` command or the SQL Server Import Wizard to load the data from the CSV/Excel file into the `ITAssets` table.

Task 2.3: Aggregate Querying for Asset Distribution

- **Objective:** To replicate the asset distribution analysis using declarative SQL queries.
- **Actions:**

    - Write a SQL query using COUNT and GROUP BY to get the total number of assets for each AssetType and Location.

Task 2.4: Querying for 'At-Risk' Assets

- **Objective:** To create a granular, reusable query to identify assets requiring immediate maintenance.
- **Actions:**
    - Write a SQL query using the WHERE clause to select all assets where NextServiceDue is between GETDATE() and GETDATE() + 30.
    - ORDER BY the results by NextServiceDue (ascending) to show the most urgent assets first.

Phase 3: Visualization & Dashboarding (Tableau)

The final phase focuses on creating an intuitive and interactive dashboard to communicate maintenance forecasts and asset status to stakeholders.

Task 3.1: Data Source Connection

- **Objective:** To link Tableau to the centralized MS SQL database for live data reporting.
- **Actions:**
    - Open Tableau and establish a new data source connection to the MS SQL Server.
    - Select the database and drag the ITAssets table (or the 'At-Risk' query from Task 2.4) into the data pane.

Task 3.2: Building Core Visualizations

- **Objective:** To create the primary charts that will anchor the maintenance dashboard.
- **Actions:**
    - Create a bar chart showing 'Asset Distribution by Type'.
    - Create a KPI (Key Performance Indicator) text card showing the total count of 'Assets Due for Maintenance (Next 30 Days)'.
    - Create a packed bubble chart showing 'Assets Under Repair', sized by count and colored by AssetType.
    - Create a heatmap showing Asset Type vs Location (Count of Assets).
    - Create a world/map visualization displaying Assets by Location.

Task 3.3: Dashboard Assembly

- **Objective:** To combine the individual visualizations into a single, cohesive dashboard.
- **Actions:**

    - Create a new dashboard.
    - Drag the worksheets onto the canvas.
    - Arrange and format the elements for clarity and visual appeal.

Task 3.4: Implementing Interactivity

- **Objective:** To empower managers to explore the maintenance schedule dynamically.
- **Actions:**
    - Add "Filter" controls to the dashboard based on Location and AssetType.
    - Configure the filters to apply to all worksheets, allowing users to drill down (e.g., see all 'Laptops' in 'Pune' due for service) and have all charts update in real-time.