Project Documentation: Skill Gap Analysis

1.0 Project Overview

The objective of this project is to perform a comprehensive skill gap analysis to identify discrepancies between the current skills of the workforce and the skills required for upcoming projects. This analysis will provide data-driven insights to guide strategic training and development initiatives. The project will be executed using Python for data processing, MS SQL for data storage and querying, and Tableau for interactive visualization.

2.0 Technology Stack

- **Data Analysis:** Python (with Pandas)
- **Database:** Microsoft SQL Server (MS SQL)
- **Visualization:** Tableau

3.0 Task Breakdown by Phase

Phase 1: Data Analysis & Preparation (Python)

This phase focuses on cleaning, analyzing, and visualizing the data using **Pandas** for data manipulation and **Matplotlib/Seaborn** for plotting.

- **Task 1.1: Data Cleaning & Standardization**
  - **Objective:** To ensure data quality and consistency before analysis.
  - **Actions:**
    - Load the `SkillGap.csv` dataset into a **Pandas** DataFrame.
    - Check for and handle any missing or null values.
    - Standardize text in the `CurrentSkill` and `ProjectSkillRequired` columns to a uniform case (e.g., uppercase).
- **Task 1.2: Core Metric Calculation**
  - **Objective:** To quantify the overall skill gap and identify the most critical areas.
  - **Actions:**
    - Use **Pandas** to calculate the overall skill gap rate as a percentage.
    - Filter the DataFrame to find employees with a skill gap.
    - Identify and list the top 5 `ProjectSkillRequired` skills that are most in-demand.
- **Task 1.3: Proficiency Level Analysis & Visualization**
  - **Objective:** To understand and visualize how an employee's proficiency affects their skill gap.
  - **Actions:**

        - Group the data by `ProficiencyLevel` using **Pandas** to calculate the gap rate for each level.
        - Use **Seaborn** or **Matplotlib** to create a bar chart that visually compares the skill gap rates for 'Beginner', 'Intermediate', and 'Advanced' employees.
- **Task 1.4: Skill Transition Heatmap (Advanced Visualization)**
    - **Objective:** To create a powerful visual matrix showing the most common gaps between current and required skills.
    - **Actions:**
        - Use **Pandas** to create a `crosstab` or `pivot_table` of employees with skill gaps, with `CurrentSkill` as the rows and `ProjectSkillRequired` as the columns.
        - Use **Seaborn's heatmap** function to plot this table. The resulting color-coded grid will instantly show the biggest "hotspots" where specific training is needed (e.g., a large number of employees with 'Excel' skills needing to learn 'Python').

Phase 2: Database Management & Querying (MS SQL)

This phase involves creating a structured database to store the cleaned 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 skill data.
    - **Actions:**
        - Write a SQL script (`CREATE TABLE`) to define a table named `EmployeeSkills`.
        - Specify appropriate data types for each column (e.g., `VARCHAR(100)`, `INT`).
- **Task 2.2: Data Ingestion**
    - **Objective:** To populate the database with the cleaned dataset from Phase 1.
    - **Actions:**
        - Export the cleaned DataFrame from Python to a new CSV file.
        - Use the `BULK INSERT` command or the SQL Server Import Wizard to load the data from the CSV file into the `EmployeeSkills` table.
- **Task 2.3: Aggregate Querying for Top Gaps**
    - **Objective:** To replicate the core analysis using declarative SQL queries.
    - **Actions:**
        - Write a SQL query using `COUNT`, `WHERE`, `GROUP BY`, and `ORDER BY` clauses.
        - The query will identify the top 5 `ProjectSkillRequired` from the `EmployeeSkills` table for records where the `Gap` is 'Yes'.
- **Task 2.4: Detailed Proficiency Breakdown Query**
    - **Objective:** To create a granular view of the proficiency levels associated with a specific skill gap.
    - **Actions:**

        - Write a SQL query that, for a given required skill (e.g., 'Python'), counts the number of employees who need it, grouped by their current ProficiencyLevel.

Phase 3: Visualization & Dashboarding (Tableau)

The final phase focuses on creating an intuitive and interactive dashboard to communicate the findings to stakeholders.

- **Task 3.1: Data Source Connection**
    - **Objective:** To link Tableau to the centralized 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 EmployeeSkills table into the data pane.
- **Task 3.2: Building Core Visualizations**
    - **Objective:** To create the primary charts that will anchor the dashboard.
    - **Actions:**
        - In a new worksheet, create a bar chart showing the count of employees for the 'Top 5 Skill Gaps'.
        - In a second worksheet, create a donut chart showing the percentage breakdown of ProficiencyLevel for all employees who have a gap.
        - Create a KPI (Key Performance Indicator) text card showing the overall skill gap percentage.
- **Task 3.3: Dashboard Assembly**
    - **Objective:** To combine the individual visualizations into a single, cohesive dashboard.
    - **Actions:**
        - Create a new dashboard.
        - Drag the worksheets (KPI card, bar chart, donut chart) onto the canvas.
        - Arrange and format the elements for clarity and visual appeal.
- **Task 3.4: Implementing Interactivity**
    - **Objective:** To empower end-users to explore the data dynamically.
    - **Actions:**
        - Add a "Filter" control to the dashboard based on the ProjectSkillRequired dimension.
        - Configure the filter to apply to all worksheets on the dashboard, allowing users to drill down into specific skills and see all related visuals update in real-time.