1. Project Goal:

Analyze patterns in mental health data to identify key factors influencing mental health in various environments, such as workplaces or specific populations, and develop insights that can inform policy-making or support systems.

1. Questions to Explore:
   1. What are the most common mental health issues reported?
   2. How does the environment (e.g., work or home) impact mental health?
   3. What demographic factors correlate with mental health conditions?
2. Tools and Technologies:
   1. Data Preparation and ETL: Python (Pandas) Jupyter Notebook.
   2. Exploratory Data Analysis: Jupyter Notebook, Matplotlib, Seaborn, Plotly, Bokeh(for enhanced interactive visualizations).
   3. Database and Backend Setup: PostgreSQL for data storage, Flask API for data interaction and integration.
3. Data Engineering Workflow

The following steps outline the data engineering workflow that will be implemented in this project to meet the requirements:

1. Data Collection and Preparation:
   1. Import data from CSV file
   2. Clean and transform the data to ensure consistency and completeness.
   3. Convert the data into a format suitable for database ingestion.
2. Database Setup:
   1. Create a SQL database (PostgreSQL) with tables to store structured data.
   2. Establish relationships between tables using primary and foreign keys for optimized queries.
3. ETL (Extract, Transform, Load) Process:
   1. Extract: Retrieve data from various sources, such as mental health surveys and demographics.
   2. Transform: Process the data to remove inconsistencies, format fields, and create calculated columns or features.
   3. Load: Insert the transformed data into the database, ensuring a smooth transition from raw to structured data.
4. Data Analysis and Visualization:
   1. Perform exploratory data analysis (EDA) using Jupyter Notebooks to identify trends, correlations, and insights.
   2. Visualize the data using interactive tools like Plotly, Altair, and Bokeh to create charts for gender distribution, treatment analysis, and other relevant metrics.
5. Flask API Development:
   1. Set up a Flask API to handle database interactions, data retrieval, and server-side processing.
   2. Ensure that data from the database can be accessed and displayed on the web application through API endpoints.

**Deliverables:**

• Data visualizations and analysis results.

• An interactive web application that displays insights from the data.

• Detailed README.md outlining project structure and usage.

• Complete ETL Workflow documentation.

• Final report summarizing the conclusions and recommendations from the analysis.