Smart Factory Monitoring System

Objective

Simulate and visualize a smart factory scenario using modular Python functions. Monitor motor vibration, simulate production performance, and extract analytical insights.

Part A: Vibration Monitoring System

Use modular functions to complete the following tasks:

1. 1. Function: simulate\_vibration\_data()

- Simulate motor vibration values between 2 mm/s and 12 mm/s for 20 time intervals.  
 - Return list of vibration levels and timestamps.

1. 2. Function: determine\_maintenance\_status(vibrations)

- If vibration > 8 mm/s, set status to 'Maintenance Required'  
 - Else, 'Normal'  
 - Return list of statuses.

1. 3. Function: plot\_vibration\_graph(timestamps, vibrations, status)

- Plot a line chart:  
 - X-axis: Time  
 - Y-axis: Vibration (mm/s)  
 - Horizontal orange line at 8 mm/s  
 - Highlight all 'Maintenance Required' points

Part B: Production Line Simulation

Use modular functions to complete the following tasks:

1. 1. Function: simulate\_production\_data()

- Simulate for 50 time steps:  
 - Units Produced: 80–120  
 - Defective Units: 0–10  
 - Return a pandas DataFrame.

1. 2. Function: calculate\_yield\_efficiency(df)

- Add new columns:  
 - Yield = Units Produced - Defective Units  
 - Efficiency = (Yield / Units Produced) \* 100

1. 3. Function: plot\_production\_charts(df)

- Plot:  
 - Line chart: Actual Yield  
 - Line chart: Predicted Yield using rolling average (window=5)  
 - Bar chart: Defective Units

Part C: Analysis

Answer the following in markdown cells:

1. 1. During which time intervals was Maintenance Required most frequently?
2. 2. What is the average production efficiency?
3. 3. What trend is visible in the predicted yield?

**D: Data Export to Excel**

Tasks (Use Functions Wherever Applicable):

1. **Function:** export\_vibration\_data\_to\_excel(timestamps, vibrations, status)
   * Create a DataFrame with columns:
     + Timestamp
     + Vibration (mm/s)
     + Status
   * Save it to vibration\_data.xlsx
2. **Function:** export\_production\_data\_to\_excel(df)
   * Save the complete production DataFrame (with yield and efficiency) to production\_data.xlsx

Submission Guidelines

- Submit: .ipynb file and .pdf export of the notebook& excel files  
- Include:  
 - Modular functions  
 - Headings & comments  
 - Labeled plots  
 - Final markdown cell with observations