Project Proposal:
Real-time IoT Data Pipeline

Project Description:

The Real-time IoT Data Pipeline project developed a system that collects and analyzes sensor data in real-time. Using Python and Azure tools, it generated temperature and humidity readings, cleaned and stored them in a SQL Data Lake, and triggered alerts for unusual values. A Power BI dashboard showed live results and system performance.

Group Members & Roles:

- **Youssef Amgad Youssef (Team Leader)**
  ETL Pipeline, Streaming Pipeline, Dashboard

- **Ekramy Rabie Elsayed**
  ETL Pipeline, Streaming Pipeline

- **Margret Atef Shawky**
  Data simulation, Streaming Pipeline

- **Shaza Alaa Gad Elkareem**
  Data simulation, ETL Pipeline, Report

Objectives:

- Designed and built a real-time IoT data pipeline to collect and process sensor data
- Combined both batch and streaming methods for efficient data handling
- Delivered real-time insights and automatic anomaly detection
- Created an interactive dashboard to visualize key metrics and system performance

Tools & Technologies:

Task	**Tool / Technology**
Programming	Python, Pandas
Database	MS SQL Server

Data Ingestion	Azure Event Hubs
ETL Pipeline	Azure Data Factory
Storage	Azure Data Lake
Streaming Analytics	Azure Stream Analytics
Alerting	Azure Functions
Visualization	Power BI
Reporting	PDF
Presentation	MS Powerpoint

Milestones & Deadlines:

Milestone 1: Data Simulation and Ingestion (1 Week)

Objectives: Simulate IoT data and push it into a file or message queue
Tasks: Create a Python script to generate sensor data (every 5 seconds)
Deliverables:

1. Python generator script
2. Sample data logs

Milestone 2: Batch Data Pipeline (ETL) (2 Weeks)

Objectives: Ingest data, process it, and store it in a data warehouse
Tasks: Use Azure Data Factory to

1. Extract data (CSV or stream)
2. Transform it (e.g., flag anomalies, average)
3. Load into SQL or Data Lake

Deliverables:

1. ADF pipeline
2. Processed dataset in storage

Milestone 3: Streaming Pipeline with Alerts (1 Week)

Objectives: Implement streaming analytics and alerting
Tasks: Use Azure Stream Analytics to

1. Process real-time data
2. Raise alerts for threshold breaches

Deliverables:

1. Streaming pipeline setup
2. Alert logic code and output

Milestone 4: Dashboard & Final Report (1 Week)

Objectives: Visualize metrics and summarize results.
Tasks:

1. Create a real-time dashboard with Power BI
2. Report on key findings and system performance

Deliverables:

1. Dashboard screenshot/live demo
2. Final project report

Timeline & Deliverables:

- **Milestone 1:** 1 Week - Python generator script, Sample data logs
- **Milestone 2:** 2 Weeks - ADF pipeline, Processed dataset in storage
- **Milestone 3:** 1 Week - Strimming pipeline setup, Alert logic code
- **Milestone 4:** 1 Week - Live dashboard, Final project report
- **Total Duration: 5 weeks**

KPIs (Key Performance Indicators):

- **Ingestion reliability:** Delivery success ≥ 99.5% (measure: Event Hubs / ADF success rate).

- **ETL latency:** Cleaned files available in ADLS ≤ 2 minutes from arrival (measure: ADF pipeline logs).
- **Anomaly detection:** Precision ≥ 90% & Recall ≥ 85% (measure: labeled test set comparison).

- **Dashboard readiness:** Load time < 3s & ≥ 90% required metrics visualized (measure: Power BI performance + KPI checklist).

- **Deliverables & feedback:** Report/slides 100% complete; stakeholder score ≥ 4/5 (measure: final checklist + short survey).