Microsoft Data Engineer Projects

Project Instructions for Students: -

The graduation project is a key requirement for obtaining the Digital Egypt Pioneers Initiative Completion Certificate.

- Students are free to choose any of the ideas listed in the project booklet for their respective career track without any restrictions **"With the management of the initiative being duly informed."**, they are able to choose other ideas not listed in the booklet, but it should go in the same format of the ideas given.
- The project is a group assignment, and teams should consist of 4 to 6 students.
- Within a maximum of one week from the announcement of the project booklet, students must form their groups and inform the instructor. If they fail to do so, the instructor has the right to assign groups randomly and announce the team members.
- Students must divide the work responsibilities within the group and inform the instructor within two weeks of the project booklet announcement. During the final presentation, each group must demonstrate the work completed and each member's responsibility for their assigned tasks.
- The final evaluation will be based on the final presentation, which must include the students' adherence to the deliverables and the distribution of tasks among team members.

تعليمات المشروع للطلاب:-

مشروع التخرج هو أحد المتطلبات الأساسية للحصول على شهادة إتمام مبادرة رواد مصر الرقمية.

- يتمتع الطلاب بحرية اختيار أي من الأفكار المدرجة في كتيب المشروع لمسارهم الوظيفي دون أي قيود، أو اختيار أي فكره أخرى غير مدرجه **(مع اعلام إدارة المبادرة بها)**، ولكن بنفس الطريقة المستخدمة في الأفكار المذكورة.
- المشروع عمل جماعي، ويجب أن تتكون فرق العمل من ٤ إلى ٦ طلاب.
- في غضون أسبوع كحد أقصى من إعلان كتيب المشروع، يجب على الطلاب تشكيل فرقهم وإبلاغ المدرب بذلك. في حالة عدم القيام بذلك، يحق للمدرب تقسيمهم بشكل عشوائي وإعلان أعضاء الفريق.
- يجب على الطلاب تقسيم مسؤوليات العمل داخل المجموعة وإبلاغ المدرب بها في غضون أسبوعين من إعلان كتيب المشروع. كما يجب على كل مجموعة خلال العرض النهائي توضيح الأعمال التي تم إنجازها وتحديد مسؤولية كل فرد في تنفيذها.
- سيتم التقييم النهائي بناءً على العرض النهائي، والذي يجب أن يتضمن التزام الطلاب بتسليم المخرجات وتقسيم العمل بين أعضاء الفريق.

✅ Project: Student Performance Dashboard

(Tied to Module 2: Programming Essentials + Module 3: SQL & Database Management)

📌 Project Overview:

This project helps students build a data pipeline that collects, stores, and analyzes student academic performance. The goal is to teach Python basics, file handling, and SQL queries through a meaningful real-world use case. By the end, students will have created a dashboard (even basic) showing student performance trends.

🧩 Milestone 1: Data Collection and Preprocessing

Objectives:
• Learn basic file handling and data preprocessing using Python.

Tasks:

1. **Collect Student Data:**
   - Provide a CSV file (or allow them to create one) with fields like student_id, name, subject, score, date, attendance.
2. **Preprocess Data:**
   - Load CSV using pandas
   - Remove duplicates, fix nulls
   - Convert dates, categorize performance (e.g., High/Medium/Low)

Deliverables:

- Python script to load and clean student data
- Cleaned dataset ready for SQL import

🧩 Milestone 2: SQL Integration & Querying

Objectives:
• Store the data in a SQL database and practice querying

Tasks:

1. Design a normalized relational schema (e.g., Students, Subjects, Scores)
2. Import cleaned data using Python and SQLite or PostgreSQL
3. Write SQL queries to:
   - Get top performers by subject

    - Attendance trends
    - Average scores by month

Deliverables:

- ER diagram
- SQL scripts for table creation and queries
- Query result screenshots or exported data

🧩 Milestone 3: Visualization & Reporting

Objectives:
• Use Python to visualize student performance over time

Tasks:

1. Use matplotlib or seaborn to show:
    - Score trends
    - Attendance heatmaps
2. Optional: Create an interactive dashboard using Streamlit or Plotly Dash

Deliverables:

- Python notebook with visualizations
- Simple dashboard (optional)

🧩 Milestone 4: Final Documentation and Presentation

Objectives:
• Summarize findings and show visual outputs

Tasks:

1. Document:
    - Key patterns in student performance
    - How the dashboard helps teachers or school admins
2. Present:
    - The visual report and live dashboard (if built)

Deliverables:

- Final Report PDF
- Presentation Slides

✅ Final Milestone Summary:

Milestone	**Key Deliverables**
1. Data Preprocessing	Python script, cleaned CSV
2. SQL Integration	SQL schema, queries
3. Visualization	Charts, dashboard
4. Presentation	Report, slide deck

✅ Project: Real-time IoT Data Pipeline

(Tied to Module 5: Data Pipelines + Module 6: Big Data Processing)

📌 Project Overview:

Students will build a pipeline that simulates sensor data (temperature, humidity) and processes it using batch and streaming techniques. This introduces orchestration, real-time analytics, and cloud-native processing.

🧩 Milestone 1: Data Simulation and Ingestion

Objectives:
• Simulate IoT data and push it into a file or message queue

Tasks:

1. Create a Python script to generate sensor data (every 5 seconds)
2. Write to a file or Kafka/Stream (optional)

Deliverables:

- Python generator script
- Sample data logs

🧩 Milestone 2: Batch Data Pipeline (ETL)

Objectives:
• Ingest data, process it, and store it in a data warehouse

Tasks:

1. Use Python or Azure Data Factory to:
    - Extract data (CSV or stream)
    - Transform it (e.g., flag anomalies, average)
    - Load into SQL or Data Lake

Deliverables:

- ETL script or ADF pipeline
- Processed dataset in storage

🧩 Milestone 3: Streaming Pipeline with Alerts

Objectives:
• Implement streaming analytics and alerting

Tasks:

1. Use Azure Stream Analytics or Apache Kafka to:
   - Process real-time data
   - Raise alerts for threshold breaches

Deliverables:

- Streaming pipeline setup
- Alert logic code and output

🧩 Milestone 4: Dashboard & Final Report

Objectives:
• Visualize metrics and summarize results

Tasks:

1. Create a real-time dashboard (Power BI, Streamlit, Grafana)
2. Report on key findings and system performance

Deliverables:

- Dashboard screenshot/live demo
- Final project report

✅ Final Milestone Summary:

Milestone	**Key Deliverables**
1. Data Simulation	Python generator
2. Batch ETL	ETL pipeline
3. Streaming Analytics	Real-time alerts
4. Dashboard & Report	Dashboard + PDF report

✅ Project: Customer Churn Analysis and Retention Strategy

📌 Project Overview:

This project helps students analyze customer behavior to identify patterns that lead to churn (customers leaving a service). Using historical transaction and interaction data, students will build a classification model to predict churn risk, deploy it, and use insights to inform customer retention strategies.

🧩 Milestone 1: Data Collection, Exploration, and Preprocessing

Objectives:

- Understand customer datasets and prepare for modeling

Tasks:

1. **Data Collection:**
   - Use a public or provided dataset with fields like customer_id, tenure, usage, complaints, churn_status
2. **Data Exploration:**
   - Analyze patterns across features (e.g., high complaints → churn)
   - Generate class imbalance reports
3. **Data Preprocessing:**
   - Encode categorical features (e.g., gender, plan type)
   - Handle missing data and scale numeric features

Deliverables:

- EDA Notebook with visualizations (bar charts, pie charts, histograms)
- Cleaned dataset ready for modeling
- Summary report of key customer behavior insights

🧩 Milestone 2: Predictive Model Development

Objectives:

- Train a churn prediction model and evaluate its performance

Tasks:

1. **Model Selection:**
   - Use classification models like Logistic Regression, Random Forest, or XGBoost

2. **Training and Evaluation:**
    - Split data (train/test)
    - Evaluate using Accuracy, Precision, Recall, F1-score, ROC-AUC
3. **Tuning and Interpretation:**
    - Optimize hyperparameters
    - Interpret feature importance (why customers churn)

Deliverables:

- Model evaluation report
- Trained model + training code
- Feature importance plots

🧩 Milestone 3: Deployment and Retention Strategy

Objectives:

- Deploy the model and generate actionable business insights

Tasks:

1. **Deployment:**
    - Build a simple Streamlit or Flask app to input customer data and predict churn risk
2. **Retention Strategy:**
    - Propose data-backed strategies to reduce churn (e.g., offer discounts to high-risk customers)

Deliverables:

- Deployed model interface
- Retention strategy report
- Demo presentation

✅ Project: Product Review Sentiment Analysis

📌 Project Overview:

This project teaches students how to process unstructured text data and use AI to extract sentiment from product reviews. Students will learn data preprocessing, NLP fundamentals, sentiment classification, and how to integrate these insights into a data pipeline.

🧩 Milestone 1: Text Data Preprocessing and Exploration

Objectives:

- Understand and clean text data

Tasks:

1. **Data Source:**
   - Use a dataset of product reviews with text + ratings (e.g., Amazon, Yelp)
2. **Preprocessing:**
   - Clean text (remove punctuation, lowercasing, stop words)
   - Tokenization, lemmatization (using NLTK or spaCy)
3. **Exploration:**
   - Analyze word frequency, length of reviews, rating distribution

Deliverables:

- Preprocessed dataset
- EDA notebook with word clouds, histograms

🧩 Milestone 2: Sentiment Modeling

Objectives:

- Build a model to classify reviews as positive, neutral, or negative

Tasks:

1. **Modeling:**
   - Use logistic regression, SVM, or a pretrained model like BERT or Vader
2. **Training:**
   - Evaluate with F1-score, confusion matrix, classification report

3. **Deployment:**
   - Build a Streamlit app to input a review and get sentiment prediction

Deliverables:

- Model training code + saved model
- App interface for sentiment prediction
- Sentiment dashboard (optional)

🧩 Milestone 3: Integration into Data Pipeline

Objectives:

- Integrate sentiment analysis into broader data workflow

Tasks:

1. Automate the pipeline to analyze incoming reviews (batch or real-time)
2. Store the results in a database
3. Trigger alerts for negative reviews

Deliverables:

- Automated pipeline script
- Database with sentiment-labeled reviews
- Alert system (log or email notification)

✅ Project: Real-Time Traffic Analytics Using Azure Stream Analytics

---

📌 Project Overview:

In this project, students simulate vehicle traffic data and build a real-time analytics system using Microsoft Azure services. This introduces them to streaming data, real-time alerts, dashboards, and integrating data into cloud storage and visualization tools.

🧩 Milestone 1: Traffic Data Simulation and Ingestion

Objectives:

- Generate and stream traffic data into Azure Event Hubs or Blob Storage

Tasks:

1. **Simulation Script:**
   - Create a Python script to simulate vehicle count, speed, timestamp, location
2. **Azure Setup:**
   - Push data to Azure Event Hubs or Blob Storage

Deliverables:

- Python data generator
- Azure Event Hub with data streams

🧩 Milestone 2: Real-Time Processing

Objectives:

- Analyze traffic patterns and detect anomalies

Tasks:

1. **Azure Stream Analytics:**
   - Create real-time queries to detect congestion, high-speed vehicles, or accidents
2. **Output:**
   - Store insights in Azure SQL or Data Lake

Deliverables:

- Real-time query configurations
- Database or data lake with processed insights

🧩 Milestone 3: Dashboard and Alerting

Objectives:

- Visualize live traffic data and trigger alerts

Tasks:

1. **Visualization:**
   - Use Power BI or a Streamlit dashboard to visualize traffic
2. **Alerts:**
   - Send alerts when traffic exceeds defined thresholds

Deliverables:

- Live dashboard screenshots
- Alert log or notification system
- Final report summarizing system architecture

🟩 Summary Table

Project Title	Key Learning Areas	Cloud Tools	Final Outcome
Customer Churn Analysis	Classification, model deployment, business strategy	Streamlit, optional Azure	Predict churn and propose retention
Sentiment Analysis of Product Reviews	NLP, text processing, AI integration	Python, Streamlit	Real-time sentiment analyzer
Real-Time Traffic Analytics	IoT, streaming, cloud pipelines	Azure Event Hubs, Stream Analytics, Power BI	Traffic dashboard + live alerts