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, 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، 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 Idea 1: Customer Data Management and Analysis

Week 1: Data Management and SQL Database Setup

- **Tasks:**
  - **Database Design:** Design a SQL database schema to manage customer data, including tables for customer information, transactions, and interactions.
  - **Implementation:** Create and populate the SQL database using Microsoft SQL Server.
  - **SQL Queries:** Write SQL queries to extract, update, and analyze customer data.
  - **Tools:** Microsoft SQL Server, SQL Management Studio.
- **Deliverables:**
  - A well-designed SQL database schema and populated database.
  - SQL queries for data extraction and basic analysis.

Week 2: Data Warehousing and Python Programming

- **Tasks:**
  - **Data Warehouse Implementation:** Implement a SQL Data Warehouse to aggregate and manage large volumes of customer data for analytical purposes.
  - **Data Integration:** Load data from various sources into the data warehouse.
  - **Python Programming:** Develop Python scripts to interact with the SQL database, perform data extraction, and prepare data for analysis.
  - **Tools:** Microsoft SQL Data Warehouse, Python (Pandas, SQLAlchemy).
- **Deliverables:**
  - A functioning SQL Data Warehouse with integrated data.
  - Python scripts for data extraction and preparation.

Week 3: Data Science and Azure Integration

- **Tasks:**
  - **Data Science with Python:** Perform data analysis and build predictive models (e.g., customer churn prediction) using Python.
  - **Azure Data Fundamentals:** Utilize Azure Data services to manage and analyze customer data.
  - **Model Development:** Develop and evaluate machine learning models using Azure Machine Learning or similar services.
  - **Tools:** Python (Scikit-learn, Matplotlib), Azure Data Studio, Azure Machine Learning.
- **Deliverables:**
  - Analysis report with insights and predictive models.
  - Integrated Azure Data services setup and documentation.

Week 4: MLOps, Deployment, and Final Presentation

- **Tasks:**
  - **MLOps Implementation:** Use MLflow to track experiments and manage machine learning models.

  - **Model Deployment:** Deploy the machine learning model using Azure services or create a web application for model predictions.
  - **Final Report and Presentation:** Prepare a comprehensive report covering data management, analysis, model development, and deployment. Create a presentation to showcase the project results.
  - **Tools:** MLflow, Azure services, web frameworks (e.g., Flask, Streamlit).
- **Deliverables:**
  - Deployed machine learning model or web application.
  - Final report and presentation.

Project Idea 2: Retail Inventory Management and Forecasting

Week 1: SQL Database Design and Implementation

- **Tasks:**
  - **Database Design:** Design a SQL database schema for managing retail inventory, including tables for products, suppliers, sales, and inventory levels.
  - **Implementation:** Create and populate the SQL database with sample data.
  - **SQL Queries:** Develop SQL queries for inventory tracking and sales reporting.
  - **Tools:** Microsoft SQL Server, SQL Management Studio.
- **Deliverables:**
  - SQL database schema and populated tables.
  - Sample SQL queries for inventory and sales analysis.

Week 2: Data Warehousing and Python Integration

- **Tasks:**
  - **Data Warehouse:** Implement a data warehouse to consolidate inventory and sales data for reporting and analysis.
  - **Data Loading:** Use ETL (Extract, Transform, Load) processes to integrate data into the warehouse.
  - **Python Scripting:** Write Python scripts to extract data from the SQL database, perform data cleaning, and prepare it for analysis.
  - **Tools:** Microsoft SQL Data Warehouse, Python (Pandas, SQLAlchemy).
- **Deliverables:**
  - Data warehouse setup and integrated data.
  - Python scripts for data extraction and preparation.

Week 3: Forecasting and Analysis with Azure Integration

- **Tasks:**
  - **Demand Forecasting:** Use Python to build forecasting models for inventory demand (e.g., time series models like ARIMA).
  - **Azure Data Services:** Utilize Azure services for data storage and analysis, such as Azure SQL Database or Azure Synapse Analytics.
  - **Model Evaluation:** Evaluate the performance of forecasting models and make necessary adjustments.
  - **Tools:** Python (Statsmodels, Scikit-learn), Azure Data Studio, Azure Machine Learning.
- **Deliverables:**
  - Forecasting model performance report.
  - Integrated Azure setup for data storage and analysis.

Week 4: MLOps and Deployment

- **Tasks:**
  - **MLOps:** Use MLflow to manage and track forecasting models.
  - **Deployment:** Deploy the forecasting model as a web service or integrate it into a dashboard for inventory management.

  - **Final Report and Presentation:** Document the entire project, including database management, data warehousing, forecasting models, and deployment.
  - **Tools:** MLflow, Azure services, web frameworks (e.g., Flask, Streamlit).
- **Deliverables:**
  - Deployed forecasting model or web app.
  - Final project report and presentation.

Project Idea 3: Customer Feedback Analysis and Improvement

Week 1: SQL Database Setup and Data Collection

- **Tasks:**
  - **Database Design:** Create a SQL database to manage customer feedback, including tables for feedback forms, customer profiles, and feedback categories.
  - **Data Collection:** Import historical customer feedback data into the database.
  - **SQL Queries:** Write queries to extract and summarize feedback data.
  - **Tools:** Microsoft SQL Server, SQL Management Studio.
- **Deliverables:**
  - SQL database schema and populated tables.
  - SQL queries for feedback extraction and analysis.

Week 2: Data Warehouse and Python Data Processing

- **Tasks:**
  - **Data Warehouse:** Implement a data warehouse to aggregate and store feedback data for in-depth analysis.
  - **Data Integration:** Load feedback data into the data warehouse.
  - **Python Processing:** Use Python to clean and preprocess feedback data, including text processing for sentiment analysis.
  - **Tools:** Microsoft SQL Data Warehouse, Python (Pandas, NLTK).
- **Deliverables:**
  - Data warehouse setup with integrated feedback data.
  - Python scripts for data cleaning and preprocessing.

Week 3: Sentiment Analysis and Azure Integration

- **Tasks:**
  - **Sentiment Analysis:** Build sentiment analysis models using Python to classify feedback into positive, neutral, or negative categories.
  - **Azure Integration:** Utilize Azure Data services for enhanced analysis and storage.
  - **Model Evaluation:** Assess sentiment analysis models and refine as needed.
  - **Tools:** Python (Scikit-learn, SpaCy), Azure Data Studio.
- **Deliverables:**
  - Sentiment analysis report with model performance metrics.
  - Integrated Azure setup for feedback data storage and analysis.

Week 4: MLOps and Final Presentation

- **Tasks:**
  - **MLOps:** Track and manage sentiment analysis models using MLflow.
  - **Deployment:** Deploy the sentiment analysis model via a web application or dashboard to visualize feedback trends and insights.
  - **Final Report and Presentation:** Summarize the project work, including database management, data warehousing, sentiment analysis, and deployment.

  - **Tools:** MLflow, Azure services, web frameworks (e.g., Flask, Streamlit).
- **Deliverables:**
  - Deployed sentiment analysis model or web app.
  - Final project report and presentation.

Project Idea 4: Real-Time Traffic Monitoring and Analysis

Week 1: SQL Database and Data Collection

- **Tasks:**
  - **Database Design:** Design a SQL database schema for managing traffic data, including tables for traffic sensors, traffic logs, and incidents.
  - **Data Collection:** Simulate or use real-time traffic data to populate the database.
  - **SQL Queries:** Develop SQL queries for querying and summarizing traffic data.
  - **Tools:** Microsoft SQL Server, SQL Management Studio.
- **Deliverables:**
  - SQL database schema and populated tables.
  - SQL queries for traffic data analysis.

Week 2: Data Warehouse and Python Integration

- **Tasks:**
  - **Data Warehouse:** Set up a data warehouse to handle and analyze large volumes of traffic data.
  - **Data Loading:** Use ETL processes to load traffic data into the warehouse.
  - **Python Scripting:** Write Python scripts to extract traffic data, perform preprocessing, and prepare it for analysis.
  - **Tools:** Microsoft SQL Data Warehouse, Python (Pandas, SQLAlchemy).
- **Deliverables:**
  - Data warehouse setup and integrated traffic data.
  - Python scripts for data extraction and preprocessing.

Week 3: Traffic Analysis and Azure Services

- **Tasks:**
  - **Traffic Analysis:** Perform real-time traffic analysis using Python, including traffic flow, congestion patterns, and incident detection.
  - **Azure Services:** Leverage Azure Data services for scalable data storage and analysis.
  - **Model Development:** Develop models to predict traffic congestion or incidents.
  - **Tools:** Python (Scikit-learn, Matplotlib), Azure Data Studio, Azure Machine Learning.
- **Deliverables:**
  - Traffic analysis report with insights and predictive models.
  - Azure integration setup for traffic data analysis.

Week 4: MLOps and Deployment

- **Tasks:**
  - **MLOps:** Use MLflow to manage and track traffic analysis models.
  - **Deployment:** Deploy the traffic analysis model as a real-time service or dashboard for monitoring traffic conditions.

    - **Final Report and Presentation:** Document the project, including database management, data warehousing, traffic analysis, and model deployment.
    - **Tools:** MLflow, Azure services, web frameworks (e.g., Flask, Streamlit).
- **Deliverables:**
    - Deployed traffic analysis model or dashboard.
    - Final project report and presentation.