Big 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: Unified Batch & Stream Analytics with Flink

Project Overview:

This project teaches students how to design and implement a unified data processing system using Apache Flink. The system will handle both batch and streaming data processing tasks, reflecting the core principle of stream and batch processing convergence. Students will simulate batch datasets and streaming events, process them using Flink APIs, and visualize insights to demonstrate real-time analytics capabilities.

Milestone 1: Batch Processing with Flink

Objectives:

- Ingest a batch dataset and compute summary statistics using Flink's batch API.

Tasks:

1. Prepare a batch dataset (e.g., sales transactions, sensor readings).
2. Load the dataset into Flink's batch environment.
3. Write a Flink batch job to compute aggregates such as total sales, average values, counts, or grouping by category/time period.
4. Validate output correctness and performance.

Deliverables:

- Batch processing Flink job code.
- Sample input dataset and processed output files.
- A brief report explaining the batch job logic and results.

Milestone 2: Stream Processing with Flink

Objectives:

- Process simulated real-time data streams using Flink's streaming API for continuous aggregation and analysis.

Tasks:

1. Create a stream data generator to simulate events such as live sensor data or user clicks.
2. Set up Flink streaming environment and ingest the live data stream.
3. Implement a Flink streaming job to calculate rolling metrics such as moving averages, counts within windows, or anomaly detection.
4. Test the streaming job and validate real-time output.

Deliverables:

- Streaming data generator script/code.
- Flink streaming job implementation.
- Screenshots or logs showing real-time aggregated results.

Milestone 3: Comparative Analysis of Batch vs Stream Processing

Objectives:

- Compare the batch and stream processing outputs to understand latency, accuracy, and use case suitability.

Tasks:

1. Analyze the differences in processing time, output freshness, and complexity between batch and stream jobs.
2. Summarize scenarios where batch processing or streaming is preferable.
3. Prepare visual or tabular comparison of key metrics.

Deliverables:

- Comparative analysis report with charts or tables.
- Discussion of trade-offs and recommendations for different business scenarios.

Milestone 4: Visualization and Final Presentation

Objectives:

- Create a dashboard or visualization of batch and streaming analytics results.
- Present the project end-to-end to demonstrate understanding and application.

Tasks:

1. Use visualization tools such as Apache Zeppelin, Grafana, or Power BI to display analytics results.
2. Integrate batch and streaming outputs in the dashboard for real-time monitoring and historical insights.
3. Prepare a presentation highlighting project goals, architecture, challenges, results, and future extensions.

Deliverables:

- Visualization dashboard screenshots or live demo link.
- Final project presentation slides.
- Documentation summarizing architecture, setup, and learnings.

Final Milestones Summary:

Milestone	Key Deliverables
1. Batch Processing	Batch job code, dataset, batch processing report
2. Stream Processing	Stream job code, data generator, streaming results
3. Comparative Analysis	Analysis report comparing batch vs streaming
4. Visualization & Presentation	Dashboard, presentation slides, project documentation

Conclusion:

This project reinforces fundamental big data concepts by demonstrating how Apache Flink unifies batch and streaming analytics. Students gain hands-on experience building scalable, real-time data processing pipelines, enhancing their readiness for real-world big data developer roles aligned with HCIA and HCIP certifications.

Project: Data Lake Implementation with Azure Data Lake Storage Gen2 and Spark

---

Project Overview:

Build a scalable data lake solution on Azure using Azure Data Lake Storage Gen2 and Spark. This project will focus on ingesting diverse datasets, organizing raw and curated zones, and performing batch data transformations.

Milestone 1: Data Lake Setup & Data Ingestion

Objectives:

- Set up Azure Data Lake Storage Gen2 environment.
- Ingest sample raw data (CSV, JSON) into the data lake.

Tasks:

1. Create Azure Data Lake Storage Gen2 account and containers.
2. Upload raw datasets into the “raw” zone.
3. Use Azure CLI or SDK scripts for automation.

Deliverables:

- Data Lake Storage Gen2 setup documentation.
- Scripts/notebooks for data ingestion.
- Uploaded datasets in raw zone.

Milestone 2: Data Processing & Transformation with Spark

Objectives:

- Use Apache Spark (on Azure Synapse or Databricks) for data cleansing and transformation.

Tasks:

1. Read raw data from the data lake using Spark.
2. Perform transformations: filtering, aggregation, and schema validation.
3. Store cleaned and processed data in a “curated” zone within the data lake.

Deliverables:

- Spark transformation scripts.
- Cleaned datasets in curated zone.
- Transformation report summarizing operations.

Milestone 3: Metadata Management & Data Cataloging

Objectives:

- Implement data cataloging and metadata management for easy data discovery.

Tasks:

1. Integrate Azure Data Catalog or Apache Atlas.
2. Register datasets and create metadata tags.
3. Create documentation of dataset schema and lineage.

Deliverables:

- Metadata catalog setup report.
- Registered dataset entries with tags and descriptions.

Milestone 4: Final Documentation & Presentation

Objectives:

- Document entire data lake lifecycle and demonstrate end-to-end data flow.

Tasks:

1. Prepare architecture diagrams and pipeline descriptions.
2. Present lessons learned, challenges, and potential improvements.

Deliverables:

- Final project report.
- Presentation slides.

Project: Scalable ETL Pipeline with Apache Airflow and Azure Data Factory

---

Project Overview:

Design and implement a scalable ETL pipeline orchestrating data workflows using Apache Airflow and Azure Data Factory, demonstrating batch and event-driven data pipeline management.

Milestone 1: Workflow Design & Airflow Setup

Objectives:

- Set up Apache Airflow environment and design ETL workflows.

Tasks:

1. Install and configure Airflow (local or cloud-based).
2. Define DAGs to automate data ingestion and transformation tasks.
3. Include task dependencies, retries, and failure handling.

Deliverables:

- Airflow environment setup documentation.
- DAG scripts for ETL workflows.

Milestone 2: Data Ingestion & Transformation via Azure Data Factory

Objectives:

- Implement data ingestion and transformation using Azure Data Factory pipelines.

Tasks:

1. Create linked services and datasets in ADF.
2. Design data copy activities from source systems (Blob storage, SQL DB) to target.
3. Add data flow transformations within ADF pipelines.

Deliverables:

- ADF pipeline JSON definitions.
- Screenshots and documentation of pipeline runs.

Milestone 3: Integration & Monitoring

Objectives:

- Integrate Airflow and ADF for end-to-end orchestration and implement monitoring.

Tasks:

1. Trigger ADF pipelines from Airflow using REST API or Azure operators.
2. Set up alerts and logging for pipeline failures and performance metrics.
3. Implement retry strategies and failure notifications.

Deliverables:

- Integration scripts/code.
- Monitoring and alert configuration report.

Milestone 4: Final Presentation & Best Practices

Objectives:

- Showcase pipeline architecture and discuss operational best practices.

Tasks:

1. Present ETL pipeline workflow with challenges and solutions.
2. Share recommendations for scalability, security, and cost optimization.

Deliverables:

- Presentation deck.
- Final project summary report.

Project: AI-Enhanced Data Pipeline for Customer Churn Prediction

---

Project Overview:

Create a data pipeline that integrates AI models to predict customer churn. Use Python for data processing, SQL for feature extraction, and deploy a pre-trained AI model for prediction.

Milestone 1: Data Collection & Feature Engineering

Objectives:

- Collect customer data and prepare features for churn prediction.

Tasks:

1. Acquire customer behavioral and transaction datasets.
2. Use SQL queries to extract features such as usage frequency, complaints, payment history.
3. Clean and preprocess data with Python (handle missing data, normalize).

Deliverables:

- Feature dataset in CSV or database table.
- Python preprocessing scripts.
- SQL query scripts for feature extraction.

Milestone 2: AI Model Integration

Objectives:

- Integrate a pre-trained churn prediction model into the pipeline.

Tasks:

1. Load a pre-trained model (e.g., XGBoost, Random Forest) using Python.
2. Develop inference scripts to predict churn probability for each customer.
3. Validate prediction results against a test dataset.

Deliverables:

- Model inference code.
- Evaluation report with accuracy, precision, recall metrics.

Milestone 3: Pipeline Automation & Deployment

Objectives:

- Automate the pipeline execution and deploy as a service or scheduled job.

Tasks:

1. Create a workflow with tools like Apache Airflow or Azure Data Factory to automate feature extraction and prediction.
2. Deploy the model and pipeline as a REST API using Flask or FastAPI.
3. Test end-to-end execution and API responsiveness.

Deliverables:

- Workflow automation scripts or DAGs.
- Deployed REST API endpoint.
- API usage documentation.

Milestone 4: Monitoring & Reporting

Objectives:

- Implement monitoring for model performance and generate business reports.

Tasks:

1. Set up logging to track prediction results and model drift indicators.
2. Create dashboards (e.g., Power BI, Grafana) showing churn trends and predictions.
3. Document monitoring setup and insights.

Deliverables:

- Monitoring configuration and logs.
- Business intelligence dashboard screenshots.
- Final project report and presentation.