Project #2: Agentic data work flow:

Assigned to: Ahmed Shaheer, Abbas Bukhari

Background:

Data engineering is a critical function for modern organizations, enabling the seamless flow of data from sources to sinks for analytics, reporting, and machine learning. However, building and managing data pipelines is a complex, multi-stage process requiring expertise in tools, configurations, and logical workflows. Traditionally, data engineers rely on manual methods to select tools, define workflows, and configure pipelines. This approach is time-consuming and prone to errors. An Agentic data workflow system offers a promising alternative, automating pipeline construction by systematically prompting users for input and leveraging a knowledge-driven approach to select and configure tools for each stage of the pipeline.

Problem statement:

Building data pipelines involves multiple stages, including source integration, transformation, orchestration, and sink configuration.

These stages present challenges such as: Tool Selection Complexity, Fragmented Workflows, Lack of Guidance, Error-Prone Manual Processes. This project seeks to address these challenges by creating a deterministic, agent-driven system that guides users through pipeline creation with structured prompts and ensures the pipeline is logically designed and correctly configured.

This project aims to develop a deterministic agentic system that simplifies pipeline creation by:

Capturing the entire scenario via user input in structured forms.
Using agents to recommend and shortlist tools based on the provided scenario.
Enabling a human-in-the-loop process to finalize tools and configurations.
Building a logical pipeline using finalized tools and asking additional questions when necessary.
Setting up tools for the pipeline based on the logical plan, with step-by-step guidance.

Tasks:

a. Scenario Agent:
   Create a form-based system to capture the pipeline requirements, including:
   Data source types (e.g., databases, APIs (optional), file systems).
   Data transformation needs (e.g., batch processing, streaming (optional)).
   Data sinks (e.g., dashboards, storage, multi zones (optional)).
   Non-functional requirements (e.g., scalability, latency, cost constraints).

b. Tool Selection / Logical plan Agent:
   Develop an agent to analyze the scenario

Select a list of tools for each stage of the pipeline (source integration, transformation, orchestration (optional), sinks).
Provide reasoning for the tool recommendations.
Shortlist tools based on compatibility and scenario-specific requirements.
Generate a logical pipeline plan based on the finalized tools.
Prompt users with additional questions if specific configurations are required.
(optional) Visualize the logical pipeline (e.g., a flowchart showing data flow and tool interactions).

c. Tool Setup Agent:
   Configure each tool as per the logical pipeline plan.
   Ask tool-specific configuration questions (e.g. what ports, IPs, DNS, Admin, password etc).
   Validate the configurations to ensure alignment with the logical plan.

d. Defined Questionnaire System and Human-in-the-Loop for each stage finalization:
   Design a system where users can (where applicable):
      Review the shortlisted tools.
      Tool selection and finalization.
      Make changes to the scenario or tool selection if needed.
   Develop structured questionnaires for:
      Scenario gathering.
      Tool setup and specific configurations can be mentioned.

e. Testing and Validation:
   Test the system with various* pipeline scenarios to ensure:
      The agents select and configure tools accurately.
      Users receive appropriate prompts and guidance at each stage.

f. Documentation and Deployment:
   Provide detailed documentation, including:
      System architecture and design.
      User guides for operating the system.
      Deploy the system as a web-based or desktop application.

Deliverables:

a. A fully functional system capable of automating data pipeline creation using structured prompts and agent-driven decisions.
b. A user-friendly form (predefined, reusable questionnaires for capturing inputs at various stages) to capture pipeline requirements in detail and mechanism for users to review and modify tool selections.

c. Comprehensive documentation software design diagrams, covering system functionality, installation, and usage.

Constraints:

Data inputs: sheets, csv, json, xmls, structured, semi-structured, graphs
Ingestion tools: Sqoop, Python
Storage: Postgres, MySQL, Hadoop standalone, MongoDB, Cassandra, Neo4j
Processing: Spark – Visualization: Superset – Orchestration: Airflow, Cron