Functional Programming with Scala

Project Title: Test Project

Group Name: My Group : 3

Addou, Amel

IG5_Polytech

November 19, 2024

About the Project

- **Your Specific Contribution:**
  - Implemented data extraction and cleaning for CVE JSON files, focusing on nested data structures.
  - Developed a pipeline to filter relevant CVE data (e.g., ID, Description, Scores).
  - Built and populated a Neo4j graph database to visualize relationships within the CVE data.
  - Created a basic account on MongoDB Atlas for data storage but focused the analysis and queries in Neo4j.

- **Methodology or Approach:**
  - Used ‘json4s‘ in Scala to parse and clean complex JSON structures.
  - Designed a graph data model for Neo4j to store CVE nodes with their attributes (e.g., ID, Severity, Scores).
  - Inserted data into Neo4j using the Bolt protocol and Cypher queries via the Neo4j Java Driver.
  - Implemented queries to explore CVE relationships and identify trends.

Challenges and Learning

- **Challenges Faced and Their Resolutions:**
  - Challenge: Cleaning and extracting nested JSON data structures.
  - Resolution: Used Scala case classes with ‘json4s‘ to map nested fields into a structured format.
  - Challenge:Creating an efficient graph model for Neo4j.
  - Resolution: Focused on a node-centric model, treating each CVE as a node with attributes, and ensuring all data was correctly indexed.
  - Challenge: Debugging Neo4j connection issues during data insertion.
  - Resolution: Ensured the Neo4j Bolt protocol was configured correctly and used appropriate session management in Scala.
- **Learning Gained:**
  - Mastered the use of graph databases (Neo4j) for data analysis.
  - Learned how to clean and process real-world JSON datasets using Scala.
- **Future Improvements:**
  - Automate data updates in Neo4j with real-time ingestion from JSON feeds.
  - Expand the Neo4j graph model to include relationships between CVEs and their references.