**Code Modernization using GenAI (Cobol to ETL Conversion)**

**Objective:**

* Use GenAI to generate pseudocode and comments for each division and procedure in the Cobol program.
* Convert the pseudocode into ETL code in Python.
* Achieve code modernization from legacy Cobol to cloud based ETL code/Python.
* Demonstrate the benefits of using GenAI and ETL for code modernization.
* Showcase the scalability, cost-effectiveness, and ease of understanding and support provided by the modernized code.
* Create a prototype that can be leveraged for future modernization efforts in other legacy codebases.

**Expected Output:**

* Python code generated based on the pseudocode generated by GenAI.
* The Python code should replicate the functionality of the original Cobol program.
* Well-documented Python code with comments for each division and procedure, ensuring ease of understanding and support.
* Successful migration of the legacy Cobol code to a cloud based ETL solution.
* Improved scalability and flexibility of the modernized code.
* Cost savings in terms of time and effort required to understand and debug the code.
* A prototype that demonstrates the potential of using GenAI and ETL for code modernization.
* A foundation for future modernization efforts in other legacy codebases.

**Criteria:**

To successfully complete this hackathon challenge, participants should focus on the following key points:

1. Utilize GenAI:
   1. Input the Cobol program into GenAI to generate pseudocode.
   2. The pseudocode should provide a summary of each step in the Cobol program, capturing the logic and functionality.
2. Generate Comments for Each Division and Procedure:
   1. Enhance the pseudocode generated by GenAI by adding comments for each division and procedure in the Cobol program.
   2. The comments should provide additional context and explanation for each section of the code, making it easier to understand and support.
3. Convert Pseudocode to ETL Code in Python:
   1. Transform the pseudocode generated by GenAI into ETL code in Python.
   2. The ETL code should replicate the functionality of the original Cobol program.
4. Well-Documented Python Code:
   1. Ensure that the generated Python code is well-documented with comments.
   2. Follow best practices for code readability and maintainability.

**Business Benefits:**

The adoption of this code modernization approach using GenAI and ETL offers several business benefits, including:

1. Mainframe Offloading:
   1. Migrating legacy Cobol code to a cloud based ETL solution reduces dependency on mainframe systems.
   2. This leads to cost savings and improved performance.
2. Cost Effectiveness:
   1. Utilizing GenAI to generate pseudocode and comments helps in understanding and debugging the code more efficiently.
   2. This saves time and effort, resulting in cost savings during the modernization process.
3. Scalability and Flexibility:
   1. The migration to a cloud based ETL solution provides scalability and flexibility.
   2. Organizations can handle large volumes of data and adapt to changing business needs.
4. Ease of Understanding and Support:
   1. The migration to a cloud based ETL solution provides scalability and flexibility.
   2. Organizations can handle large volumes of data and adapt to changing business needs.
5. Prototype for Future Modernization:
   1. The successful implementation of this prototype can serve as a foundation for modernizing other legacy codebases, such as FOCUS and PL1 codes, using similar approaches.

**Tools/Technology:**

The following tools and technologies will be utilized for this challenge:

* GenAI: will be used to input the Cobol program and generate pseudocode and comments.
* Python: will be the programming language used for generating the ETL code using the pseudocode generated by GenAI. Python is a widely used, open-source language known for its simplicity and readability.