GenAI Chatbot

# Project Overview:

* For the capstone project, you will build a chatbot application utilizing Retrieval Augmented Generation (RAG).
* Chat Interface will be provided by Streamlit
* The core functionality will be achieved via LangChain Agent, an open-source LLM equipped with tools.
* Some tools will retrieve further information and provide additional context to LLM for generation. (i.e. retrieve data from document, web, database, etc.)
* One of the tools will be retrieving information from the vector embeds stored in a vector database.
* The Agent will have a memory of the chat and can recall earlier conversations had the user in the current session.

# Functional Requirements:

1. Create a Streamlit project and import LangChain
2. Obtain a vector database and documents to load.
3. Load and split the documents, and generate the vector embeds utilizing an embedding model. Store the embeds in the vector database.
4. Create a tool in LangChain to access the vector database for RAG.
   1. Provide the system prompt to the tool to further refine its behavior.
5. Create a LangChain agent and equip the agent with the earlier tool
   1. Provide the system prompt to the agent to give it a persona and influence its behavior.
   2. Provide tool description to let agent know when to use the tool.
6. Create a conversation buffer memory and provide it to the agent so it has context of prior conversation.
7. Create a simple chat interface in Streamlit so users can interact with the agent.

# Architecture:

# Technology Stack: A screenshot of a computer Description automatically generated

1. LLM

* LLM trained and deployed in GCP Vertex AI

1. Orchestrator:

* Langchain

1. RAG:

* Retrieval
  + Loaders
    - AWS Textract (Any type of document)
    - CSV Loader
    - Directory Loader
    - PDF loaders
    - HTML Loaders
    - JSON Loader
    - Markdown loader
  + Transformers
  + Vector Database
    - Pinecone
    - Chroma DB
  + Embedding Model
    - gte-large from Hugging Face
* Generation
  + LLM chain with Prompt Engineering
* Apache Airflow for the document embedding process.

1. AI Tools

* Document retrieval using Vector DB
* Document retrieval from GCP BigQuery
* Search Engine tool

1. Agent

* Conversational Memory
* Persisting the Conversational Memory