GenAI Stack Demo - Chat with Multiple Websites

Installation

In [ ]:
```
!pip install genai_stack
```

Setup your API Key

In [ ]:
```
import os
from getpass import getpass
```

In [ ]:
```
api_key = getpass("Enter OpenAI API Key:")
os.environ['OPENAI_API_KEY'] = api_key
```

Import required modules

In [ ]:
```
from genai_stack.stack.stack import Stack
from genai_stack.etl.langchain import LangchainETL
from genai_stack.embedding.langchain import LangchainEmbedding
from genai_stack.vectordb.chromadb import ChromaDB
from genai_stack.prompt_engine.engine import PromptEngine
from genai_stack.model.gpt3_5 import OpenAIGpt35Model
from genai_stack.retriever.langchain import LangChainRetriever
from genai_stack.memory.langchain import ConversationBufferMemo
```

ETL - "Extract, Transform, and Load." Add your data here. Check documentation for the required loaders

In [ ]:
```
etl = LangchainETL.from_kwargs(name="WebBaseLoader",
                               fields={"web_path":
                                "https://huggingface.co/google/
                           }
)
```

Create Embeddings to store in VectorDB

In [ ]:
```
config = {
    "model_name": "sentence-transformers/all-mpnet-base-v2",
    "model_kwargs": {"device": "cpu"},
    "encode_kwargs": {"normalize_embeddings": False},
}
embedding = LangchainEmbedding.from_kwargs(name="HuggingFaceEmb
```

Define the VectorDB

In [ ]:
```
chromadb = ChromaDB.from_kwargs()
```

Define your LLM - Large Language Model

In [ ]:
```
llm = OpenAIGpt35Model.from_kwargs(parameters={"openai_api_key"
```

Add Retrieval and Stack all the components

In [ ]:
```
prompt_engine = PromptEngine.from_kwargs(should_validate=False)
retriever = LangChainRetriever.from_kwargs()
memory = ConversationBufferMemory.from_kwargs()
Stack(
    etl=etl,
    embedding=embedding,
    vectordb=chromadb,
    model=llm,
    prompt_engine=prompt_engine,
    retriever=retriever,
    memory=memory
)
```

Run your ETL

In [ ]:
```
etl.run()
```

In [ ]:
```
prompt1 = "what is the google/gemma-7B?"
```

In [ ]:
```
response = retriever.retrieve(prompt1)
```

In [ ]:
```
print(response['output'])
```