Project Report

Project Title: LangChain-based Document QA Application (Task 2: AI-Powered Virtual Analyst)

Objective:

To create an interactive chatbot that processes documents, retrieves relevant content, and answers user queries effectively using LangChain tools and Azure OpenAI models.

Key Features:

1. **Document Loading**:
   Utilizes PyPDFLoader to load annual report PDFs for processing.
2. **Text Splitting**:
   Applies RecursiveCharacterTextSplitter to divide large documents into manageable chunks of 1000 characters with 200-character overlap for efficient embedding.
3. **Vectorization and Retrieval**:
   Implements Chroma for embedding the document chunks using Azure OpenAI embeddings, enabling fast and accurate retrieval.
4. **Natural Language Understanding**:
   Leverages the Azure GPT-4 model with a tailored prompt to generate concise and context-aware answers.
5. **User Interaction**:
   Integrates Streamlit for a clean and dynamic UI, allowing users to input questions and receive immediate responses.

Architecture:

1. **PDF Preprocessing**:
   - Load documents using PyPDFLoader.
   - Split into smaller text chunks.
2. **Embedding and Storage**:
   - Generate embeddings using Azure OpenAI embeddings.
   - Store in a Chroma vector database.
3. **Retrieval-Augmented Generation (RAG)**:
   - Use a retriever to fetch relevant document chunks.
   - Combine them with user queries in the LLM chain.
4. **Streamlit Chat Interface**:

    - Chat-based interface for interactive Q&A.

Tools & Technologies:

- **LangChain** for text processing and chaining.
- **Azure OpenAI GPT-4** for NLP tasks.
- **Chroma** for vector search and retrieval.
- **Streamlit** for UI/UX.

Future Enhancements:

1. Multi-document processing.
2. Enhanced summarization capabilities.
3. Advanced visualization of results.