Report

Approach Taken:

- **Frameworks Used:** LangChain and Streamlit were selected for building the AI chatbot. LangChain likely served as the framework for integrating the Retrieval-Augmented Generation (RAG) model, while Streamlit was utilized for creating the user interface and deploying the application.
- **Functionality:** The chatbot was designed to incorporate Retrieval-Augmented Generation (RAG), a technique that enhances response accuracy by retrieving relevant documents before generating responses. This approach ensures that the chatbot's answers are not only contextually relevant but also accurate based on retrieved information.

Challenges Faced:

- **Complexity of RAG:** Implementing RAG can be challenging due to the need for efficient retrieval mechanisms and seamless integration with generative models. This complexity arises from:
- **Retrieval Efficiency:** Retrieving relevant documents quickly and accurately can be computationally intensive, especially when dealing with large datasets.
- **Integration with Generative Models:** Ensuring that the retrieved information effectively enhances the generative model's responses without compromising response time or accuracy.

Overcoming the Challenges:

To overcome the complexities associated with RAG implementation, I employed:

Iterative Testing and Validation: Iteratively testing the chatbot's performance against diverse datasets and scenarios to validate the accuracy and efficacy of RAG implementation. This iterative approach allows for continuous refinement and improvement of the chatbot's capabilities.