Computer Market Hub - AI Intern Assignment

a. Overall approach:

The overall approach to creating this chatbot involves integrating several advanced technologies:

1. **Data Preparation**: The chatbot uses **Retrieval-Augmented Generation (RAG)** to combine information retrieval with text generation. It starts by loading and splitting a PDF document into manageable chunks, which are then embedded and stored using **Chroma**.

2. **Model Integration**: For generating and understanding text, the chatbot utilizes **Llama 3**, an advanced Large Language Model (LLM) provided by **Ollama**. The Llama 3 model is open-source and completely free, making it accessible for development and ensuring high-quality text generation without licensing costs.

3. **User Interaction**: The chatbot features a user-friendly web interface built with **Streamlit**. This allows users to input questions and receive responses based on the stored document content.

4. **Response Generation**: When a user asks a question, the chatbot searches the document chunks for relevant information and generates a detailed response using the Llama 3 model. If the question is outside the document's scope, it suggests contacting the business.

By leveraging RAG for enhanced response accuracy, Llama 3 for powerful text generation, and the open-source nature of Llama 3 for cost-effective development, this approach ensures an effective and efficient chatbot solution.

b. Frameworks/libraries/tools:

Here's a summary of the frameworks, libraries, and tools used in making this chatbot, highlighting the advanced technologies and the significance of Llama 3 being open-source and free:

1. **Langchain**
    - **Used For**: Integrating document processing, text splitting, and embedding functionalities.
    - **Purpose**: Manages how documents are split into chunks and processed for embedding.
2. **Chroma**
    - **Used For**: Storing and retrieving vector embeddings.
    - **Purpose**: Enables efficient similarity search and retrieval of relevant document chunks.
3. **Ollama**
    - **Used For**: Providing the Llama 3 model for text generation and embeddings.
    - **Purpose**: Powers the chatbot’s ability to generate and understand natural language.
4. **Llama 3**
    - **Used For**: Generating responses based on document content and user queries.
    - **Significance**: As an open-source and completely free Large Language Model (LLM), Llama 3 offers high-quality text generation without licensing fees, making advanced AI accessible for development.
5. **Streamlit**
    - **Used For**: Creating a web interface for user interaction.
    - **Purpose**: Provides a user-friendly platform for asking questions and receiving answers.
6. **Retrieval-Augmented Generation (RAG)**
    - **Used For**: Combining information retrieval with text generation.

    - **Purpose**: Enhances the chatbot's ability to provide accurate and contextually relevant responses by integrating document search with language generation.
7. **Python Libraries**
    - **Used For**: Various utilities and integrations.
    - **Purpose**: Facilitates the implementation of the chatbot's core functionalities and interactions.
    -

Significance of Llama 3:

Llama 3's open-source and free status significantly reduces development costs and provides access to advanced language processing capabilities without the constraints of licensing fees. This makes it an ideal choice for integrating powerful text generation and understanding features into the chatbot, ensuring high performance and accessibility for developers.

c. Problems faced:

During the development of the chatbot, I ran into problems with keeping track of conversation history. I initially used Langchain's ConversationBufferMemory and ConversationChain to handle this. However, this approach caused errors and didn't work as expected.

Problem Encountered

- **Langchain ConversationBufferMemory**: This tool was supposed to store the conversation history. However, it had trouble working with ConversationChain, which led to problems with keeping the conversation context correct. This caused inconsistencies in the chatbot's responses.
- **Langchain ConversationChain**: This was meant to manage how the conversation flowed. When combined with ConversationBufferMemory, it led to failures and errors, making it hard for the chatbot to provide clear and relevant responses.

d. Future scope of this chatbot:

Here are some ideas for enhancing the chatbot in the future:

1. **Multi-Document Support**: Enable the chatbot to handle and retrieve information from multiple documents or sources.
2. **Natural Language Understanding**: Improve the chatbot's ability to understand and process more complex queries.
3. **Voice Interaction**: Add voice input and output capabilities for a more interactive experience.
4. **Personalization**: Incorporate user profiles to provide customized responses based on user history and preferences.