**Project Report**

**Computer Science B**

**By Yin Sannylain**

**Directed By: Dr. Abdallah Altrad**

**Document Q&A System with Cohere and Pinecone**

**Introduction**

In the era of information overload, efficiently extracting relevant data from large documents is a critical challenge. This project presents a document-based question-answering (Q&A) system that leverages Cohere for natural language understanding and Pinecone for vector database indexing. The system enables users to upload PDF documents, extract text, and query them for precise answers based on context. By utilizing document embeddings and semantic search, the system ensures accurate, contextually relevant responses to user queries.

**Problem Solved**

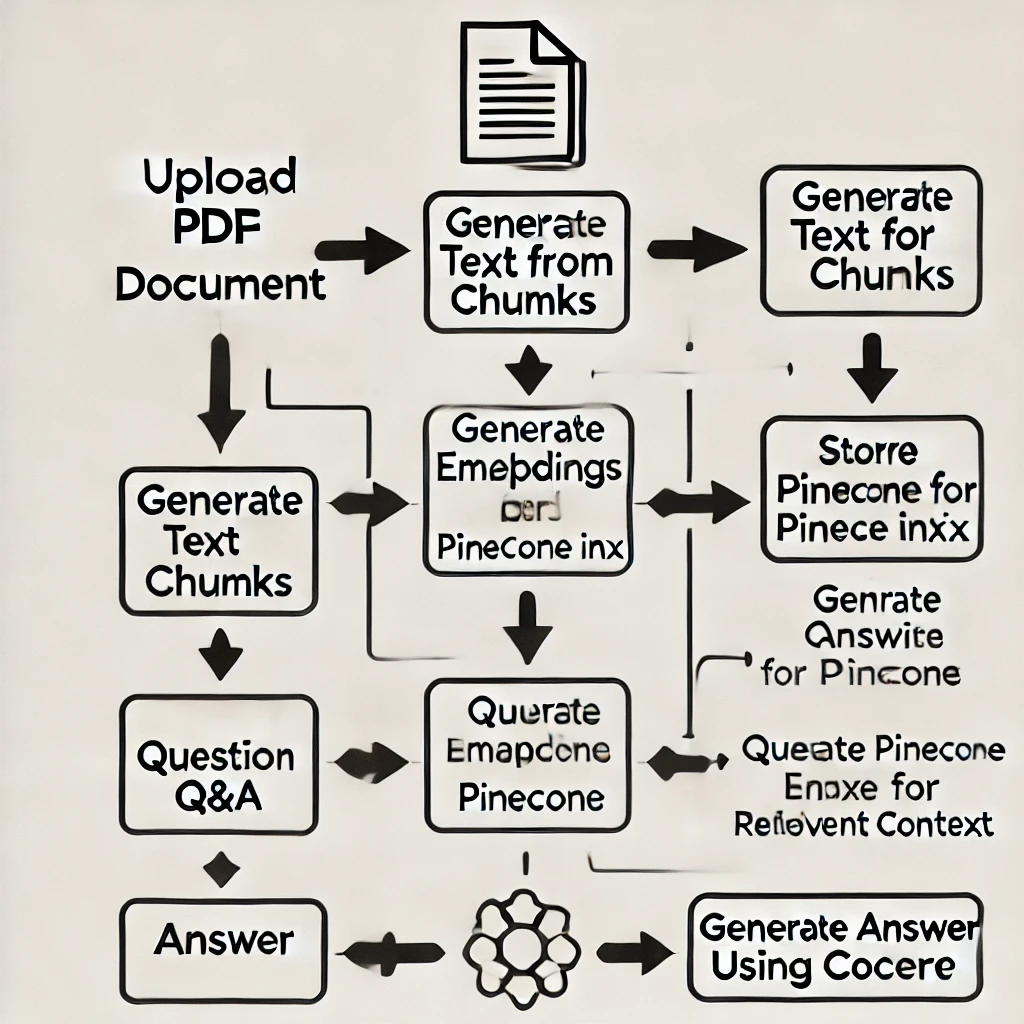
The system addresses the difficulty of extracting relevant information from large PDFs and answering questions based on their content. By combining machine learning models with semantic search, it automates the process of document understanding and response generation, allowing users to interact with documents through natural language queries. This improves the efficiency of document searches and enhances user experience by providing quick, context-aware answers.

**Similar Projects:**

* **Haystack**: A robust NLP framework designed for building search systems, including document search and Q&A.
* **GPT-3-powered Q&A Systems**: Many applications employ GPT-3 to answer questions by embedding documents and querying them efficiently.
* **Pinecone and LangChain Q&A Solutions**: Several developers have integrated Pinecone’s vector database with LangChain for document search and question-answering tasks.

**Application Concepts, Models, and Functions:**

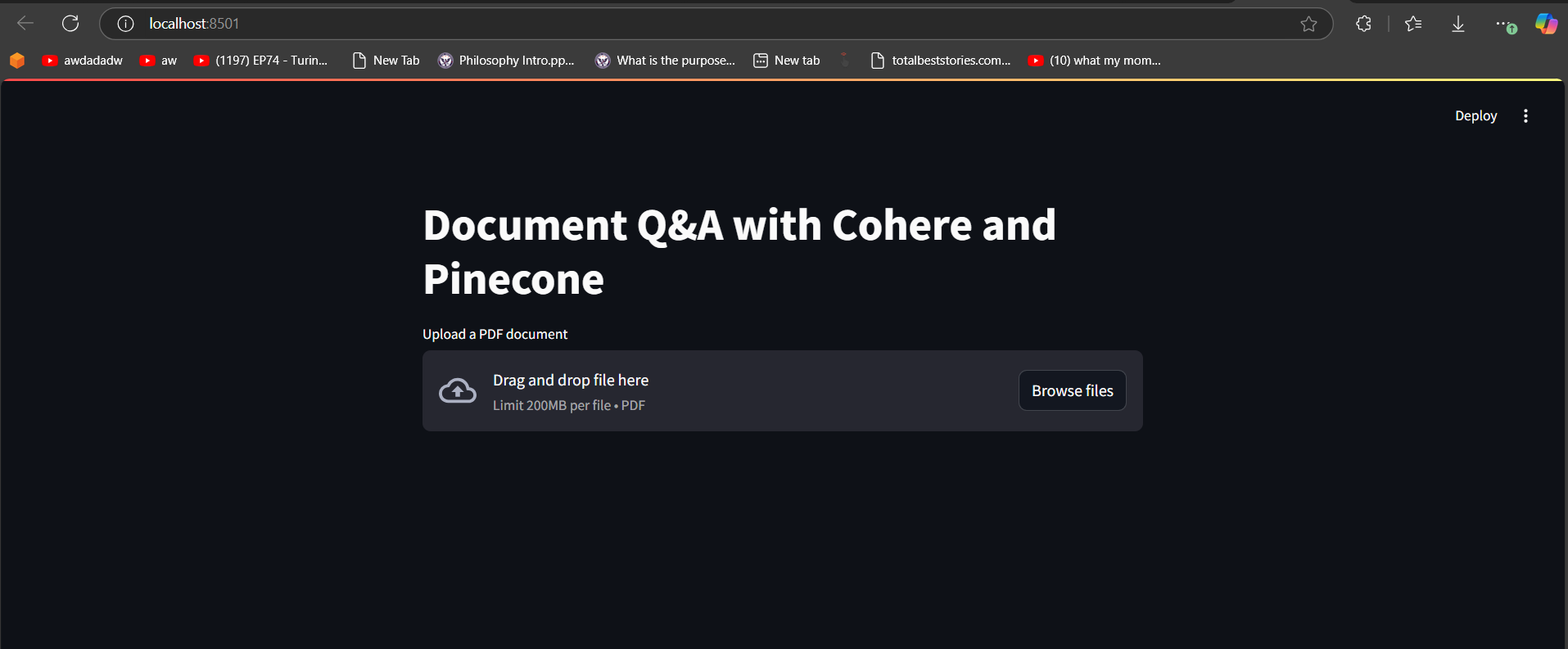
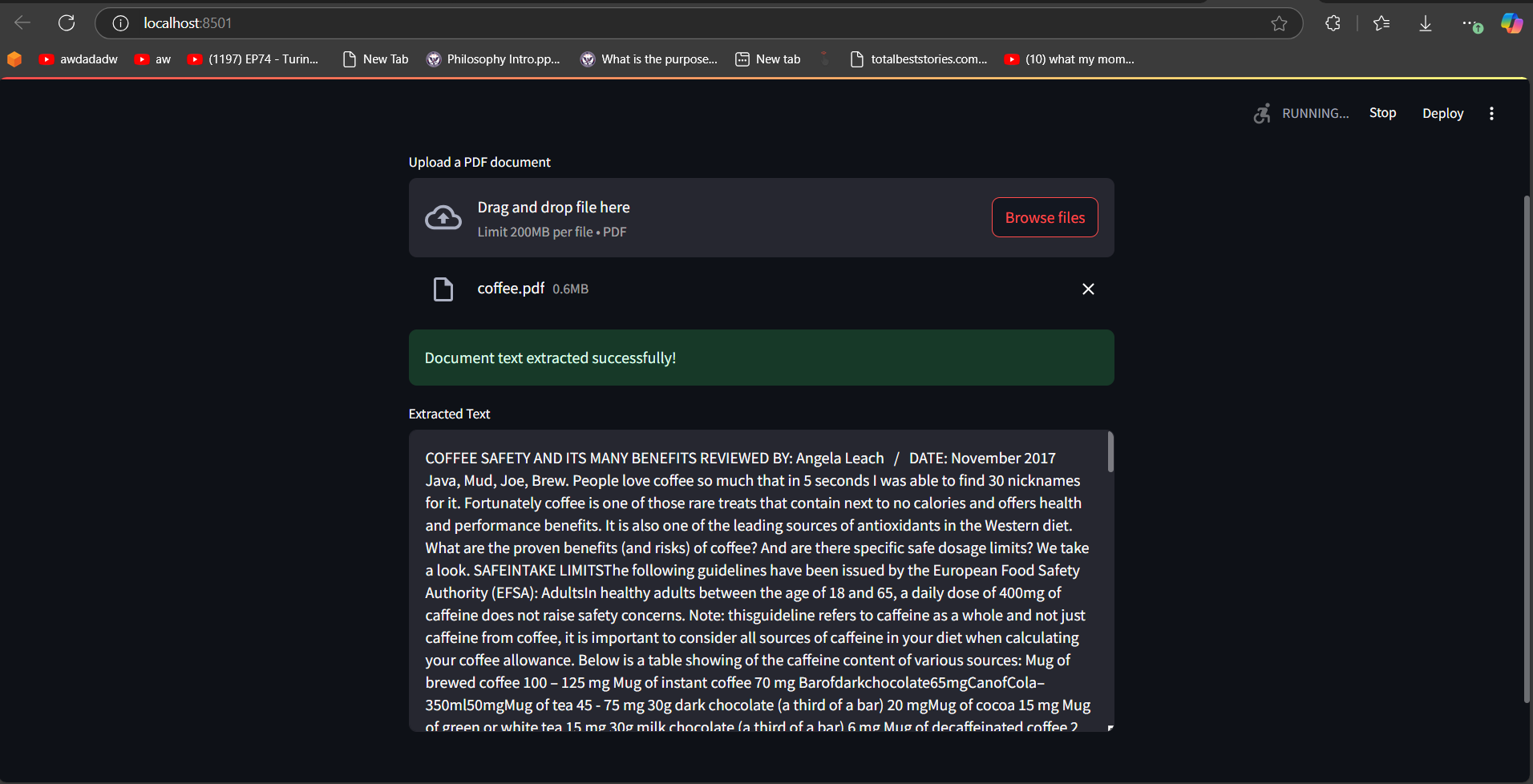
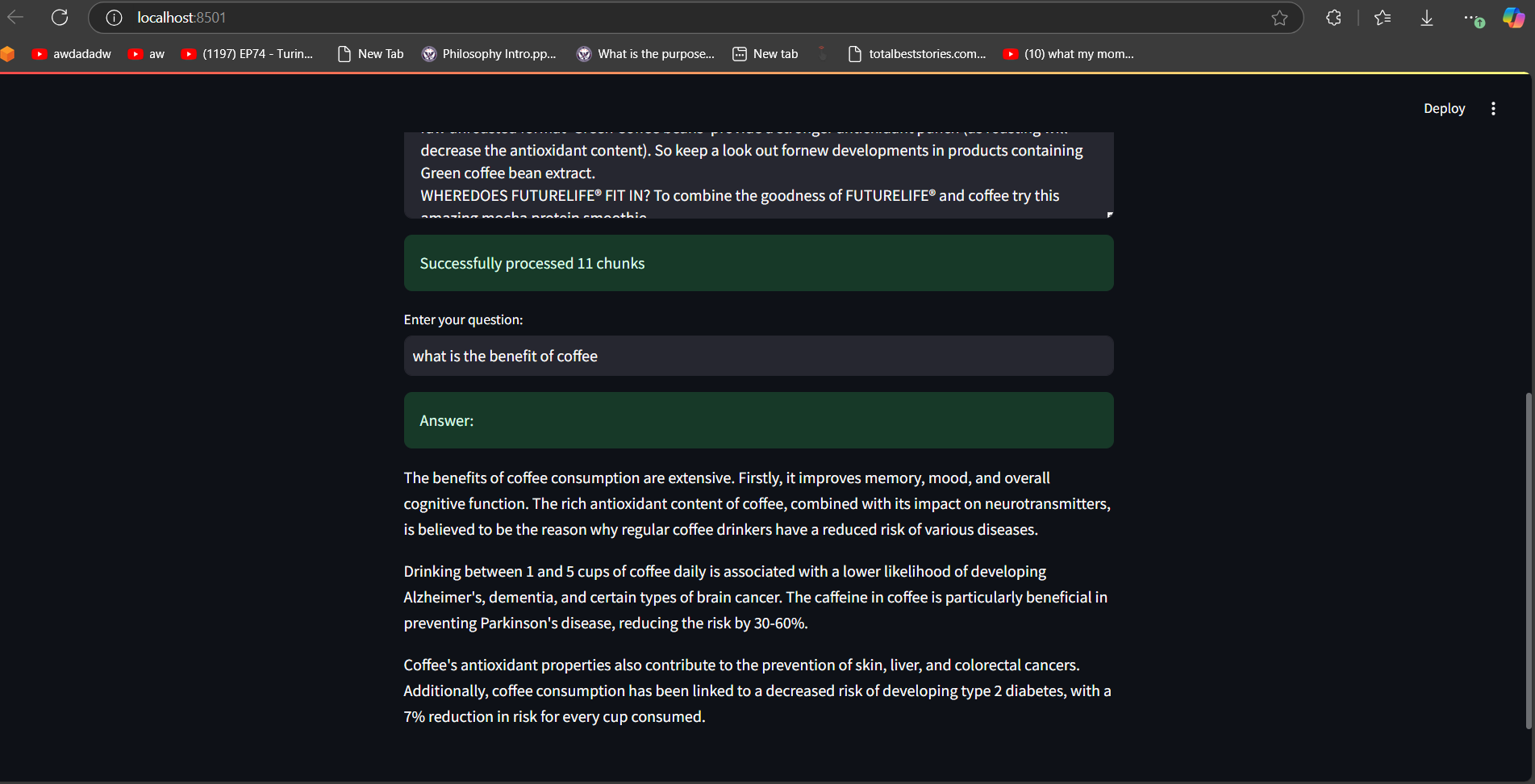
* **Text Extraction**: Using PyPDF2 to extract text from uploaded PDF documents.
* **Text Splitting**: Documents are split into smaller, manageable chunks for indexing and retrieval using RecursiveCharacterTextSplitter.
* **Embeddings**: Cohere embeddings are utilized to convert both document text and user queries into vector representations, enabling semantic search.
* **Indexing**: Pinecone stores document chunks and their embeddings, facilitating fast, similarity-based search for answering user queries.
* **Question Answering**: Cohere’s language model generates responses based on the retrieved context from the document.



**Key Contributions (Modifications):**

* **Custom Embedding Integration**: Integrated Cohere’s advanced embedding model to ensure high-quality vector representations for both text chunks and user questions.
* **Pinecone Indexing**: Developed an efficient and scalable indexing system using Pinecone, optimizing performance for quick retrieval.
* **Interactive UI**: Built a user-friendly interface using Streamlit, allowing users to upload documents, extract text, and interact with the system in real-time to get answers to their queries.

**Application Results:**

* **Data Processed**: PDF documents are processed, split into chunks, and stored in Pinecone for indexing.
* **Graphical Representation**: While the system does not generate graphs, it provides an interactive experience where users can view extracted text and retrieved answers through the Streamlit interface.
* **Outcome**: Users can upload PDF documents, input questions, and receive answers generated from the document's content. The system returns relevant text snippets and corresponding answers.

**Evaluation/Comparison:**

* **Complexity**: The system integrates multiple technologies, such as text extraction, embedding generation, and semantic search, which can be complex to implement and optimize.
* **Performance**: Response time is efficient for small sized documents, though larger documents may experience a very long delay due to the chunking process and embedding generation.
* **Scalability**: Pinecone’s serverless model ensures that the system can scale dynamically to handle varying document sizes and query volumes.
* **Usability**: The intuitive Streamlit interface allows for easy document upload and query interaction, making the system accessible to a wide range of users.

**How to Run the Application:**

1. **Required Software and Libraries**:
   * Python 3.x
   * Streamlit
   * PyPDF2
   * Pinecone
   * Cohere
   * LangChain
2. **Installation Instructions**:
   * Install required libraries:
   * bash
   * Copy code
   * pip install streamlit PyPDF2 pinecone-client cohere langchain
   * Set up environment variables:
     + Create a .env file in your project directory containing your COHERE\_API\_KEY and PINECONE\_API\_KEY:
   * bash
   * Copy code
   * COHERE\_API\_KEY=your\_cohere\_api\_key
   * PINECONE\_API\_KEY=your\_pinecone\_api\_key
3. **Run the Application**:
   * Start the Streamlit app by running:
   * bash
   * Copy code
   * streamlit run Analysis\_doc.py
4. **Configuration Settings**:
   * Ensure your API keys are set in the .env file.
   * Pinecone should be configured to use AWS in the ServerlessSpec setup.

**Dependencies:**

* **Pinecone**: A vector database for storing and efficiently retrieving document chunks based on vector similarity.
* **Cohere**: A natural language processing API for generating embeddings and understanding document content.
* **Streamlit**: A framework for building interactive web applications with a focus on user-friendly design.

**Conclusion**

This document Q&A system significantly enhances how users can interact with large PDF documents. By combining Cohere's embedding model with Pinecone's vector database and Streamlit's intuitive interface, the system delivers a high-performing, scalable solution for document-based question answering. The project addresses the growing need for efficient document processing and querying, offering valuable potential for knowledge workers, researchers, and anyone dealing with large volumes of text-based data.