# AI RAG applications using

# LangChain-Report.

Vinod Kumar Dhanavath

# Final Project

# Machine Learning.

* Final Project : Q&A Application

Executive Summary

* This report outlines the development and functionalities of the Q&A Application, a tool designed to facilitate seamless interactions with databases and documents through a user-friendly web interface. The application utilizes advanced natural language processing and database querying techniques to provide users with instant, accurate responses to their inquiries about database content or textual documents.

Introduction

* The Q&A Application serves as an interactive platform for users to make queries regarding databases or uploaded documents. It leverages state-of-the-art technologies, including Streamlit for web application deployment, PyPDF2 for handling PDF files, and AI-powered language models for understanding and responding to user queries.

System Architecture

* The application is built on a robust architecture that integrates several Python libraries and OpenAI models. It features a backend responsible for processing user inputs, executing database operations, and handling document analysis. The front end, created with Streamlit, offers an intuitive interface where users can connect to databases, upload PDFs, and interact with the system through chat.
* Methodology

Development Tools and Libraries:

Streamlit: for crafting the web app and managing the session state.

PyPDF2: to extract text from uploaded PDF files.

dotenv: for loading environment variables safely.

Langchain and OpenAI libraries: for embedding queries and providing AI-driven conversational capabilities.

MySQL Connector/Python: for establishing and interacting with MySQL databases.

NumPy and Scikit-learn: for mathematical operations and executing cosine similarity computations.

Interface and Connectivity:

Initial Interface:

* The application presents users with a clean and straightforward initial interface, as shown in the 'App Start' screenshot (see Appendix A). It clearly prompts users for database connection details, ensuring a secure and customized experience.

Database Connection:

* A snapshot of a successful database connection, as depicted in the 'Database Connection' screenshot (see Appendix B), reassures users of the application’s capability to establish and maintain robust database connections.

Implementation Details:

The application's backend logic is encapsulated in functions that:

Initialize resources based on session states.

Handle user queries by determining their context and dispatching them to the relevant processing function.

Extract and process text from PDF documents.

Perform document searches using cosine similarity of text embeddings.

* Results:

The application successfully allows users to:

* Interact via a chat interface
* Connect to a MySQL database using credentials.
* Upload and process PDF documents.
* Make queries that are contextually understood, with responses displayed promptly.
* Application Output:

The application successfully demonstrated its capability to process natural language queries related to both documents and databases. The screenshot in the report appendix C, shows the application handling two types of queries:

Document Query: When asked about the content of an uploaded document (“What is this assignment about?”) , the application correctly provided a detailed description, indicating a robust understanding of the document's text.

Database Query: The application accurately responded to a query about game sales in the EU, formulating an appropriate SQL statement and displaying the query result.

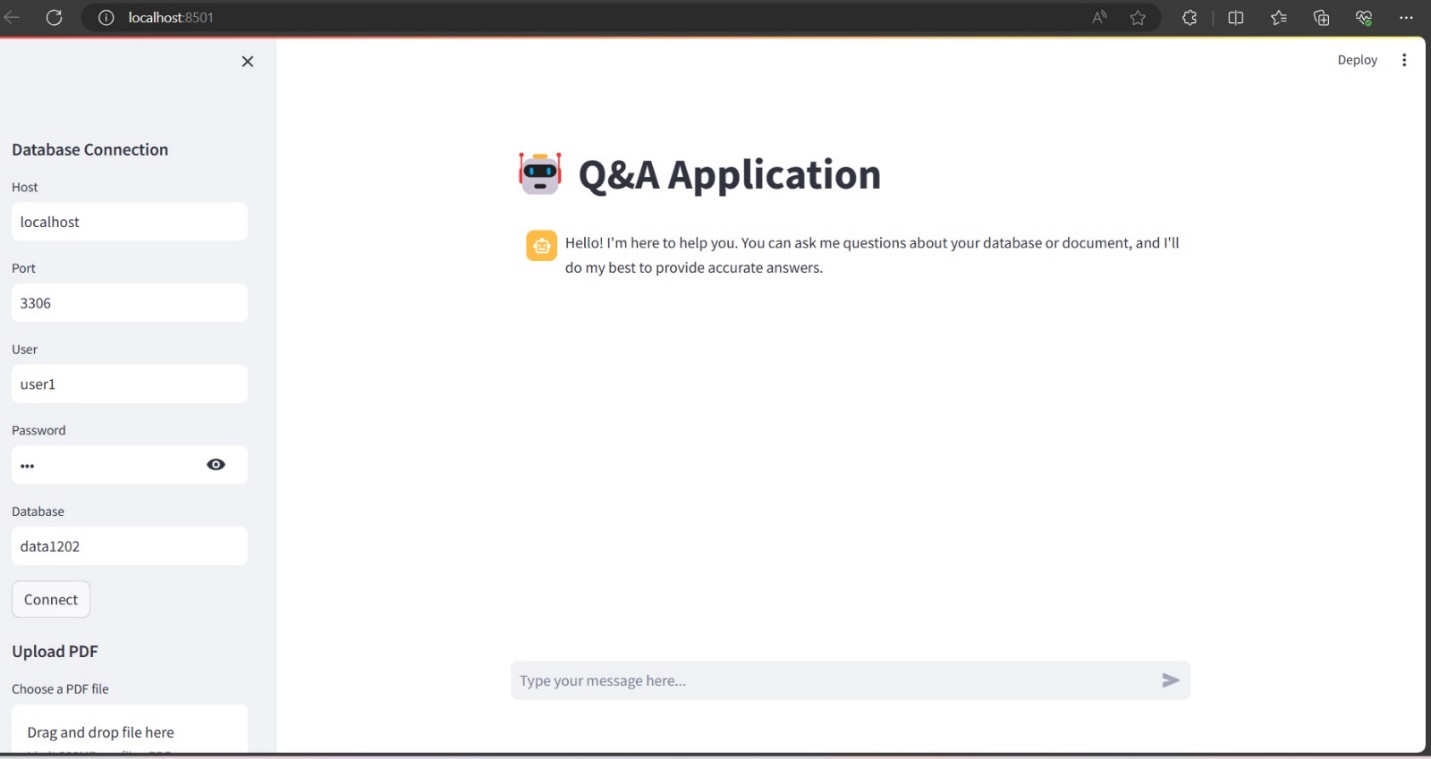
Challenges and Solutions:

The project team encountered challenges, including efficient handling of non-text PDFs and maintaining session states across interactions. Solutions involved using PyPDF2's robust text extraction capabilities and Streamlit's session state management

Appendices:

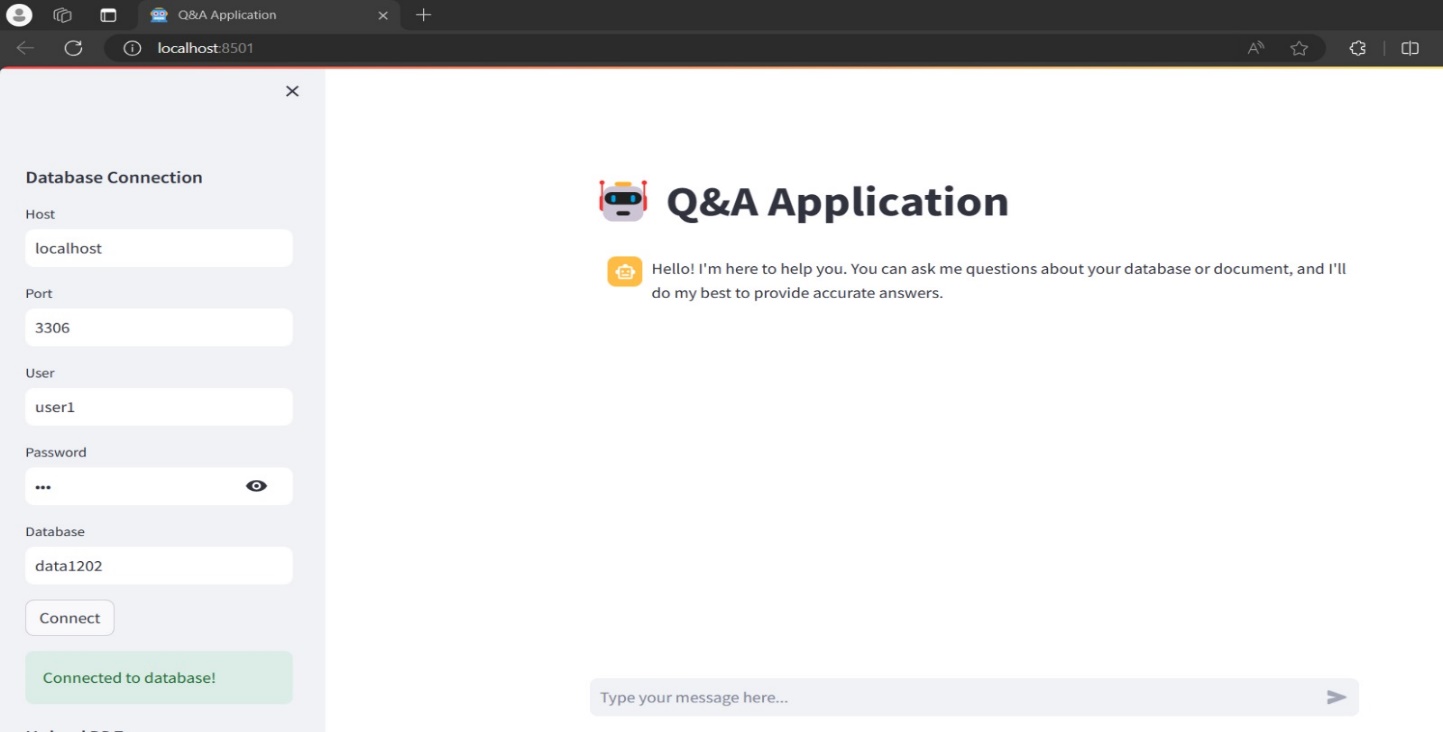
A. App Start Screenshot

The 'App Start' screenshot illustrates the application's initial state, awaiting user input for database connection parameters.

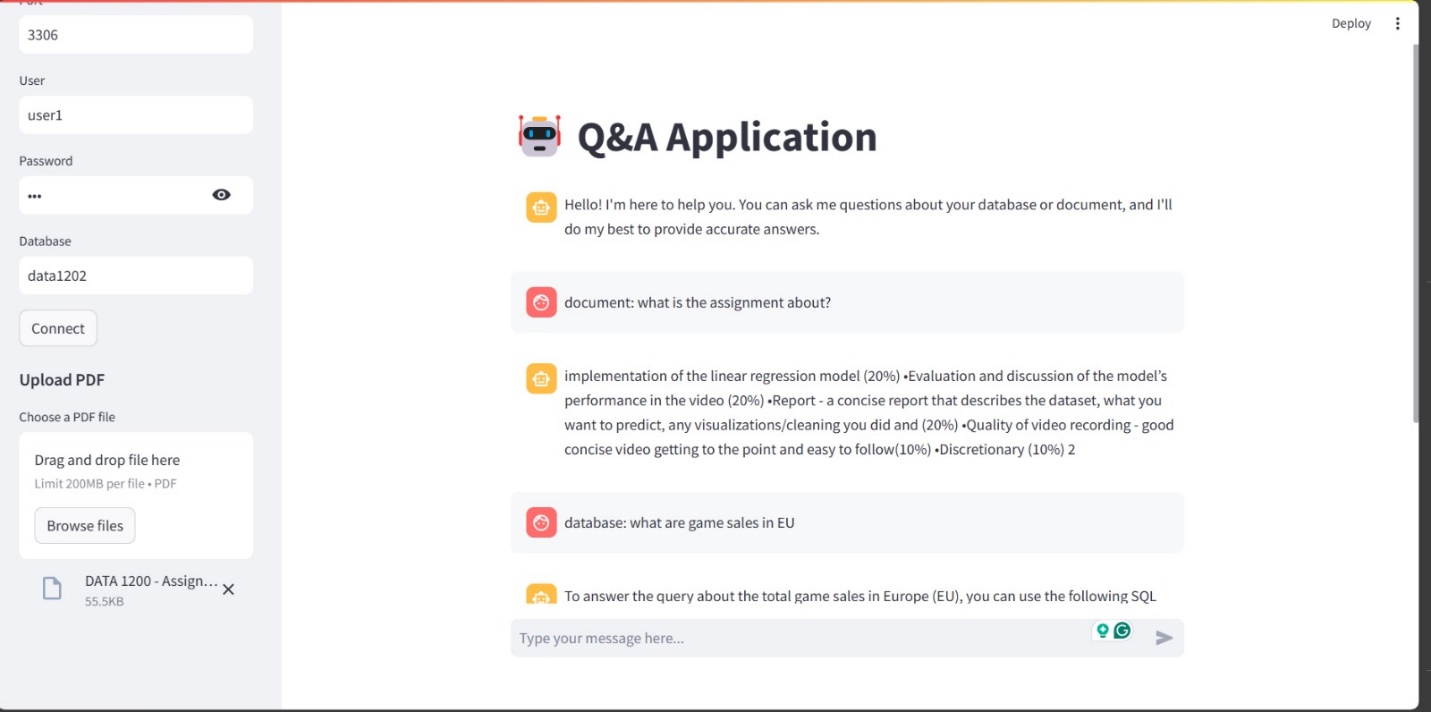


B. Database Connection Screenshot

The 'Database Connection' screenshot captures the moment post-connection, indicating a successful linkage and readiness for user queries.



C. Application Output



Thank you.