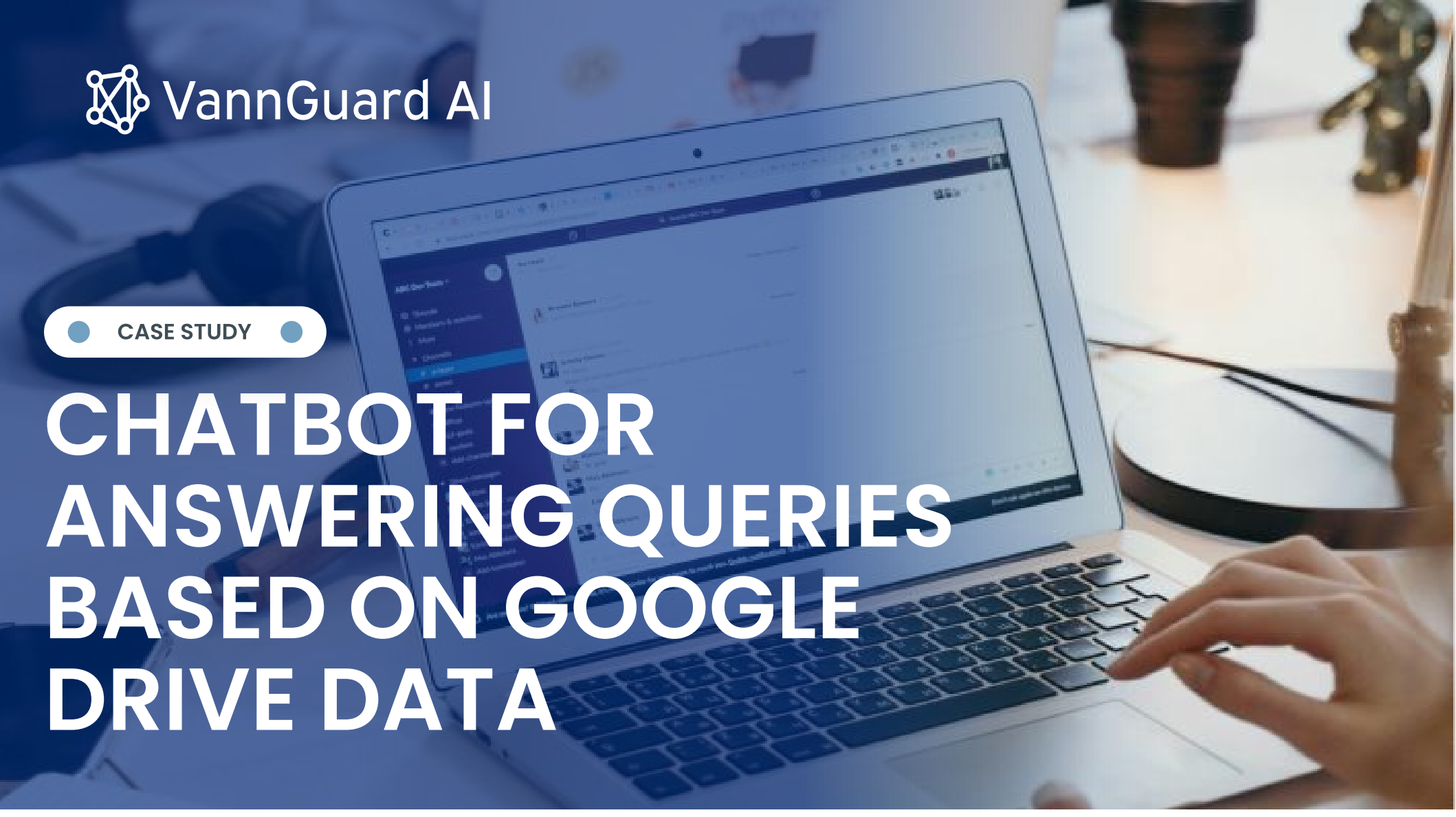
VannGuard AI



CASE STUDY

CHATBOT FOR ANSWERING QUERIES BASED ON GOOGLE DRIVE DATA

Overview

The goal of this project was to develop an intelligent assistant that can interact with the vast amount of data stored in Google Drive, including spreadsheets, DOCX, PPTX, PDFs, and TXT files using Generative AI. To address this challenge, an automated Retrieval Augmented Generation (RAG)-powered Document Q&A system was designed to efficiently process and retrieve the required data from the documents with minimal technical expertise. This sophisticated system was designed to handle complex data and provide contextually relevant responses.

Customer

Freelance

Country: USA

Customer Size: 500-1000

Industry: B2C

Publish Date: 26/08/2024

Problem Statement

We proposed a solution to efficiently navigate and extract value from a vast and diverse document repository stored in Google Drive. Quick extraction of the relevant information due to the volume and variety of documents was a challenge and it required a sophisticated tool that is capable of handling complex data and providing accurate, contextually relevant responses. We proposed a solution that was user-friendly, allowing the users with minimal technical expertise to interact with the system effectively.

Technical Solution

We developed a scalable and efficient solution by implementing a Retrieval Augmented Generation (RAG)-powered chatbot that could handle complex queries across diverse document types. The process began with data ingestion and preprocessing using LlamaIndex, which allowed us to load documents from Google Drive, clean them, and split them into manageable chunks. These processed data chunks were then ingested into a vector database, facilitating efficient querying and retrieval. For the core Q&A functionality, we employed ChatGPT-4o-mini and OpenAI embeddings to create a conversational system capable of understanding user queries in natural language and retrieving contextually relevant information from the stored documents.

To make the system accessible, we built a user-friendly interface using Flask and JavaScript, ensuring ease of use for non-technical users. Additionally, we integrated MongoDB to store user queries and responses, enabling persistent data storage and further analysis. This end-to-end architecture combined various AI and development tools to deliver a robust chatbot solution tailored to meet the client's needs for interacting with and extracting value from their document repository.

Results

The deployment of this RAG-powered chatbot significantly improved the client's ability to interact with and extract insights from their extensive document repository. The natural language interface and AI-driven processing capabilities streamlined data access, reducing the time required to find relevant information and enhancing overall productivity. Users across the organization could quickly retrieve accurate responses without technical expertise, enabling more informed decision-making and efficient operations.

Technologies

- Python
- LlamaIndex
- OpenAI GPT4o-mini
- JavaScript
- Docker
- Flask
- OpenAI Embeddings
- MongoDB
- Qdrant

Domains

- Generative AI
- Natural Language Processing (NLP)
- Chatbot Development
- Retrieval Augmented Generation (RAG)