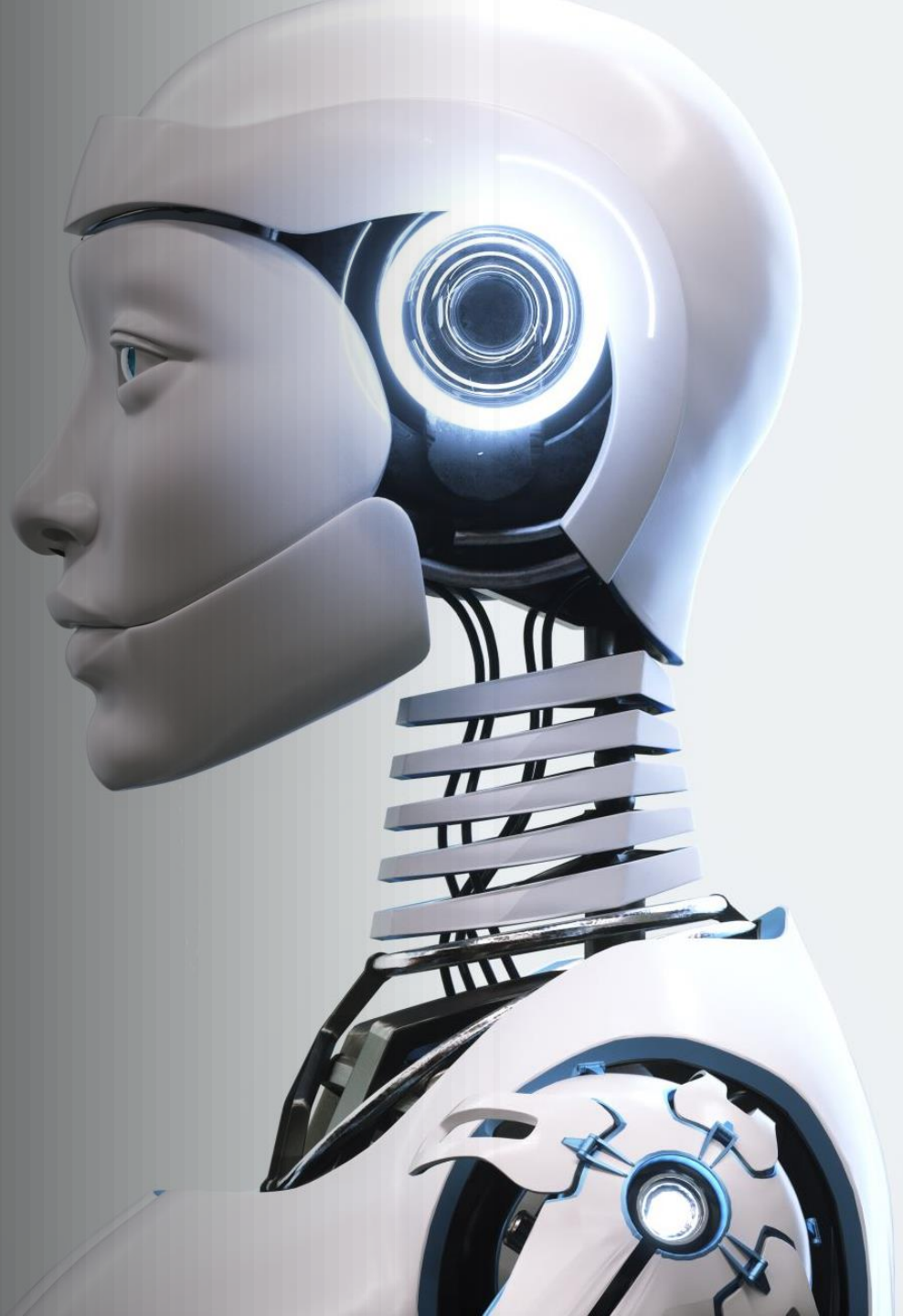
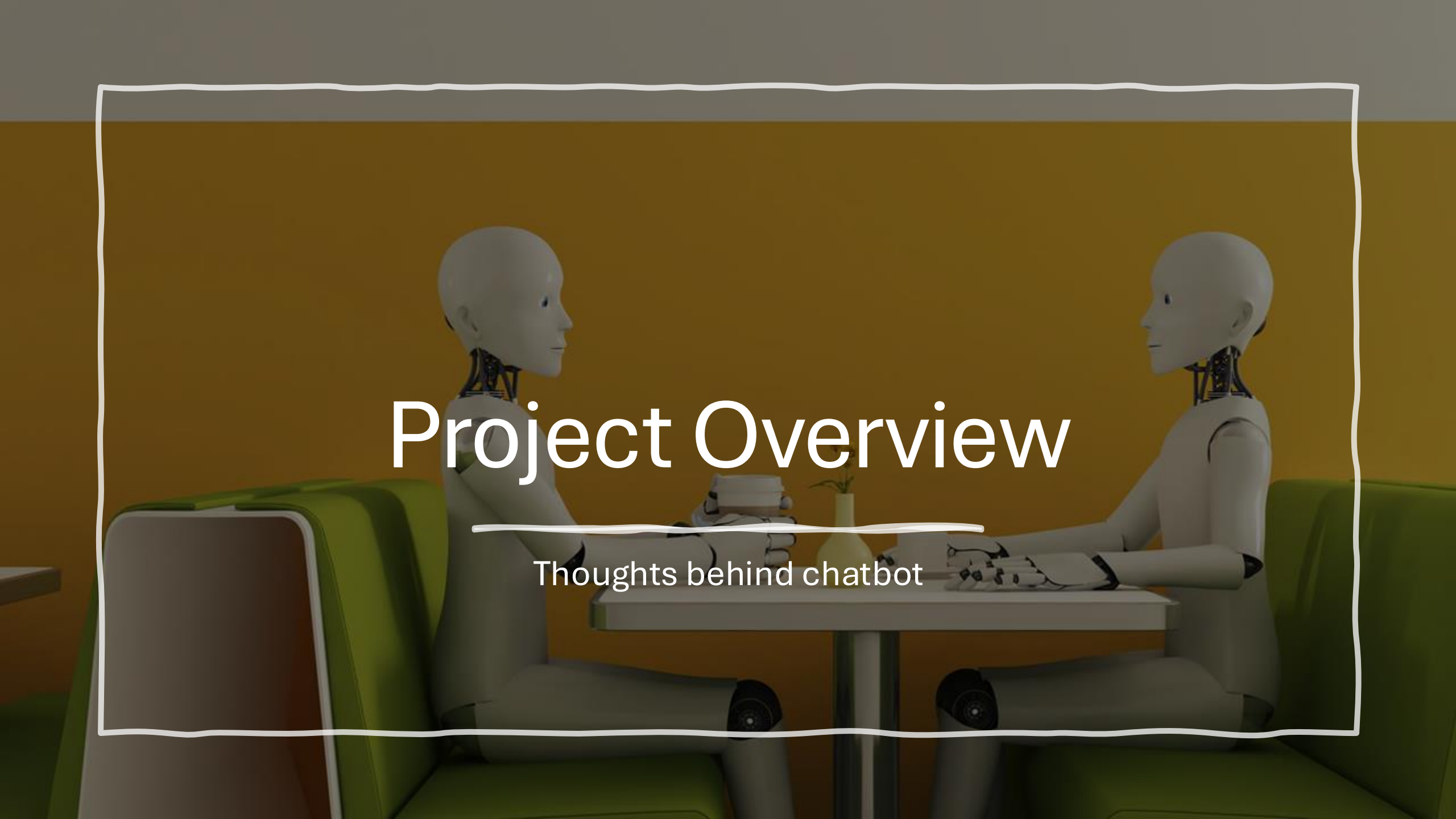
Context-Based ChatBot using RAG System



Submitted by:

Keerthana Goka

Kunal Malhan



Project Overview

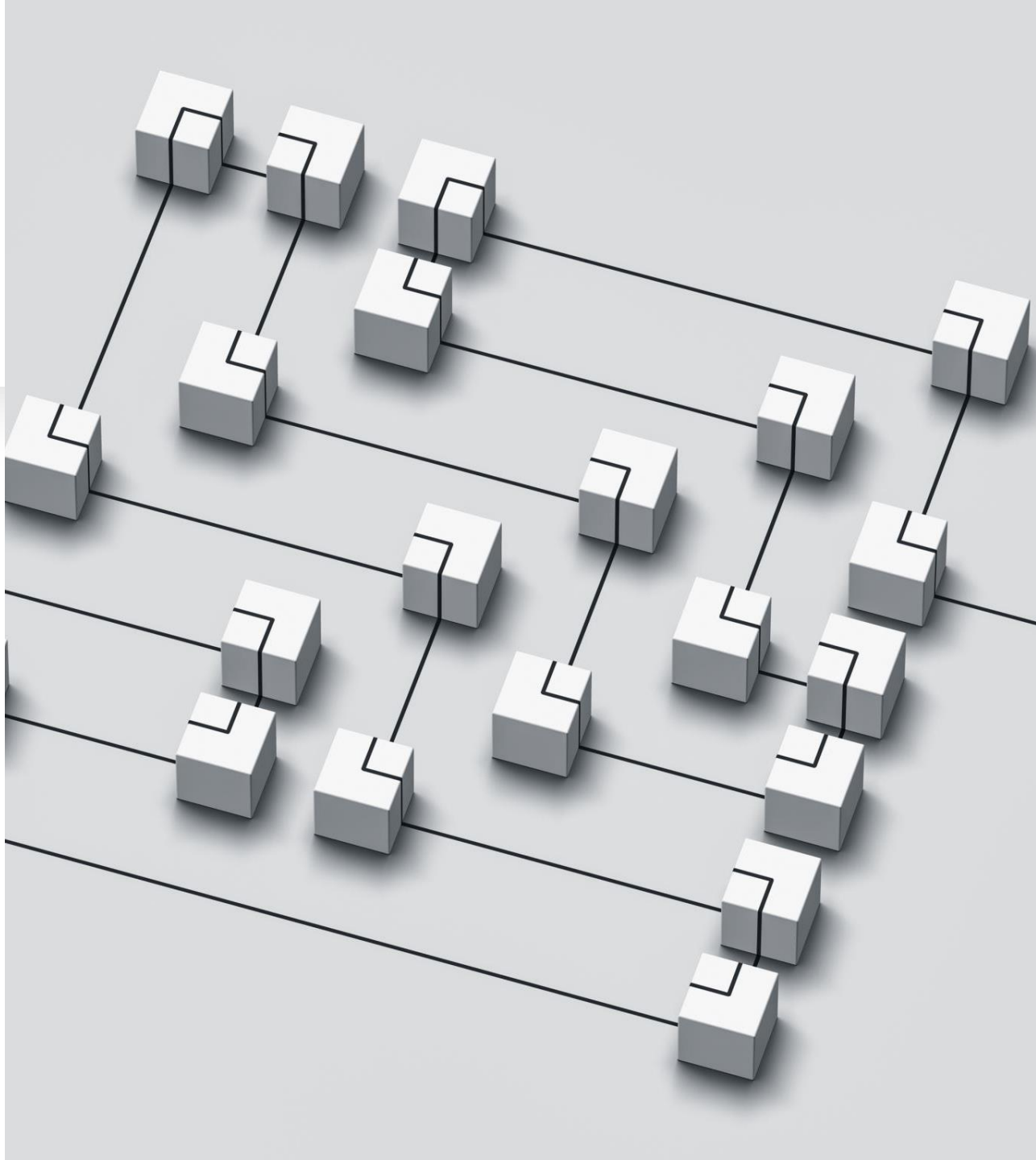
Thoughts behind chatbot

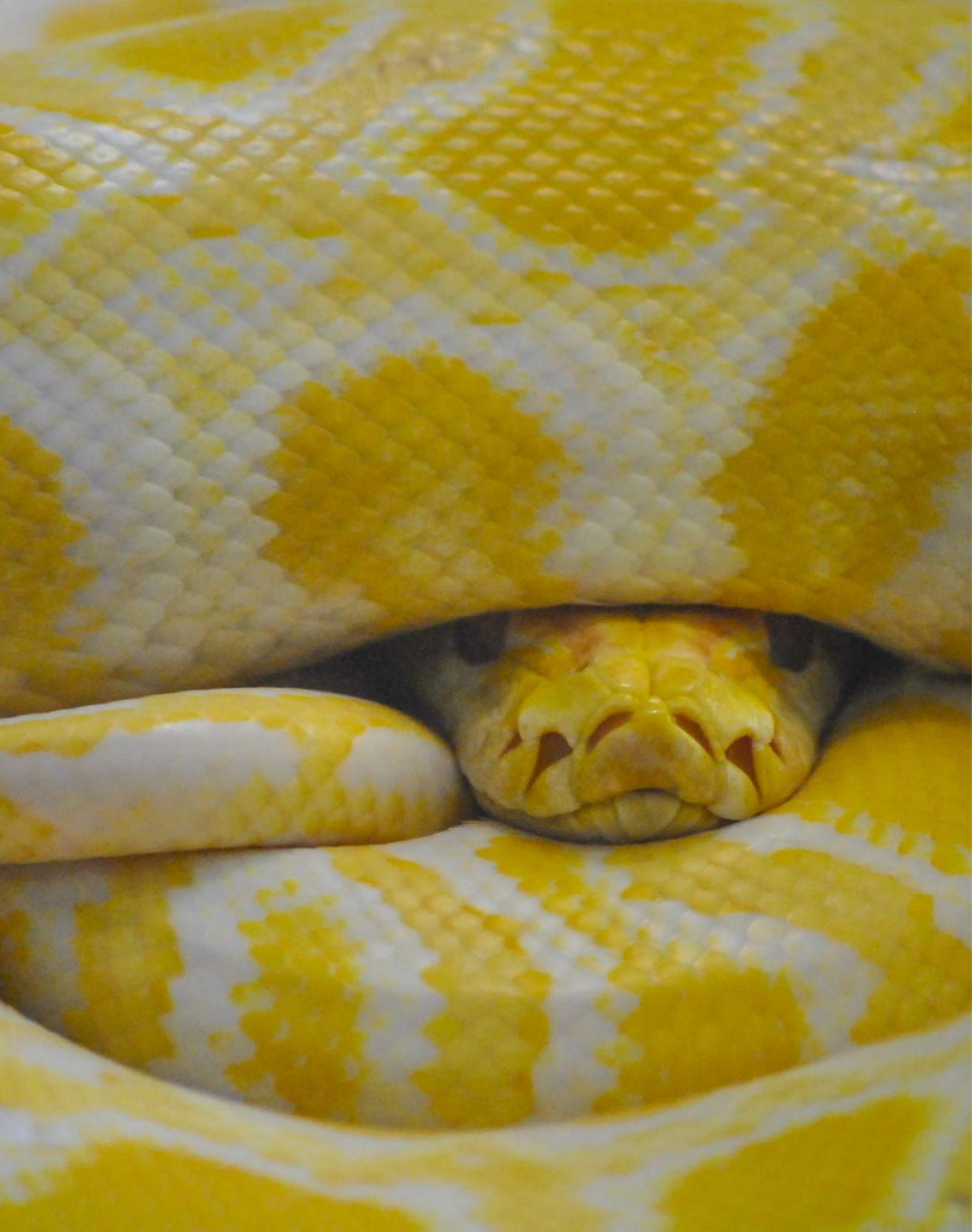
Problem Description

- This project will aim to use Retrieval Augmented Generation (RAG) system to get domain specific knowledge with base of Large Language Model (LLM) like llama2 or similar.
- The challenge for implantation will reside in understanding the user question and the responding appropriate information and developing ethical guidelines for interaction
- All communication and input/outputs will be limited to English language.

Scope of the Project

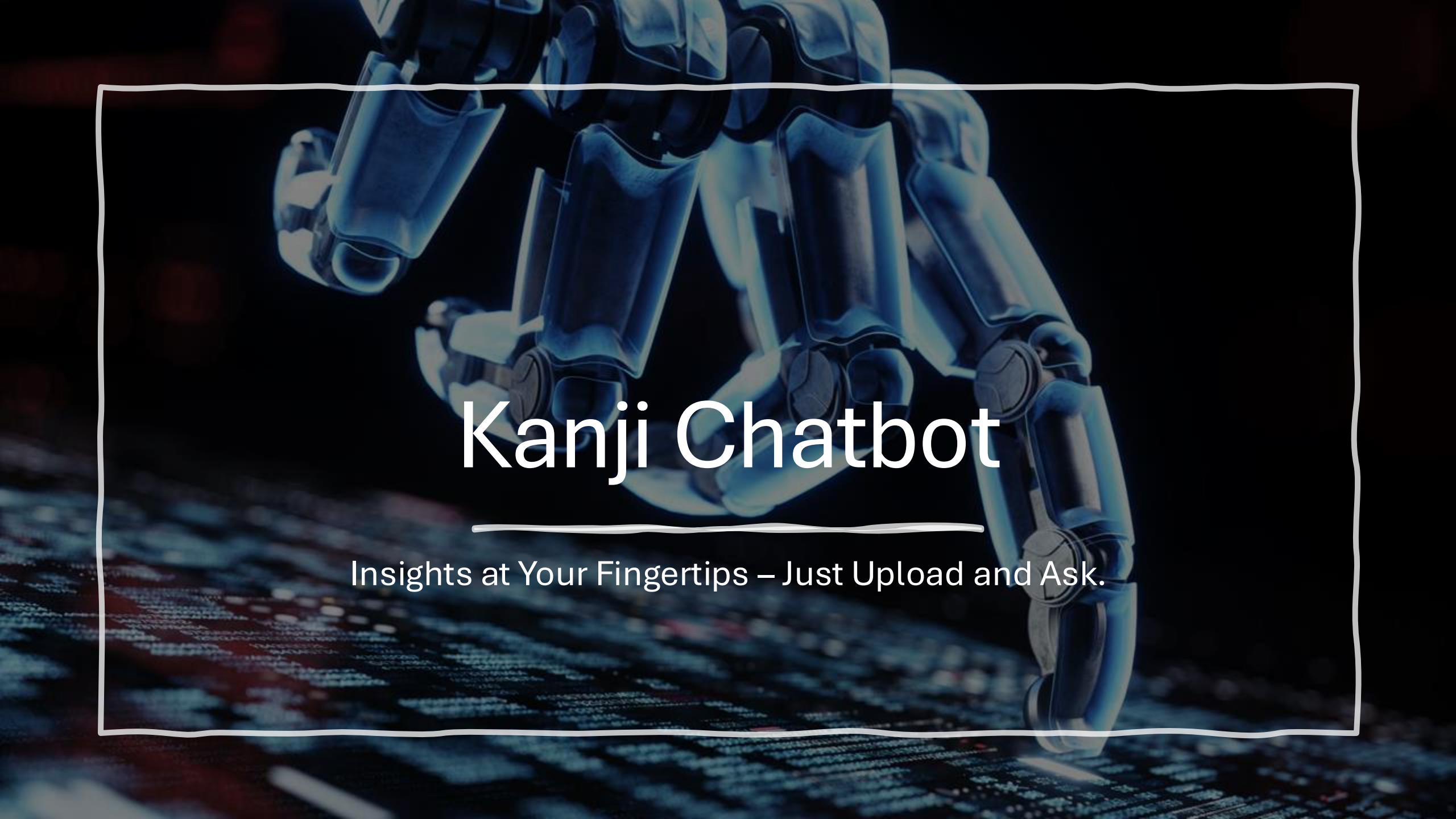
- For RAG system implementation:
  - Utilizing LLM models through Lang-Chain or Hugging-Face library
  - Loading contextual data files in form of text-chunks and store as embeddings in vector database.
  - Based on context of question retrieval of relevant data chunks.
  - Generate response for user.
  - Save response as chat-history.
- Implementation of user interface for chatbot.





Skills and Tools Required

- Python
- pandas, numpy
- LangChain library
- Hugging Face
- Streamlit

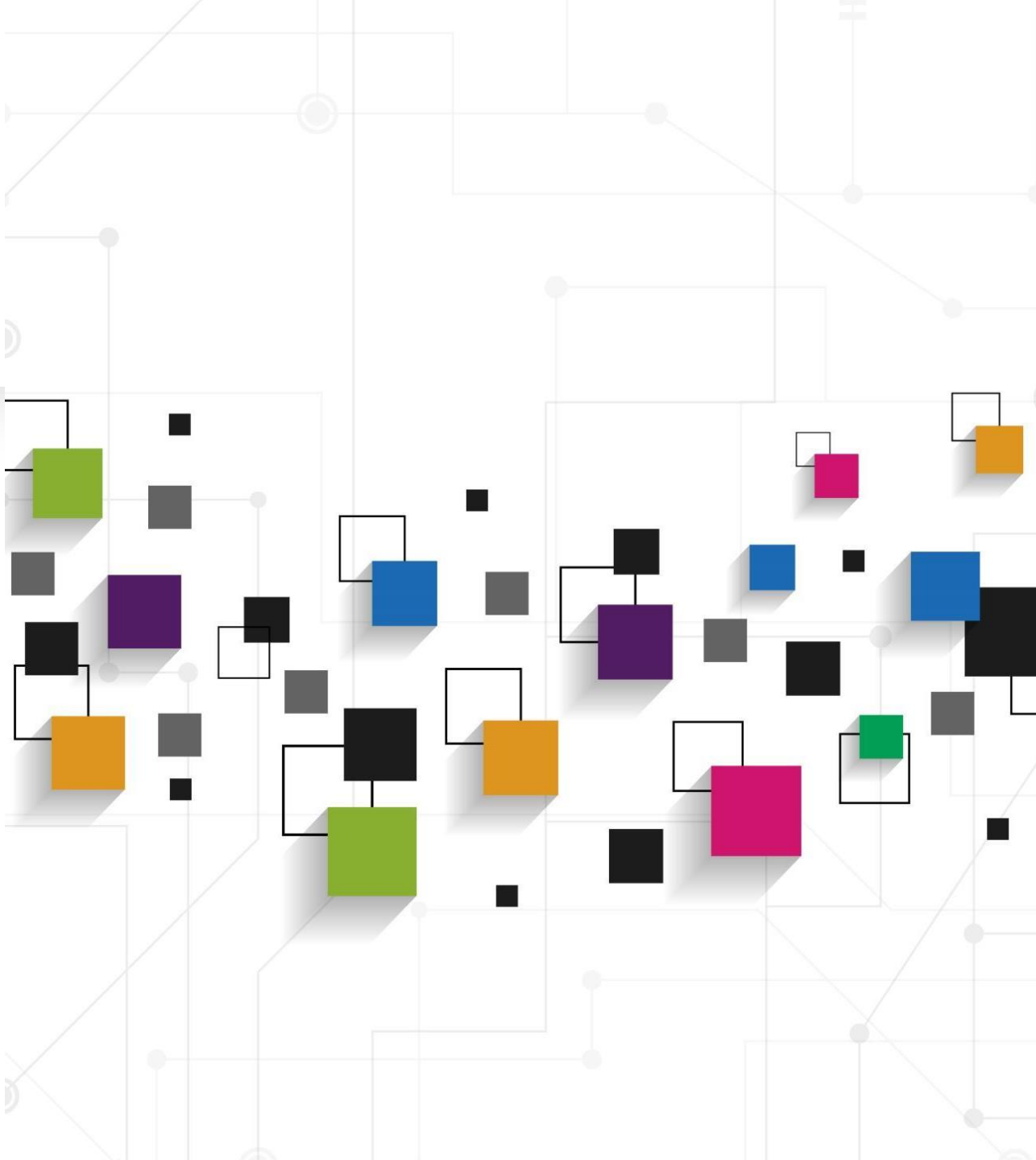


Kanji Chatbot

Insights at Your Fingertips – Just Upload and Ask.

Data Processing Pipeline

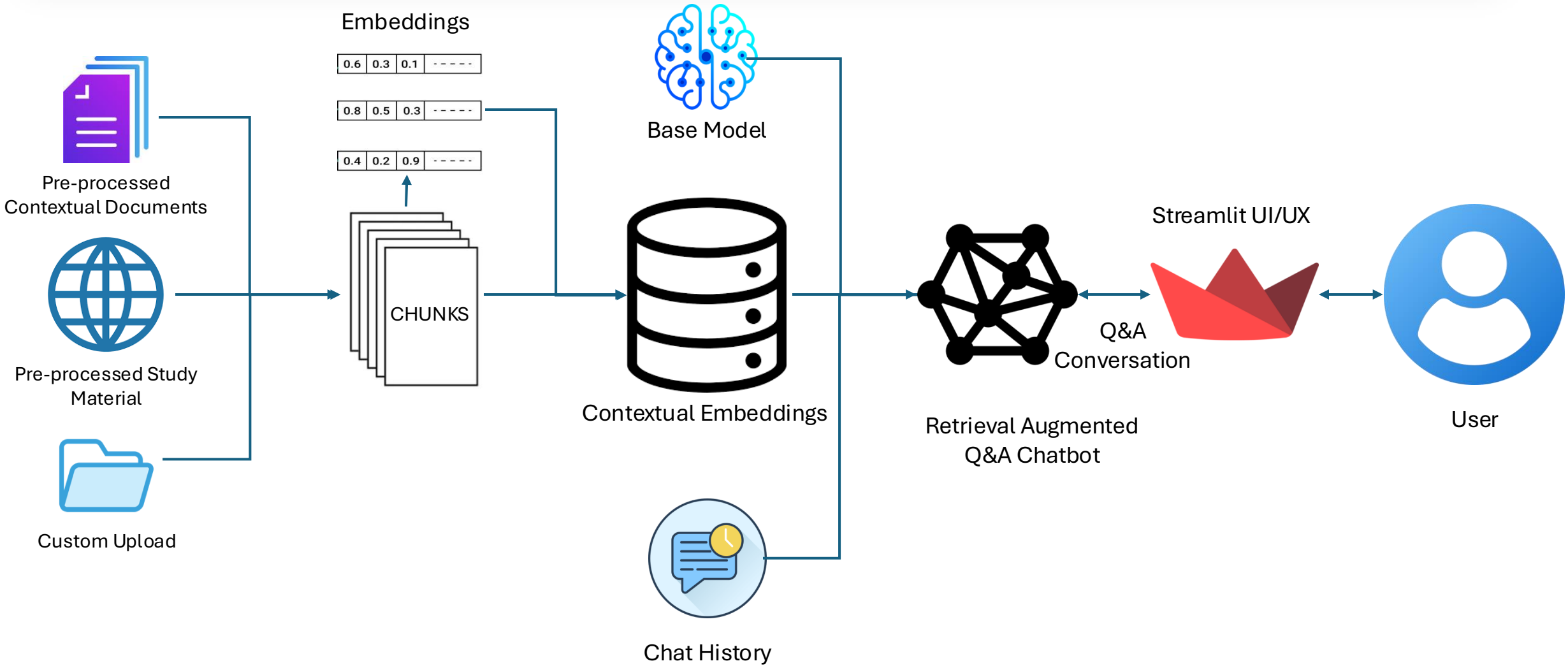
- For model training on contextual dataset:
  - Upload files
    - Virtually any text content in PDF file can be source of information for the system.
  - Create documents from files
  - Split documents into chunks
  - Create embeddings for chunks
  - Storing chunks and embeddings in vector database (ChromaDB)
- For conversional Q&A:
  - Create embeddings for user message
  - Search for relevant chunks/documents from vector database
  - Generate response and display to user
  - Add response to context for memory generation.



Methodology

- Our methodology includes the use of natural language processing, large language models and then adding contextual layer above the base model. Step by step methodology is as follows:
  - Download pre-trained model through Ollama
    - We will be using “Gemma2-9b” as base model. Advantage of using this includes its open source, easy integration, and small (need relatively less compute).
  - Setting up base model using langchain
    - langchain-community is open-source library that will be used to setup the complete layers with base model.
  - Creating Retrieval Augmented Generation (RAG) layer
  - Creating web-based user interface for chatbot using Streamlit framework

Kanji Chatbot Architecture



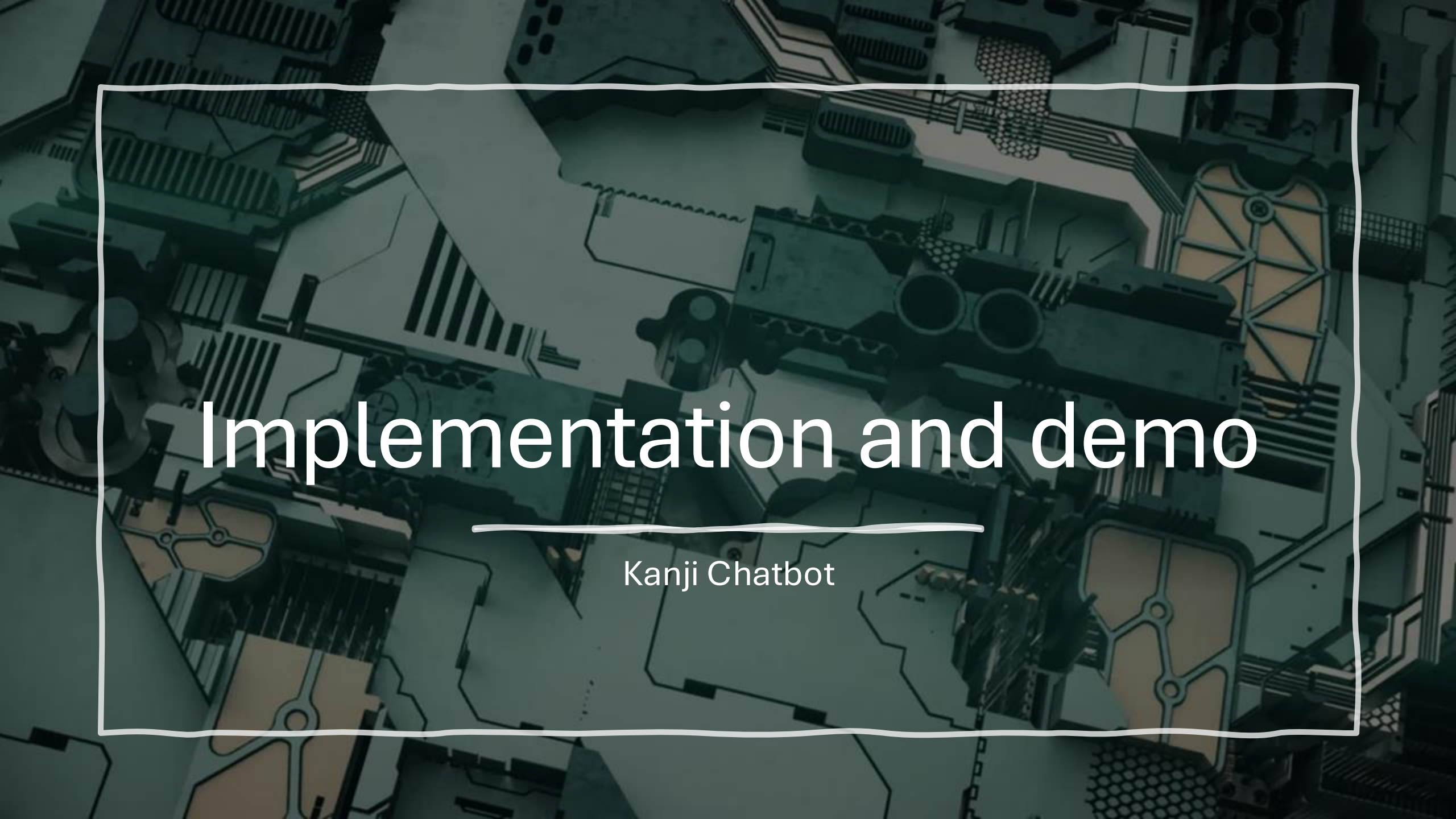
UI/UX Features of Kanji Chatbot

Chatbot access through Login or Anonymously

User based chat memorization in case of login

Multiple departmental/use-cases fixed/preloaded chat windows

Flexibility to load your own pdf documents

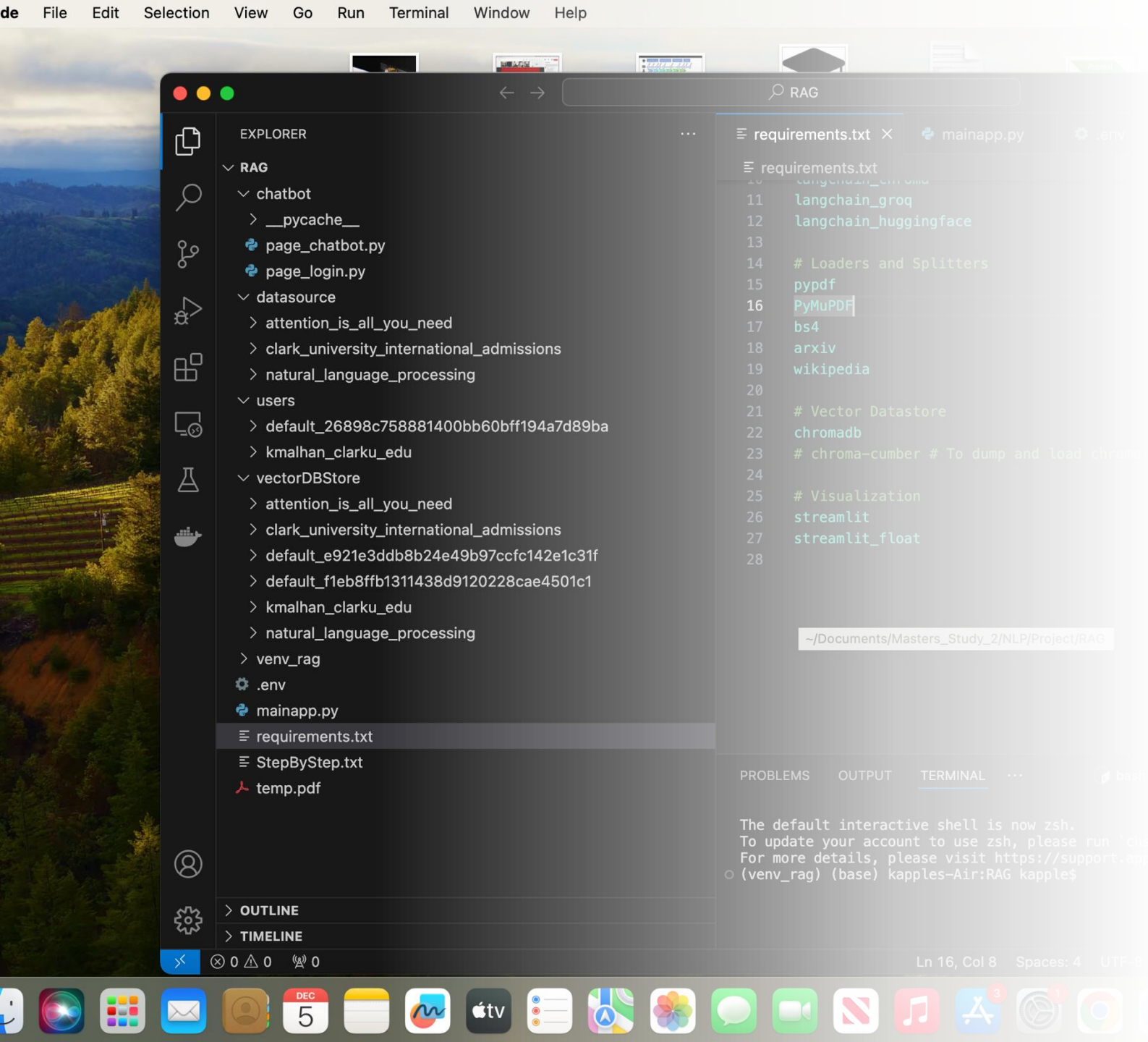


Implementation and demo

Kanji Chatbot

Libraries used

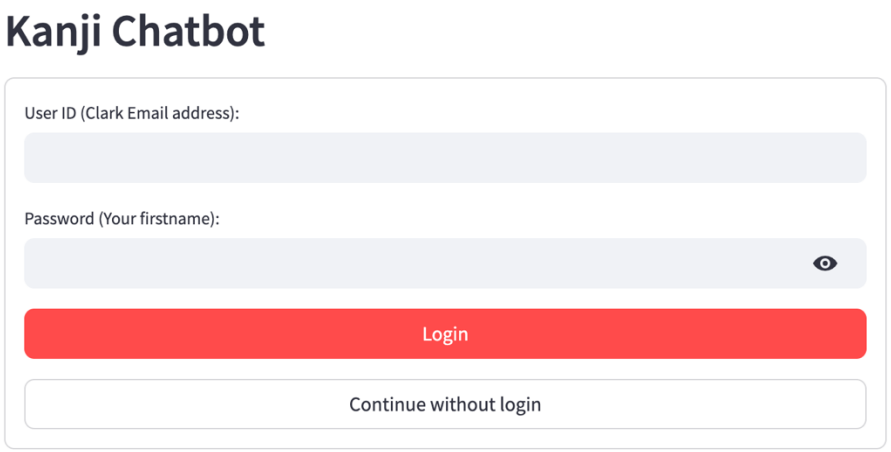
- Python Environment
  - ipykernel
  - python-dotenv
- Langchain
  - langchain
  - langchain-openai
  - langchain-community
  - langchain-text-splitters
  - langchain_chroma
  - langchain_groq
  - langchain_huggingface
- Loaders and Splitters
  - pypdf
  - PyMuPDF
  - bs4
  - arxiv
  - wikipedia
- Vector Datastore
  - chromadb
- Visualization
  - streamlit
  - streamlit_float

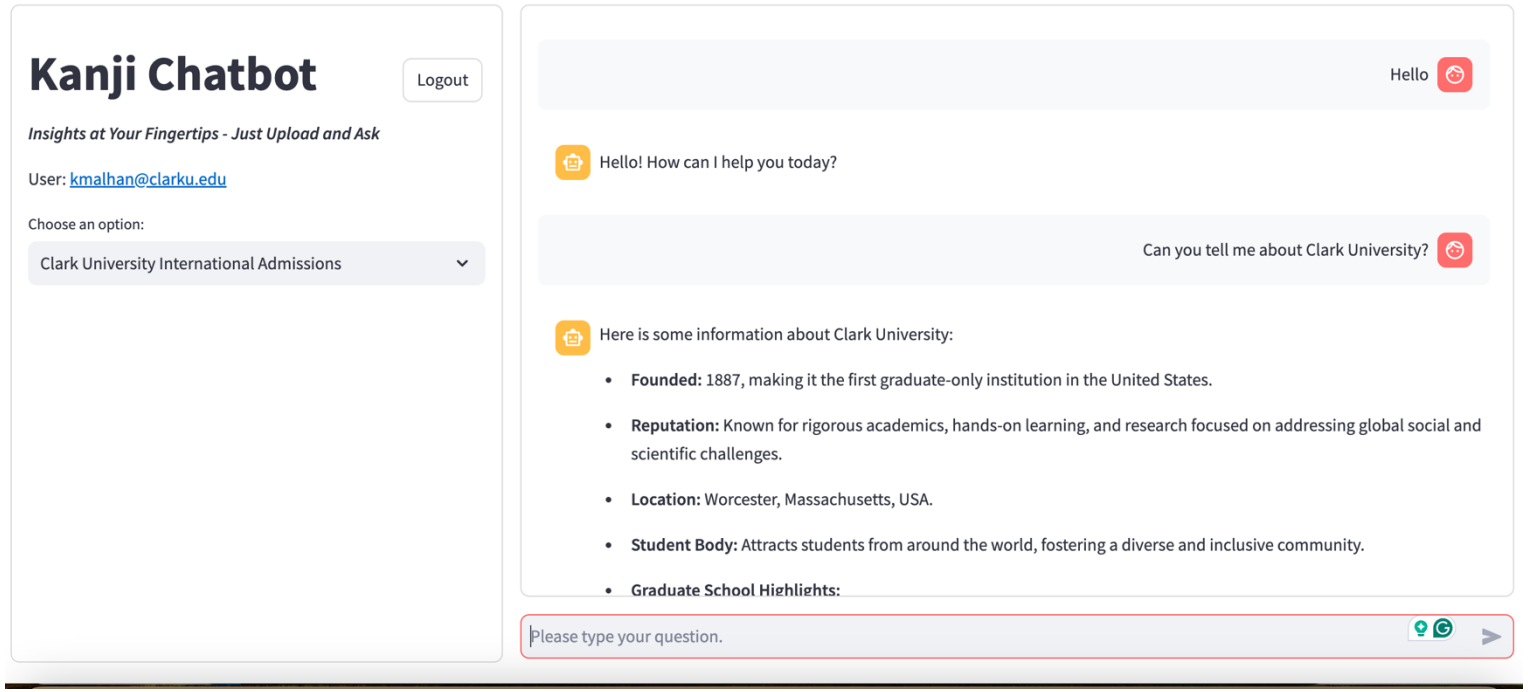


Platform

- Visual Studio
- Important files

Screenshots





Demo

Let's try Kanji

Thank You

Q & A