YMM-RESUMES SCREENER

CHATBOT FOR RUSEME SCEREENING

By:

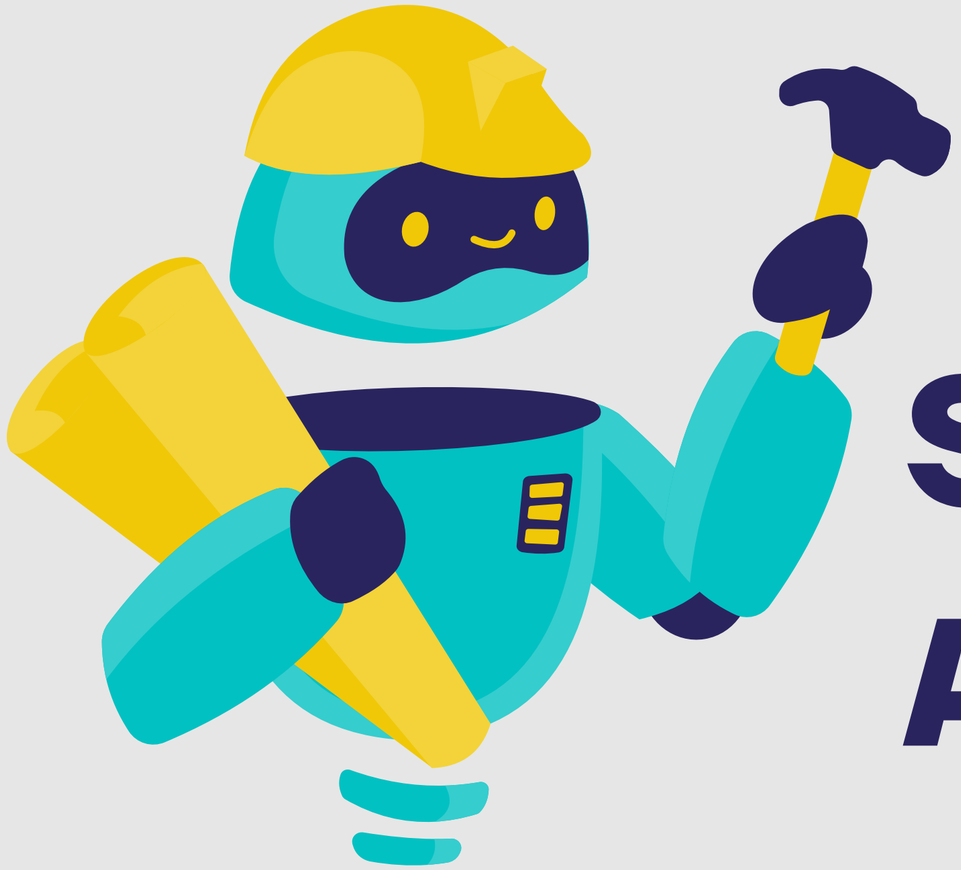
- Yara Mostafa
- Mariam Ahmed
- Menna Osman

AGENDA

YMM-RESUMES SCREENER

1. High Level Solution Architecture
2. Methodology and used models
3. The implemented features
4. Technical Challenges
5. Enhancements and Future Work
6. Demo

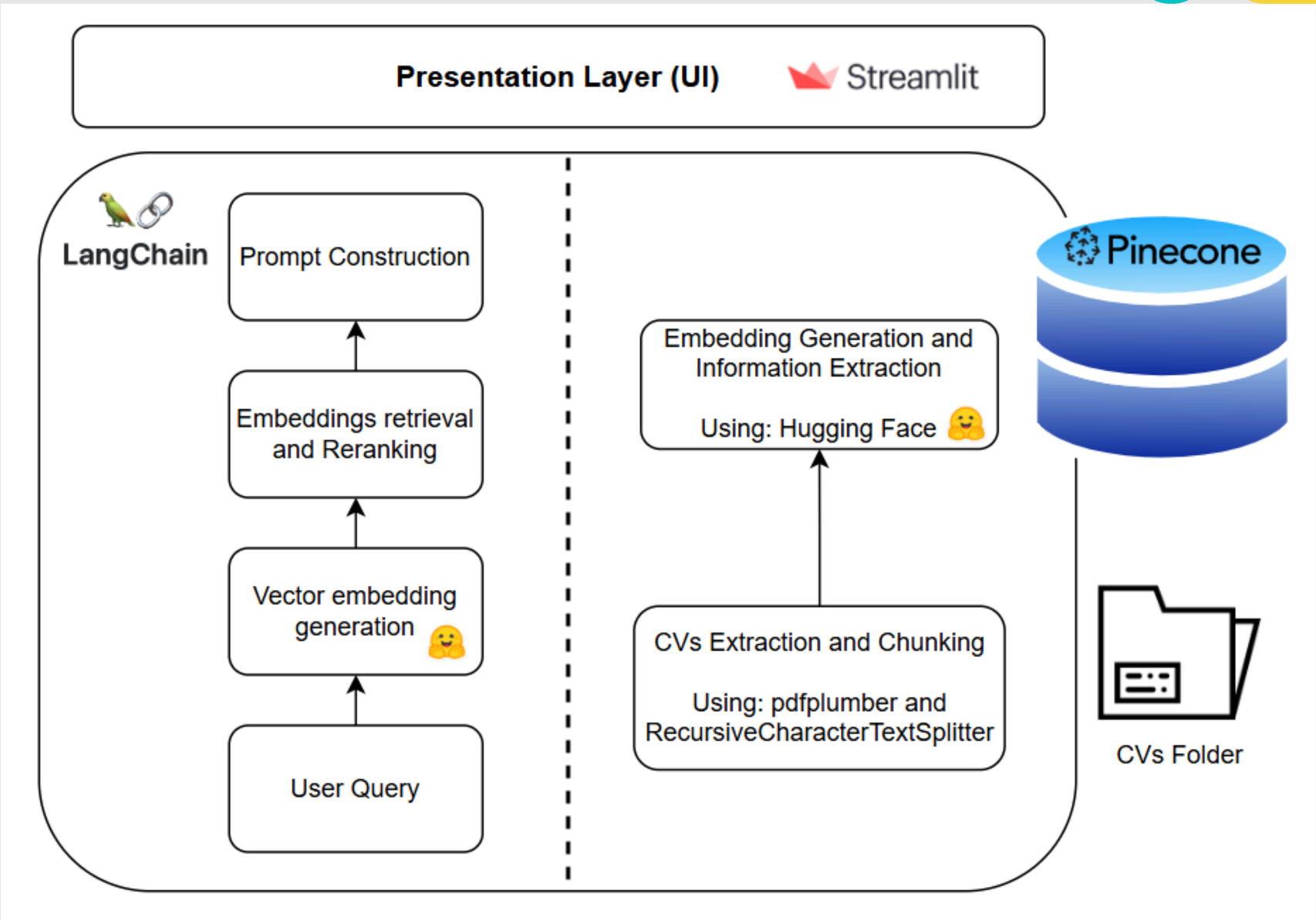
YMM-RESUMES SCREENER



Solution Architecture

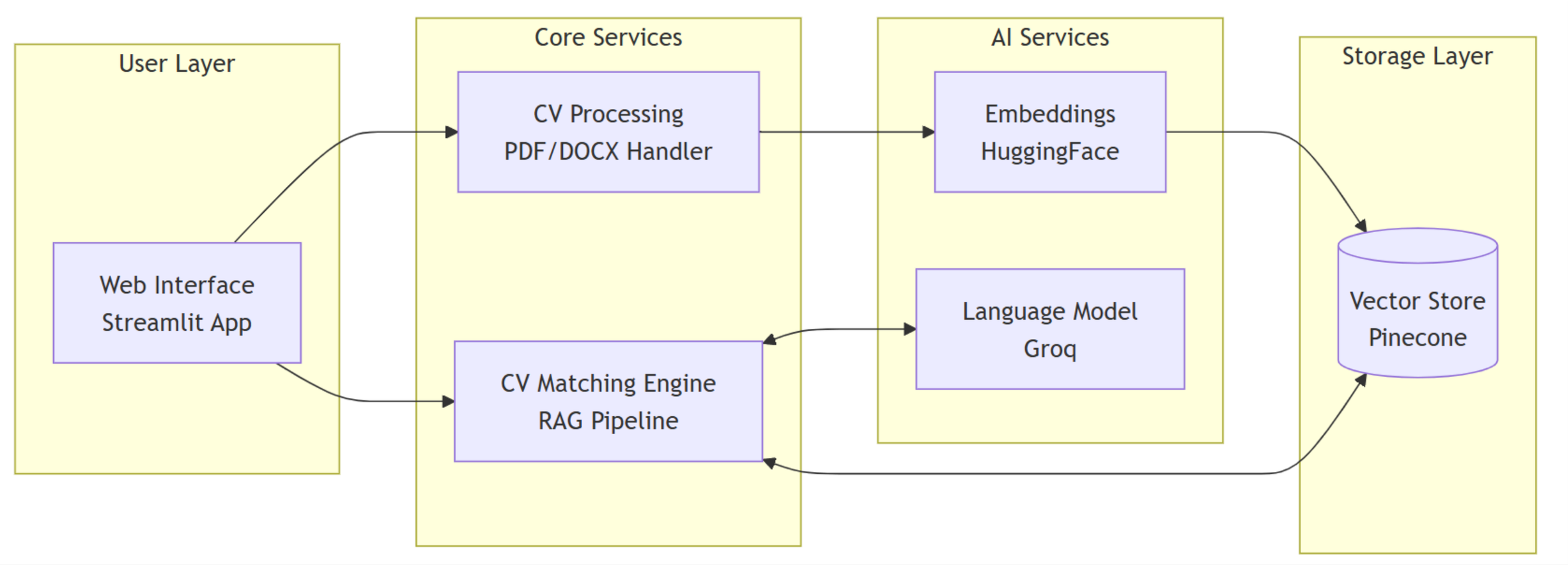
SOLUTION ARCHITECTURE

YMM-RESUMES SCREENER

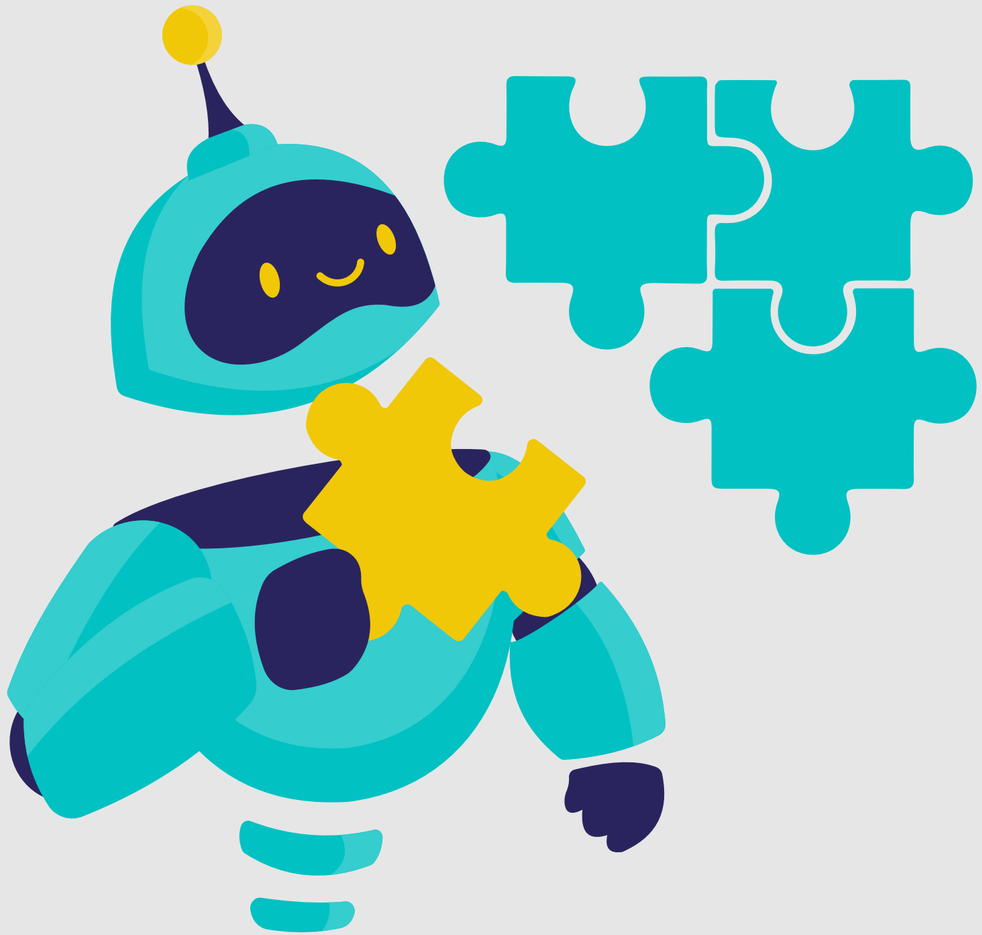


SOLUTION ARCHITECTURE

YMM-RESUMES SCREENER



YMM-RESUMES SCREENER



METHODOLOGY

DATASET AND OCR

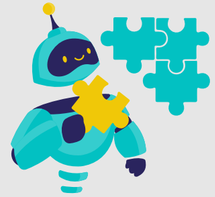
YMM-RESUMES SCREENER

- We used the Company Resumes as our **dataset**, consisting of 26 files in DOCX & PDF formats.

We explored multiple **OCR** tools:

- pytesseract
- MarkerOCR
- EasyOCR.

- Final Choice: **pdfplumber and python-docx**
- Because:
  - It outperformed others in accuracy.
  - It was more time-efficient compared to alternatives.



PREPROCESSING AND CHUNCKING

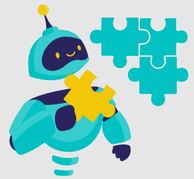
YMM-RESUMES SCREENER

- Once the raw text was extracted from resumes, it was passed through a text cleaner to remove extra spaces, special characters, and unwanted symbols.

For chucking

- we used langchain **RecursiveCharacterTextSplitter** because resumes are already structured.
- Overlapping chunks were applied to preserve sentence continuity.
- We discarded semantic splitting.
- Chunk Format:

  { 'original_file': name,
  'chunk_id': f"{name}_chunk_{i}",
  'content': chunk (prefixed with the person name) }



EMBEDDINGS AND VECTOR DATABASE

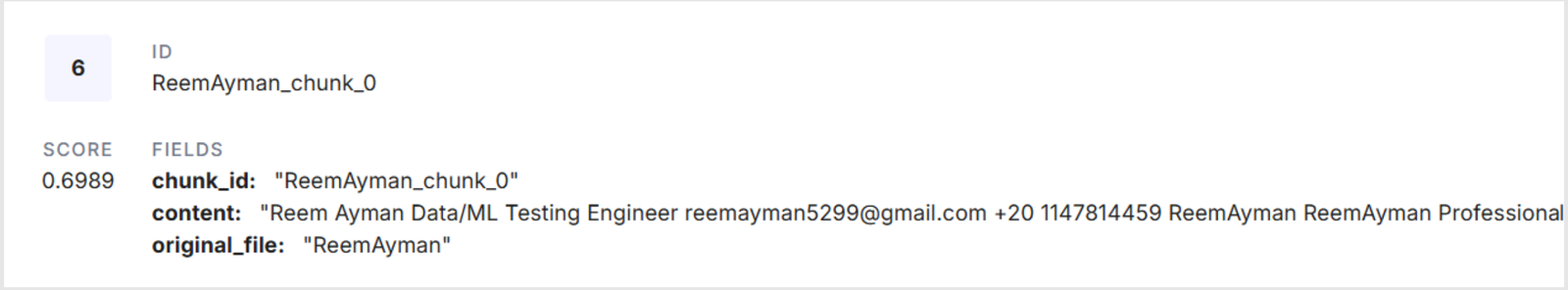
YMM-RESUMES SCREENER

Embedding Model :

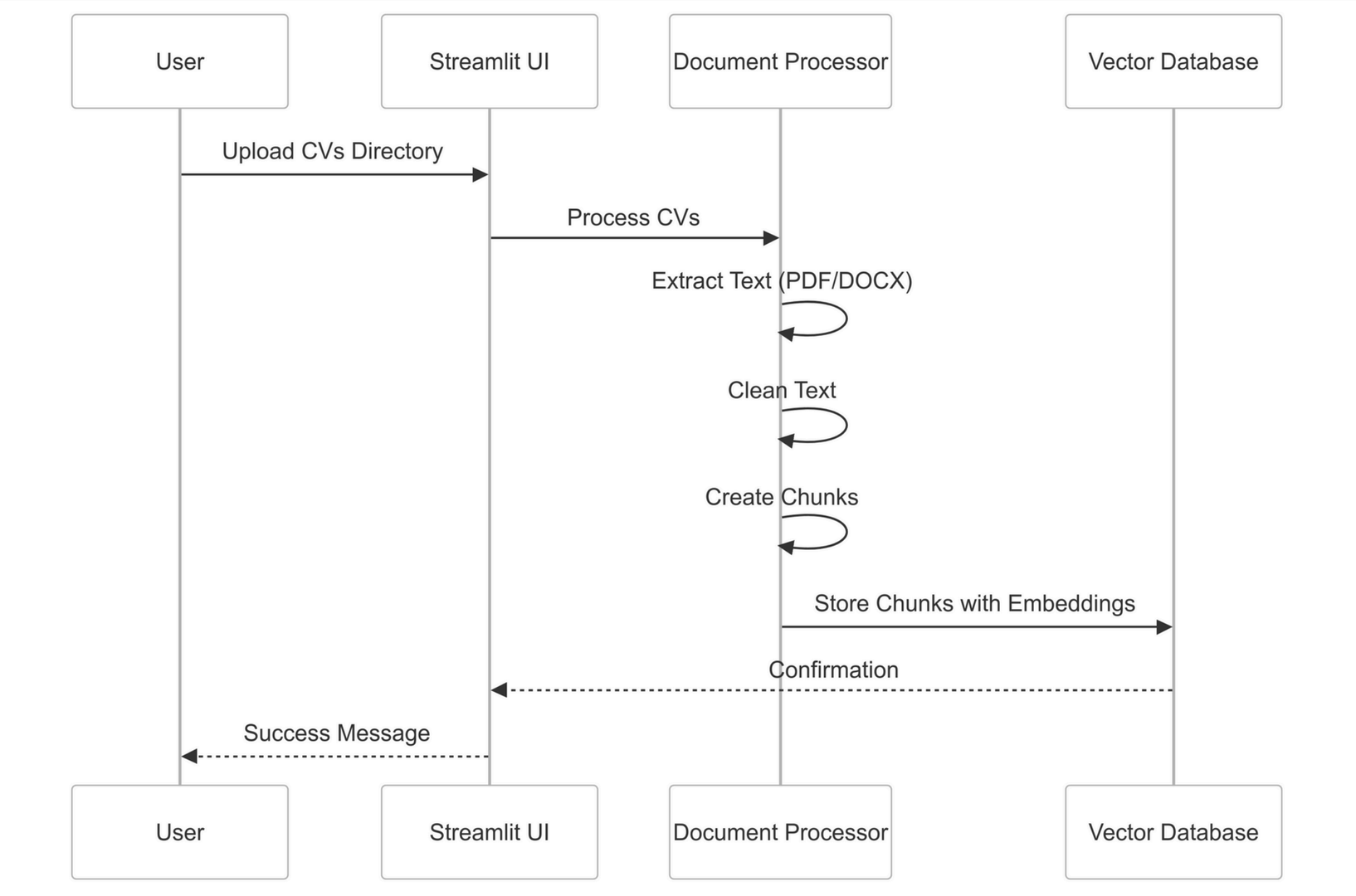
- Final Choice: **BAAI/bge-large-en-v1.5**
- Alternative Models Tried:
  - sentence-transformers/all-MiniLM-L6-v2 .
  - Cohere Embed API → Best in performance, but limited tokens per minute and month.

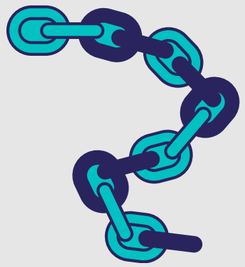
Vector Database: **Pinecone**

- Index Type Used: cosine similarity index



OFFLINE PIPELINE





RAG CHAIN

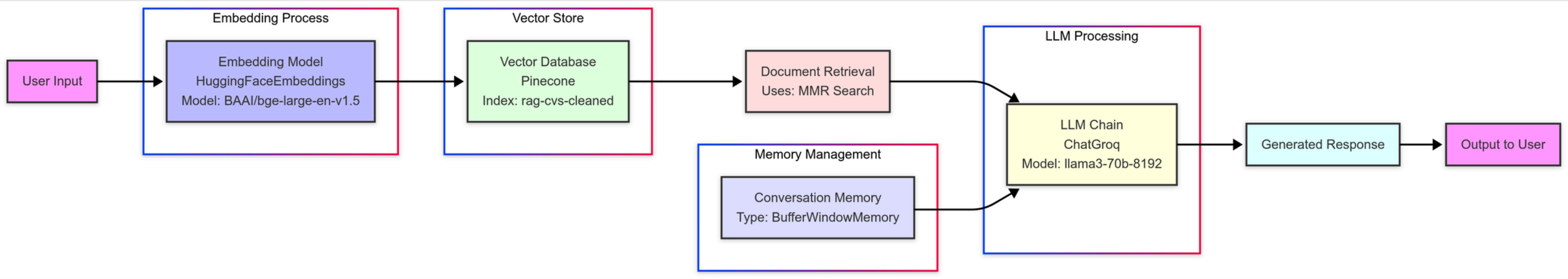
YMM-RESUMES SCREENER

Process the User Query

- Embedding the Query:
  - We use the same embedding model (**BAAI/bge-large-en-v1.5**).
- Similarity Search & Re-ranking:
  - We perform **MMR (Maximal Marginal Relevance)** search, balancing relevance &diversity.
- Retrieving and Ranking CVs:
  - The most relevant CVs are retrieved and ranked based on similarity.
- Conversational Memory:
  - We use a **conversation buffer window memory** instead of summary-based memory, as it performs better for short chat histories—ideal for our use case.
- Used Groq for Llm:
  - The chatbot leverages Meta's **Llama 3-70B,** which is one of the best free models

RAG CHAIN

YMM-RESUMES SCREENER



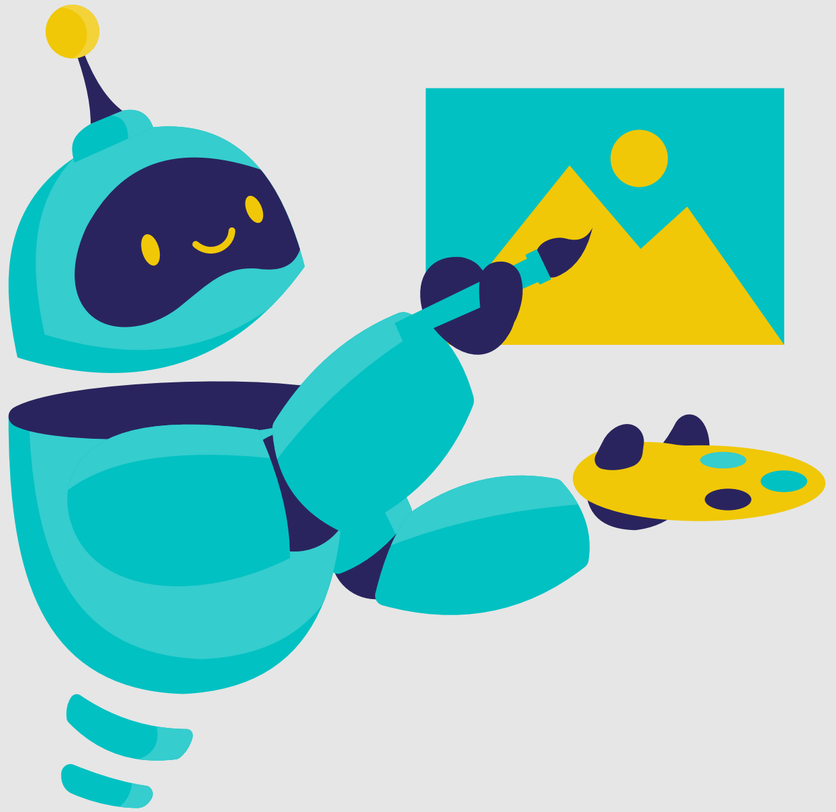
UI AND CONATINARIZATION

Streamlit

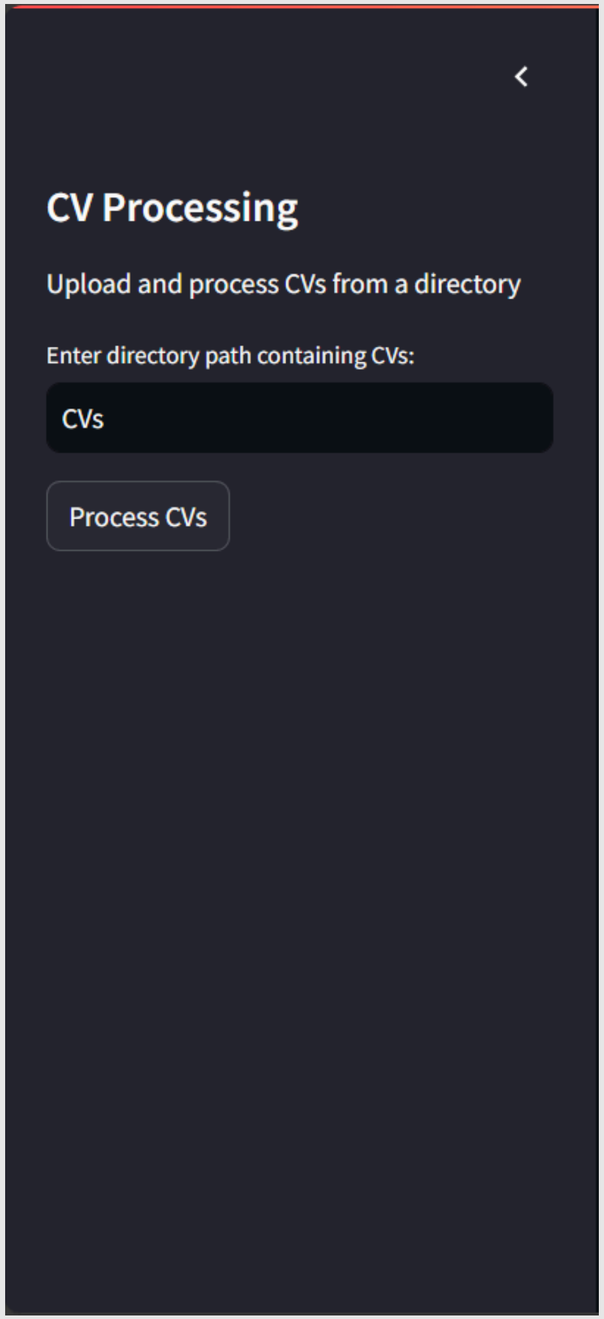
docker

YMM-RESUMES SCREENER

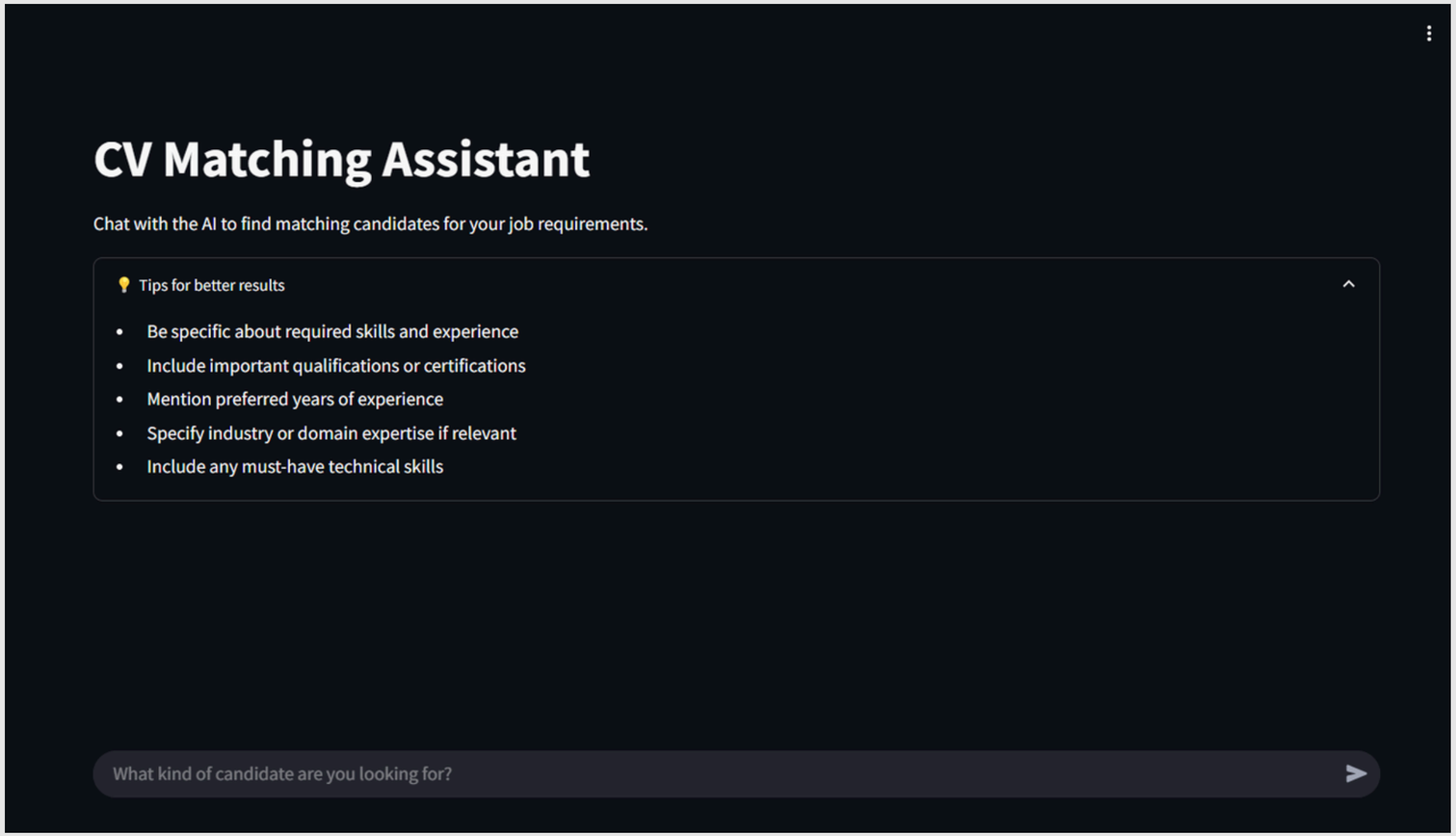
IMPLEMENTED FEATURES

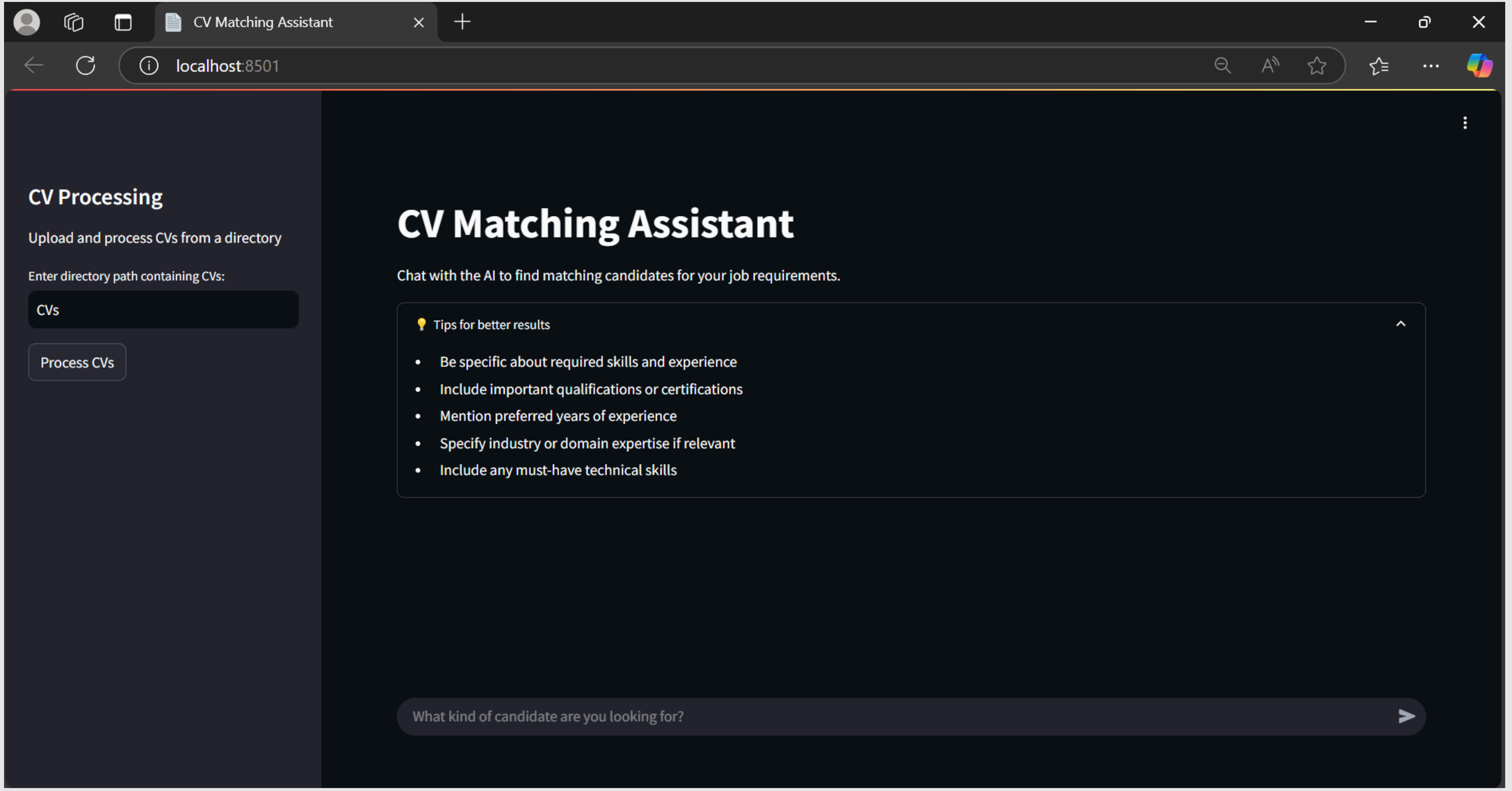


CV UPLOADING



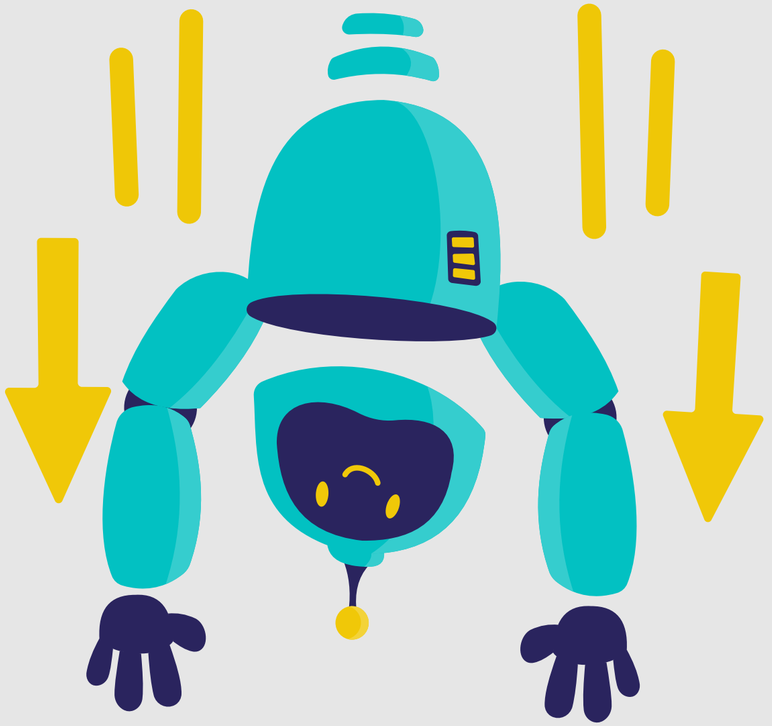
CHAT INTERFACE




CV Matching Assistant
localhost:8501
CV Processing
Upload and process CVs from a directory
Enter directory path containing CVs:
CVs
Process CVs
CV Matching Assistant
Chat with the AI to find matching candidates for your job requirements.
Tips for better results
Be specific about required skills and experience
Include important qualifications or certifications
Mention preferred years of experience
Specify industry or domain expertise if relevant
Include any must-have technical skills
What kind of candidate are you looking for?

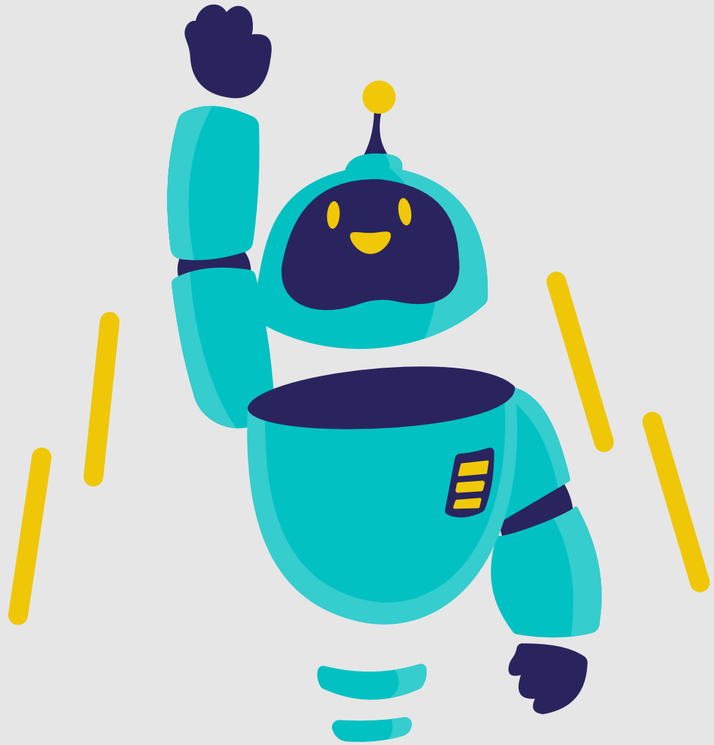
TECHNICAL CHALLENGES

- Packages confilict
- Resources constraints
- Various Options
- Returning correct person for each chunk
- Memory Management
- Selecting a good prompt



ENHANCEMENT AND FUTURE WORK

- Supporting more features in UI
- Enhance prompt
- Enhance response quality
- Latency improvement



YMM-RESUEME SCREENER

DEMO

THANK YOU