# ASSESSMENT DOCUMENTATION

By Smarika Sharma

smarikasharma89@gmail.com

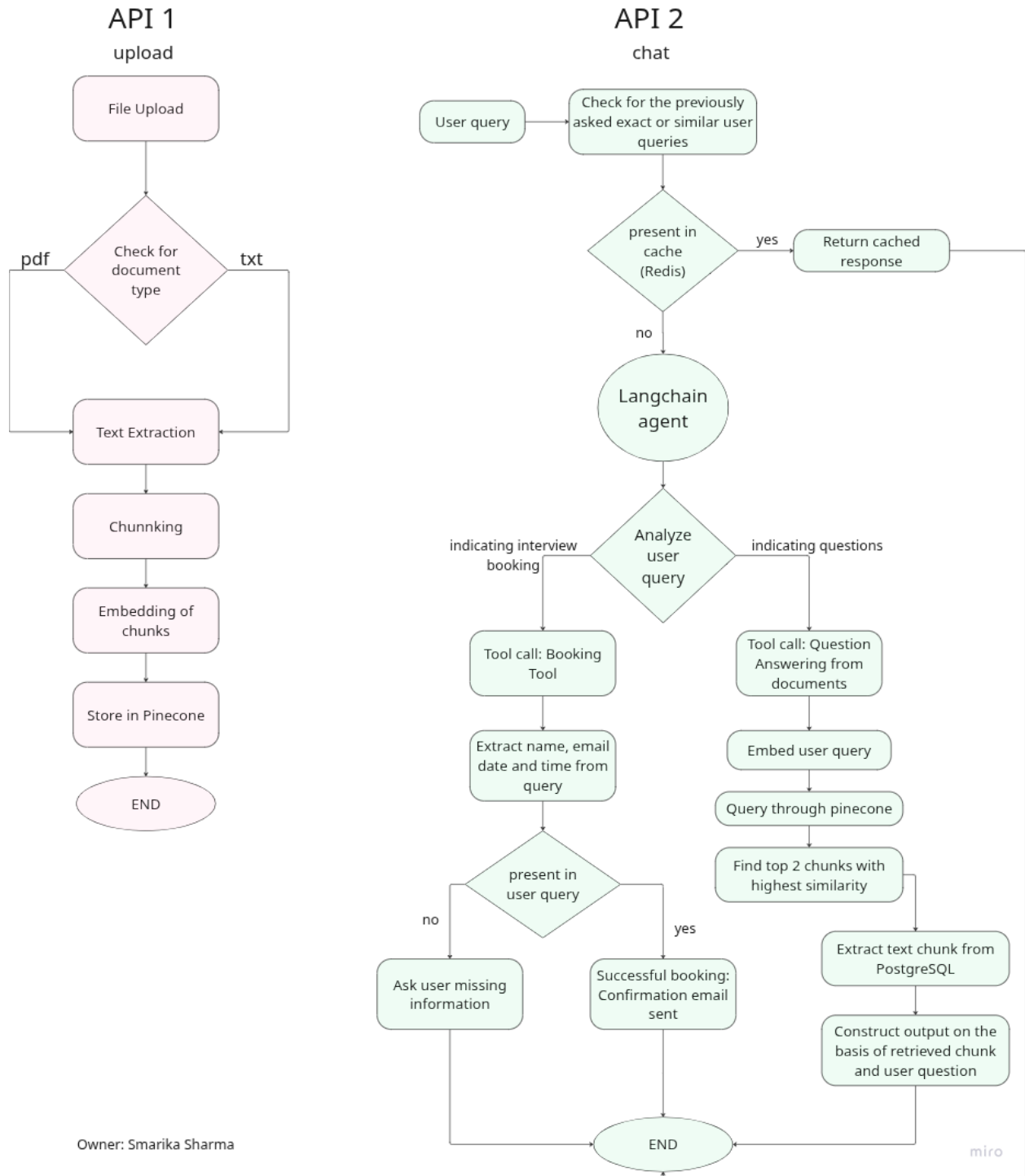# WORKFLOW

## API 1
### upload



File Upload

Check for document type

pdf / txt

Text Extraction

Chunnking

Embedding of chunks

Store in Pinecone

END

Owner: Smarika Sharma

## API 2
### chat

User query

Check for the previously asked exact or similar user queries

present in cache (Redis)

yes → Return cached response

no

Langchain agent

Analyze user query

indicating interview booking

indicating questions

Tool call: Booking Tool

Extract name, email date and time from query

present in user query

no → Ask user missing information

yes → Successful booking: Confirmation email sent

Tool call: Question Answering from documents

Embed user query

Query through pinecone

Find top 2 chunks with highest similarity

Extract text chunk from PostgreSQL

Construct output on the basis of retrieved chunk and user question

END

miro

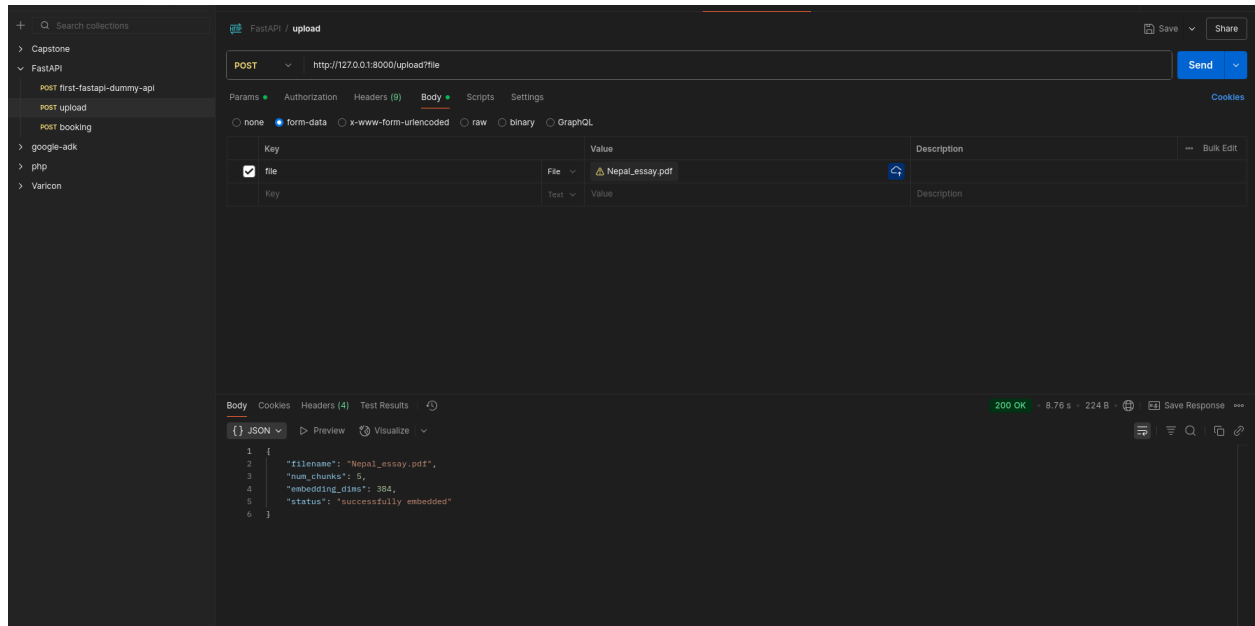Figure 1 : Project Workflow

# API 1 : UPLOAD



Figure 2: Pdf uploaded which is embedded and stored in vector database, Pinecone
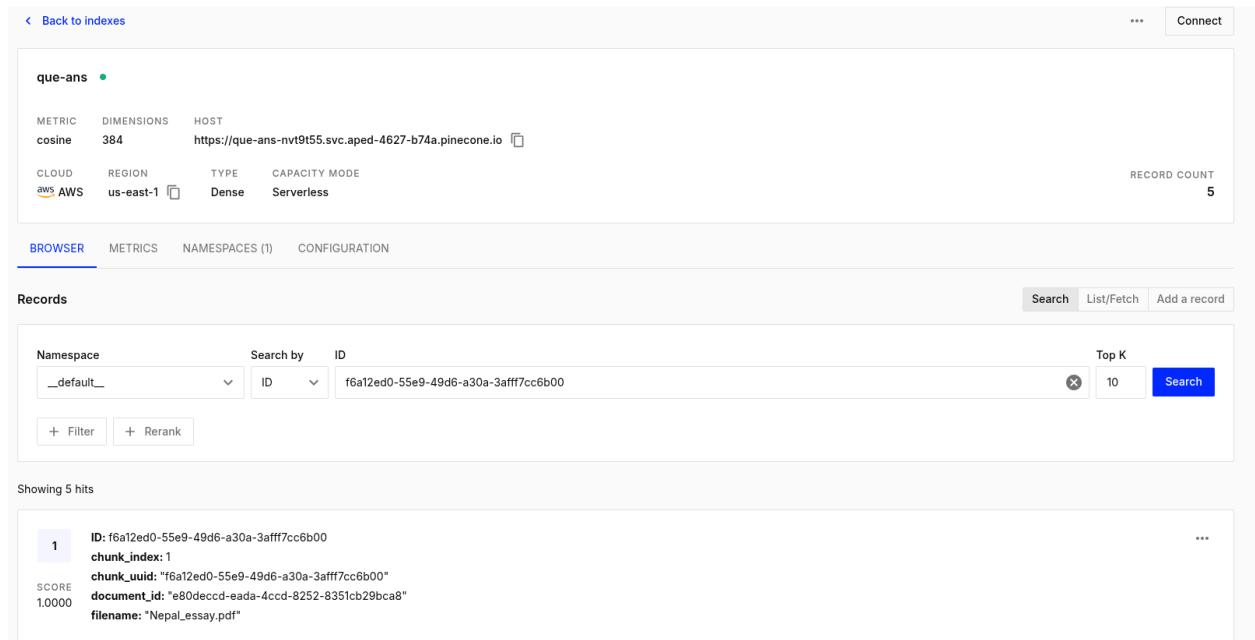


Figure 3: Text extracted from pdf, chunked, embedded and stored in Pinecone index

# API 2 : CHAT

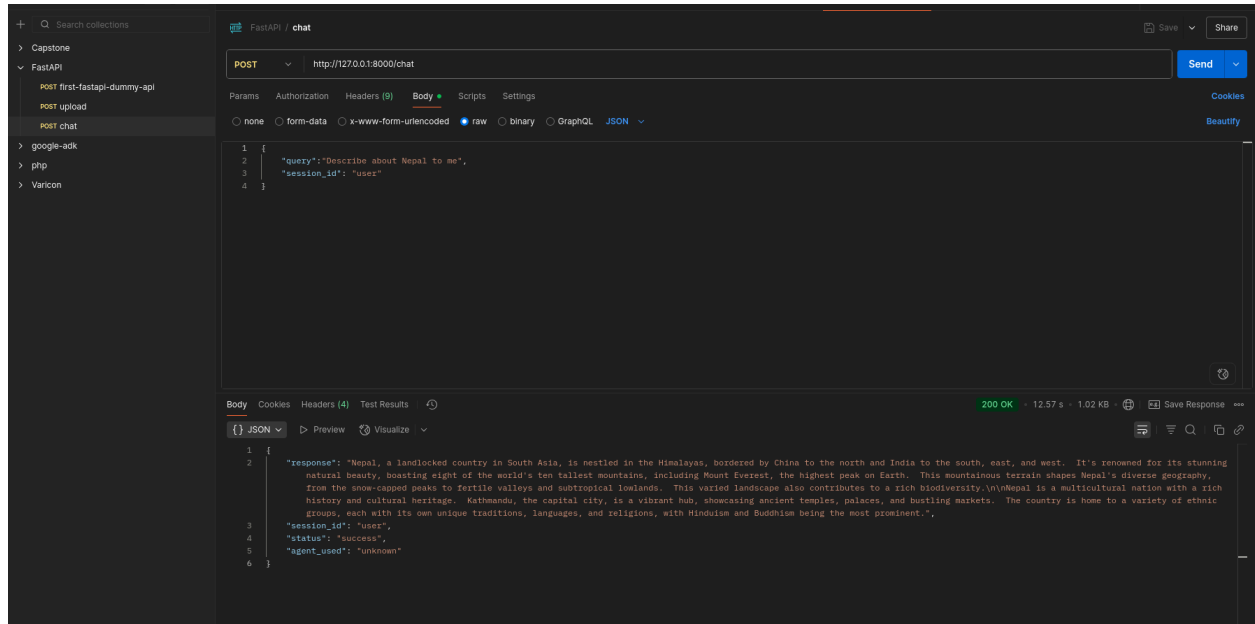This API implements RAG based agentic framework for handling user query and answering user questions.



Figure 4: User question being answered with extracted context using RAG (pinecone and metadata in PostgreSQL)
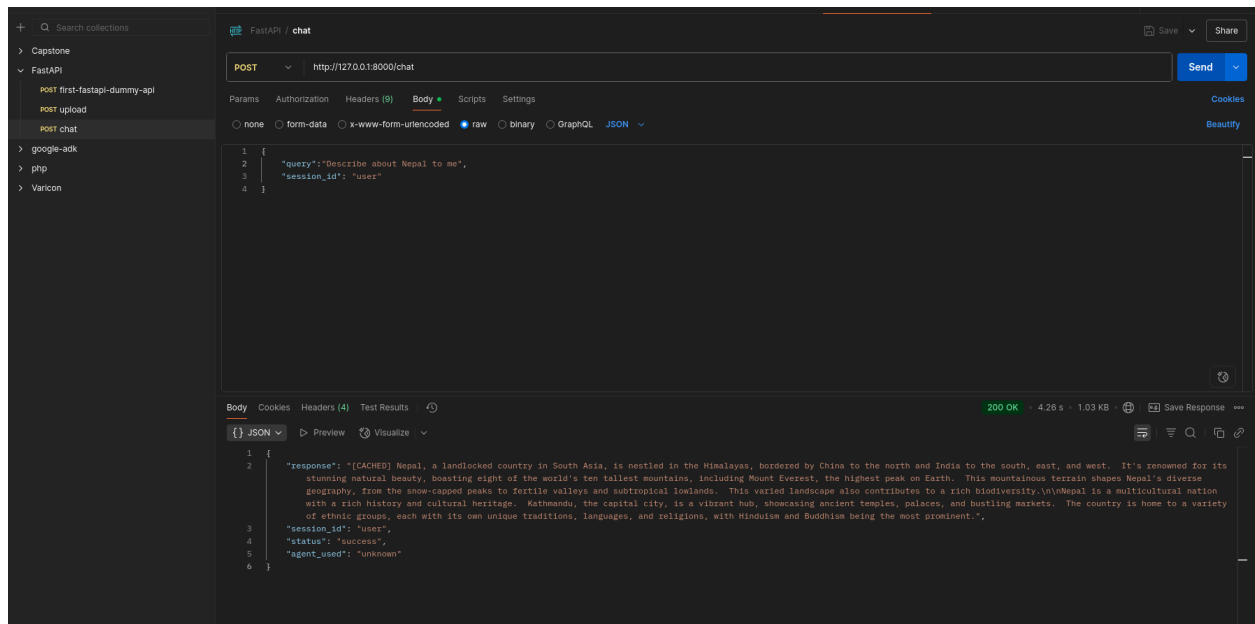
Here, the API took 12.5 seconds to be executed.



Figure 5: Quicker API response time after implementation of Redis (caching)
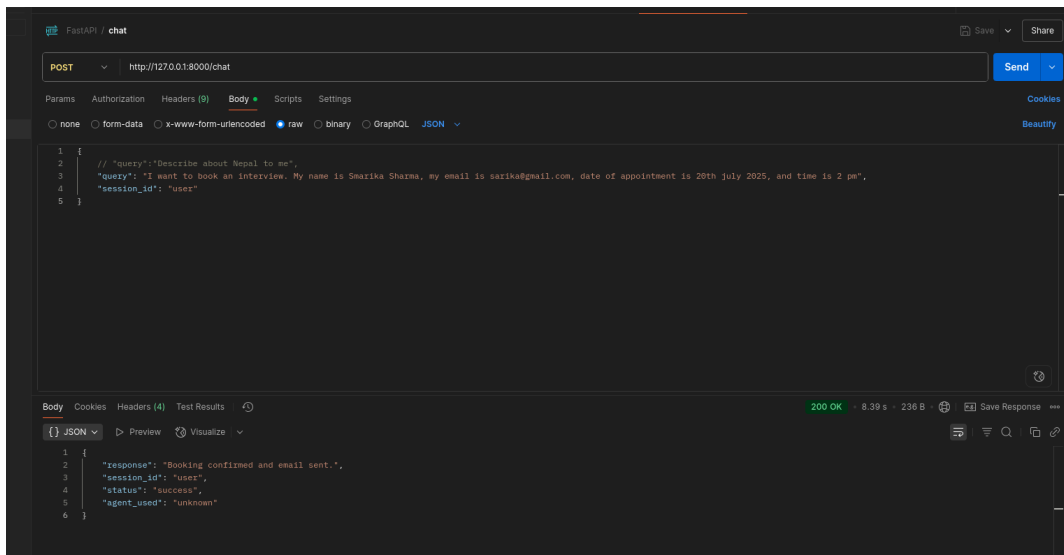
- **Booking of an interview:**

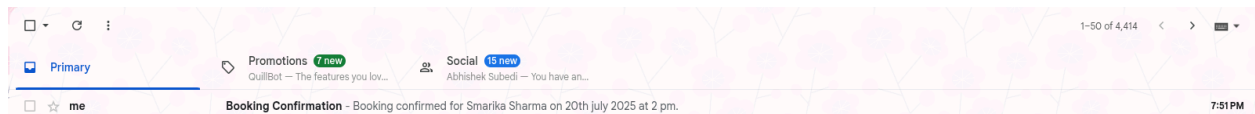Figure 6: Name, email, date and time being extracted from the query by agent to book for an interview



Figure 7: Email of booking sent to the mail

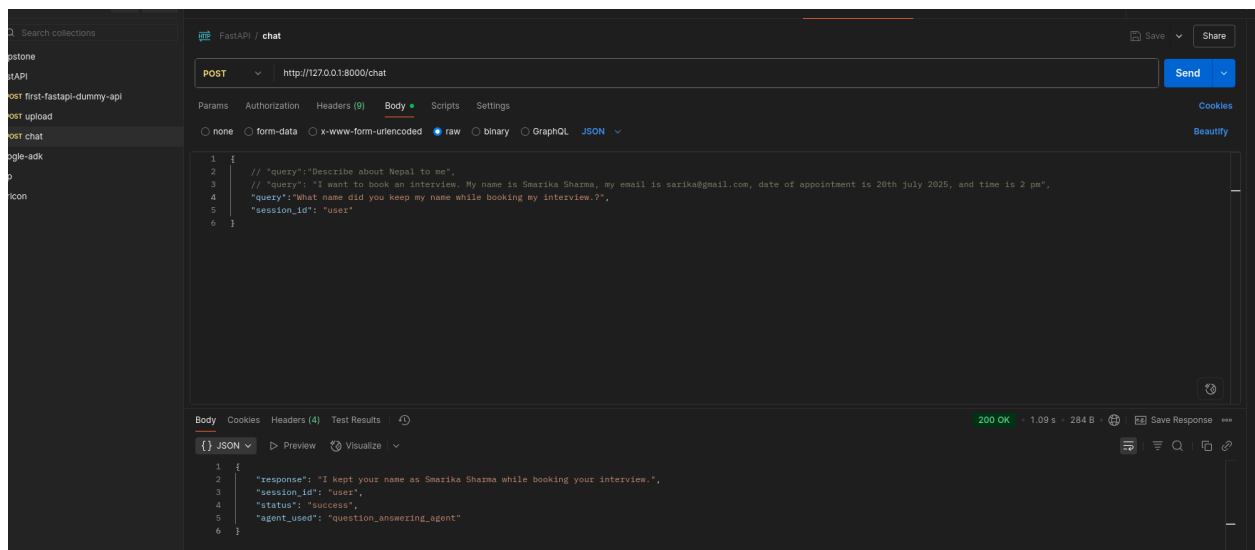● **Conversational/ Context Remembrance with the use of session**



Figure 8: Agent answers the previous conversation's context when asked