Project Title: Second Brain AI - System Architecture

1. Architectural Vision This project implements a **Local-First RAG (Retrieval Augmented Generation)** architecture. Unlike traditional cloud-native systems, this design prioritizes data privacy and zero-latency ingestion by performing vectorization on the edge (the user's machine) rather than relying on external API rate limits.

2. Core Components

- **Frontend:** React (Vite) + TailwindCSS for a responsive, real-time chat interface.
- **Backend:** FastAPI (Python) serving as the lightweight orchestration layer.
- **Vector Store: ChromaDB**. Chosen for its embedded nature, allowing the database to live within the application process. This removes the overhead of managing a separate database container (like Postgres) for a single-user application.
- **Embedding Model: all-MiniLM-L6-v2 (HuggingFace)**. A highly efficient 384-dimensional model running locally on the CPU. This ensures user documents are never sent to third-party APIs for indexing, guaranteeing privacy.
- **LLM: Google Gemini Flash**. Used strictly for the final answer synthesis, ensuring high speed and low cost.

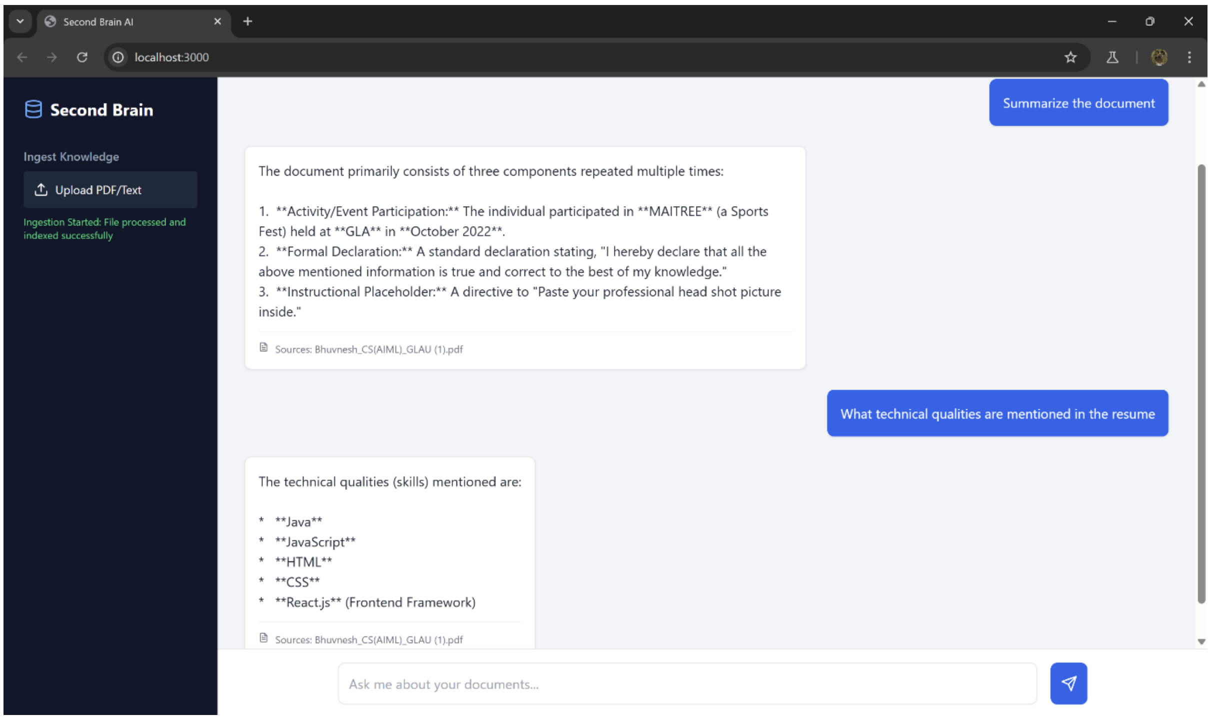
3. Data Flow

1. **Ingestion:** User uploads a PDF. The backend parses text using PyPDFLoader.
2. **Chunking:** Text is split into 500-character semantic chunks with 50-character overlap to preserve context.
3. **Local Indexing:** The local CPU converts chunks into vectors and stores them immediately in the persistent chroma_db on disk.
4. **Retrieval:** When a user asks a question, the system performs a Cosine Similarity search in ChromaDB to find the top 5 relevant chunks.
5. **Synthesis:** These chunks are injected into the Gemini system prompt to generate a grounded response.

4. Scalability & Trade-offs

- **Trade-off:** Processing is synchronous. For a massive multi-user system, I would introduce an asynchronous worker queue (Celery/Redis).
- **Decision:** For this "Personal Second Brain" MVP, the synchronous local model provides the fastest "Time-to-First-Token" and simplest deployment footprint.

Project Interface:



Ingestion Started: File processed and indexed successfully

Summarize the document

The document primarily consists of three components repeated multiple times:

1. **Activity/Event Participation:** The individual participated in **MAITREE** (a Sports Fest) held at **GLA** in **October 2022**.
2. **Formal Declaration:** A standard declaration stating, "I hereby declare that all the above mentioned information is true and correct to the best of my knowledge."
3. **Instructional Placeholder:** A directive to "Paste your professional head shot picture inside."

Sources: Bhuvnesh_CS(AIML)_GLAU (1).pdf

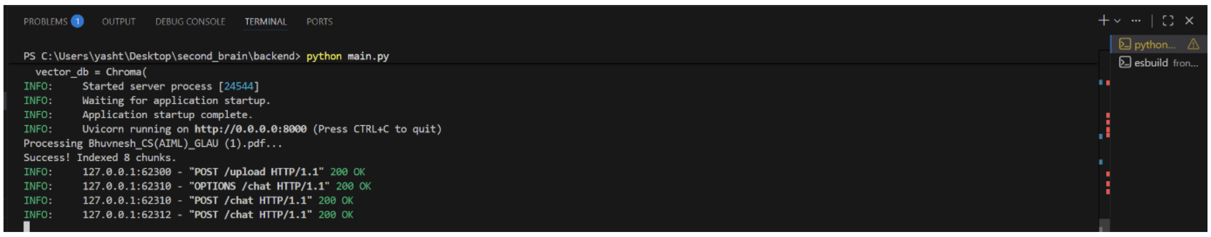
What technical qualities are mentioned in the resume

The technical qualities (skills) mentioned are:

* **Java**
* **JavaScript**
* **HTML**
* **CSS**
* **React.js** (Frontend Framework)

Sources: Bhuvnesh_CS(AIML)_GLAU (1).pdf

Terminal Log:



```
PS C:\Users\yasht\Desktop\second_brain\backend> python main.py
  vector_db = Chroma(
INFO:     Started server process [24544]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
Processing Bhuvnesh_CS(AIML)_GLAU (1).pdf...
Success! Indexed 8 chunks.
INFO:     127.0.0.1:62300 - "POST /upload HTTP/1.1" 200 OK
INFO:     127.0.0.1:62310 - "OPTIONS /chat HTTP/1.1" 200 OK
INFO:     127.0.0.1:62310 - "POST /chat HTTP/1.1" 200 OK
INFO:     127.0.0.1:62312 - "POST /chat HTTP/1.1" 200 OK
```