भारतीय सूचना प्रौद्योगिकी संस्थान सूरत
Indian Institute of Information Technology Surat
ભારતીય સૂચના પ્રૌદ્યોગિકી સંસ્થા સુરત
(An Institute of National Importance under Act of Parliament)

Team CodeBlooded

Multimodal Study Coach

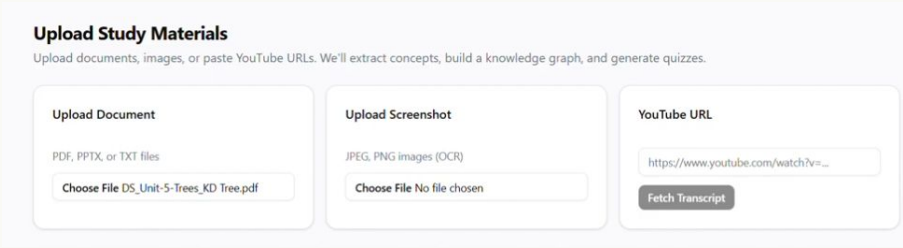
Students use multiple learning sources but cannot integrate them into structured study plans.
Built an AI tutor that merges video, text, and notes into intelligent learning paths.
A multimodal AI assistant that converts lectures and PDFs into structured knowledge.

Omkar Haryan, Panav Jogi, Keith Fernandes
Dwarkadas J. Sanghvi College of Engineering
Mumbai, Maharashtra

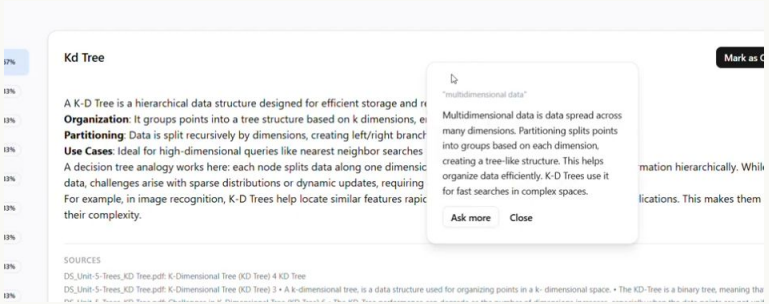
Overview of the project

Core Features: Beyond Simple Summarization

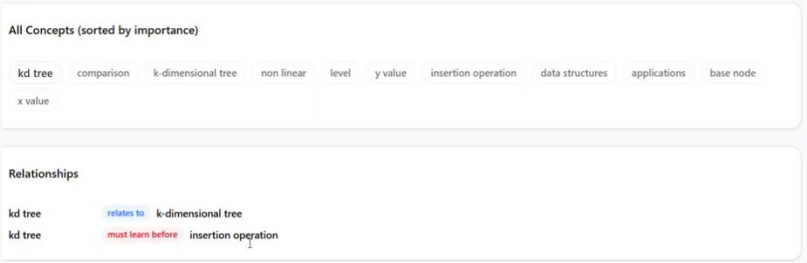
- **Multimodal Ingestion**: Automatically transforms fragmented PDFs, PowerPoint slides, and YouTube lecture URLs into a unified, searchable knowledge base.



- **In-Context Clarity**: Prevents distraction by allowing students to select text for instant, side-window definitions that maintain context without tab-switching.



- **Strategic Knowledge Graphing**: Unifies all sources—including PDFs, YouTube, and OCR-scanned screenshots—into a prerequisite-aware concept graph that prioritizes what to study next while flagging knowledge gaps based on quiz performance.



- **Adaptive Assessment**: Generates custom assessments (MCQ, conceptual, and short-answer) from raw materials to identify and close specific knowledge gaps.

Overview of the project

Project Architecture: Modular & Intelligence-Driven

- **Intelligence Layer**: Orchestrated by OpenRouter (Liquid LFM-2.5b Thinking model) for high-fidelity synthesis and automated question generation.
- **Knowledge Layer**: Employs ChromaDB for persistent vector storage and NetworkX to maintain prerequisite-aware concept graphs.
- **Student Model**: Tracks mastery via a hybrid scoring formula and the FSRS algorithm for mathematically optimized retention.

Competitive Edge: The "Closed-Loop" Advantage

- **End-to-End Efficiency**: Transitions from raw documents to adaptive quizzes in a single click, eliminating the need for manual flashcard creation.
- **Quantitative Mastery Signals**: Fuses accuracy, response time, and exposure counts into a data-driven mastery score to pinpoint weak topics.
- **Privacy-First Design**: Ensures all student data and progress remain local through persistent on-disk storage using SQLite and ChromaDB.

Tech Stack used

Core Stack: Modern, Full-Stack Excellence

- **Frontend: Next.js 16 (React)**: Chosen for its robust **Server-Side Rendering (SSR)** and **Static Site Generation (SSG)** to ensure a lightning-fast student dashboard and seamless "In-Context Definition" side windows.
- **Backend: FastAPI (Python)**: Acts as our high-performance orchestrator, handling heavy computational tasks like RAG logic and FSRS scheduling with minimal latency.

The Intelligence & Data Pipeline

- **OpenRouter (Liquid LFM-2.5b Thinking)**: Leverages advanced "thinking" models for accurate educational concept extraction and complex relationship mapping between prerequisite topics.
- **Vector & Knowledge Layer: ChromaDB & NetworkX**: We use ChromaDB for persistent, zero-config vector storage and NetworkX to maintain the logical, hierarchical relationships within the concept graph.
- **Embeddings: SentenceTransformers (all-MiniLM-L6-v2)**: Provides efficient, high-speed semantic retrieval without the high overhead of larger embedding models.

Why This Specific Stack?

- **Eliminating Context Switch**: By combining **Next.js** with **FastAPI**, we can provide a frictionless UI where students can get instant definitions in-app, preventing the distraction of switching to Google or ChatGPT.
- **Local-First Privacy**: Using **SQLite** (via SQLAlchemy) and **persistent ChromaDB** on disk ensures that student data and progress remain strictly local and private.
- **Science-Backed Learning**: The stack is specifically optimized to run the **FSRS (Free Spaced Repetition Scheduler)** algorithm, moving beyond static flashcards to mathematically optimized long-term memory retention

Approach to the project

Phase 1: Foundation & Ingestion (The Starting Line)

- **Project Skeleton:** We began by establishing a robust FastAPI backend and a Next.js frontend to ensure high-performance, asynchronous orchestration from day one.
- **Universal Document Pipeline:** The first functional component built was the ingestion engine using Unstructured.io to parse complex PDFs/PPTXs and youtube-transcript-api for video data, ensuring a "raw material" stream.
- **Vector Storage:** We prioritized the setup of **ChromaDB** for persistent vector storage and semantic retrieval, which serves as the memory for all subsequent layers.

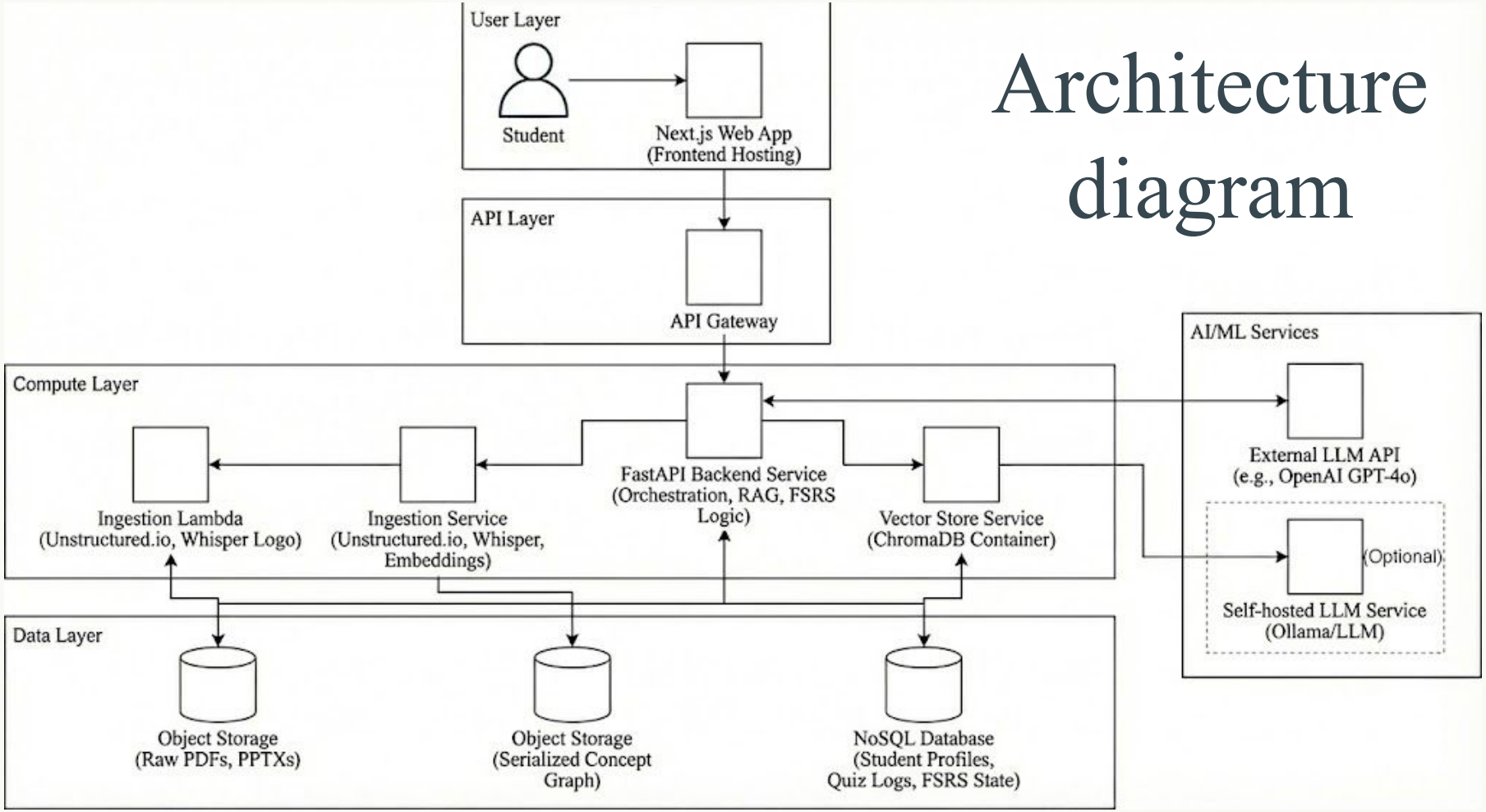
Phase 2: Intelligence & Synthesis (The Knowledge Core)

- **Concept Graph Construction:** Once data was flowing, we implemented LLM-based entity/relation extraction to build a **NetworkX** graph, mapping how educational concepts link together.
- **Automated Quiz Engineering:** We developed a multi-stage quiz generation module using LLM prompt templates to produce high-quality, Bloom's Taxonomy-aligned MCQs and conceptual questions.

Phase 3: Remediation & Personalization (Closing the Loop)

- **Student Modeling:** The final piece was the implementation of the **FSRS (Free Spaced Repetition Scheduler)** and our hybrid mastery formula to track knowledge state over time.
- **Intelligent Recommendations:** We built the recommendation engine to analyze mastery gaps and automatically suggest specific review sources (e.g., exact timestamps or slide numbers) for weak topics.

Architecture diagram



About your architecture

1. Ingestion: The Entry Point

The pipeline begins by capturing diverse study materials:

- **Unstructured.io**: Acts as our universal parser for complex document layouts like PDFs and PowerPoint slides.
- **YouTube Transcript API**: Seamlessly pulls lecture data from video URLs.
- **LangChain**: Orchestrates the initial data flow, preparing content for processing.

2. Knowledge: The Semantic Engine

Once ingested, the data is structured for deep understanding:

- **Sentence Transformers**: Converts text chunks into vector embeddings.
- **ChromaDB**: A high-speed vector store that allows the system to retrieve contextually relevant information instantly.
- **Concept Graph (NetworkX)**: Beyond simple lists, we build a prerequisite-aware map of how topics relate to each other, ensuring students learn in a logical order.

3. LLM/Quiz: The Intelligence Layer

This is where synthesis happens using **GPT-4o-mini** or local **Llama 3.1**:

- **Summarization & Mapping**: The LLM condenses long lectures into core concepts.
- **Adaptive Quizzing**: Instead of static questions, the system generates MCQs and conceptual tests dynamically based on the source material.
- **Orchestration**: LangChain manages the complex logic of feeding the right context to the LLM for accurate generation.

4. Student Model: The Feedback Loop

This final layer closes the loop, turning performance into personalization:

- **SQLite**: Stores persistent data on quiz attempts and student progress.
- **Hybrid Mastery Scoring**: We calculate a "Mastery Score" by fusing accuracy, response time, and exposure count.
- **Personalized Study Flow**: The system identifies weak topics and feeds them back into the loop, prioritizing them in the next study session to ensure no knowledge gaps remain.

Thank you

Relevant links here.

- Github https://github.com/Meanwhile-omkar/IIIT-Surat-Multimodal-edutech/tree/main
- Fill Form: https://forms.gle/bfCz861sDD8DHYUz6
- Hosted project link https://iiit-surat-multimodal-edutech.vercel.app/
- YT Video Explain https://youtu.be/Rj11_ZpqCDs