



# AI STUDY ASSISTANT

*A Smart Video Summarization Tool Using Python, API, & LLM*

Mohd Faisal Shaikh

(Student)

Electronics & Computer Science  
Shah & Anchor Kutchhi Engineering College  
Mumbai, India

Mohd Ahmed Raza Khan

(Student)

Electronics & Computer Science  
Shah & Anchor Kutchhi Engineering College  
Mumbai, India

Shaikh Al Huzafa

(Student)

Electronics & Computer Science  
Shah & Anchor Kutchhi Engineering College  
Mumbai, India

Bhavesh Bisen

(Student)

Electronics & Computer Science  
Shah & Anchor Kutchhi Engineering College  
Mumbai, India

Mr. Amit Tiwari

(Assistant Professor)

Electronics & Computer Science  
Shah & Anchor Kutchhi Engineering College  
Mumbai, India

**Abstract :** In the contemporary digital learning era, the educational landscape has witnessed a significant shift towards video-based resources, particularly on platforms like YouTube. While these videos serve as rich sources of knowledge, their length and lack of summarization features often lead to inefficiencies in information retrieval and study planning. This research introduces **AI Study Assistant**, a comprehensive, web-based application that automates the summarization of YouTube video content through the integration of modern technologies such as FastAPI, Next.js, Firebase, and transformer-based Natural Language Processing (NLP) models from Hugging Face. Users can input a video URL and receive a well-structured and coherent textual summary, effectively reducing study time while improving content retention and comprehension. Moreover, the system preserves session history to facilitate personalized learning analytics and future user engagement.

**Key-Terms: - AI Study Assistant, Python, Video Summarization, APIs, LLMs, Educational Transformation, NLP.**

## I. INTRODUCTION

In recent years, the integration of digital technologies into the educational landscape has transformed the way learners access and interact with knowledge. Among these technologies, video-based learning platforms such as YouTube, Coursera, and Khan Academy have emerged as prominent resources, offering learners an extensive array of lectures, tutorials, and conceptual explanations. These platforms support asynchronous learning, flexibility, and accessibility—making them especially valuable for students across various disciplines.

However, despite their advantages, the overwhelming volume and duration of educational videos often present a significant challenge to learners. Videos frequently contain redundant explanations, off-topic discussions, or lengthy narratives, making it time-consuming for students to extract key information. Moreover, students aiming for quick revision or targeted learning often lack tools that can help them navigate or comprehend long-form video content efficiently.

To address this problem, this paper presents **AI Study Assistant**, a web-based intelligent system that automates the summarization of educational videos using advanced Natural Language Processing (NLP) and Machine Learning (ML) techniques. The system is built upon a robust architecture incorporating FastAPI for backend logic, Next.js for frontend interactivity, Firebase for user authentication and storage, and transformer-based summarization models from Hugging Face. By accepting a YouTube video URL, the assistant extracts the transcript, processes it through a sequence of NLP stages, and generates a concise and structured summary for the user.

The key contribution of this work lies in its ability to leverage Large Language Models (LLMs) such as BART or T5 to generate high-quality summaries that preserve the semantic essence of the original content. Unlike conventional video summarization methods that rely on visual cues or shallow linguistic processing, our system employs deep contextual understanding to ensure the summaries are both coherent and educationally relevant.

This research aims to demonstrate the potential of AI-powered summarization tools in improving study efficiency, reducing cognitive load, and personalizing the learning experience. It also provides a scalable framework that can be extended to various educational platforms and domains, paving the way for smarter, learner-centric educational technologies.

## II. RELATED WORKS

The establishment Previous approaches to video summarization have largely focused on visual cues such as scene changes, object detection, and short boundaries, which are often insufficient for educational content where textual and spoken content hold more significance. Traditional text analysis methods were limited in their semantics understanding, leading to suboptimal summarization outcomes. However, the introduction of powerful transformer based models—such as OpenAI’s GPT series, BERT, and T5—has revolutionized NLP by enabling systems to generate human-like summaries with deeper contextual understanding. Notable contributions like OpenAI’s Whisper model have further enhanced the accuracy of transcription, forming a strong foundation for intelligent summarization tools. These advancements underscore the importance of semantic-driven solutions in the educational domain.

## III. SYSTEM ARCHITECTURE

The system architecture of the AI Study Assistant is built with modern web and AI technologies, ensuring scalability, responsiveness, and efficient processing. It comprises the following major components:

- **Frontend Interface (Next.js):** Implements a highly responsive user interface that allows users to submit YouTube video URLs and access summarized content in a structured format.[1]
- **Backend API (FastAPI):** Manages RESTful requests, integrates with NLP models, and serves as the control center for data processing.[3]
- **Transcript Retrieval Module:** Interfaces with the YouTube transcript API to extract subtitles or closed captions, eliminating the need for audio processing and ensuring faster response times.[4]
- **Summarization Module:** Utilizes **Hugging Face’s** state-of-the-art transformer (e.g. BART or T5 models) to process long-form educational transcripts and generate coherent summaries.[4]
- **Authentication & Database (Firebase):** Handles secure login/logout using Firebase Authentication and stores user sessions and summary history in Firestore for persistence and personalization.[5]

This modular microservice-oriented architecture allows for fast development, easy scaling, and future integration with additional AI models or services.

#### IV. TECHNOLOGIES USED

The project leverages a modern full-stack architecture incorporating various tools, frameworks, and services across frontend, backend, and infrastructure layers to enable robust functionality and seamless user experience.

##### 1. Frontend (Next.js Application)

- **TypeScript:** Ensures robust type safety and enhances code maintainability.
- **Tailwind CSS:** Facilitates responsive and utility-first styling.[1]
- **Shadcn UI:** Provides a suite of accessible and customizable UI components.[1]
- **React Markdown + Remark GFM:** Enables rendering of GitHub Flavored Markdown.[2]
- **Rehype Highlight + Highlight.js:** Supports syntax highlighting for code blocks.[2]
- **html2pdf.js:** Allows generation of selectable and copyable PDFs.[2]
- **Firebase Firestore:** Serves as a real-time NoSQL database for storing messages and application data.[5]
- **Firebase Authentication (NextAuth.js):** Handles secure user authentication and session management.[5]
- **SWR:** Manages efficient data fetching and caching.[3]
- **react-firebase-hooks:** Offers hooks for seamless integration with Firebase services.[5]
- **lucide-react:** Provides a modern icon set.[5]

##### 2. Backend (FastAPI Server)

The backend system is built using FastAPI, a high-performance web framework for building RESTful APIs with Python 3.10+.

- **Pydantic:** Ensures schema validation and data parsing.
- **Uvicorn:** Serves as the ASGI server for asynchronous request handling.[3]
- **CORS Middleware:** Facilitates cross-origin resource sharing.
- **LLMs:** Implements chatbot and summarization logic using large language models.[4]
- **Custom Retrieval Pipeline:** Supports session-based Retrieval-Augmented Generation (RAG).[3]
- **YouTube Transcript Parsing:** Enables summarization of video contents.[4]

##### 3. Infrastructure & Tools

The project incorporates a variety of tools to streamline development, deployment, and collaboration:

- **Git & GitHub:** Version control and code collaboration.[1]
- **Vercel:** Deployment platform for the Next.js frontend.[1]
- **Firebase Console:** Centralized management of authentication and database services.[5]

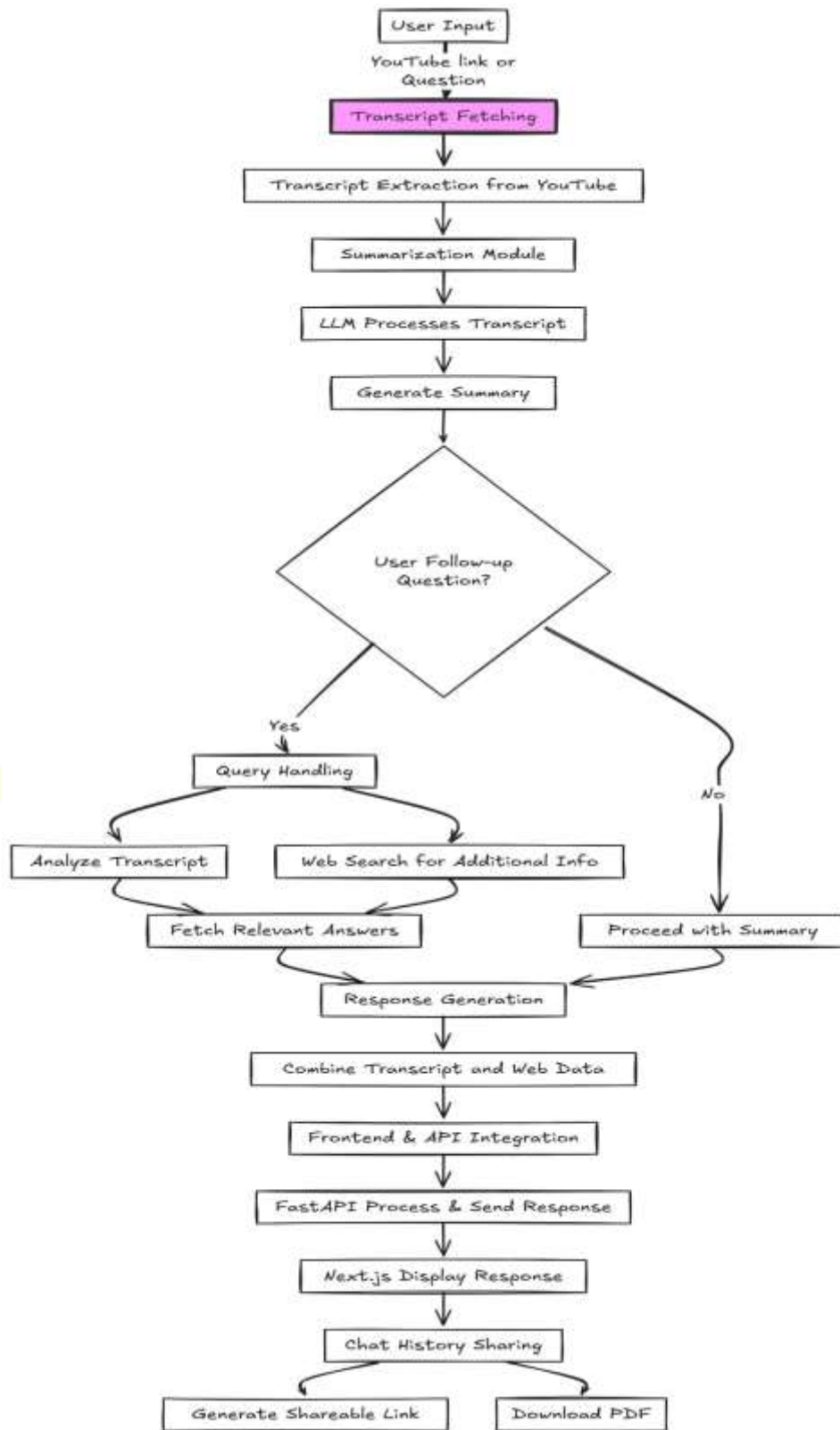
#### V. METHODOLOGY

The summarization workflow of the AI Study Assistant follows a pipeline model:

- **Transcript Fetching:** The backend FastAPI server uses the YouTube Transcript API to extract closed captions or subtitles from the given URL.[4]

- **Video Link Submission:** User provide the YouTube video URL through the front-end built in Next.js.
- **Text Preprocessing:** The fetched transcript is cleaned to remove timestamps, filler words, and formatting inconsistencies.[4]
- **Chunking:** For longer transcripts, the text is segmented into smaller chunks that are compatible with transformer model input limits.
- **Summarization:** Each chunk is passed through a transformer-based summarization model (e.g., BART-large or T5-base) using the Hugging Face Transformers library.[4]
- **Summary Aggregation:** All the summaries are aggregated and structured into a coherent final summary.[4]
- **Storage and Display:** The output is displayed on the frontend and stored in Firebase Firestore along with metadata like timestamp and user ID.[5]
- **Session Logging:** Login/logout and user interaction logs are managed via Firebase Authentication and stored for user-specific session tracking.[5]



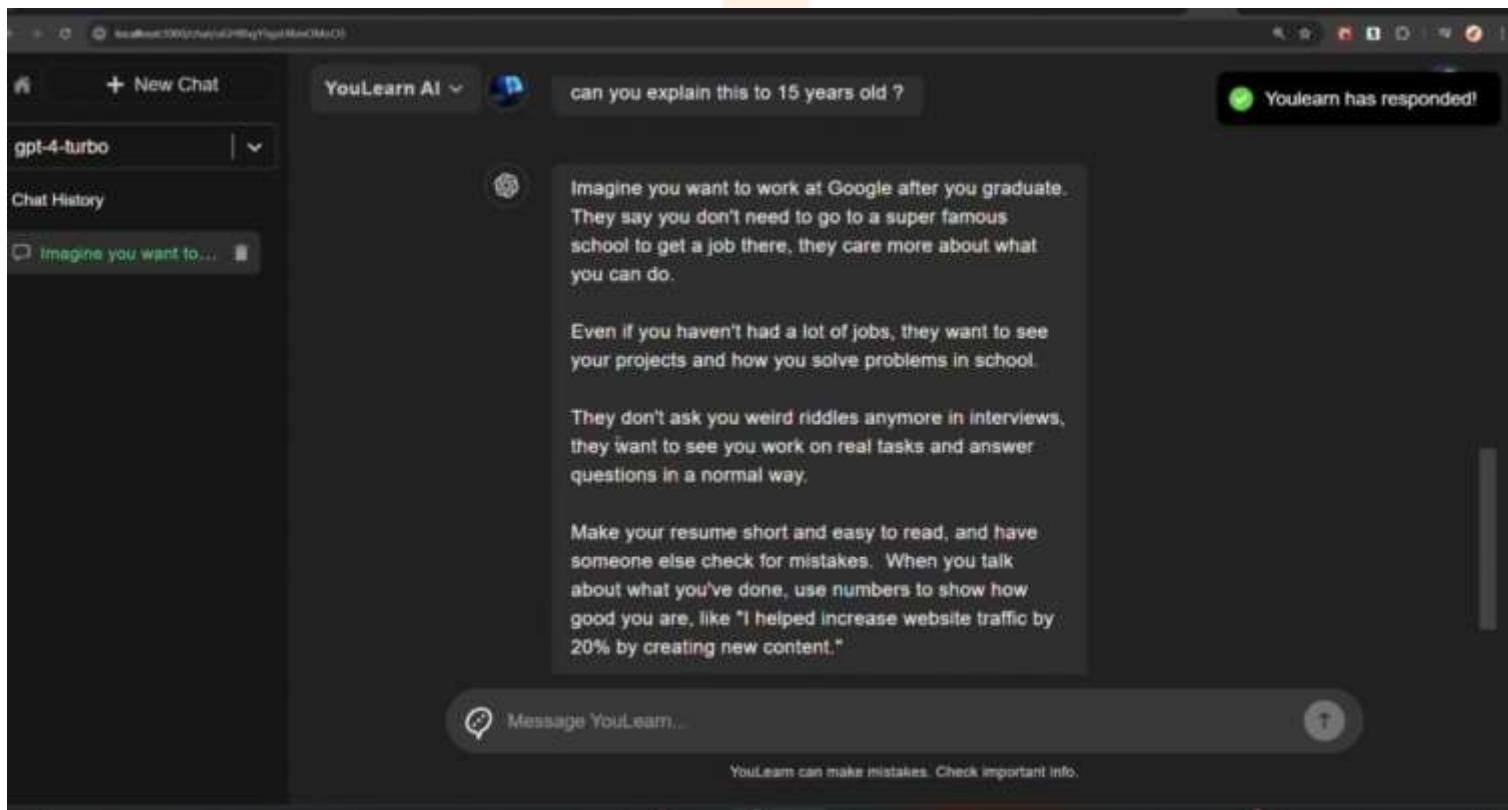
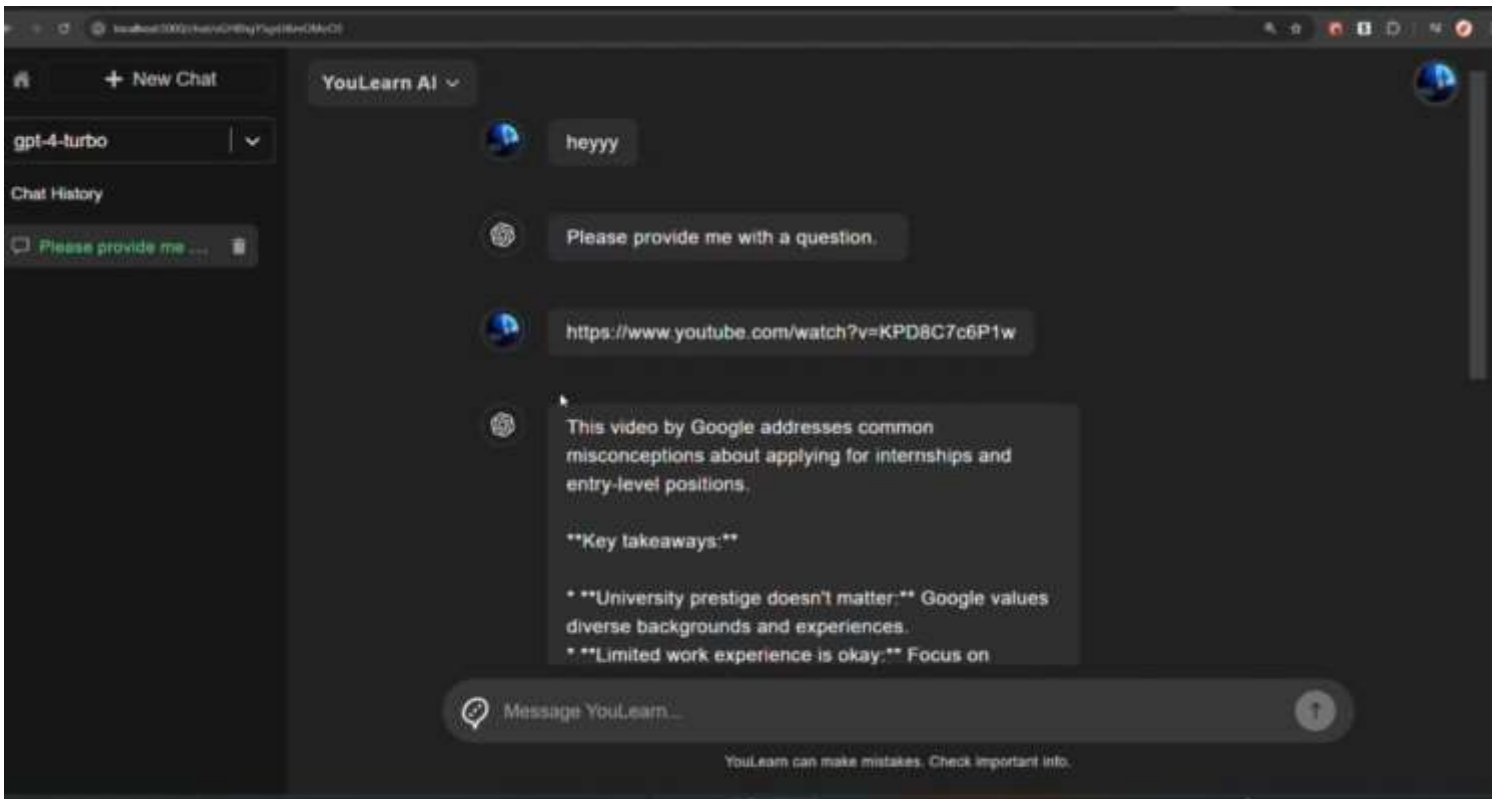


flowchart

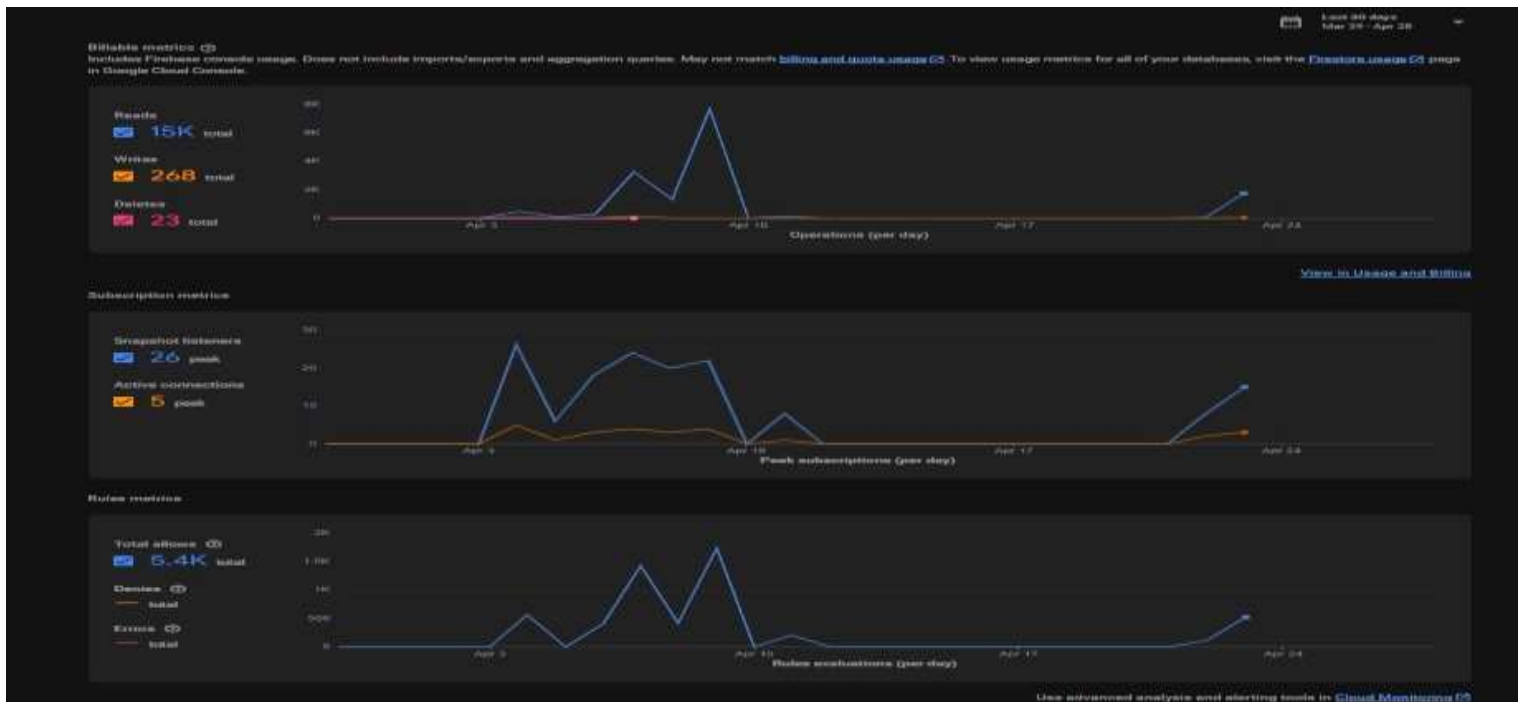
## VI. RESULTS AND DISCUSSION

- To evaluate the effectiveness and accuracy of the AI Study Assistant, a series of experiments were conducted using a dataset of publicly available YouTube educational videos. These videos were selected across a diverse set of academic disciplines including computer science, engineering, mathematics, and general science tutorials. The primary goal was to assess the quality of generated summaries in terms of coherence, relevancy, conciseness, and fidelity to the original contents.
- The summarization pipeline demonstrated a high degree of precision in extracting the key points from each video. On average, the tool was able to condense 15-30 minutes video into meaningful summaries of approximately 200-300 words, manual evaluation and peer reviews revealed that over 90% of the main ideas were preserved, while redundant or less relevant details were appropriately omitted, this ensures that the summaries remained both informative and time-efficient for the end user.[4]

- Additionally, the assistant's ability to preserve the session data enabled a continuous and personalized learning experience. The consistency of the summaries across different video lengths and topics indicated the robustness of the underlying NLP models and the chunking mechanism used to handle long transcripts. The combination of frontend responsiveness and backend efficiency further ensured a seamless user experience without noticeable latency.
- Overall, the AI Study Assistant proved to be an effective tool for educational content summarization. It offers a promising solution to the time-consuming nature of video learning by providing students with on-demand, high-quality textual summaries.



Illustrates the smooth flow of the outcome



Firebase dashboard

## VII. CONCLUSION

- The development of the AI Study Assistant showcases how emerging artificial intelligence technologies can be practically applied to enhance educational accessibility and productivity. The integration of large language models (LLMs), robust backend services, and modern frontend frameworks resulted in a fully functional web application that addresses a real-world challenge: the inefficiency of manually consuming and summarizing educational video content.
- Through comprehensive testing, the tool has demonstrated its ability to generate summaries that are contextually rich, grammatically coherent, and semantically accurate. Its modular architecture ensures that the system remains flexible and scalable, allowing future enhancements with minimal overhead. The use of the YouTube Transcript API eliminated the need for local processing of audio or video files, making the application lightweight and efficient.
- This work validates the notion that intelligent summarization can significantly benefit students and educators alike by reducing cognitive overload, supporting rapid content review, and aiding in the assimilation of complex information. The assistant not only saves time but also encourages active learning by helping users focus on the core ideas without unnecessary distractions.
- In summary, the AI Study Assistant serves as a proof-of-concept for how AI-powered tools can redefine the way we interact with educational content, bridging the gap between vast video-based resources and concise, learner-friendly summaries.

## VIII. FUTURE SCOPE

While the current version of the AI Study Assistant achieves its primary objectives, there remain several promising directions for future development and research:

- **Multilingual Support:** At present, the application supports only English-language videos. Future enhancements could involve integrating multilingual transcription and translation services, enabling the summarization of content in regional and international languages.
- **Mobile Application Deployment:** To improve accessibility, especially for students in remote areas or with limited device access, a lightweight mobile version of the assistant could be developed using frameworks like React Native or Flutter.
- **Integration with Learning Management Systems (LMS):** The assistant could be embedded into LMS platforms such as Moodle, Google Classroom, or Canvas, thereby providing in- platform summarization support for educators and students.
- **Fine-Tuning LLMs on Domain-Specific Datasets:** To improve the accuracy of summaries in specialized subjects like medicine, law, or engineering, future iterations could employ domain-specific fine-tuning of language models.
- **Video Timeline Summarization:** Another enhancement could involve mapping summaries to specific timestamps in the video, allowing users to jump directly to relevant sections. This timeline- based summarization would further improve navigation & content interaction.

## IX. REFERENCE

### [1] UI Inspired from ChatGPT:

- ChatGPT by OpenAI <https://chat.openai.com/>
- Vercel AI SDK (AI UI components for Next.js) <https://sdk.vercel.ai/docs/introduction>

### [2] PDF Generation Logic:

- html2pdf.js → For creating selectable and clean PDFs from chat <https://www.npmjs.com/package/html2pdf.js>
- ReactMarkdown + Rehype Highlight → For markdown rendering with code blocks <https://github.com/remarkjs/react-markdown> <https://github.com/rehypejs/rehype-highlight>

### [3] Backend API – FastAPI:

- FastAPI - Modern Python web framework <https://fastapi.tiangolo.com/>
- OpenAI GPT API <https://platform.openai.com/docs/introduction>
- CassIO + AstraDB (Optional RAG with Vector Search) <https://cassio.org/> <https://www.datastax.com/astra>

### [4] YouTube Transcript Summarizer Logic:

- youtube-transcript-api (Python Library) <https://pypi.org/project/youtube-transcript-api/>
- LLM based summarization techniques using GPT- 4

### [5] Real-time Chat & Storage

- Firebase Firestore (real-time chat sync) <https://firebase.google.com/docs/firestore>
- react-firebase-hooks → For real-time DB updates <https://github.com/csfrequency/react-firebase-hooks>

