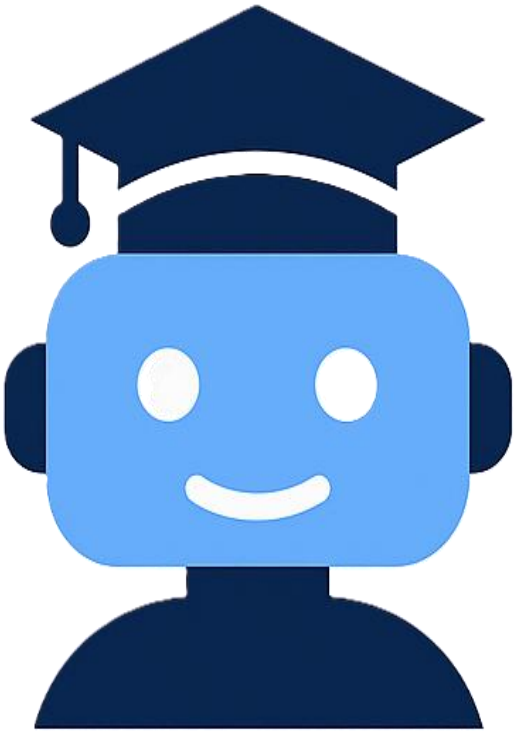


LectureAI

– Your Personal AI Tutor

Summer Internship Report

Topic: AI Agents
for automation of tasks

By: Ayushman Anupam
Tejaswi Sherawat

Mentors:
Animesh Guchhait, Chandrashish Prasad,
Hithesh KK, Rahul Shah

AlgoLabs

Abstract

LectureAI, an agentic AI solution designed to transform how users interact with and learn from video content. The last decade has seen an exponential growth in educational video content, often presented linearly, leading to key drawbacks such as information overload, time-consuming searches, and limited interaction. LectureAI addresses these challenges by transforming passive video consumption into an interactive, insightful, and personalized learning experience.

The core of LectureAI is its agentic AI chatbot, which operates on lecture content and uses APIs to process information. In addition to the chatbot which t offers several features to help learning. It can generate multiple-choice questions (MCQs) and provide detailed evaluation. The application also creates a visual mind map of the video's structure and key concepts by generating. Finally, it generates well-structured notes, including an introduction, key takeaways, and a conclusion.

Introduction

LectureAI is a personal AI tutor designed to transform how users interact with and learn from video content. Instead of just watching videos passively, users can now interact with the content in a smarter, more engaging way. At its core LectureAI is built on an *agentic* AI framework —a system made up of network of multiple AI agents that work together to process user requests and provide intelligent responses. These frameworks transform passive content into an interactive and personalized experience. Each **AI agents** are the individual components within an agentic AI framework. They are programmed to operate on specific content, such as lectures, and use API calls with context to remotely query and process information.

The LectureAI chatbot is an agentic AI system with four specialized AI agents at its disposal. The **RAG_Chatbot Tool –** for answering questions using retrieval-based methods, **Find_Timestamp_Tool -** – to locate the exact part of the video being referred to, **Explain_Timestamp_Tool** – for explaining what's happening at a specific point in the video**, General_Chatbot_Tool-** for broader conversations and support beyond Video content. The entire system is controlled by a central agent, which uses an LLM to decide which tool to invoke, and a validation component ensures the final response is accurate. This multi-agent approach allows LectureAI to offer a comprehensive and dynamic learning environment, moving beyond the limitations of simple video players.

Along with these features, It also has several features to help learning. It can generate multiple-choice questions (MCQs) and provide detailed and comprehensive evaluation. The application also creates a visual mind map of the video's structure and key concepts by generating. Finally, it generates well-structured notes, including an introduction, key takeaways, and a conclusion - all powered by AI agents.

Problem Statement

In traditional lectures, videos, and other forms of media, information is usually presented in a linear way. This leads to several challenges for learners:

- **Linear Information Flow** - Videos present information in a fixed, step-by-step manner. So, one can't easily jump to what they need, making it hard to learn at their own pace or based on their current understanding.
- **Information Overload** - Videos are often rich in content, but users struggle to quickly find specific insights or important moments.
- **Time-Consuming Search** - Finding relevant answers or sections requires manually scrolling through long parts of the video, which is time-consuming and inefficient.
- **Limited Interaction** - Most video players only offer basic features. They don't support smart search, real-time summaries, or contextual question-and-answer interactions.
- **Workflow Breaks from Knowledge Gaps -** When a video doesn't answer a user's question, they have to leave the platform and search elsewhere. This disrupts focus and breaks the learning flow.
- **No Personalized Feedback or Understanding Metrics** - There's no way to check what the viewer has understood, where they're struggling, or how their learning could be improved.

Our Agentic AI solution

To solve the problems faced in traditional video-based learning, we developed **LectureAI**, an agentic AI system that turns passive video content into an interactive, insightful, and personalized learning experience.

Key features of LectureAI include:

- **Query-Based Interaction -** Users can ask questions directly related to the video and get instant answers.
- **Timestamp-Based Summarization and Concept Locator -** The system summarizes specific parts of the video and helps locate key concepts with exact timestamps.
- **Beyond Video Knowledge -** If the answer isn't in the video, LectureAI can use web knowledge or stored memory to respond intelligently.
- **MCQ Generator and Evaluator-** Automatically creates multiple-choice questions and provides detailed evaluations to assess understanding.
- **Mind Map Generator & Word Cloud -** Visual tools that show the structure of the video and highlight important topics and keywords.
- **Generates Well-Structured Notes -** Produces notes with a clear structure, including an introduction, key takeaways, conclusion, and suggestions for further learning.

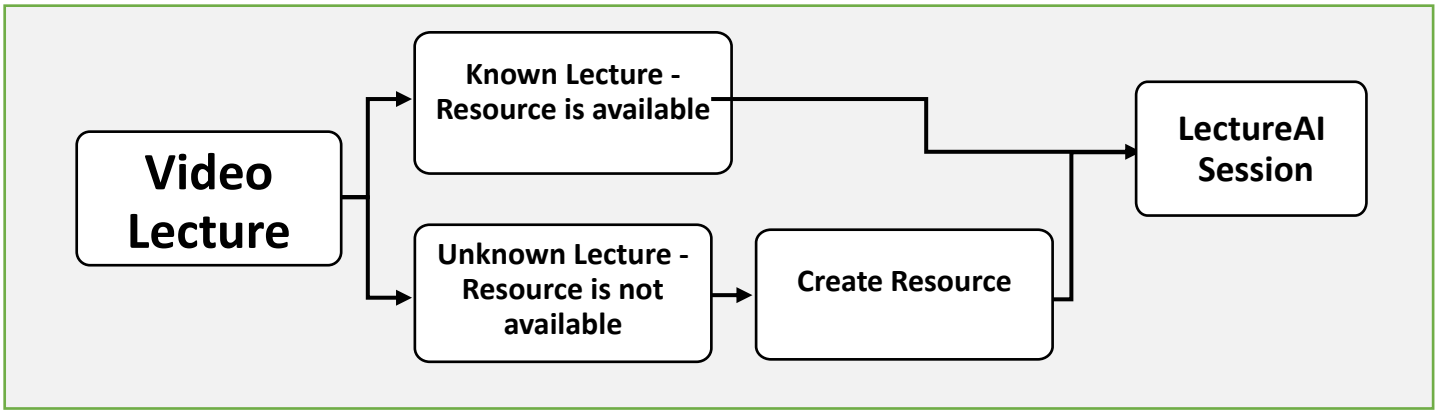
With LectureAI, users don't just watch videos — they ask questions, explore topics, get summaries, receive evaluations, and actively engage with the content. The system uses multiple intelligent agents to provide real-time assistance, making learning more effective and efficient.

Workflow of Agentic Lecture-AI App

To provide an intelligent and personalized learning experience, **LectureAI** follows a smart and adaptive workflow. This ensures that users can interact meaningfully with both familiar and new video lectures.

1. **Video Lecture Input**
   The user provides a video lecture as input to the system.
2. **Check for Existing Resources**
   The system checks whether learning resources (such as summaries, notes, or concept maps) for the lecture are already available.
   - If the **lecture is already known**, the existing resources are loaded instantly.
   - If the **lecture is new**, system generates the required resources automatically and saves in directory.

Fig. 01: WorkFlow of LectureAI



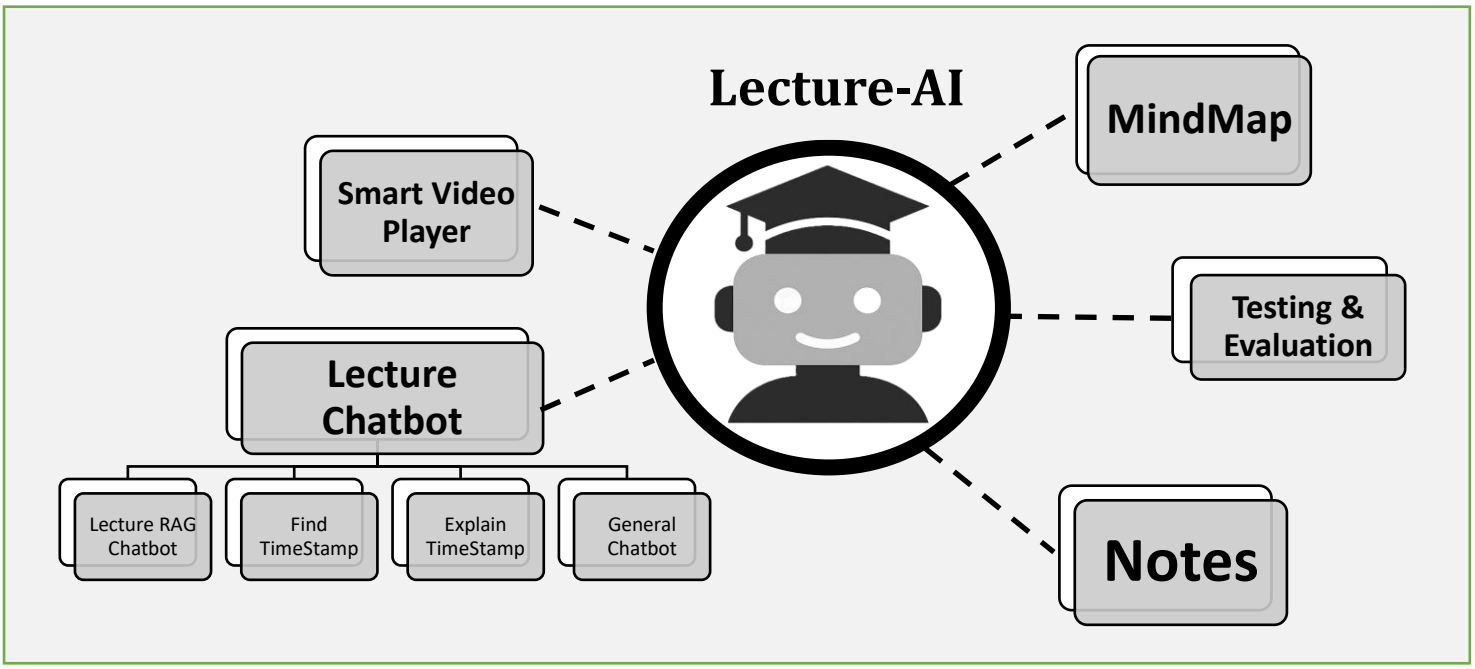
3. **Lecture-AI session**
   Session start, all data and dependencies including Vector-store, Mindmap, etc are loaded in App.

LectureAI Session

Once resources are ready—either fetched or generated—the user enters a **LectureAI session**, where they can interact with the video using four core features:

- **Chatbot** – Allows users to ask questions, get instant answers, and explore concepts from the video using natural language.
- **Mind Maps** – Visual representation of the video's structure and key concepts for quick understanding and navigation.
- **MCQ Testing & Evaluation** – Automatically generated quizzes to test comprehension, with instant feedback and scoring.
- **Notes** – Well-structured summaries including an introduction, key takeaways, conclusion, and further resources.

Fig. 01: LectureAI Session



Resource Creation from Video Lecture

The creation of data for LectureAI is a two-step process that transforms a linear video into a usable, interactive learning resource.
Step 01: Raw Data is Extracted from the Lecture Video.
Step 02: That Raw Data is Converted into a Usable Form.

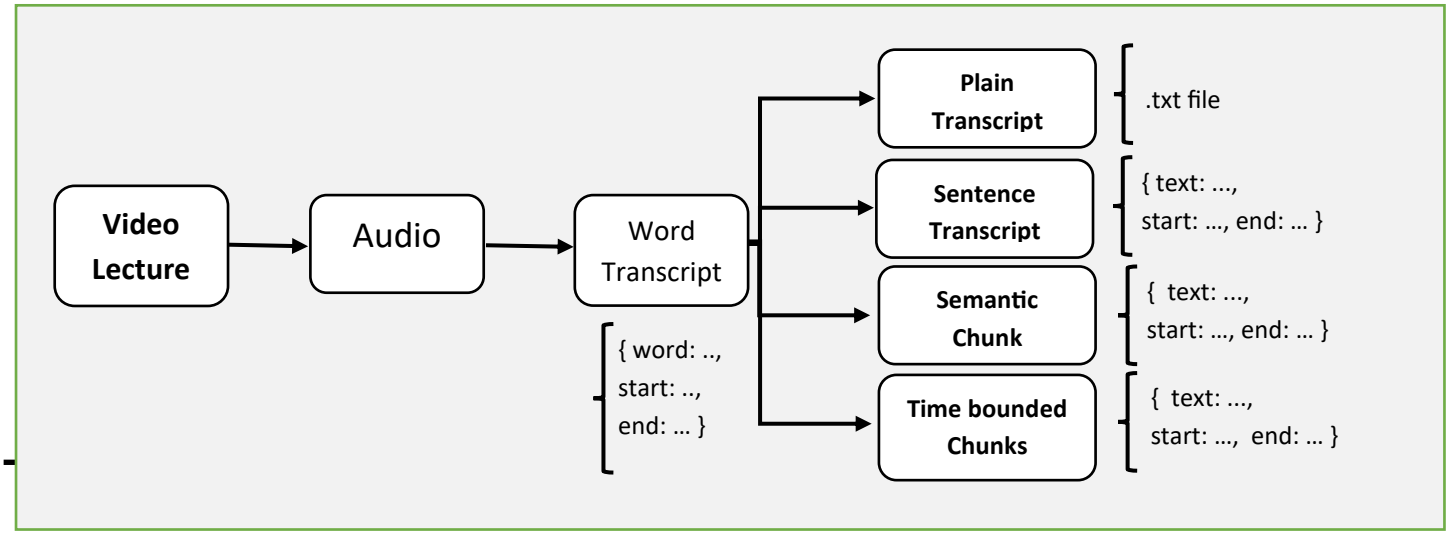
Step 01: Raw Data is Extracted from the Lecture Video.

The first step in our process is to extract raw data from a video lecture. This begins by using a Python library like “ffmpeg” to extract the audio. This audio is then processed by an API call to a model like OpenAI or Hugging Face's "Whisper" to generate a **Word Transcript**. This transcript is a list of dictionaries, with each dictionary containing a word and its corresponding start and end times. For large video files, the video is broken into smaller chunks to obtain the word transcript, and these chunks are then merged.

From the word transcript, several other types of raw data are created:

- **Plain Transcript**: This is a simple .txt file where the words from **Word Transcript** are joined together without any timestamps.
- **Sentence Transcript**: This is a list of dictionaries, where each dictionary contains a full sentence along with its start and end times. To create this, the plain transcript is partitioned into sentences using an LLM via an API call, and the sentences are then mapped back to the correct start and end timestamps.
- **Semantic Chunk Transcript**: This is also a list of dictionaries with text, start time, and end time. To create these chunks, we use an embedding model like "all-MiniLM-L6-v2" to group consecutive, semantically similar words together. The similarity between words is determined using cosine similarity.
- **Time-bounded Chunk**: This is a list of dictionaries containing text, start time, and end time. Unlike the semantic chunks, these are not based on meaning; they are created by merging consecutive sentences from the sentence transcript until a specific time limit is reached.

Fig. 02: Raw-Data extraction



Step 02: That Raw Data is Converted into a Usable Form.

After a video is uploaded, the raw transcript is processed to generate various learning resources for the user. This is done using both the **Plain Transcript** and a **Semantic Chunk Transcript**, which is a more structured and meaningful version of the original content.

1. LectureAI Chatbot (Vector Store Generation)
The **Semantic Chunk Transcript** is used to create a **vector store/database**, which powers the **LectureAI chatbot** for intelligent, context-aware question answering. We have used the embedding model **mixedbread-ai/mxbai-embed-large-v1**, which produces **1024-dimensional vectors**. These embeddings enable efficient semantic search and retrieval during chatbot interactions.

2. MCQ Generation and Evaluation

To generate **Multiple-Choice Questions (MCQs)**, we utilize both the **Plain Transcript** and the **Semantic Chunk Transcript IDs**. The generated MCQs are stored in JSON format with the following fields -question, options, topic, difficulty, explanation. An **AI agent** with a specialized prompt is used to create these MCQs and associated metadata.

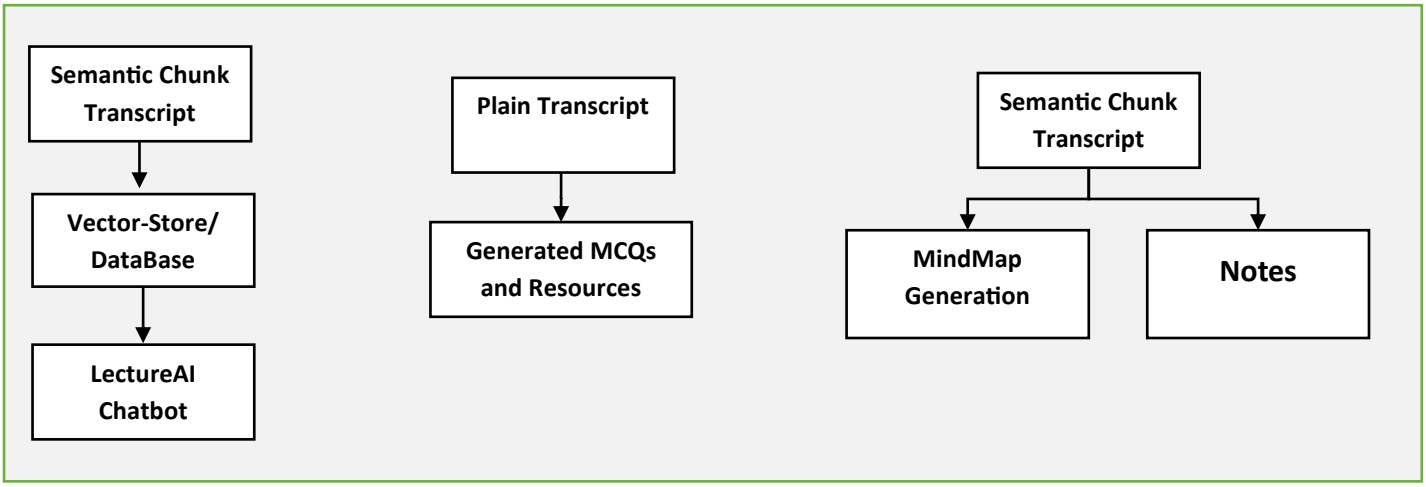
3. Notes Generation

The **Semantic Chunk Transcript** is also used to generate well-structured notes. An API call is made with a suitable prompt to create notes that include an **introduction**, **key takeaways**, **conclusion**, and **further resources**.

4. Mind Map Creation

The **mind map** is generated from the **Semantic Chunk Transcript** in a two-step process. In first step **Graphviz DOT Code** is Generated – A prompt is passed through an API to generate the DOT code representing the video's structure and concepts. In final step, we have done **Graph Rendering** – The DOT code is then converted into a visual mind map using a simple Python script and the Graphviz library.

Fig. 03: Converting Raw-data into usable form



RAG-Chatbot Tool

The LectureAI Chatbot is an agentic AI system designed to operate on lecture content. It uses APIs to process and query information remotely. The system has four agentic tools and a central tool that selects the appropriate agent for a given query. A validation component is also included, which uses an API-based LLM call to validate the response.

The four agentic tools are:

- **RAG_Chatbot Tool**: This tool uses Retrieval-Augmented Generation (RAG) with LLM API calls. It incorporates chat history to answer general lecture-related queries based on both relative and fixed context. Here, relative means context or query that was in previous chats. For example: “explain the first point?”, to answer this tool fetches history gets the context and then answer it.
- **Find_Timestamp_Tool**: This tool identifies where a specific context, whether fixed or relative, occurred in the lecture video.
- **Explain_Timestamp_Tool**: This tool is designed to explain concepts taught at specific timestamps. It provides answers grounded in the exact or relative moment of the lecture by combining RAG, LLM, and chat history.
- **General_Chatbot_Tool**: This tool answers broader academic or general queries. It uses parametric knowledge, web scraping, and the Tavily Search API to fetch accurate and current information.

These tools are controlled by a central agent called LectureAI-Agent, which uses an LLM to decide which tool to invoke. The Validation component ensures the final response is accurate by making an API-based LLM validation call.

Conclusion

LectureAI is an agentic AI solution that addressed the key drawbacks of traditional video-based learning by transforming passive video content into a personalized and interactive learning experience. The project tackled challenges like information overload, time-consuming searches, and limited interaction. LectureAI's workflow was built on a two-step process: first, raw data was extracted from a video lecture, and then this data was processed into usable resources like notes, mind maps, and MCQs.

The heart of the experience is **LectureAI session**, where users engage directly with the content. The project leveraged AI technologies like Retrieval-Augmented Generation (RAG) and LLM APIs to build this framework.

Ultimately, LectureAI solved the problem of linear and non-interactive video consumption by providing a comprehensive and dynamic learning environment. The framework converted lecture videos into interactive learning experiences, offering a personalized chatbot, mind maps, notes, and other materials to enhance engagement and effectiveness. This agentic AI tool is poised to revolutionize learning by helping learners save 30-50% of their time through personalized, automated academic support.

Future Works

We have several plans for the future to enhance the user experience. These include:

- **Real-Time Interaction**: Adding live captions and instant question-and-answer capabilities during video playback or meetings.
- **Multilingual Video Support**: The ability to understand and respond to videos in different languages.
- **Multimodal Input Support**: The capacity to process not just video and audio, but also text, PowerPoint presentations, and clipart.
- **Subject-Specific Fine-Tuned Models**: Creating specialized educational models tailored for different subject areas.
- **LMS Integration**: Ensuring seamless compatibility with popular learning management systems (LMS) such as Moodle and Canvas.