PROJECT

RAG BASED AI TEACHING ASSISTANT

Presented By:
1. Student Name- Ayush Kottalwar
2. College Name- Priyadarishini college of Engineering
3. Department- Electronics and telecommunication

OUTLINE

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Algorithm & Deployment
- Result
- Conclusion
- Future Scope
- References

PROBLEM STATEMENT

With the increasing use of online video-based learning platforms, a large amount of educational content is available in the form of video lectures. However, students often face difficulty in searching, revising, and extracting specific information from long video playlists. Watching entire videos repeatedly to find answers is time-consuming and inefficient, especially when a course consists of multiple videos in a playlist.

Although video platforms provide basic search and timestamps, they do not offer intelligent, question-answering capabilities based on the actual lecture content. Traditional AI assistants are unable to accurately answer questions because they do not have direct access to the spoken content of video lectures and may generate responses that are not aligned with the instructor's explanations.

To address this challenge, there is a need for an intelligent system that can convert video lectures into searchable knowledge and provide accurate, context-aware responses directly from the lecture material. In this project, MP4 video lectures from a playlist of 20 educational videos are converted into MP3 audio, transcribed into text, and then processed using text chunking and embedding techniques. A Retrieval-Augmented Generation (RAG)-based AI Teaching Assistant is developed to retrieve relevant information from these processed lecture transcripts and generate precise answers to student queries.

This approach enables students to interactively ask questions and receive responses grounded in the actual lecture content, reducing time spent on manual searching, improving comprehension, and enhancing personalized learning. The proposed system transforms passive video learning into an interactive, efficient, and intelligent educational experience.

PROPOSED SOLUTION

Video to Text: Convert lecture videos into text using speech-to-text transcription.

Text Chunking: Split transcripts into small, meaningful chunks for better context handling.

Text to Vector: Convert each text chunk into vector embeddings and store them in a vector database.

Query to Vector: Convert user questions into vector form using the same embedding model.

RAG Setup: Retrieve the most relevant text chunks based on vector similarity.

LLM Response: Generate accurate, context-aware answers using an LLM grounded in retrieved content.

SYSTEM APPROACH

- The system follows an intelligent and content-driven learning approach.
- Lecture videos from a playlist are converted into text using speech-to-text technology.
- The extracted text is processed and divided into smaller chunks for effective understanding.
- Each text chunk is converted into vector embeddings and stored in a vector database.
- Users interact with the system through an AI-based interface to ask questions.
- User queries are transformed into vector form for semantic comparison.
- The RAG framework retrieves the most relevant lecture content based on similarity search.
- A Large Language Model generates accurate, context-aware answers from retrieved data.
- The final responses are presented to users in a clear and interactive format.

ALGORITHM & DEPLOYMENT

ALGORITHM

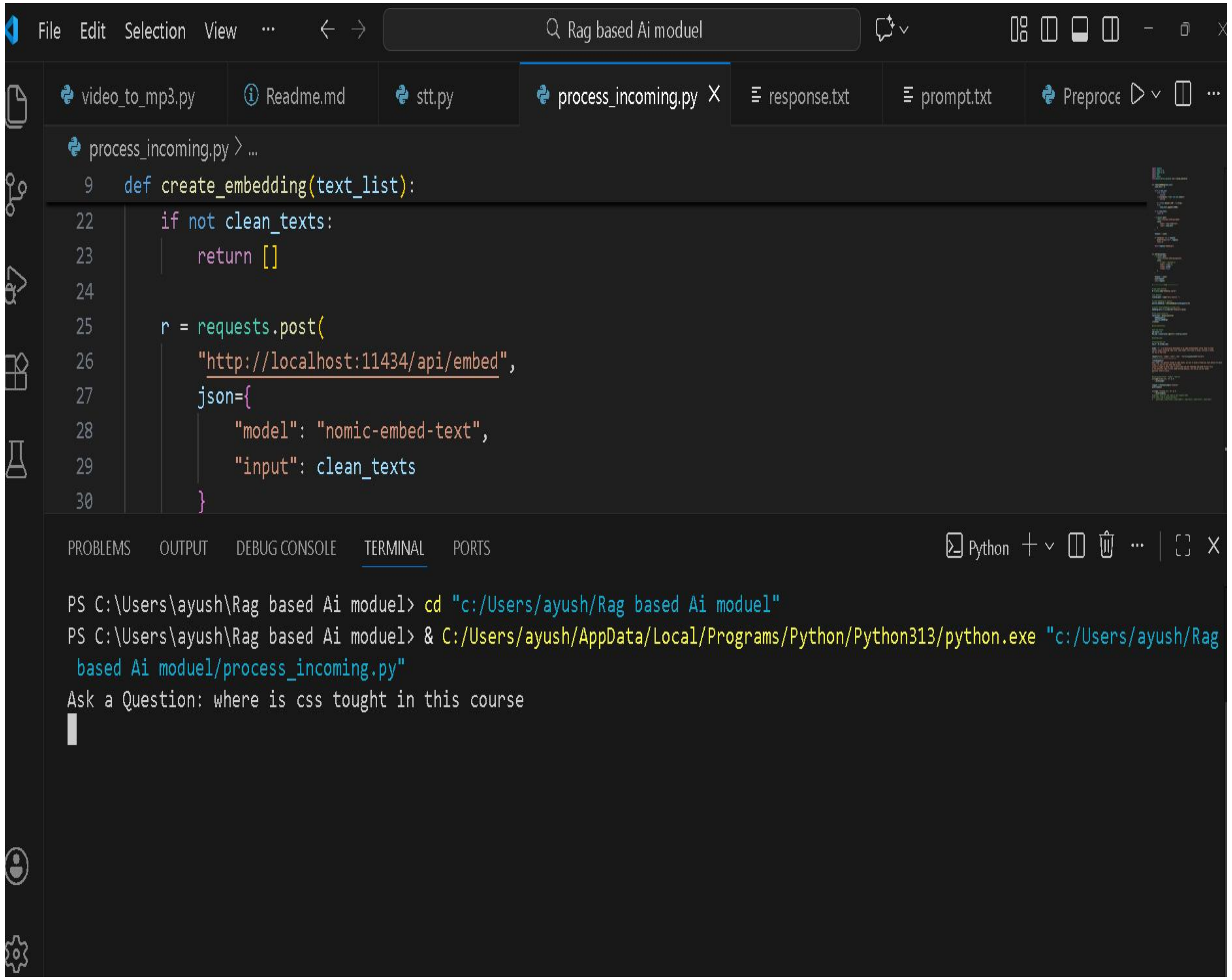
- Accept user learning queries.
- Convert learning videos into audio and then into text.
- Preprocess and split text into chunks.
- Generate vector embeddings for text chunks.
- Store embeddings in a vector database.
- •. Convert user query into embedding.
- Retrieve relevant content using RAG.
- Generate answer using LLM.
- Display response to the user.

DEPLOYMENT

DEPLOYMENT

- Deploy a web-based interface for user interaction.
- Host the backend server to manage the RAG pipeline.
- Deploy speech-to-text, embedding, vector database, and LLM services.
- Integrate the vector database for fast semantic retrieval.
- Deploy the system on cloud or on-premise infrastructure.
- Ensure secure access to learning content and user queries.
- Provide real-time, context-aware responses to users.

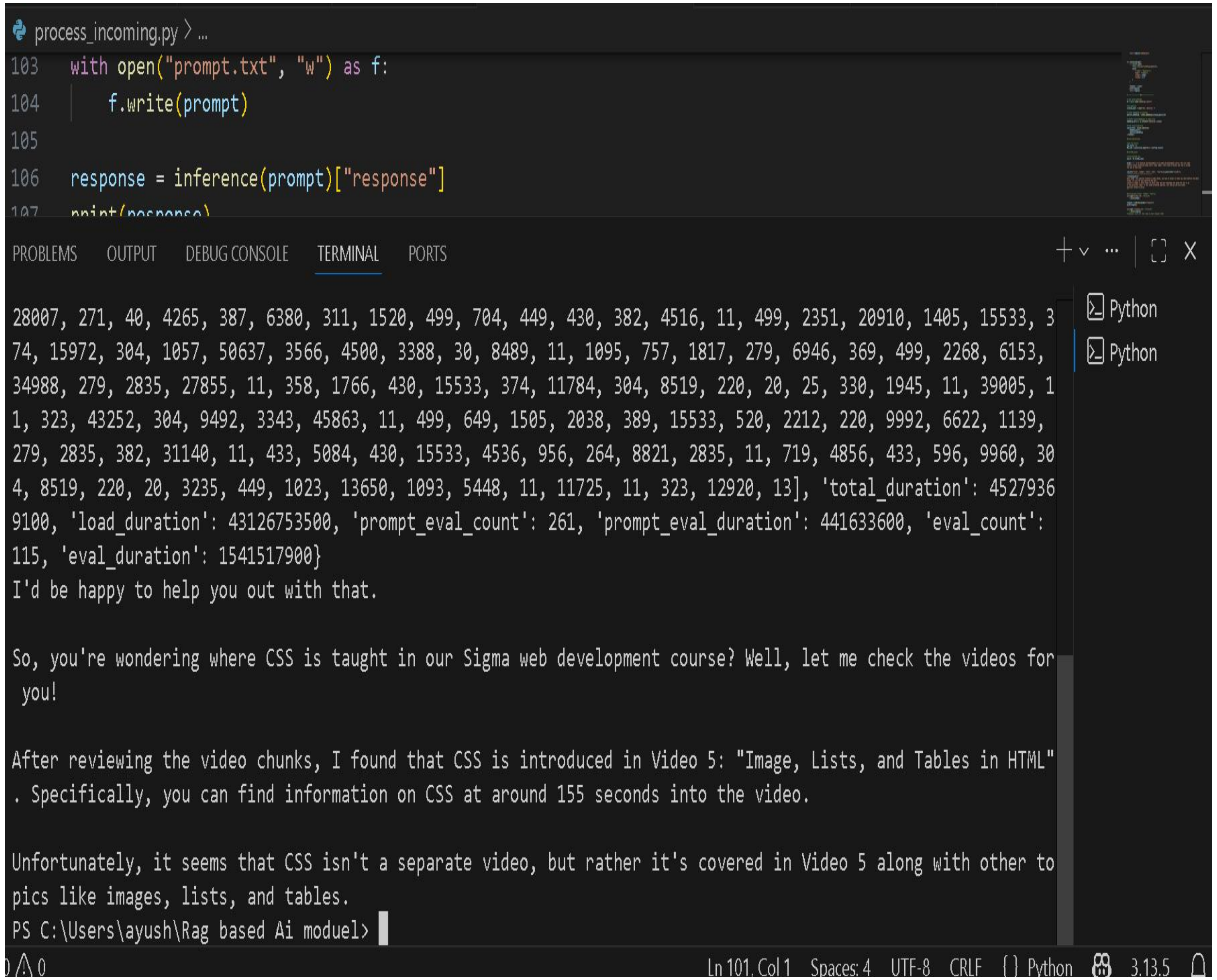
RESULT



```python
def create_embedding(text_list):
    if not clean_texts:
        return []

    r = requests.post(
        "http://localhost:11434/api/embed",
        json={
            "model": "nomic-embed-text",
            "input": clean_texts
        }
```

```
PS C:\Users\ayush\Rag based Ai moduel> cd "c:/Users/ayush/Rag based Ai moduel"
PS C:\Users\ayush\Rag based Ai moduel> & C:/Users/ayush/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/ayush/Rag
 based Ai moduel/process_incoming.py"
Ask a Question: where is css tought in this course
```



```python
with open("prompt.txt", "w") as f:
    f.write(prompt)

response = inference(prompt)["response"]
print(response)
```

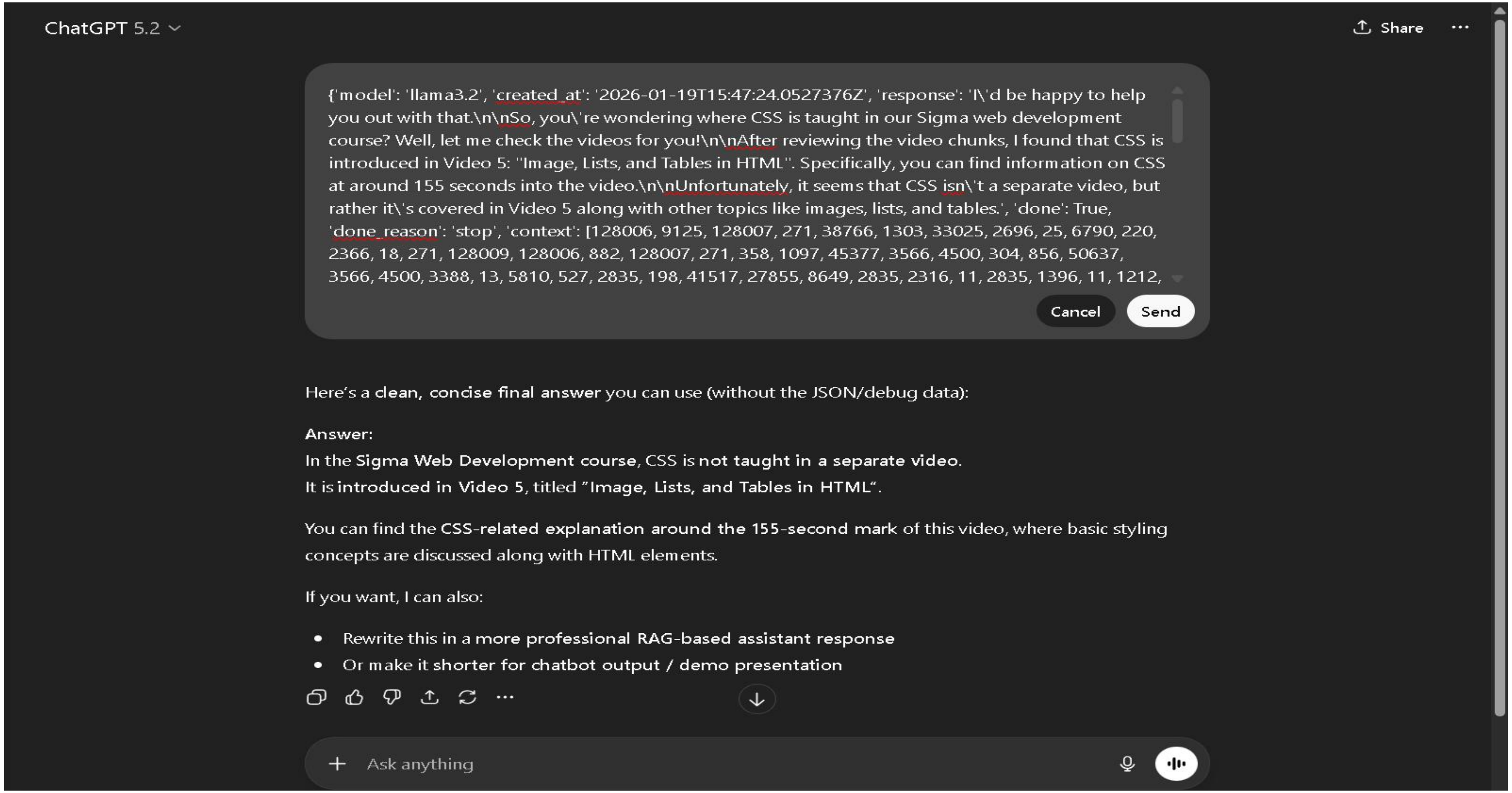
```
28007, 271, 40, 4265, 387, 6380, 311, 1520, 499, 704, 449, 430, 382, 4516, 11, 499, 2351, 20910, 1405, 15533, 3
74, 15972, 304, 1057, 50637, 3566, 4500, 3388, 30, 8489, 11, 1095, 757, 1817, 279, 6946, 369, 499, 2268, 6153,
34988, 279, 2835, 27855, 11, 358, 1766, 430, 15533, 374, 11784, 304, 8519, 220, 20, 25, 330, 1945, 11, 39005, 1
1, 323, 43252, 304, 9492, 3343, 45863, 11, 499, 649, 1505, 2038, 389, 15533, 520, 2212, 220, 9992, 6622, 1139,
279, 2835, 382, 31140, 11, 433, 5084, 430, 15533, 4536, 956, 264, 8821, 2835, 11, 719, 4856, 433, 596, 9960, 30
4, 8519, 220, 20, 3235, 449, 1023, 13650, 1093, 5448, 11, 11725, 11, 323, 12920, 13], 'total_duration': 4527936
9100, 'load_duration': 43126753500, 'prompt_eval_count': 261, 'prompt_eval_duration': 441633600, 'eval_count':
115, 'eval_duration': 1541517900}
I'd be happy to help you out with that.

So, you're wondering where CSS is taught in our Sigma web development course? Well, let me check the videos for
 you!

After reviewing the video chunks, I found that CSS is introduced in Video 5: "Image, Lists, and Tables in HTML"
. Specifically, you can find information on CSS at around 155 seconds into the video.

Unfortunately, it seems that CSS isn't a separate video, but rather it's covered in Video 5 along with other to
pics like images, lists, and tables.
PS C:\Users\ayush\Rag based Ai moduel>
```

edunet
foundation



```
{'model': 'llama3.2', 'created_at': '2026-01-19T15:47:24.0527376Z', 'response': 'I\'d be happy to help
you out with that.\n\nSo, you\'re wondering where CSS is taught in our Sigma web development
course? Well, let me check the videos for you!\n\nAfter reviewing the video chunks, I found that CSS is
introduced in Video 5: "Image, Lists, and Tables in HTML". Specifically, you can find information on CSS
at around 155 seconds into the video.\n\nUnfortunately, it seems that CSS isn\'t a separate video, but
rather it\'s covered in Video 5 along with other topics like images, lists, and tables.', 'done': True,
'done_reason': 'stop', 'context': [128006, 9125, 128007, 271, 38766, 1303, 33025, 2696, 25, 6790, 220,
2366, 18, 271, 128009, 128006, 882, 128007, 271, 358, 1097, 45377, 3566, 4500, 304, 856, 50637,
3566, 4500, 3388, 13, 5810, 527, 2835, 198, 41517, 27855, 8649, 2835, 2316, 11, 2835, 1396, 11, 1212,
```

Here's a **clean, concise final answer** you can use (without the JSON/debug data):

Answer:
In the **Sigma Web Development course**, CSS is **not taught in a separate video**.
It is **introduced in Video 5**, titled **"Image, Lists, and Tables in HTML"**.

You can find the **CSS-related explanation around the 155-second mark** of this video, where basic styling concepts are discussed along with HTML elements.

If you want, I can also:

- Rewrite this in a **more professional RAG-based assistant response**
- Or make it **shorter for chatbot output / demo presentation**

CONCLUSION

- The RAG-based AI teaching assistant effectively converts learning videos into structured knowledge using speech-to-text and vector embeddings. By combining semantic retrieval with a large language model, the system provides accurate, context-aware, and reliable answers to user queries. This approach reduces misinformation, improves learning efficiency, and offers a scalable solution for intelligent, video-based education support.

FUTURE SCOPE

- The system can be expanded to include more learning videos and subjects.
- Personalization can be added based on user learning patterns and performance.
- Multilingual support can be implemented for wider accessibility.
- Integration with quizzes, assessments, and LMS platforms can enhance learning.
- Advanced language models can be used to improve accuracy and response quality.

REFERENCES

1. Lewis, P. et al., Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, NeurIPS, 2020.

2. OpenAI, Embedding Models and Large Language Models Documentation.

3. Vaswani, A. et al., Attention Is All You Need, NeurIPS, 2017.

4. Radford, A. et al., Language Models are Few-Shot Learners, OpenAI, 2020.

5. Karpukhin, V. et al., Dense Passage Retrieval for Open-Domain Question Answering, EMNLP, 2020.

6. LangChain Documentation, Building RAG Applications.

7. FAISS Documentation, Efficient Similarity Search and Clustering of Dense Vectors.

GITHUB LINK

- Github Link :https://github.com/Ayush-ko/Rag-Based-AI-Teaching-Assistant.git

THANK YOU