TubeTalk

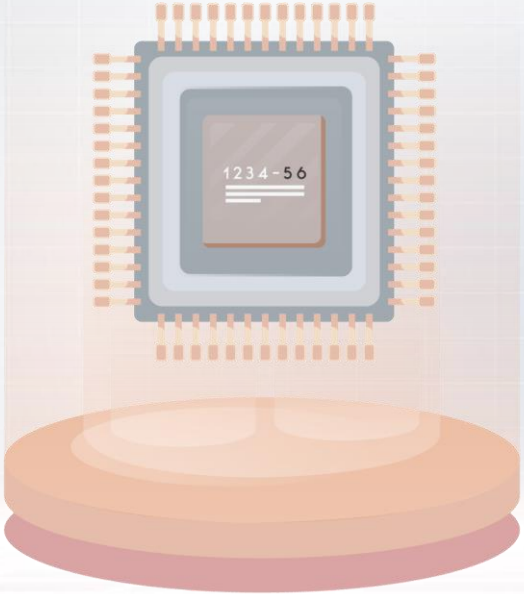
Ask Any YouTube Video

By:

- Elaf Alkhaldi
- Nada Alnemari

Course info: Saudi Digital Academy Cooperation with IronHack, AI Engineering Bootcamp

TABLE OF CONTENTS

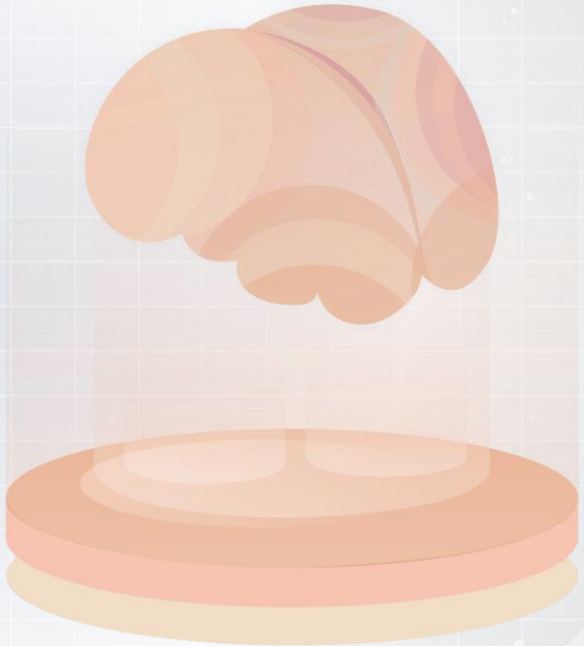


INTRODUCTION

INTRODUCTION

In today's digital age, **video content dominates** platforms like YouTube — but it's inherently **non-searchable** and **time-consuming to consume**.

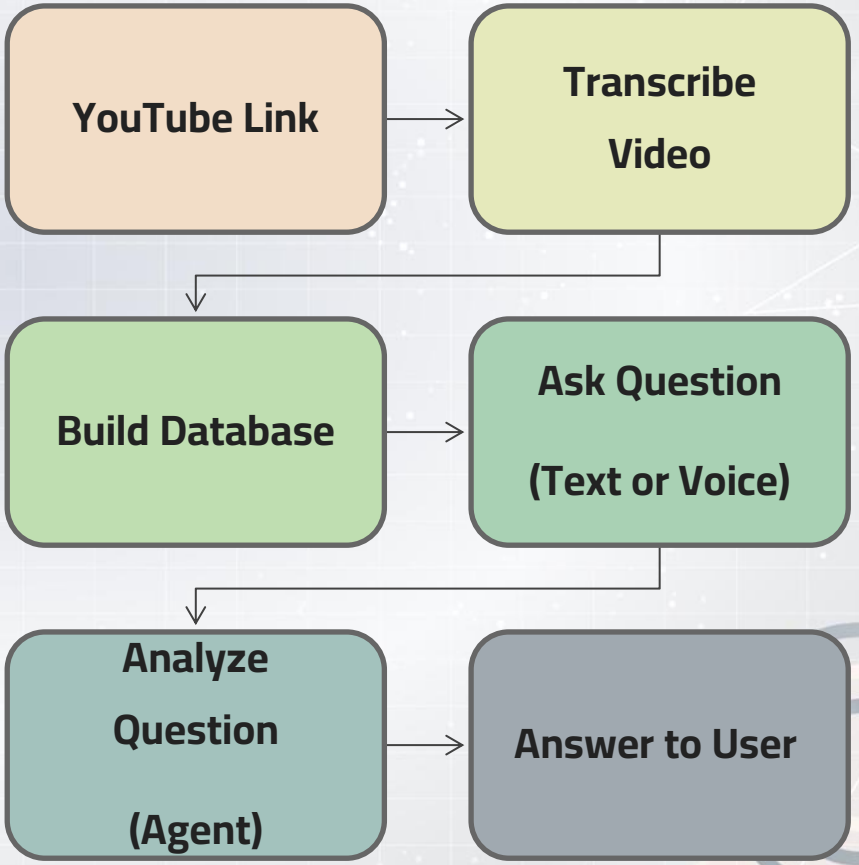
- Our project introduces an **AI-powered chatbot**, called **TubeTalk**, that transforms any YouTube video into an interactive experience.



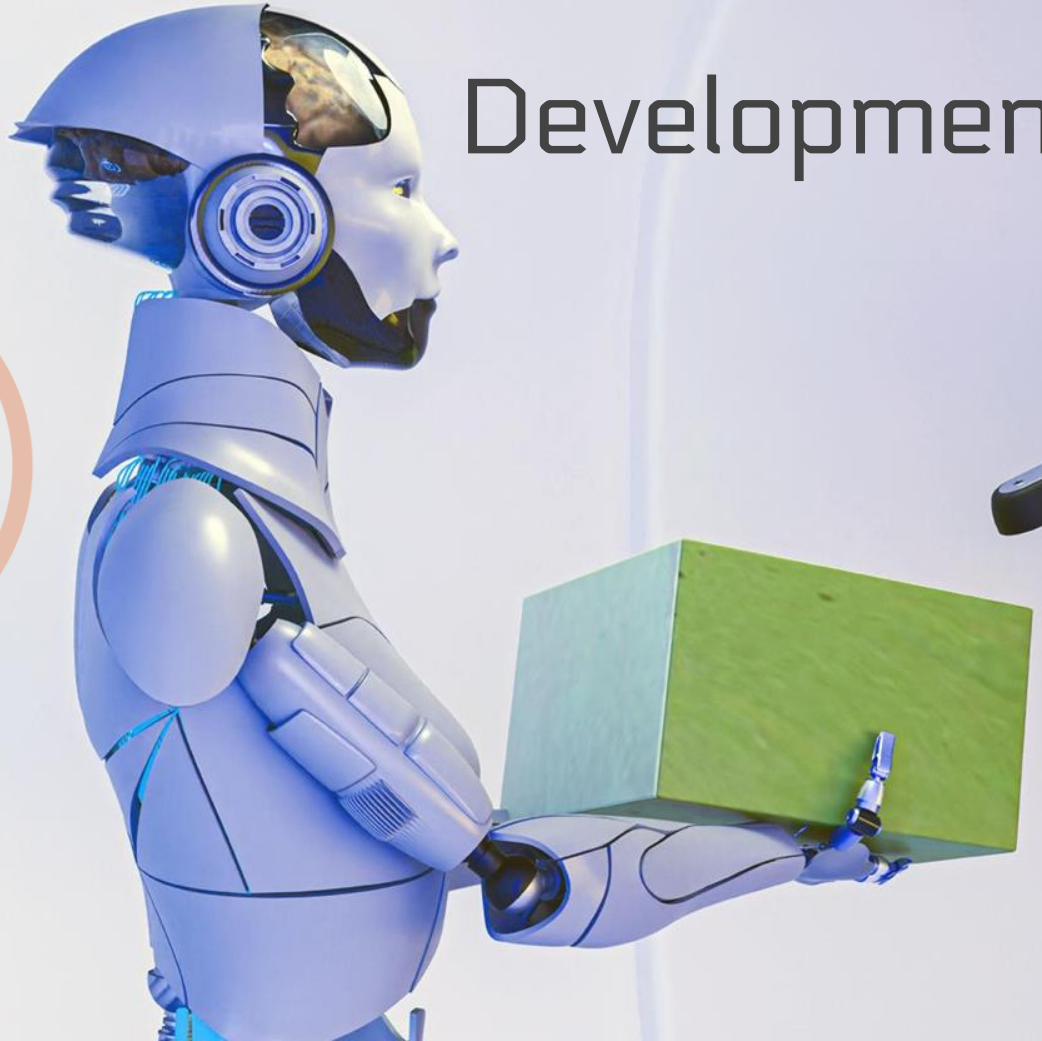
Motivation Behind the Project

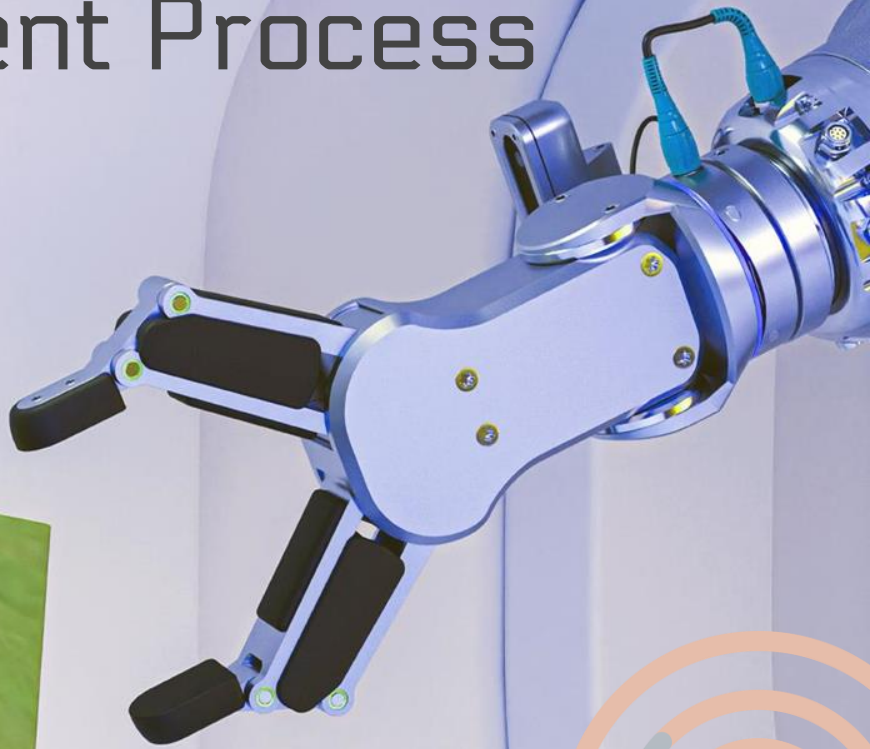
- People often want to get value from long YouTube videos **without** watching everything.
- There is no easy way to ask questions or get summaries directly from video content.
- **Our goal:** Build an AI chatbot that can **understand and summarize** YouTube videos through **natural language**.

Block Diagram :



Development Process





Development Process

- **Data Input:** User provides YouTube link or voice input.
- **Audio Processing:** Extract & transcribe audio (Whisper).
- **Text Chunking:** Tokenized using HuggingFace for vector DB efficiency
- **Vector Store:** Stored using Chroma for semantic search.
- **LangChain Agent:** Handles user queries with Search and Summarizer tools.
- **Chainlit UI:** Engaging frontend with support for voice or text queries.

Why Chainlit? It's a framework built for LLM-based apps, perfect for interactive agents and real-time chat interfaces.

Use Cases

Use Cases

- **Educational videos:** Quickly get a summary or ask concept-based questions.
- **Podcasts / Interviews:** Jump to key moments through question-answering.
- **Accessibility:** Voice interaction & transcription enhance usability..

Main Challenges

Challenge :	How We Overcame It :
❖ Whisper's latency	Used base model for speed + async threads
❖ Manual text chunking	Switched to HuggingFace tokenizer
❖ Conversional Interface	Used Chainlit instead of Gradio
❖ Audio input in Chainlit	Integrated soundfile + Whisper pipeline
❖ Agent errors	Smart response type handling & fallback messages

Results & Achievements :

- **GPT-4 agent** with Summarization & Retrieval capabilities.
- Accurate **transcriptions** using Whisper (multi-language).
- Semantic search over transcripts with **fast response time.**
- Supports both **text and voice** queries.
- Fully working **demo interface** using **Chainlit**.

Key Features

- **Multilingual** support
- **Voice and text** interaction
- Fast Q&A and summarization from transcript
- **GPT-4** based
- Real-time response via **Chainlit**

WHY Voice Input ?

- Easier for people who prefer to **speak instead of type**
- Useful for **mobile use** or users with accessibility needs
- Works in **real time** with accurate transcription

DEMO

"The app is directly deployable because it uses local Chainlit and can run on any machine that has Python."

Evaluation

To evaluate the **quality and behavior** of our AI agent, we used **LangSmith** to trace and analyze its responses based on three key criteria:

- **Accuracy**
- **Context Relevancy**
- **Hallucination**

- We tested the Chatbot on a short 3min Youtube video about Python language .

Test Case	User Question	Accuracy	Context Relevancy	Hallucination	Notes
Q1	What is Java?	☑ Rejected properly	Not relevancy	None	Bot correctly stated it's unrelated to the video.
Q2	Who invented Python?	☑ Historically correct	☑ Mentioned in transcript	None	"Guido van Rossum" and date were part of the video.
Q3	Is Python interpreted or compiled?	☑ Correct	☑ Covered in video	None	Bot's explanation matches transcript info on interpreted nature.
Q4	Can you summarize the video content?	☑ Relevant & concise	☑ Fully relevant	None	Summary accurately reflects points like ML, readability, popularity, job demand.

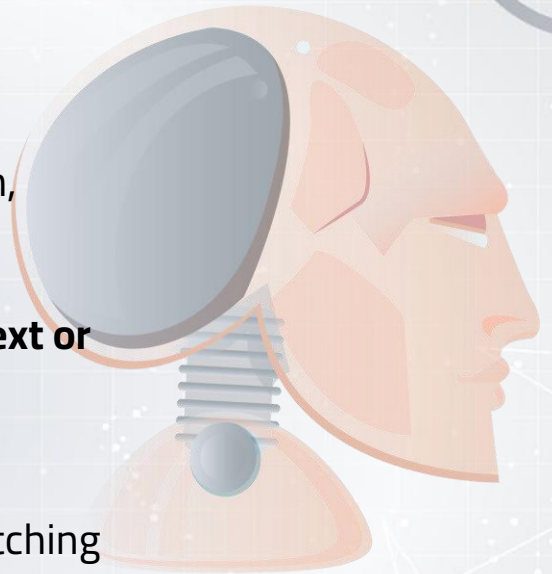
Future Work

What will happen in the future?

- Deploy as a **web app** or **browser extension.**
- **Long Video Handling:** Add support for multi-part video indexing and summarization.
- **Feedback Loop:** Let users rate answers to improve accuracy over time.
- **Mobile Compatibility:** Build a mobile app version with voice assistant integration.

Conclusion

- We built an AI chatbot that makes YouTube videos **searchable and interactive** using Whisper, LangChain, GPT-4, and Chainlit.
- Users can ask questions or get summaries through **text or voice**, making video content faster and easier to understand.
- This project shows how AI can turn passive video watching into an **active learning experience**.



THANKS!

Any Questions?

www.linkedin.com/in/elaf-alkhaldi

www.linkedin.com/in/nadaaln