Certificate

This is to certify that the mini-project entitled Tubetalk: An Intelligent LangChain System for Extracting and Analyzing YouTube Content for User Queries is a bonafide work of Jonathan Gomes(9900), Mark Lopes(9913), Vivian Ludrick(9914) submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering in Computer Engineering (Semester- V).

Prof Prajakta Dhamanskar

Guide/ Supervisor

Dr. Sujata Deshmukh
HOD, Computer Dept.

Dr. S. S. Rathod
Principal

Approval Sheet

Mini Project Report Approval for T.E. (Semester-V)

This mini-project report entitled Tubetalk: An Intelligent LangChain System for Extracting and Analyzing YouTube Content for User Queries submitted by Jonathan Gomes(9900), Mark Lopes(9913), Vivian Ludrick(9914) is approved for the degree of Bachelor of Engineering in Computer Engineering (Semester-V).

Examiner 1.———————————

Examiner 2. ———————————

Date:

Place: Mumbai

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission hasnot been taken when needed.

Jonathan Gomes(9900)

Mark Lopes(9913)

Vivian Ludrick(9914)

Abstract

This project introduces a web application that brings YouTube videos to life in a new way, making them more interactive and engaging for users. The platform is built for anyone who wants to dive deeper into video content by transforming audio into searchable text and allowing users to ask questions directly about the content. Using AssemblyAI, the app transcribes video audio, creating a text-based format that makes it easier for people to find answers within the video without having to watch it all.

A major strength of this tool is its use of FAISS (Facebook AI Similarity Search), which helps store and retrieve answers efficiently. When users ask questions about a video, the app finds the most relevant parts of the transcript and gives clear, context-driven answers. LangChain, a framework for natural language processing (NLP), powers this capability, making the interaction feel intuitive and natural.

But the experience goes beyond simple Q&A. The app also generates quizzes from the video's content, encouraging users to actively learn and retain what they've watched. This feature can be especially valuable in educational settings, where students and teachers can turn any video into a learning experience. Through quizzes and interactive questions, users can test their understanding and reinforce key points from the video.

The app was designed with ease of use in mind, from audio extraction and transcription to an intuitive interface and thorough documentation. The goal is to make it as accessible as possible, enabling users of all backgrounds to engage deeply with video content, gain insights, and learn more effectively. Whether for personal enrichment, classroom learning, or professional development, this application aims to make video content more accessible, interactive, and impactful.

Keywords:

- YouTube video analysis
- Transcript-based question answering
- AssemblyAI transcription
- Interactive quiz generation
- Natural language processing

Acknowledgments

We have great pleasure in presenting the report on “Tubetalk: An Intelligent LangChain System for Extracting and Analyzing YouTube Content for User Queries” I take this opportunity to express my sincere thanks to our guide, Proff. Prajakta Dhamanskar, C.R.C.E, Bandra (W), Mumbai, for providing the technical guidelines and suggestions regarding the direction of this work. We enjoyed discussing the work progress with her during our visits to the department.

We thank Dr. Sujata Deshmukh, Head of the Computer Engineering Department, along with the Principal and the management of C.R.C.E., Mumbai, for their encouragement and for providing the necessary infrastructure to pursue this project.

We also extend our gratitude to all non-teaching staff for their valuable support in completing our project.

Date: 24/10/24

Jonathan Gomes(9900)

Mark Lopes(9913)

Vivian Ludrick(9914)

Table of contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

The rising demand for video accessibility and knowledge extraction has driven innovation in tools for understanding video content. YouTube has a lot of information, but it takes sophisticated processing and interactive features to glean insights from videos.

The goal of this project is to create an online tool that lets people examine and engage with the transcripts of YouTube videos. The system uses natural language processing and vector-based storage to extract audio and provide precise transcriptions, which enable it to deliver insightful answers to user inquiries based on video footage. The platform also creates interactive tests that let users see how well they comprehend the subject matter.

The project's main objectives are to transcribe audio, store the transcripts in an effective vector format, create an interactive interface for easily accessible visualizations, and construct a question-answering-system.

This project offers a useful tool for researchers, students, and casual viewers by fusing cutting-edge NLP algorithms with user accessibility. This encourages deeper learning and more interesting ways to interact with video information.

Chapter 2

Review of Literature

2.1 Comparison table

Paper Title	**Algorithm**	**Database**	**Results**	**Summary (Advantage & Disadvantage)**
Dense Passage Retrieval (DPR) for Open-Domain Question Answering[1]	Dense Passage Retrieval (DPR)	Open-domain Q&A datasets	Significant improvements in retrieval-based QA tasks	**Advantage**: Improves retrieval of relevant text segments for Q&A. **Disadvantage**: Requires extensive training data and computational resources.
The Faiss Library[2]	FAISS (Facebook AI Similarity Search)	Large-scale datasets	Efficient similarity search in high-dimensional spaces	**Advantage**: Extremely fast and efficient for similarity search in large datasets. **Disadvantage**: Requires high computational power for very large datasets.
Sentence Meaning Similarity Detector Using FAISS[3]	FAISS with Sentence Embeddings	Sentence-level embeddings	Enhanced semantic similarity detection	**Advantage**: Uses FAISS for semantic search, crucial for efficient transcript search. **Disadvantage**: Requires accurate embeddings for effectiveness.
YouTube Transcript Synthesis[4]	Deep Learning for Transcript Synthesis	YouTube videos, transcription data	Improved quality of video transcript synthesis	**Advantage**: Provides high-quality transcript generation. **Disadvantage**: Computationally expensive and time-consuming for large video datasets.
Video Summarization using Speech	Speech recognition + text summarization	Video and audio content	Improved video summarizatio	**Advantage**: Combines speech recognition and text

Recognition and Text Summarization[5]	techniques		n	summarization for concise summaries. **Disadvantage**: May miss context or nuances in longer videos.
Embedding-based Retrieval in Facebook Search[6]	Embedding-based retrieval	Facebook user-generated content	Enhanced relevance and retrieval accuracy	**Advantage**: Embedding techniques enhance search performance. **Disadvantage**: Performance highly depends on the quality of embeddings used.
Word Embeddings: A Survey[7]	Word embeddings (Word2Vec, GloVe)	Text corpora	Overview of word embedding techniques and their applications	**Advantage**: Provides foundational knowledge for vectorizing text, crucial for transcript analysis. **Disadvantage**: Word embeddings might not capture all nuances, especially for specialized video content.
Understanding Human Preferences: Towards More Personalized Video to Text Generation[8]	Deep learning for personalized video-to-text generation	Video datasets, user interaction data	Personalizatio n of video transcript generation	**Advantage**: Improves personalization of transcript generation based on user preferences. **Disadvantage**: Requires detailed user data for customization.
VX2TEXT: End-to-End Learning of Video-Based Text Generation From Multimodal Inputs[9]	End-to-end deep learning model for video-based text generation	Video datasets with multimodal inputs	High-quality text generation from video data	**Advantage**: Generates high-quality transcripts using multimodal data. **Disadvantage**: Requires complex model architecture and computational resources.
Language as the Medium: Multimodal Video Classification through Text Only[10]	Text-based multimodal classification	Video and text data	Classification using text-only modalities	**Advantage**: Allows for text-based classification without using video data directly. **Disadvantage**: May not capture the full range of video

				information.
An Effective Query System Using LLMs and LangChain[11]	Large Language Models (LLMs) with LangChain	Text-based datasets	Improved query processing and response accuracy	**Advantage**: Enhances Q&A functionality using LangChain. **Disadvantage**: Requires high-quality datasets and model fine-tuning.
Automating Customer Service using LangChain[12]	LangChain for customer service automation	Customer service datasets	Automated customer service tasks	**Advantage**: Automation of service tasks. **Disadvantage**: May not handle highly complex or nuanced queries effectively.

2.2 Problems in existing system

- Reliance on Pre-existing Captions: The applicability of current solutions is limited for videos without captions because they only function with videos that already have captions.
- Absence of Advanced AI Integration: Most existing systems do not have sophisticated AI models for creating dynamic quizzes from video transcripts or answering questions based on context.
- Inaccuracy in Transcription: Tools that depend on captions frequently have flaws, particularly when the captions are manually or automatically generated, making the transcripts unfit for analysis.
- Limited Interactivity: Users of current tools are only given static transcriptions without individualized learning experiences like Q&A or quizzes, and there is no real-time interactivity with video content.

Chapter 3
3.1 PROBLEM DEFINITION

The goal of this project is to provide an interactive online application for YouTube video transcript analysis. YouTube videos' audio will be downloaded by the system, which will then use AssemblyAI to transcribe the content and save the finished transcript for later use. Through a Question & Answer (Q&A) system, viewers will be able to interact with video material and search the transcript for pertinent responses. In order to improve users' learning and engagement with the material, the system will also create interactive multiple-choice tests using the transcript of the video. The objective is to provide a strong and intuitive platform that facilitates knowledge retrieval and educational reinforcement using clever AI-powered tools such as Google Generative AI and LangChain.

3.2 OBJECTIVES OF THE PROJECT

1. Create a Transcript Processing System: Put in place a reliable system that records YouTube audio, uses AssemblyAI to transcribe it, and stores the transcript for later study.
2. Turn on Interactive Q&A Functionality: Make a vector store of video transcripts that can be searched so that users may pose queries and get pertinent, context-based responses.
3. Create Interactive Quizzes: To assist users in solidifying their comprehension of the subject matter, create multiple-choice tests automatically using video transcripts.
4. Promote Knowledge Retrieval from Videos: Make it simpler for users to find specific information in lengthy video transcripts by leveraging Google Generative AI and LangChain's capabilities for intelligent information retrieval.

3.3 SCOPE OF THE PROJECT

1. In order to provide accurate and easily accessible transcripts for analysis, this project will construct a system to manage the processing, transcription, and storage of YouTube video audio.

2. A robust Q&A interface is developed where users can ask questions and receive targeted answers based on the video's transcript, enhancing learning and content understanding.

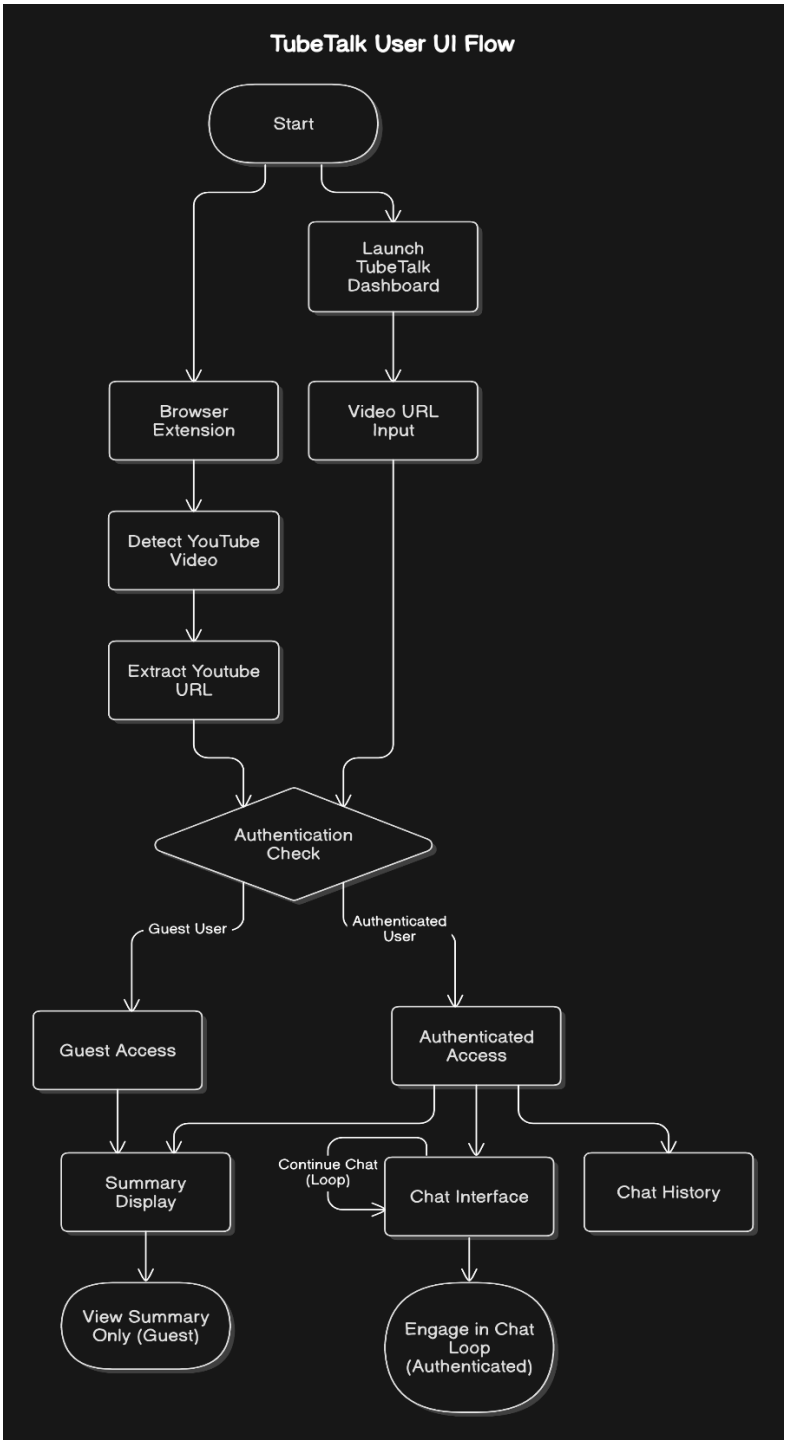
3. The system will automatically generate quizzes from transcripts, providing users with an engaging way to review key concepts, reinforcing retention through interactive multiple-choice questions.

4. By integrating LangChain and Google Generative AI, the project will support advanced information retrieval methods, allowing users to efficiently search for and extract relevant details from lengthy transcripts.

Chapter 4

SYSTEM DESIGN

4.1 BLOCK DIAGRAM



4.2 MODULE DESCRIPTION

1. User Start: The user initiates the process by accessing the application via a Chrome Extension or Web Application interface.

2. Register or Sign In: The user is required to either register a new account or sign in with existing credentials through the Authentication Deck.
- Register: The user creates a new account in the system.
- Sign In: The user logs in using their credentials

3. Authentication: The system checks if the login or registration is successful.

4. Content Interaction and Querying:
- Upload Video: The user uploads a YouTube video or provides a link, initiating transcript extraction.
- Summary Display: The system presents a summary of the video transcript, generated through backend services.
- Chat Portal: Users can ask questions related to the video transcript via a chat interface, which interacts with the Chat Response Generator to provide relevant responses.

5. Transcript Service:
- YouTube Transcription: The system fetches and parses video transcripts via a Transcript Engine.
- Audio Transcription: For videos without transcripts, the system performs audio processing, converting speech to text using ASR (Automatic Speech Recognition) to create a transcript.

6. Backend Processing:
- Processors (LLM Service/Gemini API): The system utilizes a Large Language Model to perform tasks such as:
- Chat Response Generation: Generate responses to user queries based on the video transcript.
- Summary Generation: Create concise summaries of the video content.

7. Database and Caching (PocketBase Database):
- Collections: Stores various data types, including:
- Transcripts: Parsed and processed video transcripts.
- Video Summaries: Summaries generated from transcripts.
- Chat History: Records of conversations between users on the chat platform.

8. Result Display and Feedback:
- Summary Display and Chat Interaction: Users can ask inquiries and examine created summaries using the chat feature
- Major Providers: Details about pertinent video or content suppliers.

9. End: The process concludes.

4.2.1 ALGORITHMS

4.2.1.1 THEORETICAL ANALYSIS

- Video Transcription: AssemblyAI is used in this project to transcribe the audio after it has been extracted from a YouTube video using yt_dlp. Spoken language is transformed into text through the transcription process, which forms the foundation for further research. Users can search for certain material in the video using this text data without having to view it.

- Text Segmentation and Embedding: The RecursiveCharacterTextSplitter is used to break up the transcribed text into digestible pieces, and Google Generative AI's embedding model is used to embed each chunk into a vector representation. This method enables efficient search based on user queries by facilitating efficient storing and retrieval in FAISS (Facebook AI Similarity Search). Long transcriptions can be handled while maintaining contextual coherence thanks to the chunking and embedding procedure.

- Question Answering (Q&A) System: To respond to user inquiries based on the embedded and transcribed video footage, a RetrievalQA Chain is utilized. The system uses Google Generative AI to create an answer after retrieving the most pertinent text passages from FAISS in response to a user's inquiry. Vector similarity is used in both the retrieval and answer generation processes to guarantee that the responses are contextually appropriate for the user's query.

- Quiz Generation: Using prompt-based quiz creation, the code generates multiple-choice questions based on the transcription content. This functionality allows users to test their comprehension of the video material in an interactive way. The quiz generation process generates questions, answer choices, and correct answers automatically, though it depends on the model's interpretation of the transcript content, which may introduce errors.

- User Interface with Streamlit: The code creates multiple-choice questions based on the transcription material using prompt-based quiz design. With the help of this feature, users can interactively assess how well they understand the video content. Although it depends on how the model interprets the content of the transcript, the quiz production process automatically creates questions, answer choices, and right answers.

4.2.1.2 ALGORITHMS USED

Audio Extraction Algorithm: This algorithm uses yt_dlp to extract audio from a YouTube video URL. It downloads the audio, which is then used for transcription.
Steps:

1. Input the YouTube video URL.
2. Use yt_dlp to extract the audio in the best available quality.
3. Store the audio file in a specified directory for further processing.

Panel Transcription Algorithm: This algorithm leverages the AssemblyAI API to transcribe the audio into text.
Steps:

1. Input the audio file obtained from the YouTube video.
2. Send the audio to AssemblyAI's transcription service, which uses machine learning to convert audio to text.
3. Receive and store the text transcript in a local directory.

Text Embedding and Retrieval technique: This technique employs the FAISS library for effective similarity search and Google Generative AI for embedding. By embedding text fragments and storing them in a vector database for retrieval, it makes question-answering possible.
Steps:

1.To preserve context and maximize retrieval, divide the transcript into smaller text segments using the RecursiveCharacterTextSplitter.
2. Make use of Google Generative AI embeddings to transform each chunk into a high-dimensional vector.
3. To enable effective similarity search, store the vectors in an FAISS index.
4. Using FAISS based on vector similarity, extract the most pertinent pieces from an input query.

The Question-Answering Algorithm pulls pertinent text passages from the vector store and feeds them into a language model to produce answers to user questions.
Steps:

1. Input the user's question.
2. Retrieve the most relevant transcript chunks from the FAISS index using similarity search.
3. Use Google Generative AI to generate an answer based on the retrieved context.

4.2.2 UML DIAGRAM(S)

4.2.3 UI DESIGN

Several static HTML, CSS, and JavaScript web pages make up the user interface (UI), which aims to provide a simple and interesting user experience.

1. Home Page
    - This is the primary application hub that informs users of the system's functionality and goal.
    - Draws attention to features including user engagement tools, chat-based communication, and video summary.
    - Makes registration and login choices easily accessible..
    - Provides links to explore particular features, such as "Chat with the AI" and "Summarize a Video."
2. Video Summary Page
    - Allows users to input a YouTube video link and retrieve a summary.
    - Displays essential metadata such as video title, upload date, and platform name.
    - Shows the generated summary along with options to refresh or edit.
    - Users can also browse a history of previous summaries, allowing quick access to past content.
3. Chat Page
    - Facilitates a conversational interface where users can interact with the AI.
    - Chat interface displays user messages and AI responses, providing context-based conversations.
    - Features a "Fetch Context" button to retrieve relevant past messages or summaries for continuity.
    - Users can view their conversation history and switch between different conversations if needed.

4. User Profile Page
    - Allows users to manage their profile details, such as username, email, and visibility preferences.
    - Includes options to update the avatar and control visibility settings.
    - Displays a summary of their activity (e.g., videos summarized, conversations started) to enhance engagement.
5. Register / Login Page
    - Provides a registration and login interface for user authentication.
    - New users can create an account, while existing users can sign in.
    - Includes password reset and token verification options for secure access.
6. Summary History Page
    - Displays a list of previously generated video summaries for easy reference.
    - Users can click on any summary to view details, edit, or delete.
    - Helps users keep track of their content analysis activities and revisit important summaries.
7. Service Explanation Page
    - Provides an overview of the system's capabilities, such as video summarization, chat functionality, and user profile management.
    - Explains the purpose and potential applications of the system to give users a clear understanding of its benefits.
    - Contains a "Learn More" section to dive deeper into the technology behind video summarization and AI chat interactions.
8. About Us Page
    - Shares information about the team behind the project, the mission, and the vision for the YouTube content analysis system.
    - Builds trust and credibility, inviting users to understand the project’s background and goals.
9. Contact Us Form
    - Provides a form for users to submit inquiries, feedback, or requests for support.
    - Includes fields for name, email, subject, and message, ensuring clear communication.

Chapter 5

IMPLEMENTATION AND RESULTS

5.1 Implementation

The implementation begins by accepting a YouTube URL input from the user. The URL is processed to download the video's audio using the yt-dlp library. The extracted audio is then sent to the AssemblyAI API, which transcribes the speech into text.

The RecursiveCharacterTextSplitter is used to split the text into digestible portions and preserve the transcript locally. For quick similarity-based search, each chunk is saved in an FAISS index and embedded using GoogleGenerativeAIEmbeddings.

Users can enter particular queries once the transcript has been processed and saved. The system uses GoogleGenerativeAI to obtain answers after searching the vector storage for pertinent areas. A quiz with multiple-choice questions based on the transcript is also generated by the system to assess the user's comprehension.

Because Streamlit is used in its development, users can browse the transcript, pose questions, and take quizzes through an easy-to-use interface. This approach provides a smooth and highly engaging experience for studying YouTube video content.

5.2 RESULTS

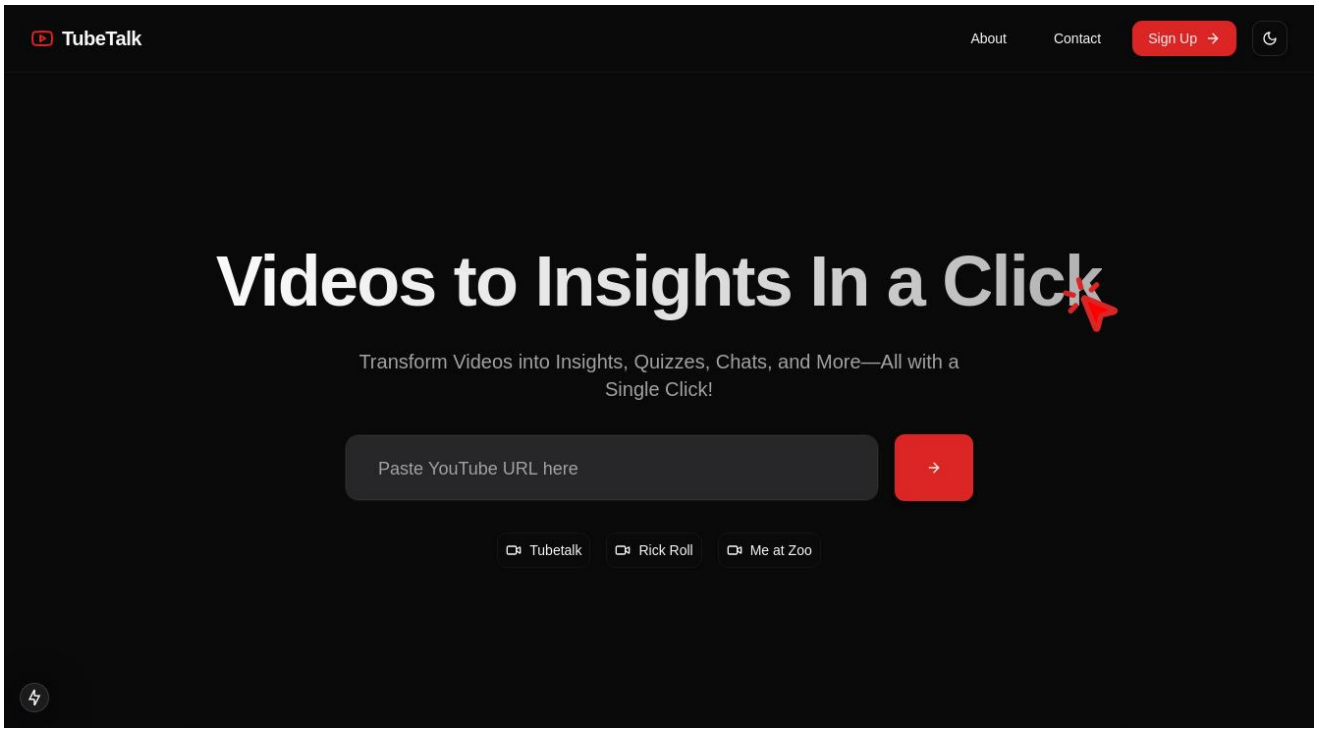


Fig 1: Home Page

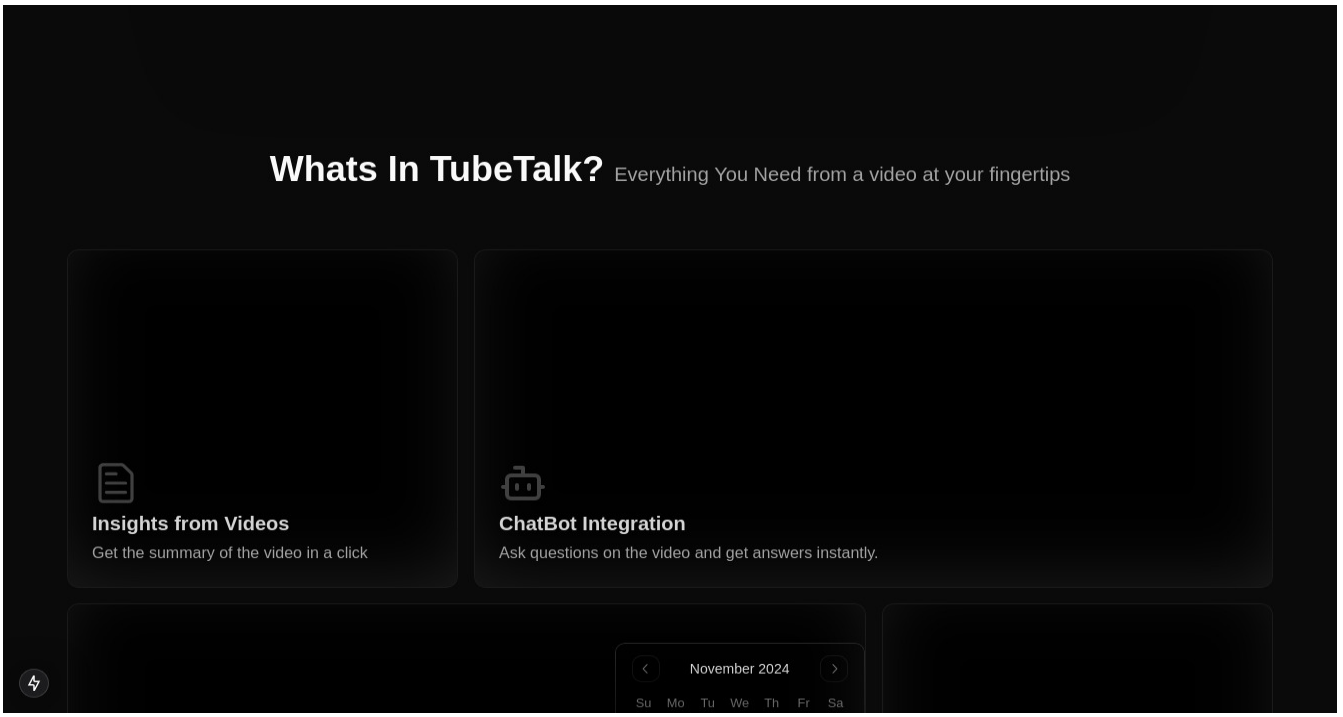


Fig 2: Features page

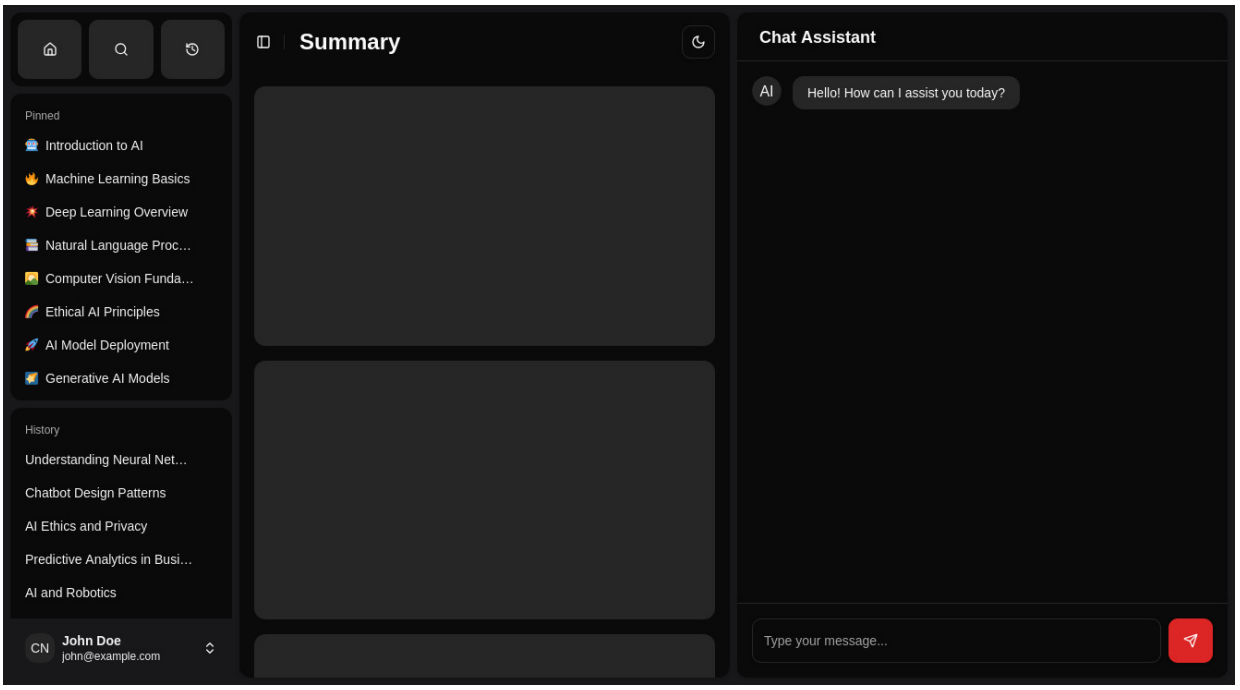


Fig 3: Chatbot and summary page

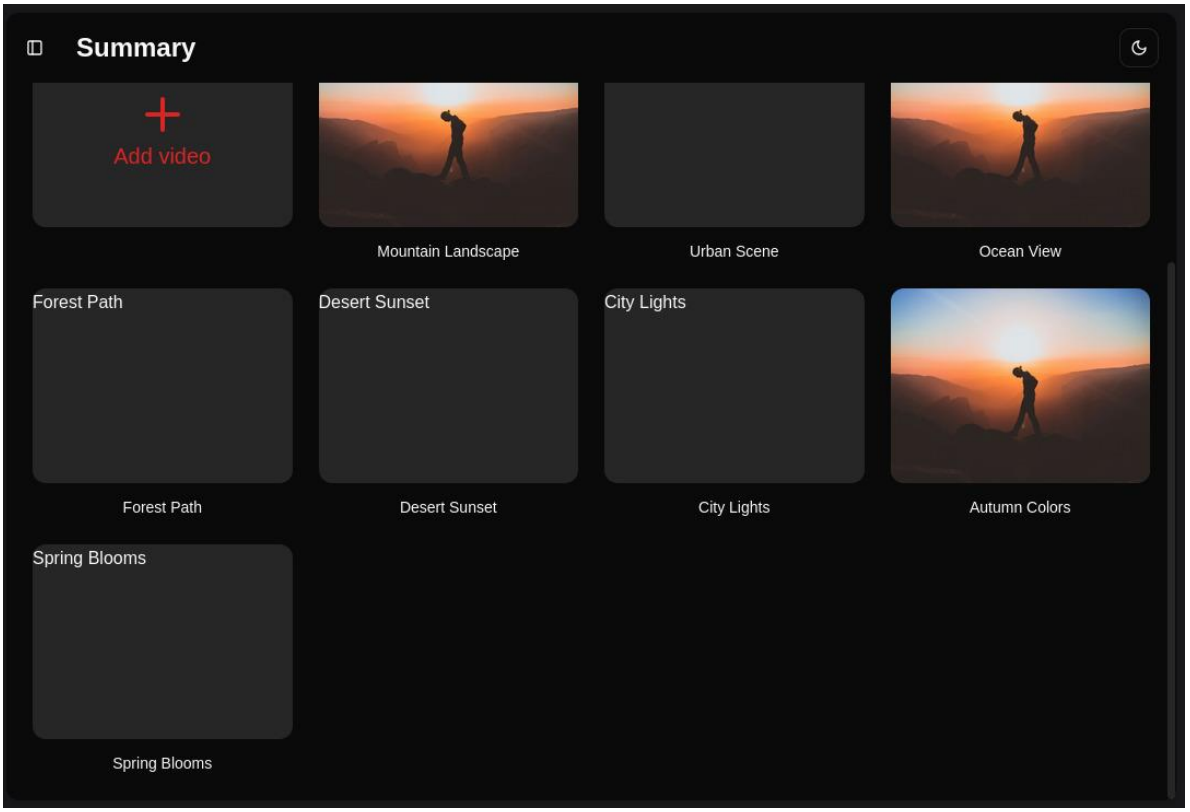


Fig 4: History page

Chapter 6

Conclusion and Future Work

6.1 Conclusion

To sum up, the YouTube Video Analyzer app offers a sophisticated, approachable platform for downloading, examining, and engaging with YouTube video material. The solution gives users an effective approach to find pertinent information and test their knowledge of video footage by utilizing cutting-edge transcription services like AssemblyAI and strong vector search capabilities with FAISS and Google Generative AI. By automating transcription, information extraction, and quiz creation, this project streamlines video analysis and enables users to learn from videos without the need for technical know-how. The application encourages deeper learning and a more interesting method to engage with video-based knowledge thanks to its interactive design.

6.2 Future Work

Personalized User Profiles: Present user profiles, which allow users to store quizzes, monitor their progress, and examine their scores.

Live Stream Transcription in Real Time: Create the ability to transcribe YouTube videos that are being streamed live so that analysis and Q&A can take place in real time.

Integration of Multilingual Transcription**:** Extend the application's functionality to support multilingual video transcriptions, making it accessible to a broader global audience.

REFERENCES

1. Vladimir Karpukhin, Barlas Oğuz, Sewon Min,"Dense Passage Retrieval (DPR) for Open-Domain Question Answering," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2004.04906

2. Matthijs Douze, Alexandr Guzhva, Chengqi Deng, "The Faiss Library," *arXiv*, 2024. [Online]. Available: https://arxiv.org/abs/2401.08281.

3. "Premanand P. Ghadekar; Sahil Mohite; Omkar More., "Sentence Meaning Similarity Detector Using FAISS," *IEEE Xplore*, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/10392009.

4. "Ankur Kumar; Priya Yadav; N. Partheeban., "YouTube Transcript Synthesis," *IEEE Xplore*, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/10541713.

5. Tirath Tyagi; Lakshaya Dhari; Yash Nigam., "Video Summarization using Speech Recognition and Text Summarization," *IEEE Xplore*, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10169901.

6. "Jui-Ting Huang, Ashish Sharma, Shuying Sun., "Embedding-based Retrieval in Facebook Search," *ACM Digital Library*, 2020. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3394486.3403305.

7. Felipe Almeida, Geraldo Xexéo, "Word Embeddings: A Survey," *arXiv*, 2019. [Online]. Available:https://arxiv.org/abs/1901.09069. Yihan Wu, Ruihua Song, Xu Chen, "Understanding Human Preferences: Towards More Personalized Video to Text Generation," *ACM Digital Library*, 2023. [Online]. Available: https://dl.acm.org/doi/10.1145/3589334.3645711.

8. Xudong Lin; Gedas Bertasius; Jue Wang., "VX2TEXT: End-to-End Learning of Video-Based Text Generation From Multimodal Inputs," *IEEE Xplore*, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9578716.

9. Laura Hanu, Anita L. Verő, James Thewlis., "Language as the Medium: Multimodal Video Classification through Text Only," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2309.10783.

10. **Adith Sreeram a**, **Jithendra Sai**;"An Effective Query System Using LLMs and LangChain," *ResearchGate*, 2023. [Online]. Available: https://www.researchgate.net/publication/372529063_An_Effective_Query_System_Using_LLMs_and_LangChain.

11. Keivalya Pandya, Mehfuza Holia, "Automating Customer Service using LangChain," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2310.05421