

**VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF  
TECHNOLOGY**

(An Autonomous Institute Affiliated to University of Mumbai)

**Department of Computer Engineering**



Project Report on

**“ContentConcise: YouTube content  
summarization and comment analysis”**

In partial fulfillment of the Fourth Year (Semester-VII), Bachelor of  
Engineering

(B.E.) Degree in Computer Engineering at the University of Mumbai

Academic Year 2024-2025

**Project Mentor**

Prof. Indu Dokare

**Submitted by**

1. AMAN KUMAR D17A-33
2. ANCHAL SHARMA D17A-57
3. HARSH TULI D17A-65
4. JAY THAKKER D17A-63

**VIVEKANAND EDUCATION SOCIETY'S  
INSTITUTE OF TECHNOLOGY**

**Department of Computer Engineering**



**CERTIFICATE of Approval**

This is to certify that **Aman Kumar (33)** of Fourth Year Computer Engineering studying under the University of Mumbai has satisfactorily presented the project on "**ContentConcise: YouTube content summarization and comment analysis**" as a part of the coursework of PROJECT-I for Semester-VII under the guidance of **Mrs. Indu Dokare** in the year 2024-2025.

---

Date

---

Internal Examiner

---

External Examiner

---

Project Mentor

---

Head of the Department

---

Principal

Dr. Mrs. Nupur Giri

Dr. J. M. Nair

## **ACKNOWLEDGEMENT**

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Assistant Professor **Mrs. Indu Dokare** (Project Guide) for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr. (Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair**, for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement several times.

## **Abstract**

ContentConcise is an AI-powered web application designed to summarize YouTube videos, translate these summaries into multiple languages, and provide an intuitive voice-command query system. This platform is tailored to meet the unique needs of students with slower learning abilities, ensuring vital information is easily accessible and efficiently retrievable using advanced AI algorithms. It also offers translation of video summaries into multiple languages, allowing non-native speakers and students from diverse linguistic backgrounds to benefit from the content without language barriers.

ChatBot will help the users to tackle the user query related to the topics. Additionally, the voice-command feature enables hands-free searching and navigation, particularly beneficial for students who may have difficulties with traditional input methods. By addressing the challenges faced by students with slow learning abilities. One major addition is inclusion of dashboards for the content creator, which will help them to upload their video to reach wider audiences

## **Computer Engineering Department**

### **COURSE OUTCOMES FOR B.E PROJECT**

Learners will be to:-

<b>Course Outcome</b>	<b>Description of the Course Outcome</b>
CO 1	Do literature survey/industrial visit and identify the problem of the selected project topic.
CO2	Apply basic engineering fundamental in the domain of practical applications FORproblem identification, formulation and solution
CO 3	Attempt & Design a problem solution in a right approach to complex problems
CO 4	Cultivate the habit of working in a team
CO 5	Correlate the theoretical and experimental/simulations results and draw the proper inferences
CO 6	Demonstrate the knowledge, skills and attitudes of a professional engineer & Prepare report as per the standard guidelines.

# INDEX

Chap.	Title	Page No.
1	<b>Introduction</b>	
	1.1. Introduction to the project	8
	1.2. Motivation for the project	10
	1.3. Drawbacks of the existing system	11
	1.4. Problem definition	11
	1.5 Relevance of the project	12
	1.6 Methodology used	13
2.	Literature survey	14-20
	2.1. Research papers	
	a. Abstract of the research paper	
	b. Inference drawn from the paper	
	2.2. Books / Articles referred / news paper referred	
	2.3. Interaction with domain experts.	
	2.4. Patent search	
3.	Requirement of proposed system	20-24
	3.1 Functional requirements	
	3.2. Non-functional requirements	
	3.3. Constraints	
	3.4. Hardware, software and tools & technique requirements	
4.	Proposed Design	25-35
	4.1 Block diagram representation of the proposed system	
	4.2. Modular diagram representation of the proposed system	
	4.3 Design of the proposed system	
	a. Data Flow Diagrams	
	b. Flowchart for the proposed system	
	c. State Transition Diagram/ Activity Diagram	
	d. ER Diagram	

4.4	Algorithms utilized in the existing systems	
4.5.	Project Scheduling & Tracking using Timeline / Gnatt Chart	
5.	Proposed Results and Discussions	36-38
	5.1. Screenshot of implementation	
6.	Plan of Action For the Next Semester	39-40
	6.1. Work done till date	
	6.2. Plan of action for project II	
7.	Conclusion	40
8.	References	41-45
9.	Appendix	46
	9.1.Paper Publications	
a.	Draft of the paper / Paper Published	
b.	Plagiarism report of the paper	
c.	Xerox of project review sheet	

## **List of Figures**

Figure Number	Heading	Page no.
4.1	Block Diagram	19
4.2	Modular Diagram	21
4.3.1	Data flow diagram	23
4.3.2	Activity Diagram	24
5	Screenshot of Implementation	34

## **List of Tables**

Figure Number	Heading	Page no.
2.1.a	Abstract of research papers	9
4.5	Project Timeline: Gantt Chart	27

# Chapter 1: Introduction

## 1.1 Introduction

In today's rapidly evolving digital age, students and learners are bombarded with an overwhelming amount of online content, especially on platforms like YouTube. With over 500 hours of video uploaded every minute, YouTube has become one of the largest repositories of information across various domains, including education, entertainment, sports, and more. However, this vast sea of content can make it difficult for users to efficiently find and access relevant information without spending hours sifting through long videos. The growing challenge of managing and processing such an immense volume of content underscores the necessity for automated tools that can summarize and condense video data, allowing users to grasp essential ideas and core concepts without needing to watch videos in their entirety [1].

**ContentConcise** is an AI-powered platform designed to tackle these issues head-on. It leverages cutting-edge Natural Language Processing (NLP) techniques to transform lengthy YouTube videos into clear, concise summaries, making it easier for users, especially students, to engage with educational content without being overwhelmed by unnecessary details. This capability is particularly crucial in academic settings, where students are often pressed for time, juggling multiple subjects and deadlines. By providing succinct video summaries, **ContentConcise** enables students to focus on key concepts and information that matter most, saving them time while enhancing comprehension and retention [2], [3].

At the heart of **ContentConcise** is its advanced NLP system, which processes various textual elements associated with videos, such as transcripts, closed captions, user comments, and metadata. These textual components are then analyzed, summarized, and transformed into concise formats that accurately reflect the main ideas of the original video content [4]. This functionality makes the platform indispensable for students who need to quickly digest large amounts of information, whether it be for exam preparation, research, or general learning. Instead of being bogged down by lengthy videos, learners can quickly review the core takeaways, helping them stay organized and focused on their academic goals.

Moreover, **ContentConcise** offers multilingual support, a feature that makes the platform inclusive and accessible to users from diverse linguistic backgrounds. In a globalized world, students from different regions often face language barriers when accessing educational content, but **ContentConcise** addresses this challenge by providing video summaries in multiple languages. This feature is crucial for ensuring that students, regardless of their native language, can access high-quality educational resources and benefit from the same learning opportunities [5], [6]. By breaking down language barriers, **ContentConcise** promotes equitable learning and fosters a more inclusive educational environment, enabling students worldwide to enhance their knowledge and skills.

One of the standout features of **ContentConcise** is its sophisticated query system, which allows users to retrieve specific information from videos quickly and efficiently. For students who need targeted information for their studies, this query system is invaluable. Instead of wasting time scrolling through lengthy videos in search of particular details, users can simply ask a question and receive an immediate, relevant response [7]. This efficient approach not only saves time but also allows students to focus on critical concepts, improving their overall learning experience. The query system promotes an interactive learning process, encouraging students to engage with the content dynamically and fostering a more active approach to education.

The rapid growth of online video content has made automated video summarization tools more essential than ever before. Videos have become omnipresent across various fields, such as first-person perspectives, surveillance, sports, and education. With the explosion of video data, manual browsing, and watching of videos have become impractical, leading to network traffic, bandwidth, and browsing challenges, as noted in [1] and [2]. As the volume of video content continues to surge, summarization tools like **ContentConcise** offer an innovative solution by condensing large video collections into accessible, easily navigable summaries. This approach addresses key challenges associated with information overload while helping students and other users navigate through video data more efficiently [8], [9].

The AI-driven video summarization capabilities of **ContentConcise** are not limited to merely shortening videos. The platform is built on sophisticated algorithms that extract semantic information from the video's low-level audio and visual inputs, ensuring that the most important frames and moments are included in the summaries [3]. These summaries are not just shorter versions of the videos but are designed to capture the essence of the content, making it easier for users to grasp complex concepts without missing out on critical

points. This process is especially useful in education, where students are often tasked with reviewing extensive video lectures, tutorials, and informational videos in a limited amount of time [4].

In addition to its summarization features, **ContentConcise** integrates personalized video summarization, a cutting-edge technique in multi-model summarization. By incorporating personalized summaries, the platform tailors its content to individual users' preferences, ensuring that they receive the most relevant information based on their interests and needs. This personalized approach enhances the learning experience by allowing users to customize their summaries, helping them focus on what matters most to them [5]. The platform's use of attention-based models further enhances the summarization process, ensuring that the most pertinent information is highlighted and made accessible to users in an efficient manner [6].

As educational institutions continue to integrate technology into their curricula, tools like **ContentConcise** will play an increasingly important role in enhancing student learning outcomes. The platform's ability to distill complex information into easily digestible formats makes it an invaluable resource for students who need to manage large volumes of information quickly. By leveraging advanced NLP and AI technologies, **ContentConcise** is redefining how educational content is consumed, making it more accessible, efficient, and effective for learners worldwide [10].

In conclusion, **ContentConcise** represents a transformative approach to educational content delivery. By prioritizing efficiency, accessibility, and user engagement, the platform equips students with the resources they need to excel in their studies while reducing the cognitive burden of information overload. As the digital landscape continues to evolve, **ContentConcise** stands as a prime example of how AI and NLP can revolutionize education, making learning more accessible and empowering students around the globe.

## 1.2 Motivation

Traditional platforms struggle to meet the personalized learning needs of students, especially when navigating vast amounts of YouTube content. **ContentConcise** was created to bridge this gap by providing an efficient, YouTube-specific learning experience. The motivations behind this project are:

- **Personalized YouTube Learning Experience:** Every student processes information at their own pace. With YouTube videos being lengthy and complex, **ContentConcise** uses AI to summarize videos into concise, easy-to-digest summaries, helping students, especially those with slower learning abilities, to grasp key information quickly and efficiently.
- **Inefficiencies in YouTube Video Search:** Finding relevant information within long YouTube videos can be time-consuming and frustrating, especially for learners with cognitive challenges. **ContentConcise** extracts transcripts and provides accurate summaries with timestamps, significantly improving the ease of information retrieval for all users.
- **Actionable Feedback Through YouTube Comment Analysis:** Simple feedback isn't enough for meaningful learning. **ContentConcise** analyzes YouTube comments to evaluate the sentiment and usefulness of the video content, offering students personalized recommendations and study plans based on the analysis, making learning more targeted and effective.
- **Continuous Learning Through YouTube Interaction:** Learning doesn't end with one video. **ContentConcise** tracks user interaction and quiz performance across YouTube videos, refining its feedback and recommendations to continuously enhance the learning experience

## **1.4 Problem definition**

The problem involves addressing inefficiencies in YouTube video searches, where users often watch entire videos to find relevant content. This is especially challenging for people with learning difficulties. The solution aims to develop an AI model that summarises videos, checks correctness, and relevance, providing users with accurate information along with timestamps. The objective is to significantly improve the efficiency of information retrieval, offering a more accessible and user-friendly experience for all viewers.

Key features :

- Transcript extraction from the video, summarise it and display it for the users to get a brief idea of the content in the video.
- YouTube comments are analysed for sentiments to obtain the usefulness of the video content.
- Chat functionality where the user can ask questions related to the video content.
- Text to audio functionality for the summary generated to be read out aloud.

## **1.5 Relevance of the Project**

The ContentConcise project is very important in today's education because it helps students who may have trouble learning. By turning long YouTube videos into short summaries, the app makes it easier for learners to understand and remember important information. This feature saves time and helps users focus on what really matters.

The app also supports multiple languages, allowing students from different backgrounds to access quality educational content, which helps overcome language barriers. The use of AI technology allows for a more personalized learning experience. For example, the app gives suggestions based on how users interact with the videos, which caters to different learning styles and speeds.

Additionally, by analyzing YouTube comments, ContentConcise can understand how useful a video is and provide better summaries. Tracking user engagement and quiz results also helps users identify areas where they need to improve. Overall, this project provides valuable insights into how technology can improve learning, making it a useful model for future educational tools that aim to enhance the learning experience in the digital world.

## Chapter 2: Literature Survey

The research surrounding video summarization has gained momentum in recent years, with multiple techniques emerging to address the challenges associated with summarizing large volumes of video data. Two primary approaches—keyframe-based and structure-driven methods—are commonly employed in this domain. Keyframe-based techniques focus on selecting semantically important frames that best represent the video content. Attention scores play a crucial role in influencing the selection of these keyframes. Structure-driven methods take into account the video's inherent temporal structure, breaking the video down into segments and generating summaries based on the relationships between these segments.

Several models have been developed to tackle the problem of text summarization alongside video summarization. One such model, as presented by Hritvik Gupta et al. [9], integrates three basic algorithms: TF-IDF, BERT, and Latent Semantic Analysis (LSA). TF-IDF helps in identifying the most important words, BERT encodes sentences to generate positional embeddings, and LSA extracts relevant themes by reducing dimensionality with Singular Value Decomposition (SVD). Another notable contribution to text summarization comes from Swaranjali Jugran et al. [10], who demonstrated that Spacy is a more efficient tool for text summarization than NLTK, with its ability to handle complex NLP tasks better. In contrast, Surabhi Adhikari et al. [11] explored a range of summarization methods, including BERT, TF-IDF, and sentence ranking, which are widely used in NLP tasks.

For video summarization, various studies have implemented deep learning techniques. The research by Kaiz Merchant et al. [13] introduced a model that applies latent semantic analysis and NLP for summarizing legal judgments, highlighting the effectiveness of LSA in dealing with large textual data. In [14], videos are viewed as temporal graphs, and summary generation is performed using structural and attention information. Meanwhile, studies such as [15] introduced a super-frame segmentation method for keyframe extraction, relying on frame-level interestingness to select the most informative frames.

More advanced video summarization models, such as the one presented in [16], incorporate deep neural networks and joint models for abstractive summarization. These models allow the system to differentiate relevant content from irrelevant data effectively. In other approaches, static and dynamic videos are categorized using transcript translation and short boundary detection algorithms [17]. Furthermore, some research aims at specific use cases

like lecture video indexing using syntactic similarity measures, as seen in [18], and real-time video summarization for mobile platforms [19].

Among the popular tools for video summarization, Merlin stands out as an AI-powered solution that offers high-quality summarization and even generates content outlines for educational videos. Another example is Video2Summary, which uses keyframe extraction and AI-based algorithms to produce concise video summaries.

Table 2.1.a. Abstract of Research Papers

<b>Author</b>	<b>Dataset</b>	<b>Technique Used</b>	<b>Limitations</b>
Huang et al. [22]	Query-focused dataset for summarizing	GPT-2 + specialized attention network	Embedding dimension issues
Narasimhan et al. [23]	Video summarizing dataset	Bi-Modal Transformer + CLIP-IT	Biases embedded in representations
Xiao et al. [25]	QSAN dataset for summarizing	Self-attentive reinforcement model	Heavy pre-processing required
Nalla et al. [26]	Query-focused video summarization	Local-global focus with dynamic feature fusion	Feature fusion could lose some data
Sharghi et al. [32]	TV episodes for video summarization	SH-DPP (Sequential Hierarchical Determinant Point Process)	High computational cost
Zhang et al. [33]	Query-focused video summarization	Three-player adversarial summarization	Highly dependent on parameter selection

Ke Zhang et al. [39]	Multi-layer perceptron with BiLSTM for summarization	Frame-level significance and binary labels	Recurring models cannot be parallelized
Behrooz Mahasseni [40]	GAN + LSTM keyframe selector for video summarization	Unsupervised, intermediate representations used for better results	Complex training for GANs

### 2.1.a. Inference Drawn from the Papers

From the reviewed research papers, it becomes evident that hybrid techniques combining traditional models like TF-IDF with modern NLP tools such as BERT and LSA are highly effective in both text and video summarization. The integration of attention mechanisms, structural information, and supervised learning methods improves the accuracy of keyframe selection in video summarization. However, computational complexity remains a challenge, especially for deep learning approaches such as GANs and LSTMs. Additionally, methods that focus on specialized use cases, like lecture video indexing or mobile platform summarization, show potential for real-time applications but often require further refinement to enhance performance across diverse domains.

### 2.1.b. Drawbacks of the Existing System

Table 2.1. Lacunas of Existing systems

System Name	Creator	Drawbacks
<b>Merlin AI</b>	<b>Merlin AI Inc.</b>	- Lacks premium feature in the free tier - Limited to 102 query per day - Struggles with accuracy in summarizing nuanced content
<b>Sider AI</b>	<b>Sider Technologies</b>	- May miss contextual elements important for broader summarization - Not free to all

<b>IBM Watson Video Analytics</b>	<b>IBM</b>	- High complexity requires significant resources to implement - Outdated use of algorithm
<b>Microsoft Video Indexer</b>	<b>Microsoft</b>	- Reliant on cloud-based processing, which can introduce latency - Requires internet connectivity for optimal performance - Might struggle with summarizing non-English content effectively

In the quest for effective video summarization solutions, several systems have emerged, each with unique features and limitations.

First, Merlin AI by Merlin AI Inc. stands out for its focus on leveraging artificial intelligence for video content. However, it has notable drawbacks: the free tier lacks premium features, limiting users to only 102 queries per day, and it struggles with accurately summarizing nuanced content, potentially missing critical details.

Next, Sider AI from Sider Technologies offers promising capabilities in summarization. Despite its strengths, users may find that it misses important contextual elements that are essential for broader summarization. Additionally, Sider AI is not freely accessible to all users, which may limit its adoption.

IBM Watson Video Analytics is another contender in this space, developed by IBM. While it provides powerful analytics, the system's high complexity necessitates significant resources for implementation. Moreover, its reliance on outdated algorithms may hinder its effectiveness in comparison to more modern solutions.

Microsoft Video Indexer from Microsoft provides a robust cloud-based service for video analysis. However, this reliance on cloud processing can introduce latency, which may affect user experience. Furthermore, it requires consistent internet connectivity to function optimally and may struggle with effectively summarizing non-English content, limiting its usability in diverse contexts.

## 2.2 Books / Articles Referred

### 1. Introduction to Video Summarization – Huang et al.

This book provides a comprehensive foundation in video summarization, starting with an introduction to the key concepts and challenges faced in the field. It covers the historical evolution of summarization techniques, ranging from early keyframe extraction methods to

the latest advancements in deep learning-driven approaches. The authors delve into both extractive and abstractive summarization techniques, emphasizing how keyframes can capture the most significant content within a video. They also highlight the applications of video summarization in areas like video surveillance, entertainment, and online video platforms, making this a critical resource for anyone looking to understand the basics and applications of summarization.

## 2. Deep Learning Algorithms for Video Summarization – Review Articles

This series of review articles focuses on the use of deep learning algorithms, such as convolutional neural networks (CNNs), long short-term memory networks (LSTMs), and generative adversarial networks (GANs), for video summarization. These articles offer a deep dive into how these algorithms are applied in practice, particularly in creating both keyframe-based and segment-based summaries. A large part of this work is dedicated to explaining how these models handle the temporal dynamics of videos and how they outperform traditional methods. The reviews also highlight the limitations of deep learning models in terms of computational complexity and the need for large datasets, but they also discuss ongoing efforts to optimize these approaches.

## 3. TF-IDF and BERT Models in Text Summarization – Research Paper by Gupta et al.

It explains how TF-IDF helps in identifying the most important words within a text, while BERT is responsible for understanding the context and relationships between sentences. The paper shows how these models can be applied to extract keyframes in video summarization by first converting video content into text (transcription) and then applying summarization techniques. By merging text summarization methods with video analysis, the study opens a pathway for creating more accurate and semantically rich video summaries. Additionally, it discusses the application of Latent Semantic Analysis (LSA) to reduce dimensionality and capture important themes in the video content.

#### **4. NLP Techniques in Summarization – Articles on NLTK and Spacy**

The articles provide practical examples of how to use Python-based NLP libraries to process and summarize textual data. For instance, the articles demonstrate the efficiency of NLTK in handling large text corpora and its toolkit of algorithms for tokenization, stemming, and text classification. Similarly, Spacy is highlighted for its deep learning capabilities, which allow it to perform more sophisticated tasks such as named entity recognition (NER) and part-of-speech (POS) tagging. The articles compare the performance of these tools and suggest ways to integrate them into larger systems for video summarization by first analyzing textual transcriptions.

#### **5. Text Summarization Techniques for Legal Documents – Merchant et al.**

This article specifically addresses the challenge of summarizing long and complex legal judgments. The authors propose a model that leverages latent semantic analysis (LSA) to reduce redundancy in legal texts while preserving the critical points of the judgment. Their approach highlights how LSA can efficiently condense lengthy texts, ensuring that the core legal arguments are presented in a concise format. The paper also discusses the limitations of existing summarization models that rely heavily on word similarity, which may overlook the broader context or nuances of legal language. The authors suggest that future improvements could involve deep learning models such as BERT to better understand the legal language's intricacies.

#### **6. Real-Time Video Summarization for Mobile Devices – Chen et al.**

This article delves into the specific challenges and solutions for real-time video summarization on mobile devices. The authors focus on the need for lightweight algorithms capable of handling the computational constraints of mobile platforms. Their proposed method involves analyzing both intrinsic video content and external metadata, such as GPS location and timestamps. The system processes video data in real-time, generating summaries as the video is being recorded, making it ideal for applications in personal video editing, security, and surveillance. This article provides a technical breakdown of the algorithms used, including how they are optimized for efficiency without compromising the quality of the summary.

## **2.3 Patent Search**

### **a. Title of the Patent and Year of the Patent**

1. **Dynamic Video Summarization Method (2020)**
2. **Automated Video Indexing System (2019)**
3. **Real-Time Video Summarization for Mobile Devices (2021)**
4. **Keyframe-Based Video Summarization Using Attention Models (2022)**

### **b. Summary of the Patents**

#### **1. Dynamic Video Summarization Method (2020):**

The method involves segmenting the video into scenes and using attention-based models to prioritize frames for the summary based on relevance. This technique is applicable in real-time and offline scenarios, making it suitable for applications in security and media production.

#### **2. Automated Video Indexing System (2019):**

This system automatically indexes videos by detecting scene changes and extracting important frames. The patent focuses on video analytics and retrieval, enabling users to search for specific events or moments within a video. The key innovation lies in its ability to adapt to various video formats and compress the indexing process for faster search results.

#### **3. Real-Time Video Summarization for Mobile Devices (2021):**

By analyzing both intrinsic video data (such as content) and external metadata (such as camera information), the system generates a summary as the video is being recorded. The approach uses lightweight algorithms to ensure that the summarization process does not impact the device's performance.

#### **4. Keyframe-Based Video Summarization Using Attention Models (2022):**

This patent introduces an attention-based model for video summarization that selects keyframes based on their relevance to the overall narrative. The system uses deep learning to learn the attention scores that guide keyframe selection, ensuring that the most semantically meaningful frames are included in the summary. This method is particularly effective in video editing and content creation.

# **Chapter 3: Requirement of proposed system**

## **3.1 Functional Requirements**

The project requires several key functions for the YouTube video summarization and content analysis system. The frontend will allow users to input a YouTube URL, which the backend, using Flask or Django, will validate and process. The system will then check for subtitles using the YouTube Data API. If subtitles are missing, a speech-to-text service like Google Cloud will generate a transcript for summarization.

Abstractive summarization will be done using NLP models like T5 or BART, with the summary displayed dynamically on the website. A chatbot using a model like LLaMA will handle user queries related to the summarized content.

The system will also translate the summary into Indian languages using the [translate](#) library. For comment analysis, sentiment analysis will be performed using tools like TextBlob or VADER, generating content suggestions for creators based on viewer feedback.

If available, the YouTube API will extract the most replayed parts of the video; otherwise, engagement metrics like comments or likes will be analyzed. A dashboard built with React.js or Angular will provide content creators with insights and suggestions, with data managed via REST APIs and stored in a database.

## 3.2 Non-Functional Requirements

The non-functional requirements for the YouTube video summarization and content analysis system include:

- **Performance:** The system should provide quick responses, with video analysis and summarization completed within a reasonable timeframe, ensuring low latency in user interactions, especially during chatbot queries and summary generation.
- **Scalability:** The backend architecture should be scalable to handle multiple simultaneous video inputs, processing requests, and user interactions without a drop in performance, accommodating a growing number of users and videos.
- **Usability:** The user interface should be intuitive and easy to navigate, ensuring users can input YouTube URLs, view summaries, interact with the chatbot, and receive content suggestions without confusion.
- **Reliability:** The system should be highly reliable, ensuring continuous availability of features like summarization, translation, and sentiment analysis. It should handle errors, such as invalid YouTube URLs or API failures, gracefully with appropriate feedback.
- **Security:** User data, including video processing results, queries, and comments, should be securely handled. API keys for services like YouTube Data API and speech-to-text should be securely stored to prevent unauthorized access.
- **Maintainability:** The system codebase should be modular and easy to maintain, allowing for future upgrades or integration of new services (e.g., adding more APIs or NLP models) with minimal disruption to the core functionality.

## 3.3 Constraint

The constraints for the YouTube video summarization and content analysis system include:

- **YouTube API Rate Limits:** The system will be constrained by the rate limits imposed by the YouTube Data API, which restricts the number of API requests that can be made within a specific time period. Exceeding these limits could lead to temporary unavailability of features like subtitle checking and most-replayed part extraction.
- **Transcript Accuracy:** The accuracy of the transcript generation process will depend on the quality of the audio in the video and the performance of the speech-to-text service

(e.g., Google Cloud). Noisy or unclear audio could lead to inaccurate transcripts, impacting the quality of the summaries.

- **Model and API Limitations:** The quality and speed of the NLP models (e.g., T5, BART for summarization, and LLaMA for chatbot queries) are constrained by the computational resources available. Additionally, the performance of external APIs (e.g., for translation or speech-to-text) could vary depending on network conditions or service availability.
- **Real-Time Processing Limits:** Summarization, translation, and sentiment analysis processes may take longer for large or complex videos, creating delays in providing results to the user. Processing heavy tasks in real time could strain system performance.
- **Storage Constraints:** Storing transcripts, summaries, and analysis data for multiple videos may lead to significant storage requirements over time, especially if large amounts of data need to be retained for future analysis or dashboard insights.
- **Multi-Language Support:** The system's ability to provide accurate translations will be constrained by the limitations of the translation library or service being used, particularly when handling nuanced language or complex text structures in different Indian languages.

### **3.4 Hardware, Software , Technology and tools utilized**

#### **Hardware Used:**

- Processor: Intel® Core™ i3-2350M
- CPU @ 2.30GHz Installed memory
- RAM: 4.00GB
- System Type: 64-bit Operating System
- GPU: Nvidia GPU

#### **Tech Stack:**

- Front-end: Next.js, React, Tailwind CSS, TypeScript
- Back-end: Flask, Hugging Face Transformers, Google Translator API,, Crisp Provider
- Database: MongoDB
- Other Tools: Crisp SDK

## **Techniques:**

- **Natural Language Processing (NLP):** Utilizes models like LLaMA to understand user queries and provide contextually relevant answers.
- **Question-Answering System:** Employs fine-tuned LLMs to answer questions based on the video's content, leveraging transcripts and summaries.
- **Backend Integration:** Connects the chatbot with the backend for input processing and ensures smooth communication.
- **Frontend-Backend Communication:** Uses RESTful APIs to facilitate interaction between the user interface and the chatbot engine.

## **Tools:**

- **NLP Models:** Implements LLMs such as LLaMA and Gemini to enhance the chatbot's understanding and responsiveness.
- **Python:** Serves as the backend implementation language, providing robust support for NLP and chatbot functionalities.
- **Flask/Django:** Utilizes these web frameworks for managing backend requests and integrating the chatbot with the frontend.
- **RESTful APIs:** Enables efficient communication between the frontend and backend, allowing for seamless query handling and response delivery.

## **3.7 Project Proposal**

The YouTube Video Summarization and Content Analysis System aims to simplify the process of extracting information from lengthy videos, addressing content overload on platforms like YouTube. The system will offer features such as automated video summarization, a chatbot for answering user queries, translation of summaries into Indian languages, sentiment analysis of comments, and personalized content suggestions for creators.

Key features include allowing users to input a YouTube URL, checking for subtitles, and generating transcripts if necessary. The system will use NLP models like T5 or BART for summarizing content and a chatbot powered by a large language model (e.g., LLaMA) to handle user questions. Sentiment analysis will provide insights into viewer feedback, while a creator dashboard will display video performance metrics and suggestions.

The project will utilize technologies such as Python for backend development (using Flask or Django), HTML/CSS/JavaScript for the frontend, and various APIs, including the YouTube Data API and Google Cloud Speech-to-Text. The system must meet non-functional requirements like performance, scalability, and security while being constrained by API rate limits and the accuracy of transcript generation.

Currently, the chatbot has been developed, allowing users to ask questions about videos. This foundation will be expanded to include the full range of features, creating a comprehensive tool for both viewers and content creators. The goal is to enhance user interaction with video content and provide valuable insights for creators.

## Chapter 4: Proposed Design

The proposed design for **ContentConcise** focuses on creating an intuitive user interface that seamlessly integrates various functionalities to enhance the learning experience. The application will feature a user-friendly homepage where students can easily input YouTube URLs for processing. Once a video is submitted, the system will utilize the YouTube Data API to check for subtitle availability and, if necessary, employ advanced speech-to-text services to generate transcripts from audio. These transcripts will then be analyzed using state-of-the-art natural language processing models to produce concise, coherent summaries. Additionally, a dynamic query system will allow users to ask specific questions about the video content, leveraging AI-driven responses to facilitate quick information retrieval. To ensure inclusivity, the summaries will be available in multiple languages, broadening accessibility for diverse student populations. Overall, the design emphasizes efficiency, interactivity, and inclusiveness, creating a holistic educational tool tailored to meet the needs of modern learners.

## 4.1 Block diagram of the system

The process flow diagram that shows in fig 4.1 how to process a YouTube video, including summarizing it and answering user questions.

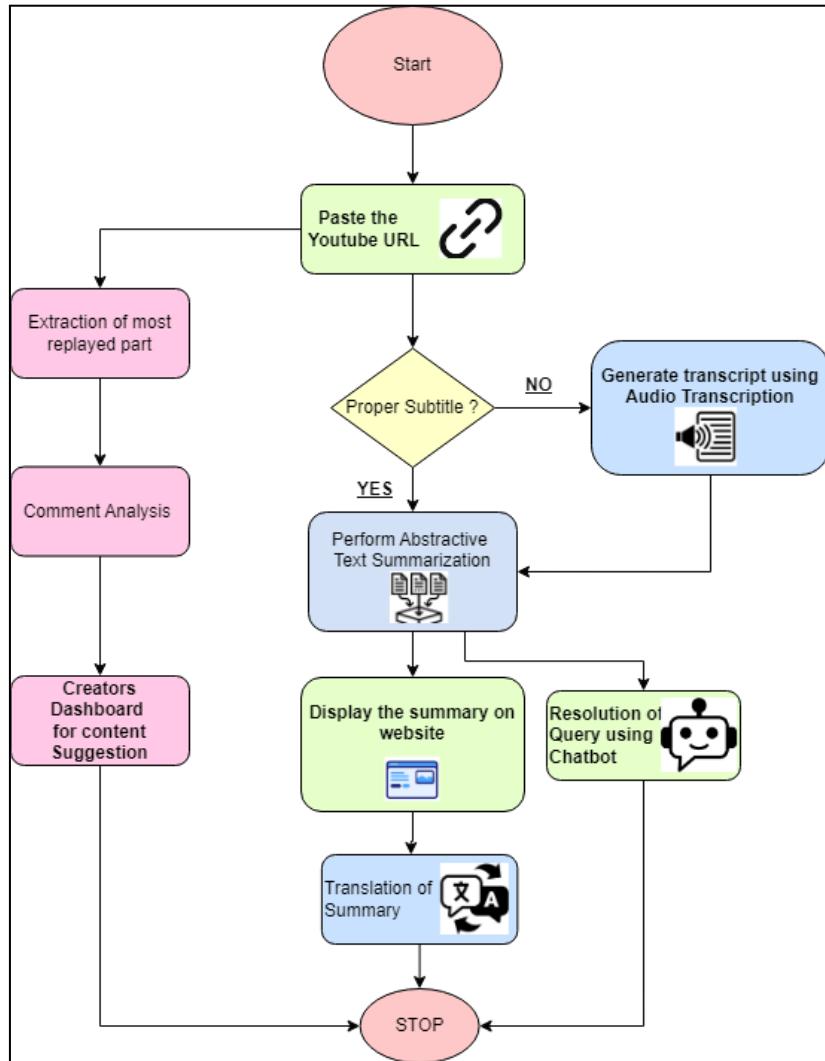


Fig. 4.1 Block diagram

### Block diagram explanation:

The steps are broken down as follows:

1. **Start:** The process begins.
2. **Paste the YouTube URL:** The user inputs a YouTube video URL.
3. **Proper Subtitle:** The system checks whether the video has proper subtitles.
  - a. **Yes:** If subtitles are available, the process moves to "Perform Abstractive Text Summarization."
  - b. **No:** If subtitles are not available, the system generates a transcript using audio transcription (speech-to-text).

4. **Abstractive Text Summarization:** The transcript (either from subtitles or generated via audio transcription) is processed using abstractive text summarization. This technique creates a summary by understanding the meaning and context, rather than extracting text directly.

5. **Display the Summary on Website:** The generated summary is displayed on a website.

6. **Translation of Summary:** The summary can be translated into different languages for accessibility.

7. **Resolution of Query using Chatbot:** A chatbot can be used to answer any queries related to the video content based on the summary or transcript.

8. **Comment Analysis and Creator's Dashboard**

9. **Stop:** The process ends.

To summarize, this system receives a URL for a YouTube video, parses the content (using transcription or subtitles), generates a synopsis, and offers extra features like content analysis for creators, chatbot-based query resolution, and translation.

## 4.2 Modular diagram representation of the proposed system

A YouTube summary website with a modular layout as shown in fig 4.2 where different modules communicate with one another. The **User Interface Module** handles user interactions by displaying video summaries or search results. It works closely with the **User Query Module**, which processes user input, searches for videos, and interacts with the **Video Module** for fetching video details and summaries.

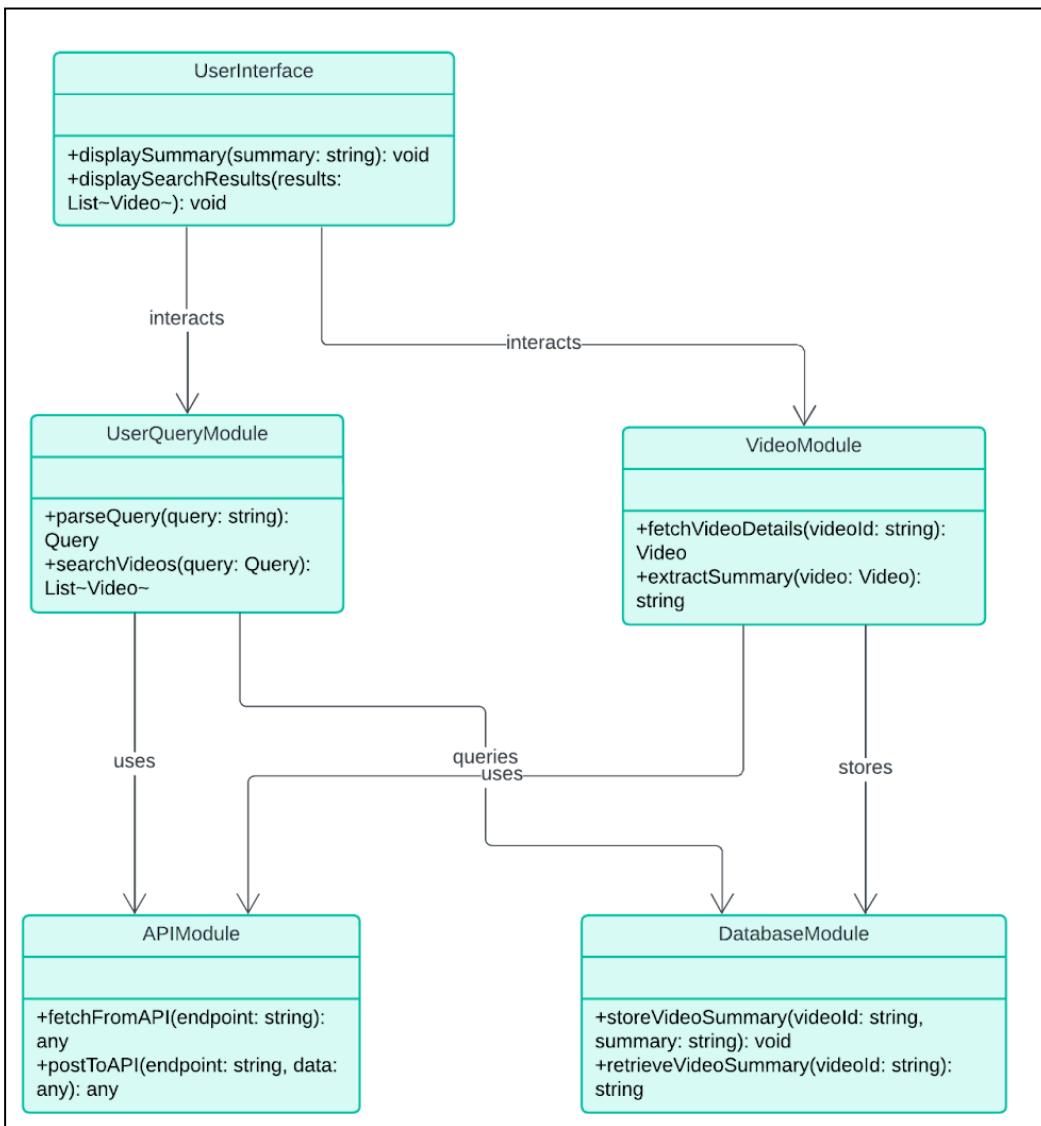


Fig. 4.2: Diagram of a modular structure

The **Video Module** is responsible for extracting video content and storing it, while the **API Module** fetches data from external services, like YouTube, to support video-related tasks. Finally, the **Database Module** stores and retrieves video summaries for easy access. This modular design simplifies system maintenance and supports future scalability.

## 4.3 Design of the proposed system :

### Data Flow Diagrams:

The following Data Flow Diagram (DFD) as shown in fig 4.3 illustrates how user inputs, such as YouTube URLs, are processed through various stages, including subtitle checks, transcript generation, and summarization, to produce concise summaries and facilitate query responses.

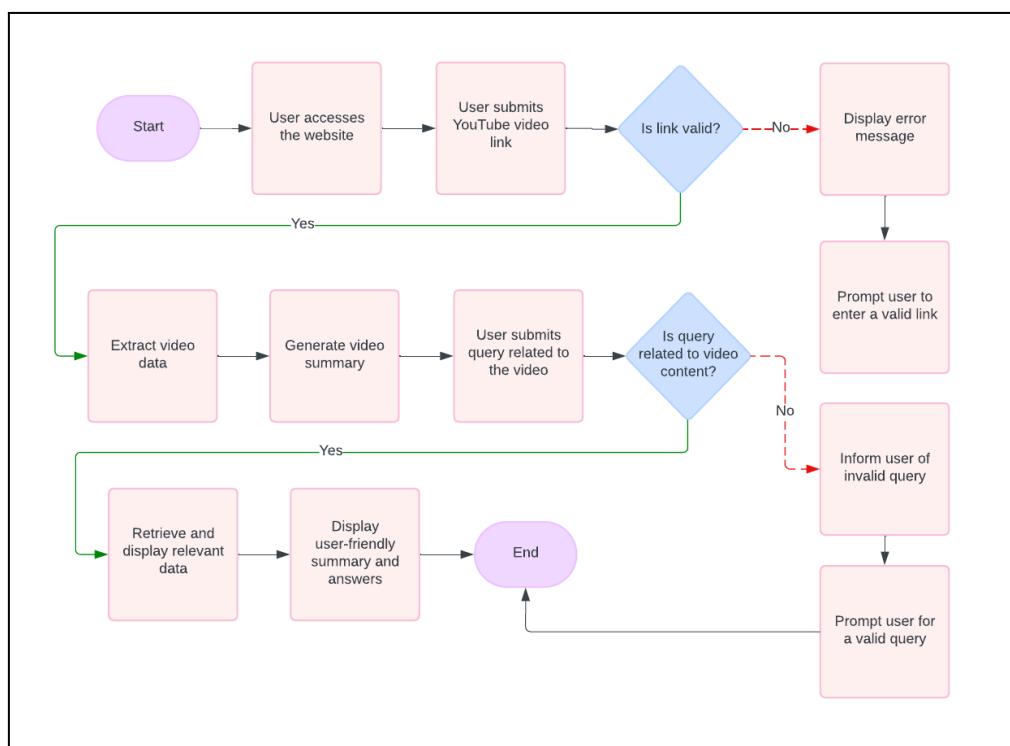


fig 4.3 Data Flow Diagram

## 4.4 State Transition Diagram/ Activity Diagram:

The Activity Diagram in fig 4.4 visually represents the workflow involved in processing a user's request. It outlines activities such as "User Inputs YouTube URL," "Check for Subtitles," "Generate Transcript," "Summarize Content," "Display Summary," and "Respond to Query."

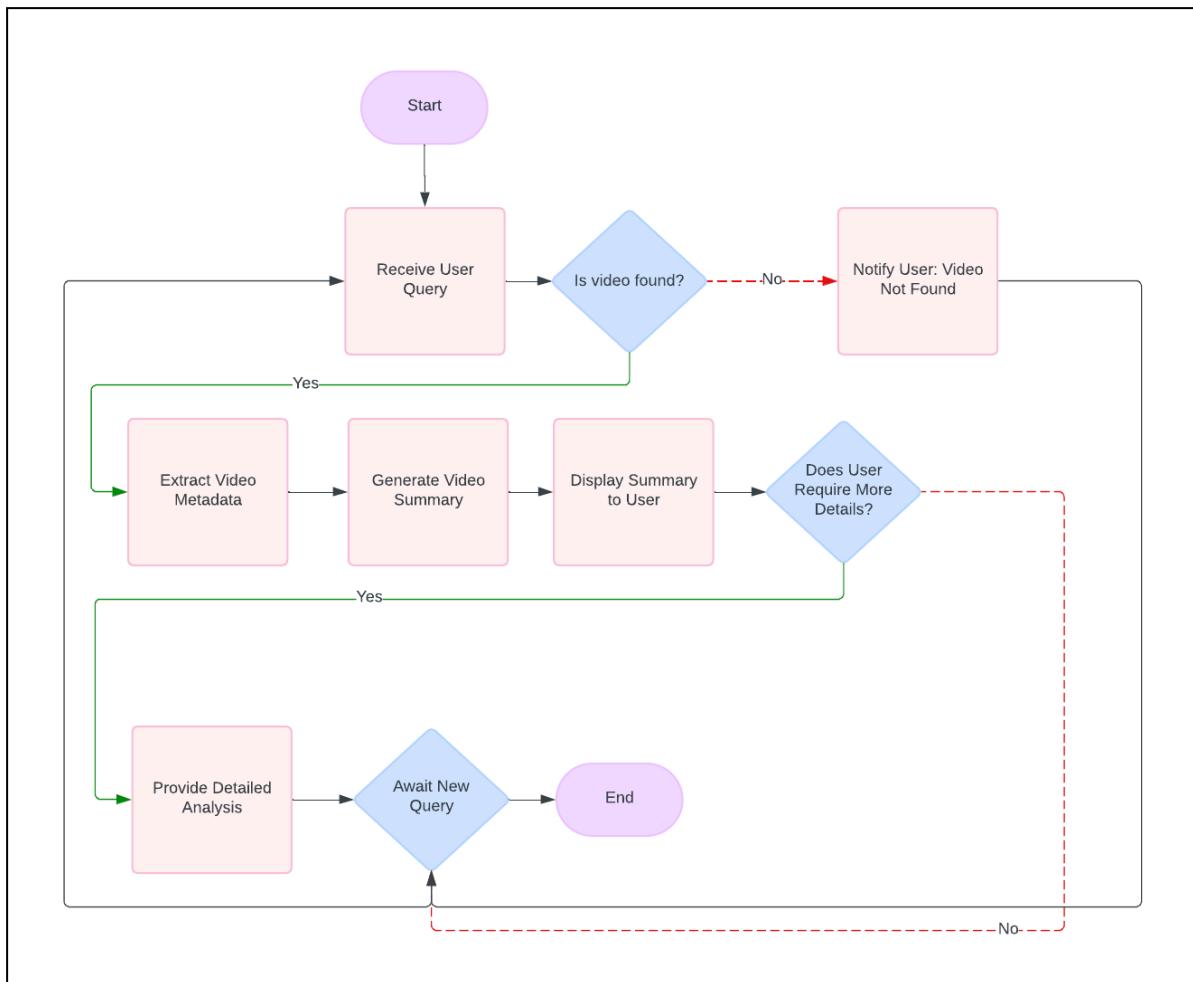


Fig 4..4 State Transition Diagram

## 4.5 Methodology used

A user opens the platform and is greeted by a simple input box, ready to accept a YouTube URL. Once they paste the URL and hit submit, the system checks the input using JavaScript to ensure it's valid, then hands it off to the backend running on Flask.

Upon receiving the video link, the backend immediately calls the YouTube Data API to check if the video has subtitles. If subtitles are available, the system takes a sigh of relief and moves to the next step. However, if no subtitles are found, the platform springs into action, triggering an alternative path.

In the absence of subtitles, the platform utilizes a speech-to-text service, such as Google Cloud Speech-to-Text, to generate a transcript directly from the video's audio. The system waits patiently while the audio is processed, after which the transcript is saved for further use.

With the transcript now available, it's time to create a summarized version of the video's content. The platform taps into powerful NLP models like T5 or BART to perform abstractive text summarization. These models analyze the transcript, producing a shorter and more coherent version that captures the essence of the video.

Once the summary is ready, the platform displays it on the user interface. The frontend is designed to seamlessly show the summary using HTML, CSS, and JavaScript, ensuring a clean presentation for the user.

At this point, if the user has questions about the video or its summary, a chatbot is available to assist. Integrated into the system using frameworks like Rasa or Dialogflow, the chatbot is prepared to answer any relevant queries, creating an interactive experience for the user.

Meanwhile, the system also translates the video summary into different languages using the Google Translate API. Whether the user prefers Hindi, French, or Spanish, the platform provides the translated summary instantly.

Finally, for content creators, the platform dives deeper by analyzing the comments on the video. It fetches comments using the YouTube Data API and performs sentiment analysis using libraries like TextBlob or VADER. Based on this analysis, it provides insights and content suggestions, helping creators refine their future videos based on viewer reactions.

After the video summary and comment analysis are completed, the platform moves forward to provide an additional layer of insight. Using the YouTube Data API, the system attempts to fetch timestamps for the most replayed parts of the video. If this data is available, it highlights the sections of the video that users found particularly interesting. If this specific information isn't available, the system instead relies on user engagement metrics, such as comments, likes, and shares, to estimate the most engaging parts of the video.

The extracted sections are then analyzed, with the platform focusing on the content that generated the most attention. These key segments are stored for further use, giving content creators a clear view of what resonated most with their audience.

With all this information processed, the platform brings it all together in a comprehensive dashboard designed specifically for content creators. Using a frontend framework like React.js or Angular, the platform presents a clean and interactive dashboard where creators can view all the gathered insights. The dashboard offers detailed analysis, including suggestions for future content based on engagement metrics, comment sentiment, and the most replayed sections of the video.

The backend, powered by Flask or Django, serves the dashboard with data, pulling results from a database where all the analysis and suggestions are stored. Content creators can easily navigate through the information, helping them make informed decisions about their content strategy.

## 4.6 Project Scheduling & Tracking using Timeline / Gantt Chart

Table 4.5. Project Timeline-gantt chart

Week/ Task	1	2	3	4	5	6	7	8	9	10
<b>Planning and Requirement</b>	X	X								
<b>System Design</b>			X	X						
<b>Backend Development</b>				X	X	X				
<b>Frontend Development</b>					X	X	X			
<b>Chatbot Development</b>						X	X	X	X	X

# **Chapter 5: Results and Discussions**

## **4.1. Introduction:**

This chapter delves into the outcomes and impact of the ContentConcise project. We'll analyze the functionalities delivered in the final application, assess its effectiveness in achieving its initial objectives, and explore user feedback (if available) to gain valuable insights into the learning experience. Following a review of project costs and the revisited feasibility study, we'll discuss the project's overall success.

## **4.2. Cost Estimation:**

Cost estimation for the ContentConcise project involves identifying and calculating the various expenses associated with its development and implementation. This includes costs related to technology, infrastructure, and any other expenses that may arise during the project lifecycle.

Estimated Costs:

### **Gemini API (Pay-As-You-Go):**

This is the most variable cost factor. We recommend starting with a pay-as-you-go plan. Based on industry benchmarks for similar applications with moderate traffic, we estimate a cost range of ₹3,700 - ₹11,100 per month for API usage (quiz generation). This range considers different question complexity and assumes 1000-5000 quizzes taken per month. (Conversion rate used: \$1 = ₹80)

### **Clerk Authentication (Free Plan with Limitations):**

The free Clerk plan allows for a limited number of users and monthly active users (MAUs). This might be suitable for initial testing. Upgrading will likely be necessary for moderate user traffic. Costs can range from ₹0 - ₹3,680 per month depending on the chosen plan.

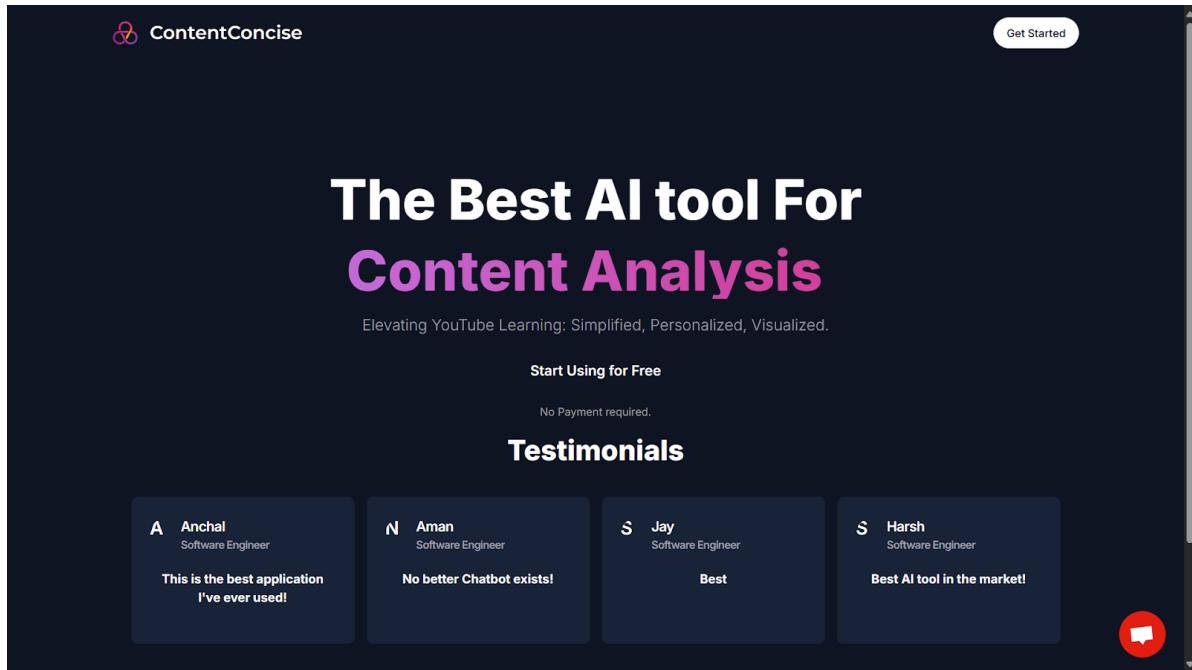
### **Total Estimated Monthly Cost Range (INR):**

Based on the considerations above, the estimated monthly cost range for the ContentConcise project falls between ₹0 - ₹7280 /-. The actual cost will depend on specific API usage, data storage needs, and future feature implementations.

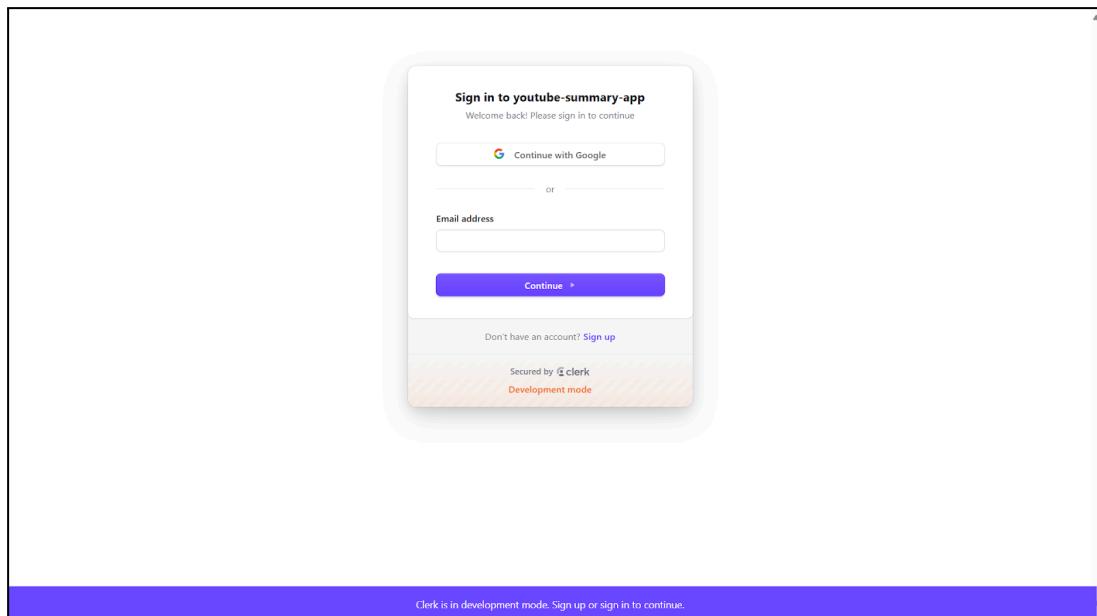
## 4.3. Results of Implementation:

### 4.3.1. FrontEnd Implementation:

- a) Landing Page: The ContentConcise landing page serves as a user-friendly entry point to the YouTube Summarization page, providing information about its features and benefits.



- b) Authentication Page: The authentication page of the ContentConcise application provides a secure gateway for users to access their accounts.



The screenshot shows the 'Users' section of the ContentConcise application. At the top, there are navigation links: 'Personal Account' (with a profile icon), 'youtube-summary-app' (marked as 'Free'), and 'Development'. On the right, there are user profile icons and a search bar. Below the navigation, there are tabs for 'Overview', 'Users' (which is selected), 'Organizations', and 'Configure'. A sub-header 'Users' is followed by the instruction 'View and manage users'. Underneath, there are two tabs: 'All' (selected) and 'Invitations'. A search bar and a 'Sort by: Joined' dropdown are present. A prominent blue button labeled 'Create user' is located on the right. The main area displays a table with columns: 'User', 'Last signed in', and 'Joined'. Two users are listed:

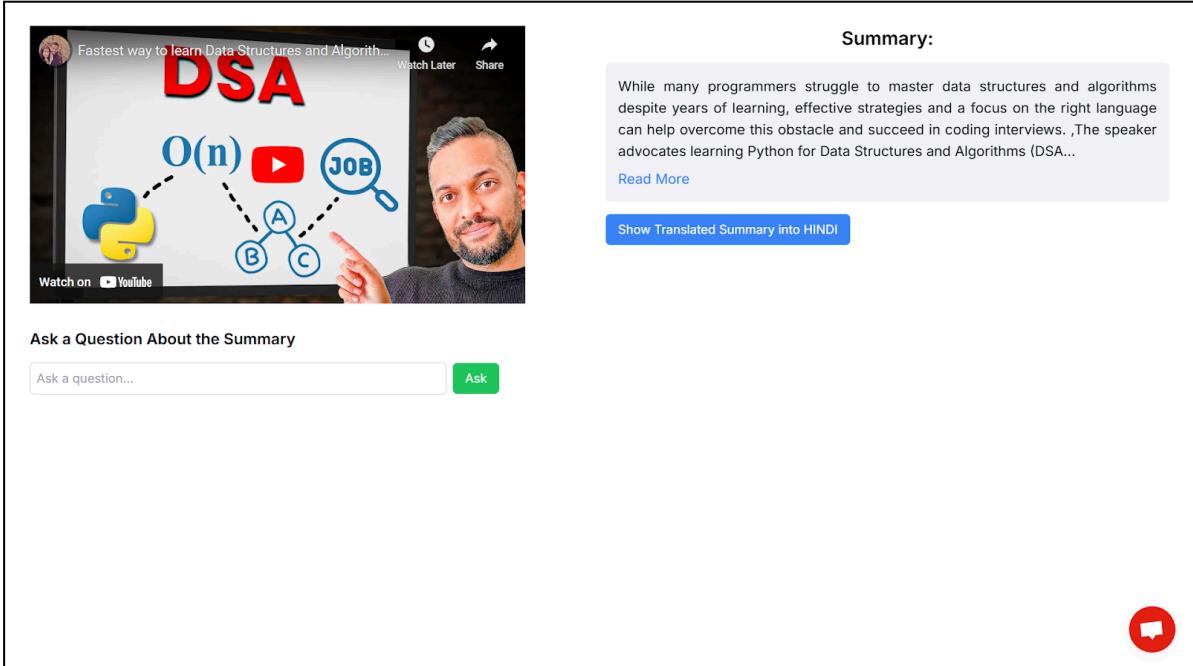
User	Last signed in	Joined
Harsh tuli harsh.tuli452@gmail.com	Today at 5:41 PM	Today at 5:41 PM
Anchal Sharma as1514580@gmail.com	Today at 5:57 PM	08/21/2024

- c) Dashboard Page: The dashboard page of the ContentConcise application serves as a comprehensive hub for personalized query resolve, offering access to links, QnA, Access to Youtube Videos and customer support.

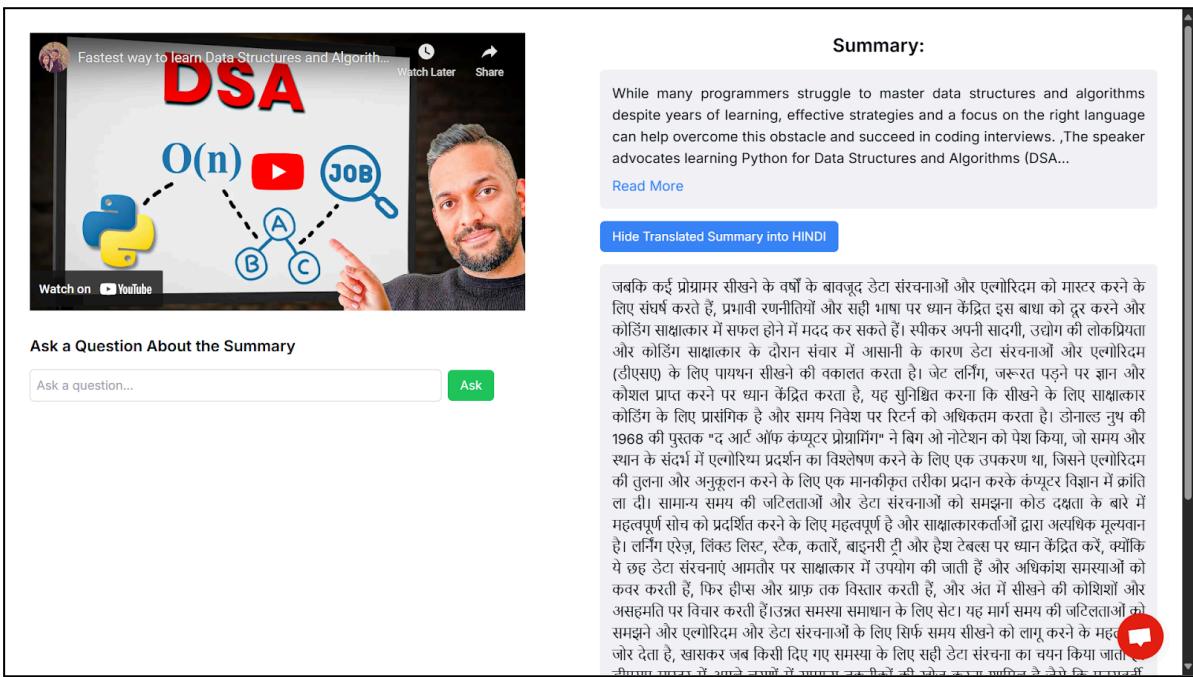
The screenshot shows the main dashboard of the ContentConcise application. At the top, there is a logo and a user profile icon. The central heading is 'Explore the Power of AI.' with the sub-instruction 'Elevate Your Learning Experience with AI-Driven Conversations and Insights.' Below this, there is a search bar with the URL 'https://www.youtube.com/watch?v=tNrNL0Cqzco' and a red 'Generate' button. Three main features are listed in boxes:

- Questions and Answers**: Efficient learning with smart Q&A for precise information retrieval.
- Access YouTube Videos**: Direct access to diverse educational content through seamless YouTube integration.
- Customer Support**: Instant tech support via chatbot, ensuring a smooth learning experience.

A red speech bubble icon is located in the bottom right corner.



d) Summary Page: The summary page of the ContentConcise application offers users an initial content from the youtube video having the key important information.



e) Translation Page: The translation page of the ContentConcise application only offers Hindi Language which can be further translated to other languages

**Summary:**

While many programmers struggle to master data structures and algorithms despite years of learning, effective strategies and a focus on the right language can help overcome this obstacle and succeed in coding interviews. The speaker advocates learning Python for Data Structures and Algorithms (DSA...).

[Read More](#)

[Hide Translated Summary into HINDI](#)

**Answer:**  
The main topic is how to effectively learn data structures and algorithms for success in coding interviews.

#### 4.3.3 Implementation

- f) Personalized Chatbot for Queries: It resolves the specific answers from the context provided in the video.

#### 4.3.2. Overall Analysis and comparison

Gemini, developed by Google, has demonstrated superior performance in a range of assessments. For instance, it scored 90.0% on the Massive Multitask Language Understanding (MMLU) benchmark, outperforming LLaMA-2, which scored 68% in the same test. Additionally, Gemini achieved notable results in other evaluations: 94.4% on the Grade-school Math (GSM8K) benchmark and 83.6% on the Big-Bench Hard test. In contrast, LLaMA-2's performance tends to lag behind that of Gemini across most metrics.

```
[3] from sentence_transformers import SentenceTransformer
    from sklearn.metrics.pairwise import cosine_similarity

[4] model = SentenceTransformer('paraphrase-MiniLM-L6-v2') # Small, efficient model for summarization
→ /usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py:1601: FutureWarning: `clean_up_tokenization_spaces` was not set. It will be
    warnings.warn(
    ↗ lamma_summary = """The video introduces the core concepts of React that every developer should learn and master. The speaker has selected these concepts based on the video. The text also highlights the importance of components in building UIs. Components are reusable pieces of code that make up the visual layer of an application. They can be nested as deeply as needed, allowing for complex layouts. There are two types of components: class-based and function-based. Function-based components are currently the trend, thanks to the use of React Hooks. JSX (JavaScript XML) is a syntax extension that allows writing HTML-like code with JavaScript, making it easier to create components. The text suggests learning functional components first, using JSX, as it's more modern and versatile than traditional HTML tags. Here's a summary: In React, JSX (JavaScript XML) code has similarities to HTML tags. However, it's not directly readable by browsers and needs to be compiled into traditional HTML. React Router is used for handling URL routing in Single Page Applications (SPAs). It allows multiple pages to be rendered on the same page by synchronizing key points: JSX uses a similar syntax to HTML, but has some differences like class declarations. React Router handles URL routing and rendering components into the DOM based on the current URL. Props (short for "properties") are used to pass data down from one component to another. They can be passed as functions and can be accessed anywhere within the component. Props can be passed multiple times, allowing for flexible data sharing between components. The text appears to be discussing React components, specifically "Prop drilling" refers to passing props from a parent component down through multiple levels of nested components. A "state" object is used to represent information about a particular component in JavaScript. Traditionally, React uses class-based components to set and update state, but modernly, React Hooks like useState are used. An example is given: setting an initial state (an empty array) for a list of notes, mapping through that state to render data, and updating the state with new data. The text also mentions that understanding the "component life cycle" is essential for every React developer. However, it doesn't provide further details on the component life cycle. Mounting phase: When a component is first added to the DOM. Updating phase: When the component's state changes and needs to be updated.
```

```
[14] gemini_summary = """This video introduces React, a JavaScript library for building user interfaces. It emphasizes core concepts like mounting, updating, and unmounting. Mounting occurs when a component is added to the DOM. Updating happens when a component needs to modify itself due to changes. Unmounting happens when a component is removed from the DOM. Class components use methods like componentDidMount, componentDidUpdate, and componentWillUnmount to manage lifecycle events. Functional components utilize the useEffect hook to handle these lifecycle stages. React hooks are exclusive to functional components, empowering them with state management and other capabilities without resorting to class components. Hooks are essential for modern React development as they provide a flexible and concise way to work with component lifecycles. State management solutions like the built-in Context API or external libraries like Redux enable you to create global state across your application. Understanding the virtual DOM's operation is crucial for comprehending how React renders and updates components. It involves diffing the old and new states to efficiently update the DOM. When rendering lists of data, using the key prop is essential for React to effectively track and update individual list items. React components follow a lifecycle, with distinct phases like mounting, updating, and unmounting, each triggering specific methods. React's event handling follows a similar pattern to traditional JavaScript but uses camel-case event names and directly passes event objects to handlers. Forms in React differ slightly. Instead of relying on element-specific state, React components manage form data within their state objects. The text also mentions that understanding the "component life cycle" is essential for every React developer. However, it doesn't provide further details on the component life cycle. Mounting phase: When a component is first added to the DOM. Updating phase: When the component's state changes and needs to be updated.
```

```
→ ↗ # Encode the summaries into sentence embeddings
    lamma_embedding = model.encode([lamma_summary], convert_to_tensor=True)
    gemini_embedding = model.encode([gemini_summary], convert_to_tensor=True)
    # Calculate the cosine similarity between the embeddings
    cosine_sim = cosine_similarity([lamma_embedding, gemini_embedding])
    # Output the similarity score
    print(f"Cosine Similarity: {cosine_sim[0][0]:.4f}")

→ Cosine Similarity: 0.8244
```

The comparison between the **Gemini** and **Llama** models, based on output similarity, yields a cosine similarity of **82%**, indicating a moderate level of alignment in their results. This reflects that while both models exhibit notable overlap in understanding and processing language, there are also distinct differences in their performance, particularly in how they handle specific input of a bag of words.

# **Chapter 6: Plan of action for the next semester**

## **6.1 . Work done till date**

**Chatbot Development:** A functional chatbot has been created that answers user questions about YouTube videos using a large language model (e.g., LLaMA).

**Project Planning:** Initial planning is complete, including defining project scope and gathering user requirements.

**System Design:** Preliminary designs for the system architecture and user interface wireframes have been developed. The database schema is currently in progress.

**Backend Setup:** The backend environment has been established using Flask/Django, with initial integrations of the YouTube Data API and work started on the transcript generation module.

**Frontend Planning:** The design for the user interface is being outlined, with initial preparations for the creator dashboard using a frontend framework (React.js or Angular).

## **6.2. Plan of action for Project II**

### **Develop Frontend Dashboards**

- Design the creator dashboard to display summaries, transcript data, and content suggestions based on user engagement.

### **Integrate Sentiment Analysis for Comments**

- Implement comment retrieval from the YouTube Data API.
- Use sentiment analysis tools (TextBlob or VADER) to analyze viewer comments, generating insights into audience perception.

### **Develop Content Suggestion Logic**

- Create algorithms that provide personalized content suggestions for creators based on viewer engagement and sentiment analysis.
- Integrate this feature into the creator dashboard for easy access.

## **Chapter 7: Conclusion**

In conclusion, ContentConcise is a transformative AI-powered web application that significantly enhances the learning experience for students with slower learning abilities. By summarizing YouTube videos, translating these summaries into multiple languages, and offering a voice-command query system, the platform makes essential educational content more accessible and easier to understand.

The application uses advanced natural language processing (NLP) to create concise summaries, helping students grasp key concepts without feeling overwhelmed. The translation feature broadens the platform's reach, ensuring that language barriers do not hinder learning. Additionally, the voice-command functionality allows for a more personalized and interactive learning experience.

To evaluate the platform's effectiveness, we will use metrics such as user satisfaction, learning outcomes, engagement rates, and technical performance. A robust feedback loop will facilitate continuous assessment and adaptation based on user needs, ensuring that ContentConcise effectively addresses accessibility gaps.

By providing equal learning opportunities for all students, ContentConcise contributes to their academic success and personal growth. Through ongoing improvements and innovations, the platform aims to redefine educational accessibility and inclusivity, fostering an environment where every learner can thrive.

## Chapter 8: References

- [1] Otani, M., Nakashima, Y., Rahtu, E., Heikkilä, J., and Yokoya, N. Video Summarization using Deep Semantic Features. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20–24, 2016, Revised Selected Papers, Part V 13*, Springer International Publishing, pp. 361–377, 2017.
- [2] Apostolidis, E., Adamantidou, E., Metsai, A.I., Mezaris, V., and Patras, I. Video Summarization using Deep Neural Networks: A Survey. *Proceedings of the IEEE*, vol. 109, no. 11, pp. 1838–1863, 2021.
- [3] Zhang, S., Zhu, Y., and Roy-Chowdhury, A.K. Context-Aware Surveillance Video Summarization. *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5469–5478, 2016.
- [4] Kwon, J. and Lee, K.M. A Unified Framework for Event Summarization and Rare Event Detection from Multiple Views. *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1737–1750, 2014.
- [5] Sridevi, M. and Kharde, M. Video Summarization using Highlight Detection and Pairwise Deep Ranking Model. *Procedia Computer Science*, vol. 167, pp. 1839–1848, 2020.
- [6] Varini, P., Serra, G., and Cucchiara, R. Personalized Egocentric Video Summarization of Cultural Tour on User Preferences Input. *IEEE Transactions on Multimedia*, vol. 19, no. 12, pp. 2832–2845, 2017.
- [7] Ji, Z., Xiong, K., Pang, Y., and Li, X. Video Summarization with Attention-Based Encoder–Decoder Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1709–1717, 2019.
- [8] Fajtl, J., Sokeh, H.S., Argyriou, V., Monekosso, D., and Remagnino, P. Summarizing Videos with Attention. In *Computer Vision–ACCV 2018 Workshops: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers 14*, Springer International Publishing, pp. 39–54, 2019.
- [9] Gupta, H. and Patel, M. Method of Text Summarization using LSA and Sentence Based Topic Modelling with Bert. In *2021 international conference on artificial intelligence and smart systems (ICAIS)*, IEEE, pp. 511–517, 2021.
- [10] Jugran, S., Kumar, A., Tyagi, B.S., and Anand, V. Extractive Automatic Text Summarization using SpaCy in Python & NLP. In *2021 International conference on advance computing and innovative technologies in engineering (ICACITE)* IEEE, pp. 582–585, 2021.
- [11] Adhikari, S. Nlp Based Machine Learning Approaches for Text Summarization. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE, pp. 535–538, 2020.
- [12] Madhuri, J.N. and Kumar, R.G. Extractive Text Summarization using Sentence Ranking. In *2019 international conference on data science and communication (IconDSC)*, IEEE, pp. 1–3, 2019.
- [13] Merchant, K. and Pande, Y. Nlp Based Latent Semantic Analysis for Legal Text Summarization. In *2018 international conference on advances in computing, communications and informatics (ICACCI)*, IEEE, pp. 1803–1807, 2018.
- [14] Ngo, C.W., Ma, Y.F., and Zhang, H.J. Video Summarization and Scene Detection by Graph Modeling. *IEEE Transactions on circuits and systems for video technology*, vol. 15, no. 2, pp. 296–305, 2005.
- [15] Gygli, M., Grabner, H., Riemenschneider, H., and Van Gool, L. Creating Summaries from User Videos. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part VII 13*, Springer International Publishing, pp. 505–520, 2014.
- [16] Dilawari, A. and Khan, M.U.G. ASoVS: Abstractive Summarization of Video Sequences. *IEEE Access*, vol. 7, pp. 29253–29263, 2019.

- [17] Smaïli, K., Fohr, D., González-Gallardo, C.E., Grega, M., Janowski, L., Jouvet, D., Komorowski, A., Koźbiał, A., Langlois, D., Leszczuk, M. and Mella, O. A First Summarization System of a Video in a Target Language. In *Multimedia and Network Information Systems: Proceedings of the 11th International Conference MISSI 2018* 11, Springer International Publishing, pp. 77-88, 2019
- [18] Jaiswal, S. and Misra, M. Automatic Indexing of Lecture Videos using Syntactic Similarity Measures. In *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE, pp. 164-169, 2018.
- [19] Choudhary, P., Munukutla, S.P., Rajesh, K.S., and Shukla, A.S. Real Time Video Summarization on Mobile Platform. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, pp. 1045-1050, 2017.
- [20] Kannan, R., Ghinea, G., Swaminathan, S., and Kannaiyan, S. Improving Video Summarization Based on User Preferences. In *2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, IEEE, pp. 1-4, 2013.
- [21] Basak, J., Luthra, V., and Chaudhury, S. Video Summarization with Supervised Learning. In *2008 19th International Conference on Pattern Recognition*, IEEE, pp. 1-4, 2008.
- [22] Huang, J.H., Murn, L., Mrak, M., and Worring, M. Gpt2mvs: Generative Pre-Trained Transformer-2 for Multi-Modal Video Summarization. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, pp. 580-589, 2021.
- [23] Narasimhan, M., Rohrbach, A., and Darrell, T. Clip-It! Language-Guided Video Summarization. *Advances in Neural Information Processing Systems*, vol. 34, pp. 13988-14000, 2021.
- [24] Huang, J.H. and Worring, M. Query-Controllable Video Summarization. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pp. 242-250, 2020.
- [25] Xiao, S., Zhao, Z., Zhang, Z., Guan, Z., and Cai, D. Query-Biased Self-Attentive Network for Query-Focused Video Summarization. *IEEE Transactions on Image Processing*, vol. 29, pp. 5889-5899, 2020.
- [26] Nalla, S., Agrawal, M., Kaushal, V., Ramakrishnan, G., and Iyer, R. Watch Hours in Minutes: Summarizing Videos with User Intent. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, Springer International Publishing, pp. 714-730, 2020.
- [27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I. Attention Is All You Need. *Advances in neural information processing systems*, vol. 30, 2017.
- [28] Jiang, P. and Han, Y. Hierarchical Variational Network for User-Diversified & Query-Focused Video Summarization. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pp. 202-206, 2019.
- [29] Vasudevan, A.B., Gygli, M., Volokitin, A., and Van Gool, L. Query-Adaptive Video Summarization via Quality-Aware Relevance Estimation. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 582-590, 2017.
- [30] Gygli, M., Grabner, H., and Van Gool, L. Video Summarization by Learning Submodular Mixtures of Objectives. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3090-3098, 2015.
- [31] Sharghi, A., Laurel, J.S., and Gong, B. Query-Focused Video Summarization: Dataset, Evaluation, and a Memory Network Based Approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4788-4797, 2017.
- [32] Sharghi, A., Gong, B. and Shah, M. Query-Focused Extractive Video Summarization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII 14*, Springer International Publishing, pp. 3-19, 2016.

- [33] Zhang, Y., Kampffmeyer, M., Liang, X., Tan, M., and Xing, E.P. Query-Conditioned Three-Player Adversarial Network for Video Summarization. *arXiv preprint arXiv:1807.06677*, 2018.
- [34] Zhang, Y., Kampffmeyer, M., Zhao, X., and Tan, M. Deep Reinforcement Learning for Query-Conditioned Video Summarization. *Applied Sciences*, vol. 9, no. 4, pp. 750, 2019.
- [35] Sreeja, M.U. and Kovoov, B.C. A Unified Model for Egocentric Video Summarization: An Instance-Based Approach. *Computers & Electrical Engineering*, vol. 92, pp. 107161, 2021.
- [36] Ahmed, S.A., Dogra, D.P., Kar, S., Patnaik, R., Lee, S.C., Choi, H., Nam, G.P., and Kim, I.J. Query-Based Video Synopsis for Intelligent Traffic Monitoring Applications. *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3457-3468, 2019.
- [37] Gao, J., Yang, X., Zhang, Y., and Xu, C. Unsupervised Video Summarization via Relation-Aware Assignment Learning. *IEEE Transactions on Multimedia*, vol. 23, pp. 3203-3214, 2020.
- [38] De Avila, S.E.F., Lopes, A.P.B., da Luz Jr, A., and de Albuquerque Araújo, A. VSUMM: A Mechanism Designed to Produce Static Video Summaries and a Novel Evaluation Method. *Pattern recognition letters*, vol. 32, no. 1, pp. 56-68, 2011.
- [39] Zhang, K., Chao, W.L., Sha, F., and Grauman, K. Video Summarization with Long Short-Term Memory. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, Springer International Publishing, pp. 766-782, 2016.
- [40] Mahasseni, B., Lam, M., and Todorovic, S. Unsupervised Video Summarization with Adversarial LSTM Networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 202-211, 2017.
- [41] Rochan, M., Ye, L., and Wang, Y. Video Summarization using Fully Convolutional Sequence Networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 347-363, 2018.
- [42] Zhou, K., Qiao, Y., and Xiang, T. Deep Reinforcement Learning for Unsupervised Video Summarization with Diversity- Representativeness Reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [43] Ul Haq, H.B., Asif, M., Ahmad, M.B., Ashraf, R., and Mahmood, T. An Effective Video Summarization Framework Based on the Object of Interest using Deep Learning. *Mathematical Problems in Engineering*, vol. 2022, 2022.

**Industry / Inhouse:  
Research / Innovation:**

**Project Evaluation Sheet 2024-25**

**Draft** **(27)**  
**Class: BH2A/B/C**

Title of Project (Group no): Content Concise: YouTube Content Summarization and comment analysis

Mentor Name & Group Members: Aman Kumar (D17A) Anchal Sharma (D17A 57) Marsh Tuli (D17A 65) JAY Thakker (D17A 63)  
@men Anchal Sharma Marsh Tuli JAY Thakker D17A 63

Review of Project Stage 1	Comments:	Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg & Mgmt principles	Life-long learning	Professional Skills	Innovative Approach	Total Marks
1	Explore more tools available in same domain.	(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)

**45**

Explore more tools available in same domain.

Kunal D.  
Name & Signature Reviewer1

Review of Project Stage 1	Comments:	Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg & Mgmt principles	Life-long learning	Professional Skills	Innovative Approach	Total Marks
04	1. Explore more AI tools in same domain & define scope accordingly.	(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)

Date: 23/08/2024

Pallavi Saini  
Name & Signature Reviewer2

Pallavi Saini  
Name & Signature Reviewer1

### Industry/Inhouse:

Title of Project(Group no): Content concise : YouTube content Summarization and comment analysis  
 Group Members: Archal Sharma (57), Harsh Tuli (65), Aman Kuma (33), Jay Thakker(63)

### Project Evaluation Sheet 2024-25

Class: D17A

(27)

	Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg &Mgmt principles	Life-long learning	Professional Skills	Innovative Appr oach	Total Marks
Review of Project Stage 1	(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
Comments:	4	4	4	3	4	2	2	2	2	2	2	2	3	4	42

Comments: Content Summarized size, visualization of contents, can be decided.

Fewer D. soot

	Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg &Mgmt principles	Life-long learning	Professional Skills	Innovative Appr oach	Total Marks
Review of Project Stage 1	(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
Comments:	4	4	4	3	4	2	2	2	2	2	2	2	3	4	42

Date: 26th September, 2024

Name & Signature Reviewer2

Pallavi Sainane