E-LECTURES SUMMARISER

Submitted in partial fulfilment of the requirements
of the degree of

Bachelor of Engineering

by

Akanksha Mansharamani	60003180002
Manav Shah	60003180025
Varun Vora	60003180058

Project Guide:
Dr. Abhijit Joshi

(Information Technology)
Dwarkadas J. Sanghvi College of Engineering
University of Mumbai

2021-2022

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

CERTIFICATE

This is to certify that the project entitled **"E-Lectures Summariser"** is a bonafide work of **"Akanksha Mansharamani" (60003180002), "Manav Shah" (60003180025) and "Varun Vora" (60003180058)** submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Information Technology".**

Dr. Abhijit Joshi
Project Guide

Dr. Vinaya Sawant
Head of Department

Dr. Hari Vasudevan
Principal

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Project Report Approval for B.E.

This project report entitled ***E-Lectures Summariser*** by ***Akanksha Mansharamani, Manav Shah and Varun Vora*** is approved for the degree of Information Technology.

Examiners 1. ---------------------

2. ---------------------

Date:

Place:

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

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 has not been taken when needed.

Akanksha Mansharamani
(60003180002)

Manav Shah
(60003180025)

Varun Vora
(60003180058)

Date:

Acknowledgement

The successful completion of this project would not have been possible without the help and guidance of many respectful individuals. It gives us great pleasure to express our sincere thanks and gratitude to all of them. We are extremely grateful to our project in-charges, Dr. Abhijit R Joshi for his constant support and encouragement throughout the course of this project. In addition, we are extremely thankful to our respected Principal, Dr. Hari Vasudevan, for giving us the support and guidance to work on this project. Our thanks and appreciation go to the college and staff for providing us with the necessary resources and valuable suggestions. Our classmates and team members have given valuable inputs in this proposal which further gave us the inspiration to improve our project. We thank them all for their help to complete our project. We express our sincere heartfelt gratitude to each of the individuals who helped us in successfully developing our project within the prescribed time.

Akanksha Mansharamani

Manav Shah

Varun Vora

Abstract

In these troubled times of pandemic, the teaching-learning process has been affected adversely at every level of education. Schools and colleges had to shift and adapt to various online platforms for achieving academic goals set by respective universities and boards and to emulate the past trends of the academic year and the overall curriculum. The whole process of conducting lectures, practical sessions and workshops has been changed and this change has its ups and downs.

The students and teachers have been familiarized with computing applications and their functionalities, making them more equipped for the era of computers. Also, for students and teachers who had to commute to the institution on a daily basis have the advantage of eliminating travelling time from their daily routine. Some cons of this online process are, lectures can be long, boring and monotonous. With the absence of a classroom environment, personal touch is lost as a result of which students have a shorter attention span than usual. Lectures recorded can be viewed later. If a student misses on a lecture or wishes to revise a concept from the recordings he/she must view the entire recording of the lecture. Moreover, the student must put in extra time and efforts into compiling notes. This is not always feasible especially when the student is on a time crunch.

Thus, a system that should aid the students in studying for their exams and utilizing studying time with efficiency seems to be something that would be welcomed by open arms from students as well as the teaching community. This aid can be acquired by using AI methodologies to generate summaries of lectures in the form of concise lecture notes and time-indexed videos according to subtopics.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the first chapter, the motivation behind the selected topic is illustrated. The major expected challenges that will be faced during the course of implementing the system are described. This is followed by the report overview demonstrating the flow of the remaining sections.

1.1 Motivation / Objective

The ongoing pandemic took a huge hit on the learning system as we know it. As a result, student interactions in the online classroom have declined. A UK Engagement Survey finds less than two-thirds of undergraduates feel universities ensured quality of academic experience during Covid-19. Many students struggle to keep up with college classes and have lost motivation to study. Lack of social interaction has further alleviated the confidence of learners. Via our system, we aim to eliminate all the above-mentioned problems. The proposed model will not only increase learner satisfaction but also boost the confidence of students lagging academically and help them catch up with their peers. The use of visual variety has proved to be aesthetically appealing. Thus, we plan on leveraging this to our advantage to grab the learner's attention. Our proposed system not only makes the learner process easier, but also helps students save onto study time. Students can easily view summaries before their exams instead of having to go through the trouble of frantically searching through study material and resources. This in turn, will boost academic grades of the students. It will also make learning a fun process. Using this application, students will be much more interested in studying rather than reading long paragraphs of black and white monotonous texts. Moreover, since the application can also use YouTube videos to generate a summary, students are not limited to the video recordings of the lecture. They can also use other reference material videos in the E-Lectures Summariser.

1.2 Major Challenges

Currently, a major challenge we may face is making our model comprehensive for all English-speaking regions across the country. The language dialect adopted by different people comes with its own set of difficulties. For example, a professor from South India will have a varied English accent as opposed to a professor conducting a lecture in a British or an American English accent (neutral accent).

Another major issue is identifying lecture content and separating it from parts of the video recording that may not be relevant to learning. Sometimes, during class, a teacher might give the students a

break or engage in a conversation not pertaining to the subject at hand. Isolating such instances and not including them in the summary is essential.
The choice of subjects used for the system also proves to be difficult. The indexing module identifies keyframes based on the snapshot of whiteboard content/ presentation used by the professor for teaching the students. However, in some subjects like mathematics that involve a lot of mathematical equations, it is challenging to extract important words which can be used as a bookmark. This also proves to be challenging even when a lecture recording consists of no presentation / board content.
For any E-learning system, there are also typical challenges like content design, assignment, and delivery, which also apply to our task.

1.3 Report Overview

The report consists of the objective of the proposed system and major challenges that must be overcome in the system. There is also an abstract of the project which explains the project.

Chapter 2 consists of the literature review done for the proposed system. It consists of the research done for the project. It provides the research papers which were referred for the project and explains the most important of them based on the algorithms, technology used, methods, etc. The section compares the difference between the algorithms used previously and analyses each algorithm used.

Chapter 3 explains the proposed methodology. It provides the method used for the implementing the proposed system. This section consists of the definition of the problem and the scope of the project. The scope of the project includes the assumptions made and constraints in the proposed system. It also includes the features of the proposed system and our proposed system architecture.

Chapter 4 explains the Project Management of the proposed solution. It consists of the Project Schedule, which includes the Task Network Diagram and the Timeline Chart of the project. It evaluates the feasibility of the project, viz, technical, operational and economic feasibility. Hardware, software and operational project requirements are listed. We performed the project estimation using the COCOMO model and functional point analysis. The chapter ended by stating the risk management mitigation planning.

Chapter 5 consists of the different design elements of the proposed system. It provides Data Flow and various UML diagrams to explain various aspects of the system design in detail. Further, it contains the Proposed System Architecture and the Description of Datasets used in this project.

Chapter 6 comprises of the implementation. This chapter explains how the rudimentary modules in the system work together in harmony. It mentions various algorithms and tools used in development of this system. Finally, it depicts the look and feel of the User Interface and the different paths a User can follow in the system.

Chapter 7 tells us about all the different kind of tests and experiments performed on the system and its various modules. It puts forth the results of these tests and experiments. Following that, Chapter 8 concludes the Project report with concrete inferences and scope for future development is mentioned.

Chapter 2

Literature Review

Currently, there has been extensive research to develop research-based products that are either unique or improve upon the existing systems in the field of educational technology. The quality in EdTech research has not been consistent. Even though the amount of research being done is considerable, the impact and quality of educational research has not been up to the mark. There is a lot of research that is being done but while doing that, the impact that this research will bring upon the world of EdTech is being overlooked and not considered. In a stagnating education system, challenging the status quo of teaching is tough. However, impact research can give the crucial support and assurance that instructors require to implement new techniques.
The following section talks about the existing work that has been currently done and the methodology behind different systems. The tools and framework that are currently in use have also been mentioned. In addition to that, observations on existing work have also been delineated which also includes what the current technologies lack.

2.1 Existing Work

This section is further divided into four sections. The first section talks about the end-to-end systems that currently exist in the market. The next section talks about the research done about systems and approaches which are like our proposed system. The last two sections talk about what algorithms, tools and framework are currently being used to develop these systems.

2.1.1 Existing Systems

Ivory Research:
Figure 2.1 shows a snapshot of the IvoryResearch Service

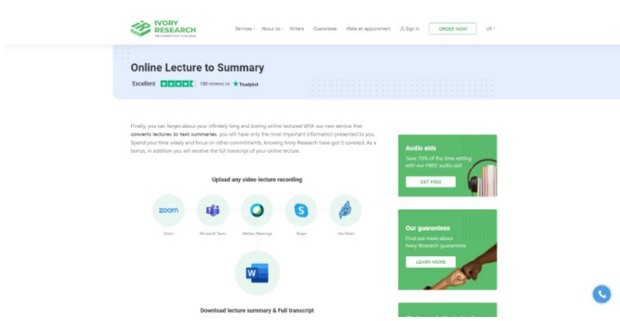

Figure 2.1: Ivory Research

Ivory Research is a service that provides a plethora of services to students that helps ease student lives and promotes academic excellence. One of the paid services they provide is a lecture to summary service as well as provide a transcript for the entire lecture.
In Ivory Research, the students are required to upload recorded lectures from different sources like Zoom, MS Teams etc. They then have to request for a order for the video uploaded and ask for a quotation for the required service. Based on the input lecture, the ivory research team provides the user with a quotation and the user is asked to pay the quoted amount to avail the team's services. Ivory research has dedicated writers for these tasks who get to work immediately and provide you with results in a few days.

Coursera:
In Coursera they have handwritten transcripts of each and every lecture with manually indexed videos. As you can see in Figure 2.2, they have a reading subsection which has the screenshots and explanation of the important topics covered in the lecture.

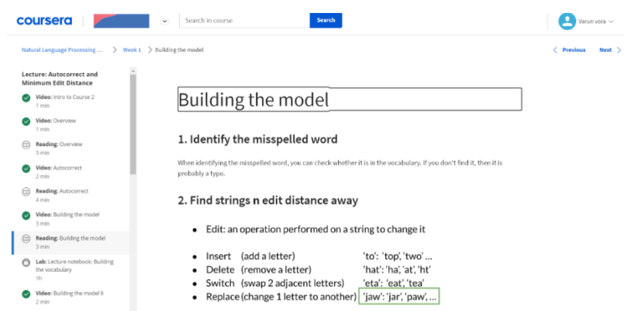

Figure 2.2: Coursera Lecture Summary

In Figure 2.3, we can also see how the transcripts can be downloaded in a language of the user's choice. They are available just below the respective video and also contain pointers to that specific part of the video. Everything in coursera is done manually by students and staff at Coursera and hence is not available for every course in the catalog.

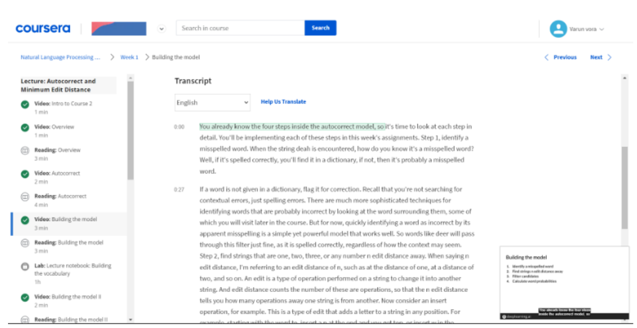

Figure 2.3: Transcripts and Video Pointers

MakeMySummary:

A mobile app with paid services to summarize items from YouTube, Blogs, Podcasts, Twitter, and Apps. It is integrated with various video processing and streaming applications from where it takes a video file as an input and generates a textual summary or a summarized clip of the video. It is not open source and the techniques the AI it claims to use for generating these summaries are not revealed.

In addition to an application, they also have browser extension capabilities that allow the user to summarize the contents on the webpage the extension is called into use at. It also allows the user to bookmark and archive important parts of the summary. Their interface (as shown in Figure 2.4) is set up in such a way that the user can seamlessly divide and navigate work into three lists namely: Archived, Important, and Starred.

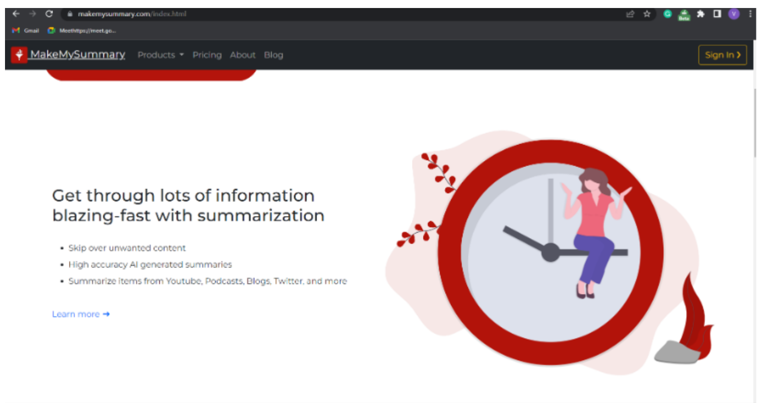

Figure 2.4: MakeMySummary Interface

IntelliVision:

Figure 2.5 shows a snapshot of the IntelliVision Interface.

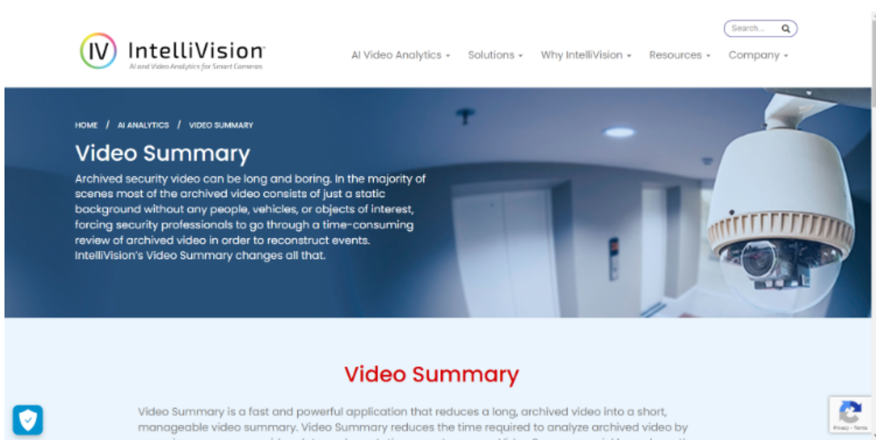

Figure 2.5: Intellivision Interface

By analysing video information, extracting metadata, sending out real-time warnings, and delivering insight to security people and other security systems, IntelliVision's AI and Deep Learning video analytics technologies bring intelligence to video cameras. They provide multiple services one of which is generating a video summary.
Their output is aimed to provide security and catch any security breaches that have occurred in the video. They isolate events that they consider high risk and point it out to the security officials concerned. They also provide real-time video summaries from live video feeds by quickly removing the unwanted noise from the video efficiently and identifying high risk events. They however are not useful when it comes to making me.

2.1.2 Literature Related to Methodology / Approaches

In [1], subtitles mentioned in video lectures are used to perform extractive summarization. The summary produced, finds its application in NPTEL videos. In the first step, the user gathers the file from the internet, carries out data-preprocessing and filters out the features. Some features used to generate the summary include content word feature, sentence location feature, and sentence-to sentence cohesion.
The TF-IDF feature,used to generate a summary, is based on the principle that the importance of a word increases when the frequency of the word in the document increases but it decreases when the frequency of the word in the entire corpus increases. TF-IDF is used in search engines to rank documents based on the query generated by the user. Here, TF-IDF is used on a single-document as opposed to the usual multi-documents. However, every single sentence is treated as an individual document for accurate results. To determine the TF-IDF weight of the sentences a sum is taken of the weights of all the unique terms in the document. This procedure is repeated for all the sentences in the document. Since all the sentences do not become a part of the summary, we carry out a normalization process in which we divide the weight of the sentence by the number of unique terms present in it.
The mean weight of the entire text is used to determine the predefined threshold. All the sentences whose weight is higher than the mean weight are selected to be a part of the summary and are extracted from the source document. The proposed system in [1] produces a summary which is around 45% of the original text and is proportion to the size of the document. This system finds its application in summary of educational videos available on NPTEL, Udemy and Coursera.

AccessMath [2] system, like its name suggests, works with math video lectures. Its goal is to aid visually impaired students learn better. There are videos of the whiteboard data from two separate sources for each lecture. An application for extracting and retrieving that content is demonstrated. After the content has been indexed, the user can utilise a piece of the whiteboard content in a video frame as a query to find video segments with similar content. The recall of matched graph edges between the query and candidate graphs is used to order the query results, and graphs of surrounding related components are used to characterise both the query and candidate regions. This is a sketch-based picture retrieval method that does not require recognition.
The output of the retrieval process gives a set of frames that are ranked according to their order of relevance. For this we require an automated process to index the video content as well as identify the similarity between different regions of the slides. Sketch-based image retrieval system (SBIR) since the mathematical formulas are treated as sketches. Both local and global features are used to equate a query and the content in the index. Locally, OCR is used to find similarity between the connected components. On a global scale, similarity is found out using recall of graph edges between the candidate regions and the GNCC of the query.

Despite the presence of several MOOCs such as YouTube and Coursera, most educational videos are too long and lack detailed annotations. This makes it challenging for the students to find content of their interests. To solve this issue, [3] has introduced a note generating method that implements semantic relationships to find correlations between the whiteboard/slide-content and the corresponding speech texts. After obtaining the visual entities from the slide-content, word mover's distance (WMD) algorithm is used to measure semantic similarity between these obtained entities and the descriptive speech text, which cannot be dealt with using time-align technology. Lastly, note-like blocks are generated depending upon the importance of the visual entities. Generation of highly structured key-points and highlighted structured notes is highly efficient for students to get an overview of the topic.

The innovative tool in [4] takes the input from a large number of students. The participants provide the start and end time duration of subtopics inside the recordings of the video. The system breaks down the videos based on these timestamps obtained from crowdsourcing. It takes approximately 17 seconds to process a one hour video. The tool also provides a mechanism for video summarization in which the audio transcripts are summarized and matched back to the resultant parts of the video. This helps us save onto storage and also reduces the length of the video by almost half.

2.1.3 Literature Related to Algorithms

The following are different algorithms used for pre-processing of textual data and summarization of content in the video lectures.

Data Preprocessing Techniques:

Lexical analysis of the text and cleaning the text. This includes the following:

- Removing digits and frame numbers
- Special character removal
- Getting rid of stop-words
- Newline Characters removal
- Lowercasing of the document

Stemming:
It is a process of extracting the root word by either removing the suffix or the prefix of the word. Different techniques employed for stemming include affix removal, lookup table and making n-grams. Stemming may or may not result in actual words.

Lemmatization:
Similar to stemming, it provides the root word. However it ensures the word is valid and belongs to the language. We add part-of-speech parameter to provide context for lemmatization. In NLTK, WordNetLemmatizer is used to get the lemma of words.

Summarization Techniques:

TF-IDF Approach:
Term frequency (TF): The weight of a term is done by considering its frequency of occurrence in that term. A term that appears in a document more number of times is likely to be more relevant to the contents of the document than a term that is present less number of times. The formula for deriving the weight in terms of term frequency is defined in Figure 2.6:

$$W_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d} & \text{if } tf_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

Figure 2.6: Weight of TF

Inverse Document Frequency (IDF): This term was proposed by Salton as the process of information retrieval began advancing. TF does not take into account that irrelevant terms that occur many times in a given document will be counted and won't produce an effective summary. Eg- Stop words have a high TF, but do not provide any contribution to the relevance of content present in the text. IDF is defined in Figure 2.7 as follows:

$$\text{IDF}(t) = \log_{10}(\text{Total no of documents / No of documents with term } t \text{ in it})$$

Figure 2.7: IDF

As shown in Figure 2.8, for normalization, the weight of the sentence is divided by the number of index terms present in that given text document.

$$\text{Norm(tf-Idf)} = \text{Sum of tf-idf score of all terms present in sentence/no of terms present in sentence.}$$

Figure 2.8: Normalization of IDF

As it is shown in Figure 2.9, the mean weight of the entire text is used to determine the predefined threshold. All the sentences whose weight is higher than the mean weight are selected to be a part of the summary and are extracted from the source document.

$$\text{A sentence is selected if : tf-idf}(sent) \geq \text{Avg(tf - idf)}$$

Figure 2.9: Sentence Selection Algorithm

Text Rank Algorithm:

TextRank is based on the principle that words that occur more often, are more significant than those that don't . This makes sentences with frequent words more important. The extractive summary generation algorithm associates each sentence with a score and the top-ranked sentences are chosen as a part of the summary.

TextRank Model: The model in TextRank is based on voting implemented by a graph-based ranking model. A vertex casts a vote for another vertex by linking to it.

Text as a graph: The text is represented as a graph displaying the interconnected words and textual entities along with the relationships between them. They are used to determine the significance of a vertex in a graph, by drawing global information recursively from the entire graph.

Text Rank includes the following Natural language processing tasks:
i. Keyword Extraction - The task is to identify a set of index terms that can be used to best identify a document. Using frequency as a criteria leads to poor results. This algorithm does not require any supervision.
ii. Sentence Extraction - Here we build a graph associated with a text. A vertex is added to the graph for each sentence present in the document. These vertices (sentences) are the entities that need to be assigned a rank. Text rank works well for applications having entire-sentences.

RNN Encoder-Decoder Summarization:
RNN encoder-decoder (Figure 2.10) is a deep learning architecture that is used to process sequential data. It has several hidden layers. In RNN, the input of a the current step depends on the output of the previous state. Thus, the last state of the layer consists of the whole input which was accumulated over all the previous layers.

(Image source: https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-text-summarization-using-deep-learning-python/)

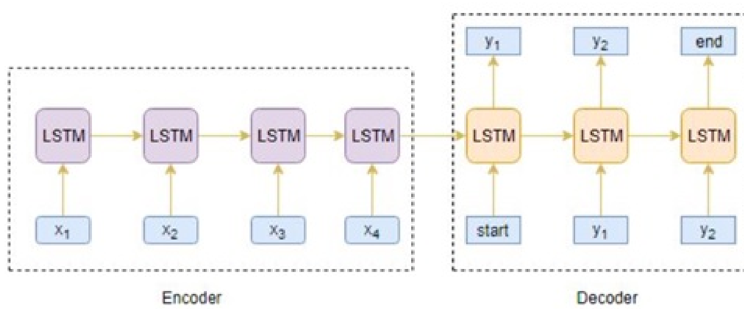

Figure 2.10: RNN Encoder-Decoder

RNN encoder-decoder is derived from the seq-to-seq model that maps the input sequence to a similar sequence. It finds its use in modern NLP applications such as text summarization. The text that is to be summarized is provided as the input and produces the actual summary as output. The encoder combines the current input vector with the output of previous hidden states to feed to the next hidden state. The output of the last hidden state of the encoder is given to the decoder in a vector format which is called the context vector. A distributed representation form of the decoder output is sent to the softmax layer and attention mechanism before generating the summary.

Long Short-Term Memory (LSTM):

As shown in Figure 2.11, an LSTM cell consists of four gates namely, Input, Forget, Memory Output gates which share information amongst themselves, forming a loop for information sharing. Its chaining structure is similar to that of RNN.

(Image source: https://bhrnjica.net/2019/04/08/in-depth-lstm-implementation-using-cntk-on-net-platform/)

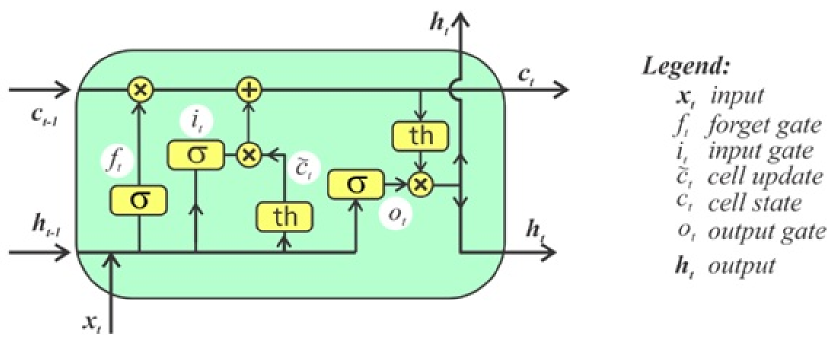

Figure 2.11: LSTM

1. The "Input Gate" takes a randomly initialized vector in the first timestep. Following that, the input of the current state is the output of the previous step. In every step, the input and output of the forget gate go under element-wise multiplication, which is stored as output of the current memory gate.
2. The inputs for the "Forget Gate" serve as the output for the previous block, while retaining the bias and the input vector information from the previous block. This gate uses the sigmoid activation function for its neural network like structure.
3. The "Memory Gate" controls the effect of old information on the new one. New information is obtained by element-wise multiplication of the old information and the output of two neural networks of the memory gate. The first neural network resembles the forget gate with a different bias.The second neural network uses a tanh activation function.
4. "Output Gate" decides how much information should be sent to the next LSTM unit. The final output is produced by passing the input vector, bias, previous hidden states and the new information to a sigmoid function and multiplying it by the tanh of new information.

2.1.4 Literature Related to Technology / Tools / Frameworks

Some technology and tools that have been used in the literature relevant to our problem definition are:

I. Speech-to-Text:

- ASR (Automatic Speech Recognition) [8]: ASR is a voice-to-computer interface technology that allows people to communicate with computers using their voices in a way that resembles typical human conversation in its most advanced forms. Natural language processing, or NLP, lies at the heart of the most advanced ASR systems currently available. Though AI has a long way to go before reaching its pinnacle of development, we're already seeing some spectacular results in the form of intelligent smart phone interfaces like the iPhone's Siri and other systems used in commercial and advanced technological contexts.
- Mel Frequency Cepstral Coefficients (MFCC) [9]: The first step in all ASR systems is to extract features and identify signal components that may be used to recognise linguistic content and discarding background noise, emotion, etc. The most crucial thing to understand about speech is that the vocal tract, which includes the tongue and teeth, filters human sounds. The sound that emerges is determined by the shape. If we can properly determine the shape, we should be able to accurately represent the phoneme being created. The short-time power spectrum envelope reflects the shape of the vocal tract, and MFCCs' job is to accurately represent this envelope. Davis and Mermelstein first presented MFCCs in the 1980s, after which they retain cutting-edge status over LPCCs which were most common feature types for ASR.
- Linear Prediction Cepstral Coefficients (LPCC) [9]: These are derived from LP analysis or a filter bank approach. Speech systems based on these coefficients are able to achieve high levels of accuracy for speech recorded in clean environments. These features are directly extracted from spectra using the energy values of linearly organised filter banks that emphasise contribution of all frequency components of a voice signal equally and are hence consided phonetic information. For speech-to-text converison, LPCCs are used to capture emotion-specific data expressed through vocal tract properties. In this work, the voice signal was subjected to LP analysis of the 10th order to obtain 13 LPCCs per speech frame of 20ms with a frame shift of 10ms. Emotion recognition in humans is based on the speaker's expression and the listener's interpretation of the emotion. The purpose of using LPCCs is to automatically recognise emotions while taking into account the speaker's vocal tract properties.

II. Word Quantization models:

- Vector Quantization (VQ) [4]: Vector quantization (VQ) is a signal processing quantifying technique that uses the distribution of prototype vectors to simulate probability density functions . Having been built for Data Compression, it separates a large number of vectors into groups represented by a centroid point, like with k-means and other clustering approaches. The density matching in VQ is particularly beneficial for calculating the density of large high-dimensional data. The error is inversely realted to the frequency of datapoints. Hence, VQ is suitable for lossy data compression and can be used to estimate density correct lossy data. As a result, VQ can help us put the ASR transcripts into context.

- Dynamic Time Warping (DTW) [4]: Dynamic Time Warping is one of the most well-known distance measures between pairs of time series. The main idea behind DTW is to use comparable sections to compute the distance between two time series. It uses dynamic programming to match two time series segments. DTW could discover walking similarities between people walking at different pace, with many accelerations and decelerations during an observation. DTW has been used to analyse different multimedia in temporal sequences, and data transformable into a linear sequences. For dealing with varied speaking rates, automatic speech recognition is a well-known application.

III. Text Summarization:

- PEGASUS [5] Self-Supervised seq2seq model: PEGASUS implements the seq2seq architecture. This design is unique in that it has a pre-training goal that is self-supervised. It essentially eliminates the need for labelled samples in data and opens up a massive amount of previously unknown, unlabeled data for training. The main idea behind this goal is that the closer the self-supervised pre-training goal is to the final downstream work, the better the fine-tuning performance will be. In PEGASUS, complete sentences are removed from a text (i.e., ”masked”), and the model is trained to predict these sentences, as seen in the image. Even for humans, the authors admit that this task appears to be nearly impossible. When it comes to crafting meaningful phrases, however, such training leads to a higher sense of comprehension.

- TF-IDF approach [1]: TF-IDF stands for ”Term Frequency - Inverse Document Frequency.” This is a method for determining the total number of words in a set of documents. Each word is frequently given a score to represent its importance in the document and corpus. This method is widely used in the disciplines of information retrieval and text mining. After vectorizing the documents, we may do a variety of tasks, such as locating relevant documents, rating, clustering, and so on. You use the same strategy when you run a Google search (now they are updated to newer transformer techniques). A query is the search term you use to find something, and documents are the web pages you find. In the search, all documents are represented by a fixed representation.

- Probabilistic Latent Semantic Analysis (PLSA) [6]: PLSA is a statistical method for analysing data with two modes and co-occurrences, also known as probabilistic latent semantic indexing in the Information Retrieval field. Based on their affinity to specific hidden variables, the observed variables represented in low-dimensions using PLSA .
  PLSA is based on a mixture decomposition produced from a latent class model as opposed to the linear algebra based Normal Latent Semantic Analysis. In order to give readers a thorough perspective of the original content, human-written summaries tend to cover a wide range of topics. As a result, we can expect a conclusion.

IV. Feature/Entity Recognition:

- Clustering of Visual/Textual Entities [7]: Agglomerative Clustering of video lecture content from sentences spoken by teacher and images of whiteboard/presentation content based on Semantic Similarity.

2.2 Observations on Existing Work

Some observations from the Existing Systems, Tools, Methodology, Approaches and Literature are:

- In this system, the modules are made up of several moving parts that involve speech-to-text generation, extractive Text Summarisation, extraction of Whiteboard/Slideshow content, Optimal Character Recognition (OCR), and recognising key frames and entities from the visual content.
- The first step in our system to is to separate the audio signal from the lecture recording and generate a transcript of everything said by the professor in that lecture. The most renowned technique or tool for generating a speech signal from audio and converting it into text form is ASR (Automatic Speech Recognition). ASR has been under constant development in recent years and many IT giants have produced a tool of their own. Example: Google Cloud's Speech-to-Text API, IBM's Watson, Facebook's Wave2Vec.
- ASR is based on phonemes of a particular language which is the frequency and utterance of words. Current ASR models train on audio that has been processed to make vocal features stand out namely the log-mel filter bank features of speech.
- The log-mel filter bank is based on [MFCC] or [LPCC] that process the speech signal to filter certain acoustic sounds or syllables and then an analysis and processing of the filtered information is performed, and they are matched with pre-fed data that sound the same with greatest probability.
- This method has not been able to deliver the best results because of inconsistent data on phonemes, different accents over the world for the same language and it would be even more ineffective in our use case because of high frequency of technical terms that may not have been encountered by the model in the training phase.
- A novel approach is adopted by Facebook AI's wave2vec tool. It produces state-of-the art results by using a self-supervised NLP DNN that extracts vectors from the audio/speech of a constant length. For each vector, the original speech samples are taken and broken into fragments that are mismatched and the NN is tasked with recognizing the original sample. Thus, the model is trained to discern certain sounds by itself, sounds that correspond to specific features of speech and language. This can cater to novel terms in a particular kind/class of videos as it can learn to decipher the technical terms of that video set.

- ASR generates the speech delivered in the video into text. The next task is to make sense out of those words. For that purpose, the speech information is punctuated with SBD (Subsidiary Boundary Detection) to render it into chronological sentences.
- These sentences are then passed to word recognition and quantisation models like VQ, ANNs, DNNs, etc. to make sense out of each word/sentence and correct any discrepancies that are rendered due to faulty speech recognition or the inarticulate diction of the speaker. These models are based on the acoustic or lexical approaches mostly.
- After the video being processed through the previous stages, we can use this reliable textual data to generate summaries using the various Text Summarisation techniques
- One of the most used approaches is to extract important sentences based on the average TF-IDF score of the sentence. The TF-IDF score is calculated for each word in the sentence, and then these values are added together to get a total TF-IDF score for that sentence. However, since the length of the sentence could incorrectly increase the score, the average of these scores is taken for the word occurrences across multiple documents in the corpus.
- TF-IDF scoring works on the frequency of terms being used in multiple lectures and this may not always lead to an efficient summary. Repetition of irrelevant terms can ruin the importance of significant terms that may be used less often.
- An improvement over the TF-IDF scoring model is the TextRank algorithm that constructs a graph from words and sentences and starts ranking them on some keywords that are extracted prior to graph construction. Following this, sentences are ranked with the number of occurrences for these keywords.
- The TextRank algorithm is a great improvement over the TF-IDF scoring model, but it cannot adjust to any subject or a use case for that matter. To counter this problem, we rely on cutting edge AI.
- PEGASUS is a state-of-the-art technique proposed by Google AI. It implements the self-supervised seq2seq architecture whose novelty lies in its pre-training objective.
- It uses Gap Sentences Generation to let the system learn from itself and produce more efficient summaries by competing with itself.
- In PEGASUS, the encoder is pretrained as a masked language model to complement the GSG.

- All the above steps should lead to the generation of an effective summary from the words spoken by the professor.
- To extract the textual/visual content from the whiteboard or slideshow, using one of state-of-the-art OCR models (like Google Cloud Vision) or a custom-trained CNN is the way to go.
- To separate the content from different times in the video, Entity Recognition is a viable option, with the use of Agglomerative Clustering at its core. But for different subjects the entities will have to be pre-defined else it will cause inconsistency in different subjects and even subtopics.
- To counter this issue, we can approach it from a different perspective. We can extract key frames from the video using Syntactic Similarity algorithms and once we have snapshots of these key instances, recognise the content in those images and cross-reference it with the textual summary.
- The OASIS model (Online Algorithm for Scalable Image Similarity Learning) trains a bilinear similarity measure over sparse representations. In OASIS, a passive-aggressive learning algorithm is used with a high margin criterion and a low hinge loss cost. Studies demonstrate that OASIS outperforms existing state-of-the-art algorithms by an order of magnitude. Thus, we can use the OASIS model to extract keyframes in the video.

Chapter 3

Proposed Methodology / Approach

This chapter highlights the problem definition, the scope of the project, and the proposed system architecture of the model along with its features. The use case diagram is included to gain a better understanding of the interactions of the users with the system. The chapter is concluded with the activity diagram depicting the behaviour of the system.

3.1 Problem Definition

Online lectures can be long and monotonous. With the absence of the classroom environment, personal touch is lost as a result of which students have a shorter attention span than usual. Lectures recorded can be viewed later. If a student misses a lecture or must refer to some concept in any given lecture, he/she must view the entire recording of the lecture. Moreover he/she must put in extra time and effort into making notes. This is not always practical especially before exams when students are on a time crunch.
Through this research we plan on creating a system that is capable of producing:

Concise Notes:

- A summary of video lectures in the form of text and images.
- Notes will be generated using subtitle transcripts of the video and the whiteboard/slide content from the lecture.
- Transcripts with snapshots and links in the video

Indexed Video:

- Bookmarks or Timestamps for particular parts or topics in the video lecture.
- These is intended to be achieved through Video Indexing probably by Entity Recognition.
- The transcripts for the video lecture and the lecture clip itself will be equipped with links to skip to specific parts or topics in the video as per request or action of the user.
- Video-clips as queried by the user.

3.2 Scope

The project aims to ease online learning and make it more student friendly. We aim to develop a summarizer that provides a clear and concise summary of recorded lectures in the form of text and notes. The scope also includes timestamping the lectures according to the disparate topics taught and providing links to it in the textual summary and notes. Video Indexing by timestamps will help to optimize query processing and simplify search of content. We will be using lecture recordings available in English. These will be available to the users 24/7. The lectures will be chosen from a single educational domain initially which can later be scaled as per the requirements of other universities and schools depending upon the subjects taught there.

3.2.1 Assumptions and Constraints

Assumptions:
We are assuming that

- The lectures recorded are purely in English and not in Hinglish(mixture of Hindi and English) or any other regional languages.
- The dialect is clear and easily understandable.

Constraints

- The system only caters to English language.
- Use videos that are available with manually generated transcripts to compare the results of our work.
- Limited data sources

3.3 Proposed Approach to build E-Lectures Summariser

This approach towards the building of the system was that the problem statement was divided into smaller entities and a solution was to be found for each part. These entities were treated as seperate modules and were approached differently. The final system was then created after integrating this different modules into a single system. The following section shall comprise of the various features that were implemented in the proposed system, the various tools used for collecting the data required to build this system and the limitations of these tools dataset.

3.3.1 Features of the Proposed System

The features of our proposed system include:

1. To use the audio extracted from the recorded lectures to generate text summaries. Transcripts will be extracted using the subtitles generated. These transcripts will then undergo feature extraction to obtain the summary of the lecture in the form of text. The student can go through this to get an overview of the content taught by the tutor.
2. Retracting back to the lecture recording from the summarized text. In case a user is facing difficulties understanding the summary or wishes to view the detailed part of the recording , he/she can be directed to the desired part of the recording from that given part of the summary.
3. The video can be played on the system interface itself which has the video chapters embedded and the user is able to skip to those specific chapters which they want to focus on.

3.3.2 Tools used for data collection, size of the sample and limitations

We have used the following sources for Data, in our case the Video Lectures:

1. YouTube Videos
2. Coursera Videos
3. Udemy Videos

There were multiple tools that were used to scrap data from these sources, pre-process the data scraped and then API's were called to produce required results. The tools used were :

1. IBM Watson API : It was used to extract audio from the video and generate a transcript for the same.
2. Punctuator: It was used to punctuate the extracted data and divide them into sentences for extractive text summarisation

3. Google's Cloud Vision API: This was used to detect important content from slides used in the lecture.

Size of the Sample and Limitations:
Typically, we have used videos lectures that have a runtime of less than 15 minutes. We have not used videos with length greater than 15 minutes because, the GPU that we have been working with, cannot efficiently handle the humongous amount of frames present in longer videos.

3.4 Benefits of Proposed Solution

The proposed system is intended to provide the following benefits to students and teachers:

- One of the major difficulties students face during lectures is taking down notes and at the same time paying attention to the concepts being taught and understanding the gist of it at the same time. Our system gives them the liberty to stay focused on what is being taught and not utilise their time taking notes
- The lectures must be uploaded by the lecturer directly hence all students can directly access those lectures without having the need to upload them every time individually. This saves time and effort making the process more efficient.
- Addition of relevant whiteboard content to the summary adds a personal touch of the professor. The student can relate to the examples being taught in the lecture and get accustomed to the method used by the professor.
- A student may not always watch the entire video lecture again and again. He may want to skip to a topic he is struggling with, and our bookmarks feature allows him to do just that.
- He can either jump to a specific part of the lecture using the bookmarks generated by the system or he can search for a specific keyword and the system allows him to jump to that specific instance of the video where the keyword is mentioned the most.

Chapter 4

Project Management

This chapter consists of the project schedule, evaluation of feasibility, and resources required. It also explains the intricacies of project estimation and risk mitigation planning.

4.1 Project Schedule

This section includes the schedule for the project work.

4.1.1 Task Network Diagram

Figure 4.1 shows the Task Network Diagram of the proposed system.

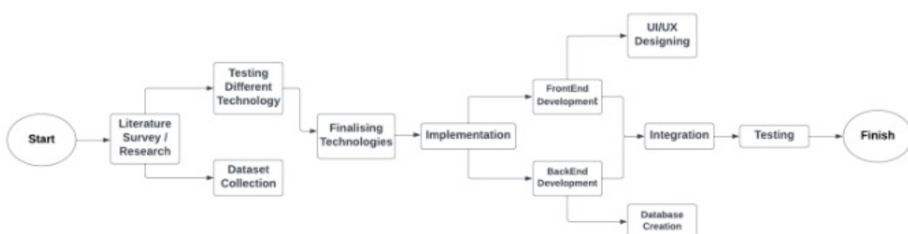

Figure 4.1: Task Network Diagram

4.1.2 Timeline Chart

Figure 4.2 shows the Timeline Chart of the proposed system.

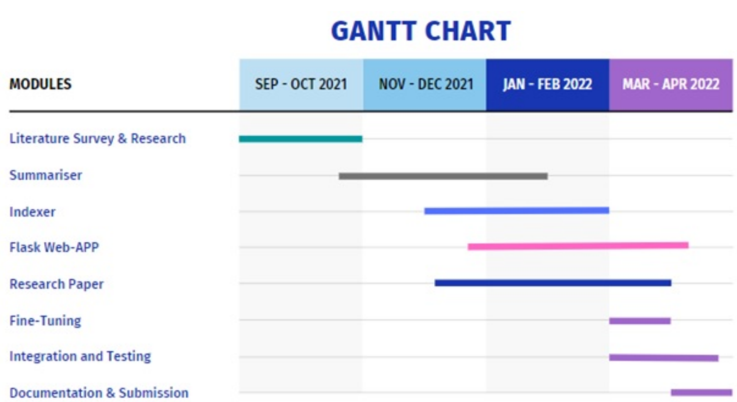

Figure 4.2: Timeline Chart

4.2 Feasibility Study

This section evaluates different kinds of feasibility for this project.

4.2.1 Technical Feasibility

For the user, the basic hardware and software requirements include having recent versions of existing operating systems like Windows, Linux etc and a preferred RAM requirement of 4 GB. A minimum free space of 16 GB on the laptop is preferred to download the custom notes from the website. Lower configurations will work as well but may give decreased performance and may impair efficiency by increasing query response time. However, the proposed system will be built on a considerably more powerful hardware platform (Intel i7 11th Gen, 16GB RAM, 4 GB NVIDIA MX250, 512 GB NVME PCIE 2.0 SSD). As a result, there is little doubt that the project will encounter technical issues during the coding and testing phases. Given the availability of hardware, it is possible to conclude that the suggested system is technically viable for development. The authors possess necessary technical expertise required to create the first prototype of the system. However, to scale the system to develop a well-defined product will require external help, preferably from an organisation. Hence, the project is feasible within the limits of current technology.

4.2.2 Economic Feasibility

Now, the cost of developing the proposed system is close to negligible as all the processing is being done locally on our laptops. When the system is deployed the only cost to use the system will be to process the video on a server. The creation of the audio file, generation of notes will carry some operational cost but will be minimal compared to the output of the system. Since most of the computation will be done using a server, no additional hardware costs for the user will be added, making the system cost effective.

4.2.3 Operational Feasibility

The proposed system will be highly efficient for the students and teachers as it helps them save time that could be better used elsewhere. Through a survey done in march, the students were receptive to this particular idea and were willing to use the system if developed. Since the system will be so easy to use, we expect minimal resistance and queries from the users about our system. Thus, considering the operational feasibility the development of the proposed system is considered as operationally feasible for development.

4.3 Project Resources

This section comprises of the different resources that must be needed for development of the project and smooth operation by user.

4.3.1 Hardware Requirements

The minimum hardware configuration required on the client side or rather on the user's device is:

- 8 GB RAM
- Intel i3 and above

The minimum hardware configuration required on the server side or rather on the developer's device is:

- 16 GB RAM
- Intel i7 and above
- NVIDIA MX/GTX/RTX GPU
- 4 GB Video-RAM
- 512 GB ROM

4.3.2 Software Requirements

Following are the software requirements for project completion.

1. Python
2. TensorFlow
3. Chrome Driver
4. MySQL
5. Web Browsers like:
    - Chrome
    - Firefox
    - Microsoft Edge

4.3.3 Operating Requirements

For smooth operation of the system, a stable internet connection with a minimum speed of 1MB/s is required.

4.4 Project Estimation

This section comprises of estimation of project parameters, function point analysis and risk management.

4.4.1 COCOMO Estimation Model

The COCOMO model is a good measure for estimating the number of person- months required to develop software. The Table 4.1 presents the COCOMO formulae for different types of programs:

Table 4.1: Formulas in COCOMO Model

TDEV	**Programmer Productivity**	**Development Time (Month)**
Application Programs	PM = 2.4 x $(KDSI)_{1.05}$	PM = 2.5 x $(PM)_{0.38}$
Utility Programs	PM = 3.0 x $(KDSI)_{1.12}$	PM = 2.5 x $(PM)_{0.35}$
System Programs	PM	PM

Using the above formula for the application programs, the programmer productivity and the development time are as follows: KDSI = 5KLOC PM = 2.4 x $(5)_{1.05}$ = 13 person-months TDEV = 2.5 x $(13)_{0.38}$ = 7 months P = PM/TDEV= 1.86. Thus, according to COCOMO model we obtained the following results shown in Table 4.2:

Table 4.2: Results of COCOMO Model

Effort Applied	13 person-months
Duration	7 months
People Required	3

4.4.2 Function Point Analysis

The Table 4.3 shows the Function Point Analysis of the proposed system.

Table 4.3: FPA Table

Measuring Parameter	Count	Weighing Factor
Number of external inputs (EI)	11	11x4=44
Number of external outputs (EO)	10	10x4=40
Number of external inquiries (EQ)	9	9x6=54
Number of internal files (ILF)	4	4x10=40
Number of external interfaces (EIf)	2	2x5=10
Total Count		188

$\sum(fi) = 4 + 1 + 0 + 3 + 5 + 4 + 4 + 3 + 3 + 2 + 2 + 4 + 5 = 43$

FP = Count-total * [0.65 + 0.01 *∑(fi)]
= 188 * [0.65 + 0.01 * 43]
= 188 * [0.65 + 0.43]
= 188 * 1.08
= 203 approx.
The estimate for developing this project would be 203 hours of work.
As there are three developers, each developer shall have to contribute approximately 68 hours of work.

4.5 Risk Management Mitigation Planning

The Table 4.4 shows the Risk Management, Mitigation and Planning for the proposed system.

Table 4.4: RMM Planning Table

Risk Event	Effect	Risk Mitigation	Contingency Plan
Interface Problems	Unstable system behavior	Test prototype	Refresh the system
System Freezing	System failure	Test prototype	Restart the server
Server load	Unreliable output	Timely testing server configuration	Load balancing of traffic

As with every project, we need to identify high-value, business-critical data assets that are vulnerable to both internal and external attacks. We should be able to provide early warnings of potential threats and vulnerabilities to sensitive enterprise information assets, data, and processes. Finally, with a business-consumable data risk control centre, it is critical to transmit meaning and value to the user. To optimise processes and manage risks, we should also allow talks with IT and security.

Chapter 5

System Design

This chapter consists of the different design elements of the proposed system.

5.1 Design Diagrams

This section contains the Data Flow Diagram and UML Diagrams to explain various aspects of the system design in detail.

5.1.1 Data Flow Diagram

Figure 5.1 shows the Data Flow Diagram of the proposed system.

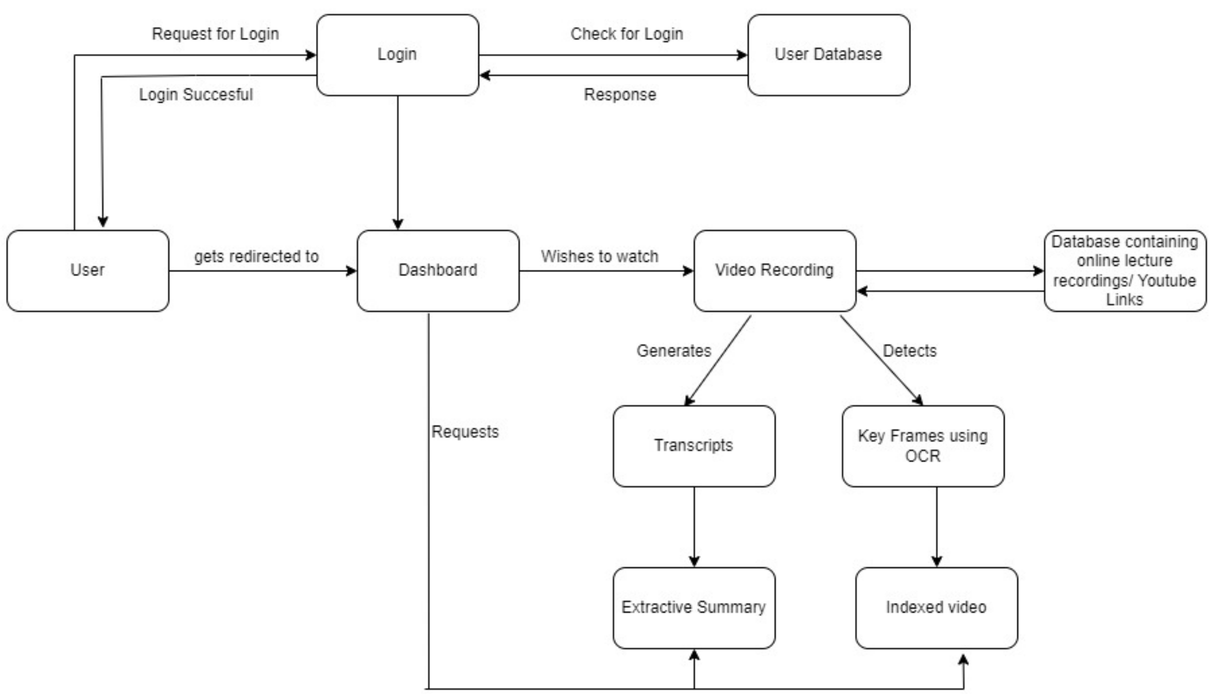

Figure 5.1: Data Flow Diagram

Whenever a user logs on to the website, he/she is brought to the dashboard page. The user has options to access the video, generated textual summary and indexed bookmarks/timestamps for key topics in the lecture. The dashboard also has links to provision upload of lecture videos or URLs for YouTube lectures. Once a new lecture is added, the user may generate a text summary or indexed/bookmarked video with certain timestamps via the buttons offered next to the uploaded video. For the purpose of generating text summary, the URL of the video or the video file itself is passed, from which, a transcript is generated using text-to-speech and passed to the Extractive Summarisation algorithm. In case of the Video indexing, the video file is parsed by a Computer Vision script to extract keyframes and Whiteboard/Slides content to finally produce bookmarks of important topics with timestamps.

5.1.2 UML Diagrams

The Use Case diagram of the proposed system is shown in Figure 5.2.

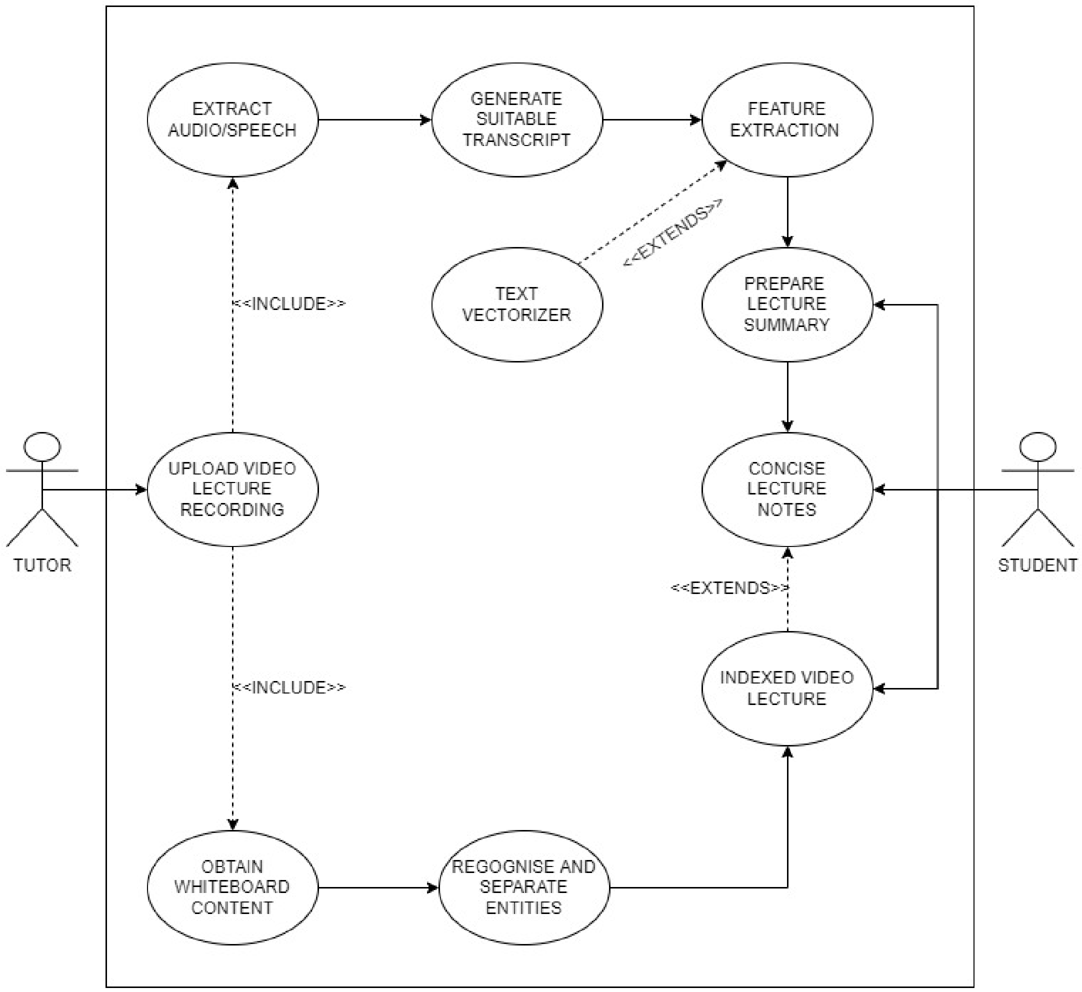

Figure 5.2: Use Case Diagram

There are two actors - Tutor and Student in the system. A tutor can upload the lecture in the form of a video file or URL (Table 5.1) followed by extraction of speech to create a transcript of the lecture and recognise content on the whiteboard/slides for Video indexing. The extracted transcipts are passed to an Extractive Summarisation algorithm which uses feature extraction of vectorized text to finally generate a Lecture Summary (Table 5.2). The content obtained from whiteboard/slides is parsed to recognise keyframes/entities to produce a Indexed Video lecture (Table 5.3). A student can access the concise notes from Extractive Summarisation (Table 5.4) and indexed video with timestamps (Table 5.5) for the videos uploaded by the Tutor.

Table 5.1: Use Case - Upload Video Lecture Recording

Use Case ID	Upload Video Lecture Recording
Xref	website/upload.html
Trigger	Tutor selects Upload Video Button
Precondition	Tutor is on website/dashboard.html
Basic Path	Tutor wants to upload new Video File
Alternative Paths	Tutor wants to work with already uploaded video
Postcondition	Video is uploaded
Exception Paths	-
Other	-

Table 5.2: Use Case - Generate Summary

Use Case ID	Generate Summary
Xref	website/summary.html
Trigger	Tutor clicks Generate Summary Button
Precondition	Tutor is on website/dashboard.html and Video is already uploaded
Basic Path	Tutor wants to generate summary for a video lecture
Alternative Paths	Tutor has already generated summary before
Postcondition	Summary is generated and stored
Exception Paths	-
Other	-

Table 5.3: Use Case - Index Video

Use Case ID	Index Video
Xref	website/indexed$_{v}ideo.html$
Trigger	Tutor clicks Index Video Button
Precondition	Tutor is on website/dashboard.html and Video is already uploaded
Basic Path	Tutor wants to index a lecture video.
Alternative Paths	Tutor has already indexed the video before.
Postcondition	Video Indexes are generated and stored
Exception Paths	-
Other	-

Table 5.4: Use Case - Access Video Summary

Use Case ID	Access Video Summary
Xref	website/summary.html
Trigger	Student selects Generate Summary Button
Precondition	Student is logged on website/dashboard.html
Basic Path	Student wants to access previously generated summary
Alternative Paths	-
Postcondition	Video Summary is displayed
Exception Paths	-
Other	-

Table 5.5: Use Case - Access Video Indexes

Use Case ID	Access Video Indexes
Xref	website/indexed$_{v}ideo.html$
Trigger	Student selects Index Video Button
Precondition	Student is logged on website/dashboard.html
Basic Path	Student wants to access previously indexed video lecture
Alternative Paths	-
Postcondition	Indexed Video Lecture is displayed
Exception Paths	-
Other	-

Figure 5.3 shows the Activity diagram of the proposed system.

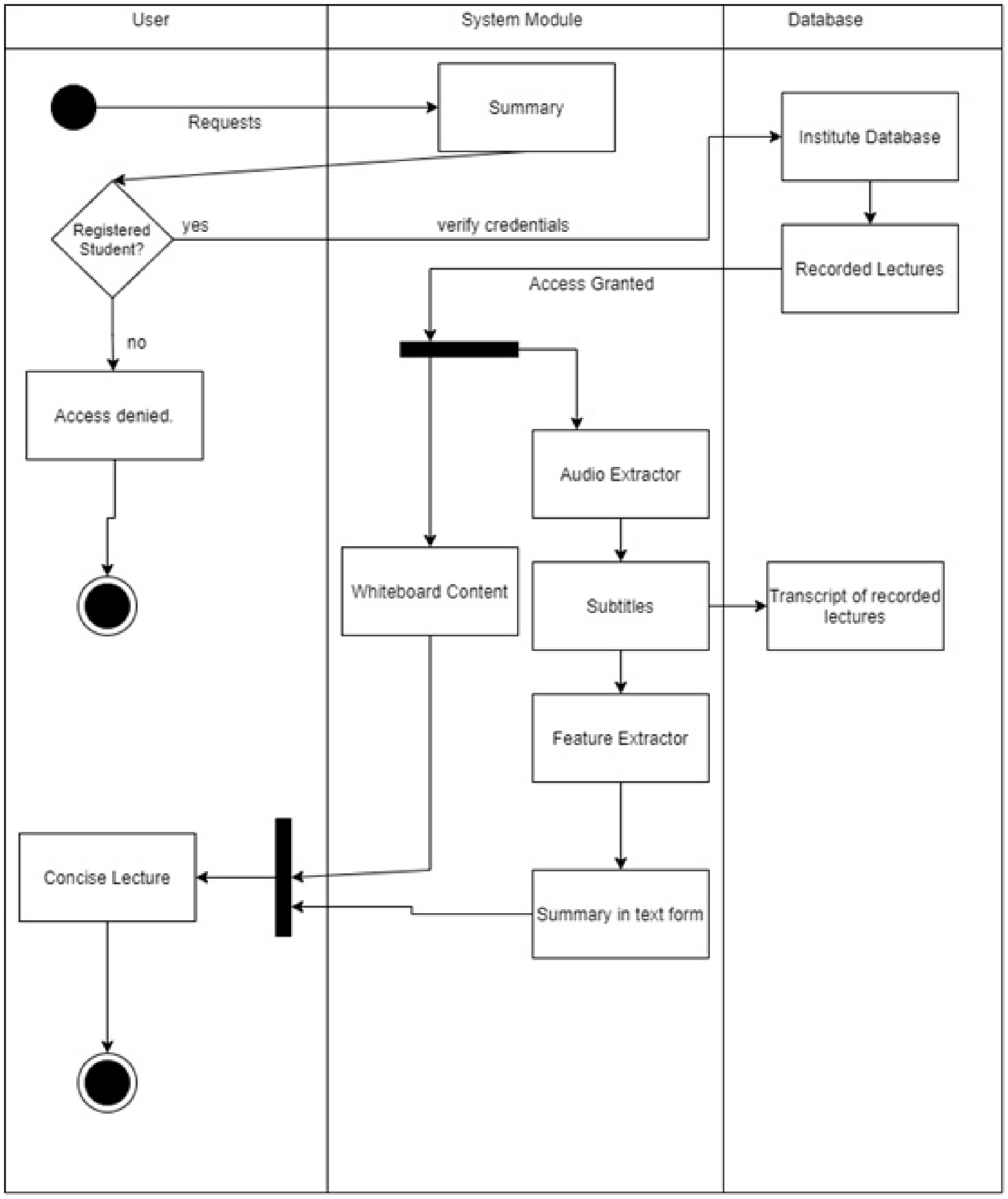

Figure 5.3: Activity Diagram

A User can access the Video Lecture files uploaded if they are logged on with appropriate credentials. After verification, a user can either access already stored summary and video indexes or initiate a process for generating them. For generating summary, transcripts are obtained from video speech and then passed to the Extractive Summarisation model to finally generate a summary text file. For video indexing, the content from whiteboard/slides is parsed to obtain keyframes/key-entities which are then bookmarked as timestamps to produce video indexes/chapters.

5.1.3 Database Schema

The database in our system is meant to store user credentials, video lecture details that were added via URL and local file in separate tables.

Table 5.6 describes the schema of the DB Table: users

Table 5.6: Schema of Table: users

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
name	varchar(100)	YES		NULL	
email	varchar(100)	YES		NULL	
username	varchar(30)	YES		NULL	
password	varchar(100)	YES		NULL	
userroll	varchar(2)	YES		s	

Table 5.7 describes the schema of the DB Table: summary

Table 5.7: Schema of Table: summary

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
teacher	varchar(100)	YES		NULL	
subject	varchar(100)	YES		NULL	
datee	varchar(10)	YES		NULL	
link	text	YES		NULL	

Table 5.8 describes the schema of the DB Table: vsummary

Table 5.8: Schema of Table: vsummary

Field	Type	Null	Key	Default	Extra
vid	int	NO	PRI	NULL	auto_increment
vteacher	varchar(100)	YES		NULL	
vsubject	varchar(100)	YES		NULL	
vdatee	varchar(10)	YES		NULL	
path	text	YES		NULL	

5.2 Proposed System Architecture

The Figure 5.4 shows the architecture of the proposed system.

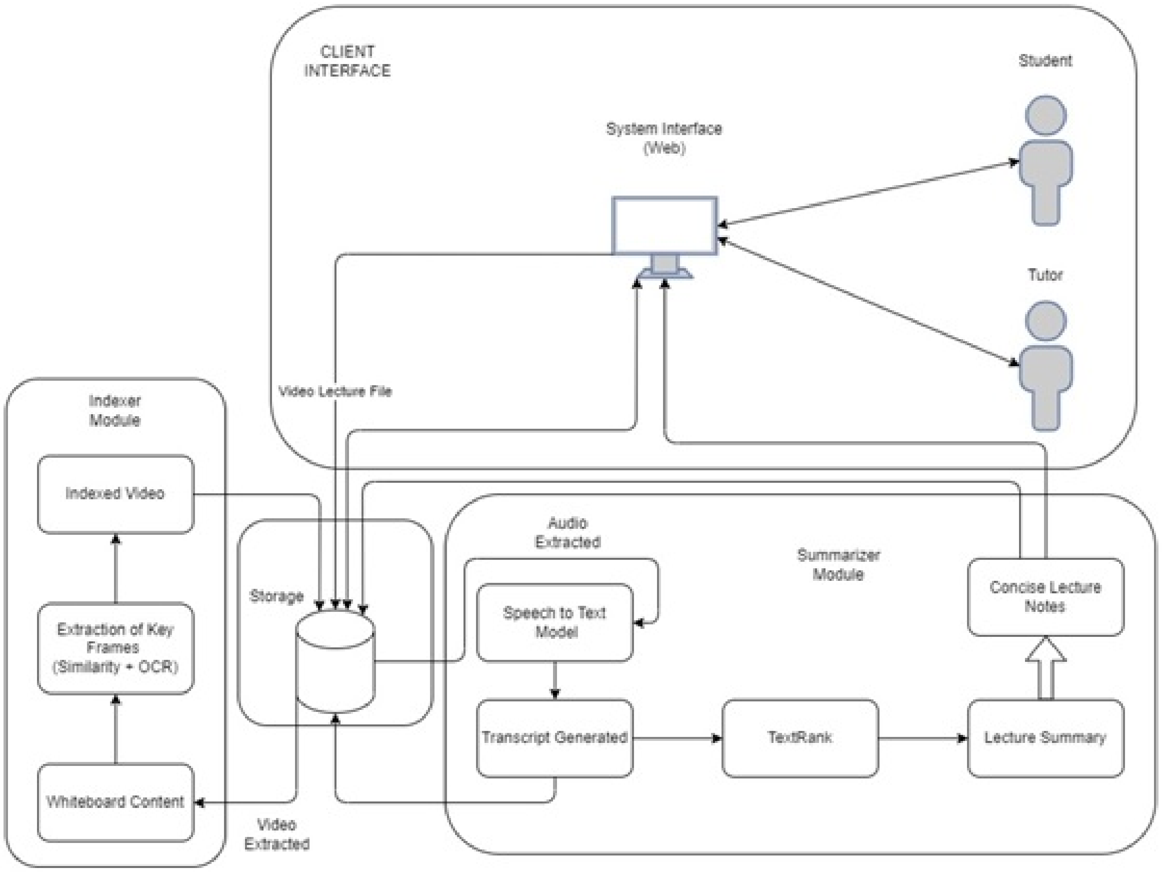

Figure 5.4: System Architecture

Our system is divided into three modules:

1. User Interface
2. Summariser
3. Indexer

A) User Interface

The User Interface is the link between the tutor or student and the system which provided the functionalities to upload, summarise and index video lectures. This UI allows the Tutor to upload a video which the students can access on their dashboard. Both tutor and student can command the generation of summary and indexing of the video. This invokes the respective AI-based scripts which produce stored files for both the summary and the video indexes. The user/s can access the summary and indexes of a video for which they have already been produced and the process need not be repeated every time a user logs on. The UI also has functionalities to register new students and also allow uploading of YouTube URLs instead of the entire video lecture file.

B) Summariser Module

The Summariser module takes the video file or the URL of the video as input to perform Extractive Text Summarisation. It leverages the IBM Watson text-to-speech model to generate transcipts of the video lecture which is given context by a punctuator that makes these transcripts viable for summarising. The transcript is then passed to the TextRank algorithm which works similar to PageRank and recognises key sentences in the video by arranging the sentences and their links in a directed text graph. The highest ranked sentences are compounded to produce a concise summary.

C) Indexer Module

The Video Indexer module is responsible for creating bookmarks or chapters with timestamp hyperlinks from the video file received as input. The video frames are parsed through with the help of the Computer Vision Library in python. While parsing the frames, each subsequent frame that is captured after a certain time interval (5 sec) is compared to the reference key-frame by the SSIM (Structured Similarity Index Measure) and if it is dissimilar by a certain threshold (<0.8 or 80% similarity), it is accepted as the next key-frame. After the whole video is parsed, all the key frames are run over with the Google Vision API's OCR tool to detect content on the whiteboard or lecture slides and then recognise chapter/topic names via a custom string manipulation algorithm. The video is parsed again to mark the first occurrence of topic/chapter names and marked as an index in the lecture. There are 2 passes on the video to optimize the detection of keyframes without compromising the efficiency of the SSIM threshold.

5.3 Description of Datasets

We are using Unsupervised Learning for both the Summariser and Indexer modules. Unsupervised Learning does not involve training a model or algorithm on any existing or historical data. For testing, we have used video lectures of various domains.

Chapter 6
Implementation

The proposed system was implemented as a web application that links and abstracts the two Unsupervised Learning modules written in Python scripts. The application comprises of the following functionalities: Login/Register for Tutors and Students, Upload Video Lecture file/URL for Tutor, Generate or Access Extractive Summary, Index Video or access previously indexed video lecture.

6.1 Working of System

The working of the system is explained in detail as follows:

- The User (Tutor/Student) can log on to the system dashboard via the login page. Students can register themselves via the Register webpage. A Tutor can be registered only via the admin. After successful verification of User credentials and permissions, the dashboard is displayed.
- On the dashboard, the Tutor has options to upload a new Video Lecture onto the application via the video file or the URL of the lecture video. This takes the Tutor to the upload_video/add_url page. After filling the respective forms, a new video filepath/URL is appended to the database. Students can access all videos uploaded by their Tutors.
- Either User (Tutor/Student) can make requests for summary generation and video indexing via the buttons next to the newly uploaded video. On clicking the respective buttons, the scripts for the process is initiated with the selected video and the summary/indexes are stored in the system. The users can always access the summary/indexes that were previously produced, hence making sure the process only happens once.
- The summary generation is done by extracting the audio/speech from the video, making transcripts of the video speech by IBM Watson's speech-to-text API, punctuating the transcripts and performing Extractive Summarisation on the transcripts.
- The video lecture is indexed or rather bookmarked with chapters by parsing the video using OpenCV, recognising key-frames by comparing subsequent snapshots of the video with the SSIM similarity measure, detecting whiteboard/slides content using OCR and finally generating an indexes file with chapters and their corresponding timestamps.

6.2 Algorithms Used

The various module of the system consist of the following algorithms.

TextRank

Pseudo Code:

Step 1: The different articles are combined into a single text input and then pre-process the data.

Step 2: The data is then split into different sentences and for each sentence a vector representation using word embeddings for every word.

Step 3: A similarity matrix is then created where similarity between each and every sentence is stored in respective cells of the matrix.

Step 4: A graph is then created where the sentences are the entities that act as nodes and the similarity score between those sentences are the weight of the edges that connect them.

Step 5: The top ranked sentences based on the sentence ranking are then selected to be a part of the summary.

Recognise Keyframes using SSIM

(SSIM - Structured Similarity Index Measure)

Pseudo Code:

Step 1: Iterate through video frames using OpenCV

Step 2: Assign first frame as key-frame

Step 3: Compare a snapshot frame every 5 seconds with the reference key-frame using SSIM

Step 4: If the SSIM measure is less than threshold (0.8), then append current frames to keyframes list and assign it as reference key frame for subsequent comparison.

Step 5: End, if all frames are parsed.

Create Chapters from Video Lecture

We use the OCR tool to detect content/text on the whiteboard/slides for the derived list of keyframes. The chapter names are recognised by eliminating unstructured text and sentences by a custom string manipulation algorithm. Then, the video is parsed again to mark the start timestamp of each chapter.

6.3 Tools Used

Flask: Flask is a Python-based microweb framework which does not need specific tools and libraries. It doesn't have rely on third-party libraries for form validation, database abstraction or any other typical components. The summariser and indexer modules are implemented using Flask. This Request-Response API function aids in the separation of duties, resulting in a fault-tolerant system.

Google Cloud Vision API: Through REST and RPC APIs, Google's Cloud Vision API's Machine Learning models assigna labels to photographs and sort them into millions of predefined categories in seconds. The model in this API can detect various objects, faces, printed and handwritten text. We are using this tool to detect the whiteboard/slides content in the video lectures.

OpenCV: We are using the python opencv library to parse the video lecture in the form of image frames, take regular snapshots to recognise keyframes and mark the first occurrence of Chapter titles in the video.

MySQL: We are using MySQL as our RDMS. It is meant to store information about the uploaded video lectures and their details. The MySQL database also has tables for recording credentials of all Tutors and Students who have access to the system. The table where the video information is stored, stores the video itslef as a URL or a path to the video file on disk.

6.4 Interface Design

This section reflects the interface design of the system.

The following images show the path a user takes from the Home page (Figure 6.1) to Register(Figure 6.2)/Login(Figure 6.3), finally to the dashboard.

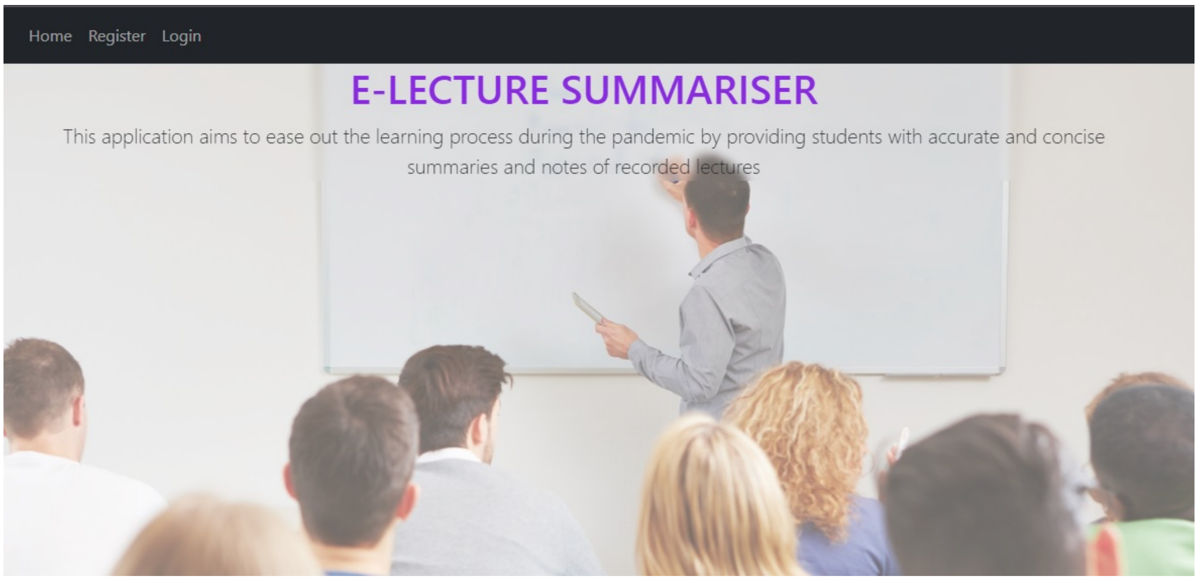

Figure 6.1: Home page

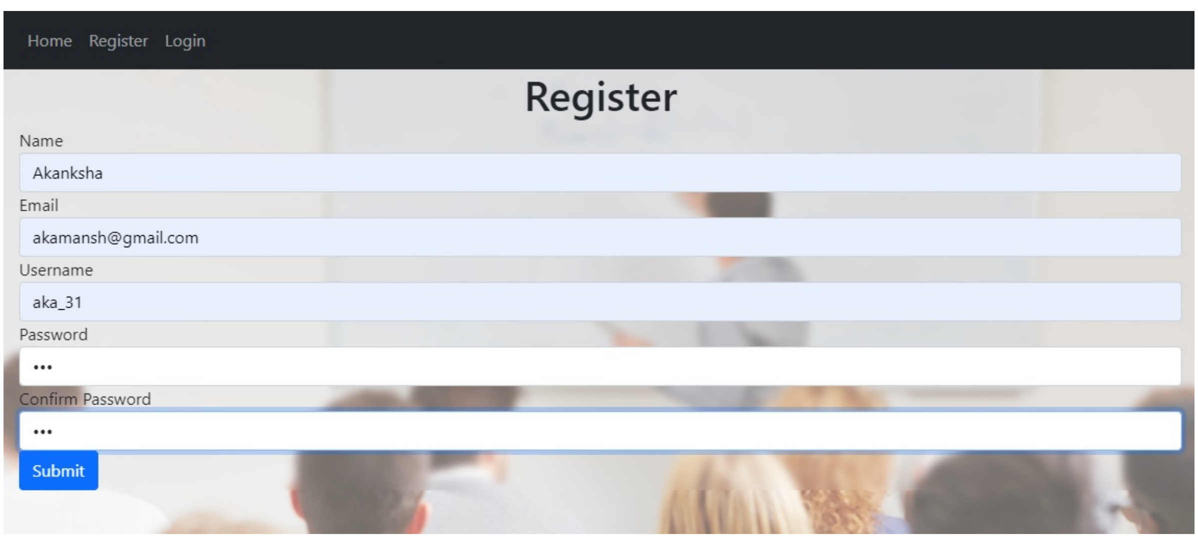

Figure 6.2: Registration page

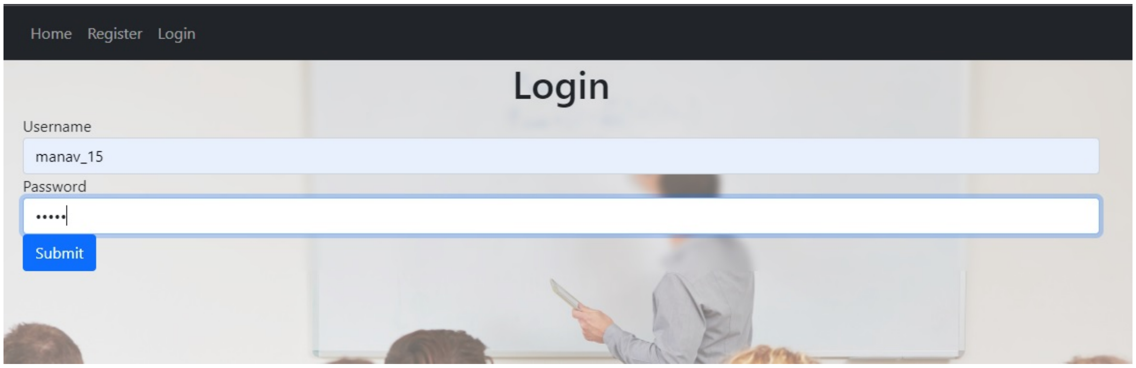

Figure 6.3: Login page

We can see the Dashboard (Figure 6.4) now.

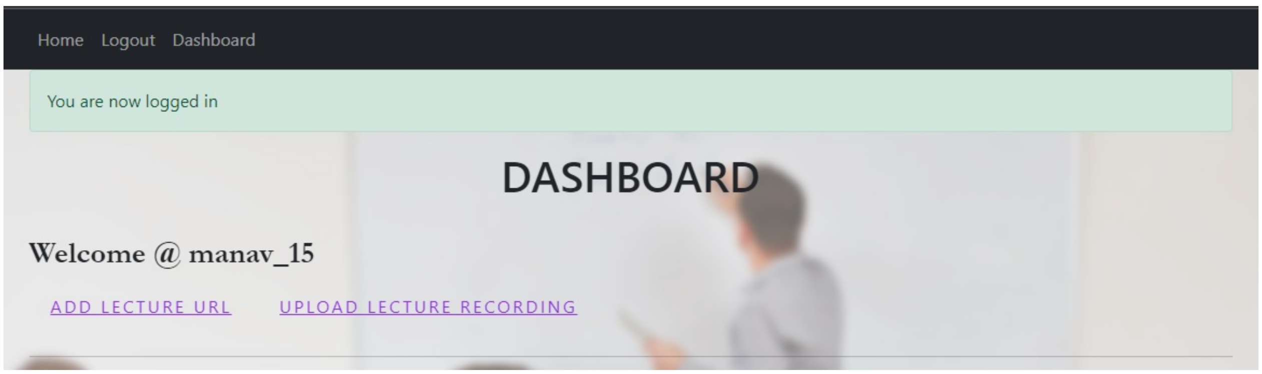

Figure 6.4: Dashboard

The tutor can use the Add Lecture URL (Figure 6.5) and Upload Lecture Recording (Figure 6.6) to add new video lectures.

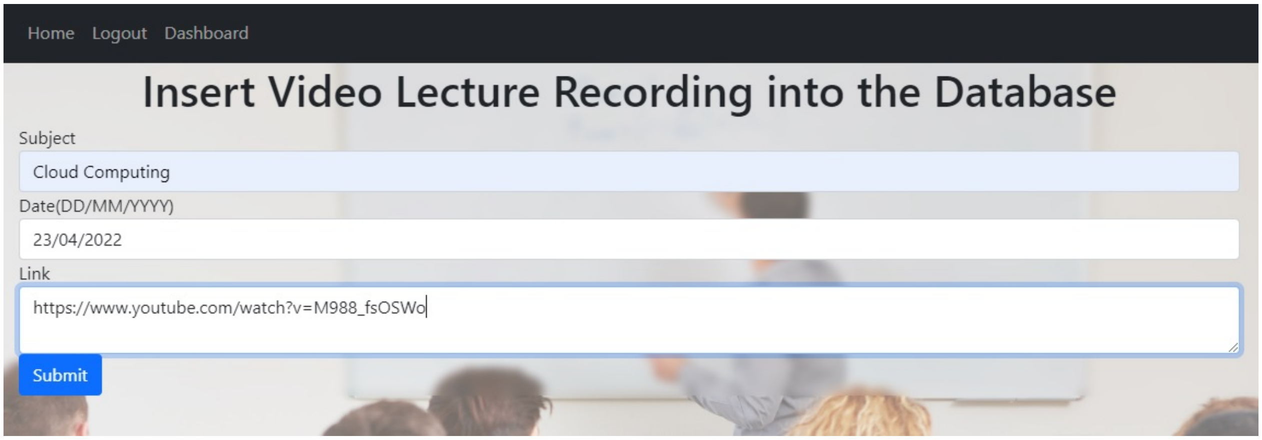

Figure 6.5: Upload Video Lecture - URL

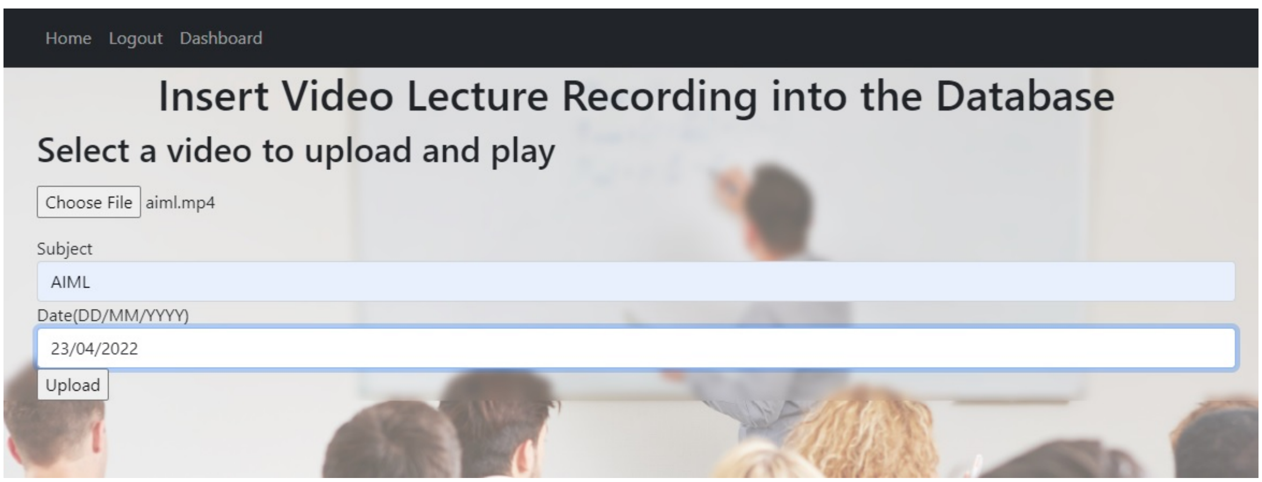

Figure 6.6: Upload Video Lecture - File

The user can request or access the summary by clicking the Generate Summary button (Figure 6.7).

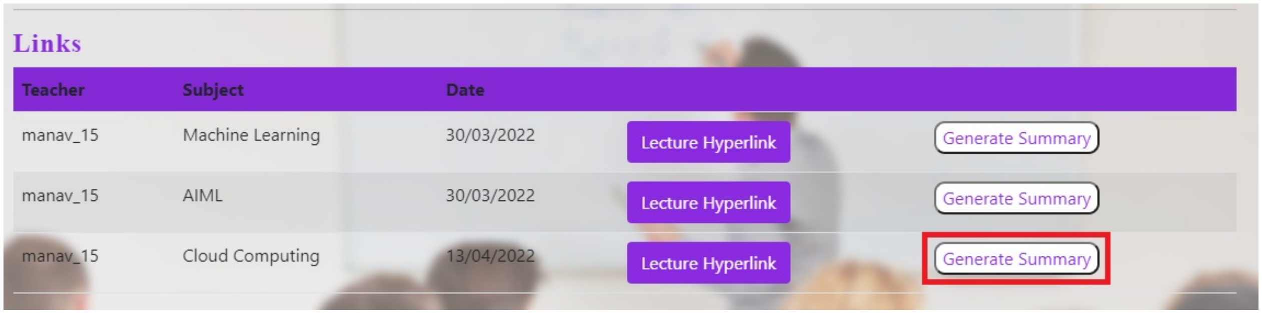

Figure 6.7: Request Summary Generation

The generated summary is displayed on a new page (Figure 6.8).

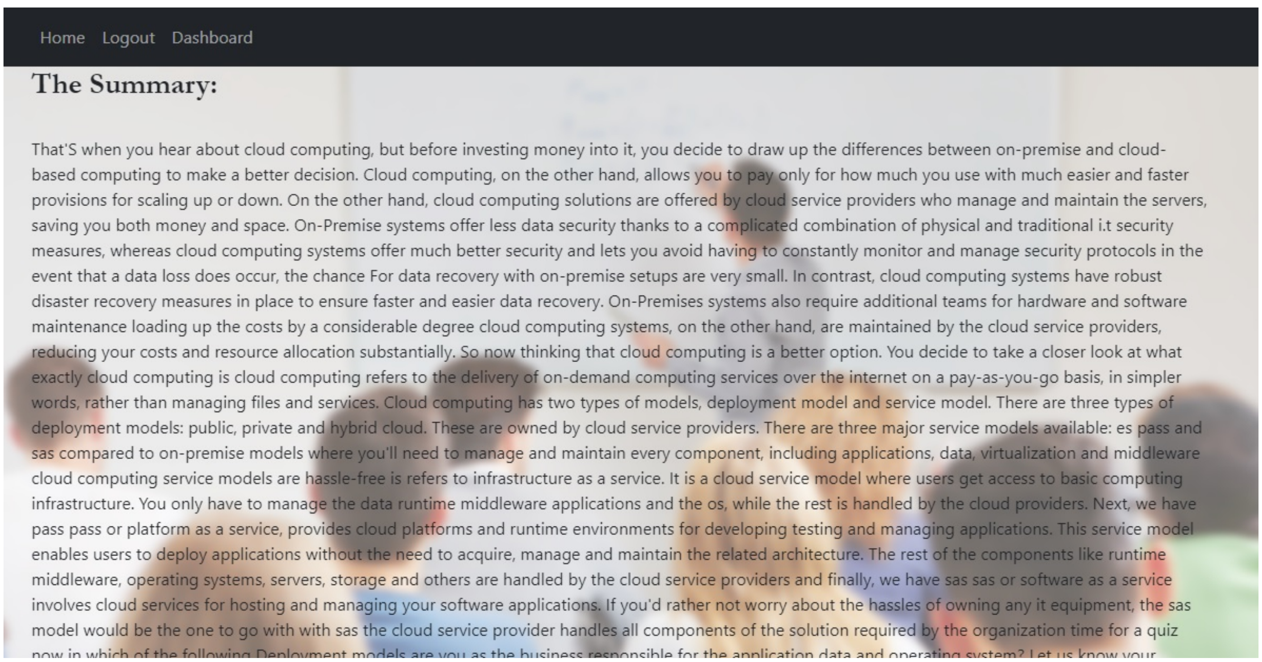

Figure 6.8: Display Summary

The user can request or access the video indexes/chapters by clicking the Index Video button (Figure 6.9).

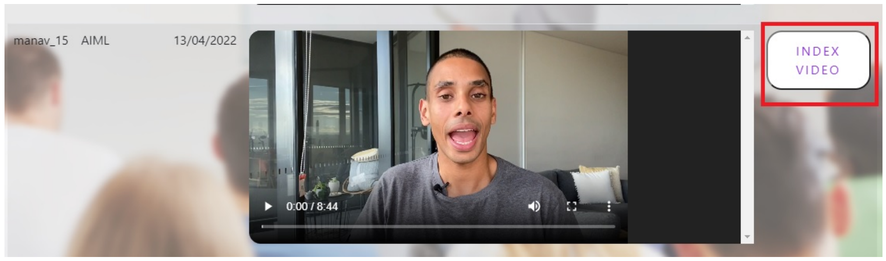

Figure 6.9: Request Video Indexing

The Video is displayed on a new page (Figure 6.10) with hyperlinks to timestamps of the chapters.

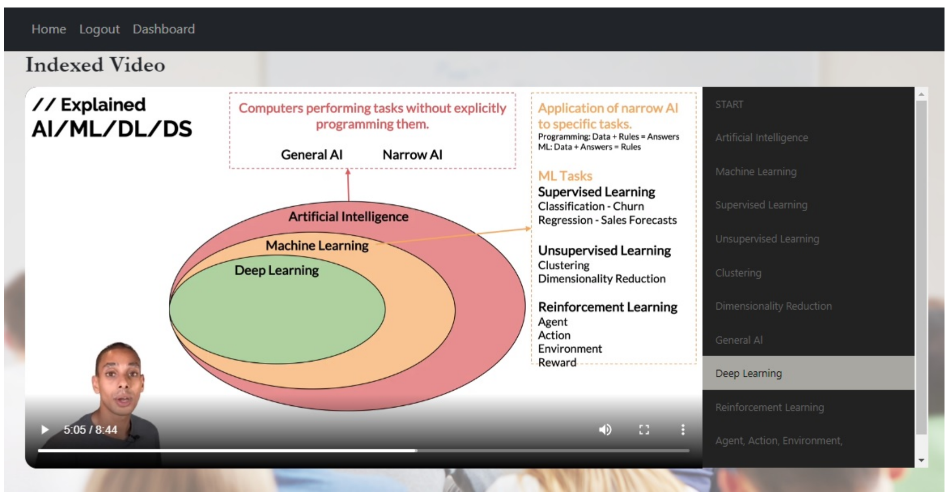

Figure 6.10: Display Indexed Video

We can see that the chapters are now visible on the dashboard too. (Figure 6.11)

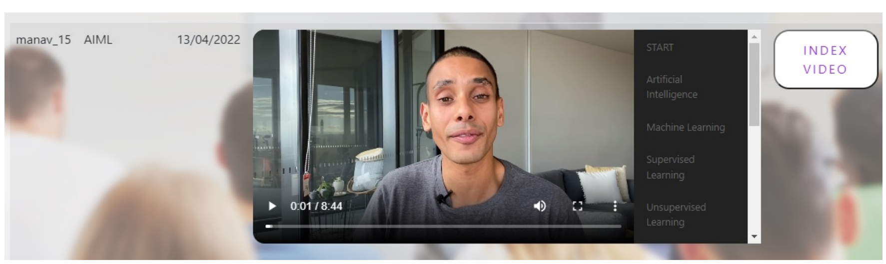

Figure 6.11: Indexed Video on Dashboard

Chapter 7

Testing and Results

This chapter includes all the different kind of tests and experiments that are to be performed on the system and its various modules.

7.1 Test Plan

This section comprises the Test Plan for the system testing.

Introduction:
Testing is a process that allows the user to evaluate the system created in order to detect faults in the system so that it can be rectified. It involves the examination of the software product to determine whether or not it meets the required criteria.
Our testing plan mainly involves verification of the type of data that the user is uploading, checking for anomalies in the data and pre-processing it accurately whenever necessary. The system architecture is divided into different modules and hence their individual efficiency as well as the integration of these modules is tested thoroughly. Depending on the requirements of users, features incorporated, and procedures required for performance, several test techniques were used to assess the system's functioning and correctness. Each approach was given its own set of test cases, and the results were compared to what the system was expected to do.
Different team members were tasked with handling different modules of the system and were hence responsible for the subsequent testing of their modules.

Purpose of The Test Plan Document (Objective):
The main objective of the test plan document is to guide the thinking of the team. It helps them create a layout for the entire process and assists them to systemically follow all the steps of the layout. It makes them confront the different challenges they might face and helps them to focus on important areas. The entire team behind the system not only involves the test team but also involves the customers, business managers etc. and this plan helps to communicate the details of the testing process to them directly.

References:
Documents and links, including the following:

- Literature survey
- Existing system literature
- Past semester documentation

Items to be Tested / Not Tested:
All the items to test and their description is mentioned in Table 7.1

Table 7.1: Items to be Tested / Not Tested

Item to Test	**Test Description**	**Test Date**	**Responsibility**
Login / Registration Page	Involves the testing of registration of user as a student to the system and includes appropriate display of dashboard functionalities	March 2nd,2022	Akanksha Mansharamani
Speech to Text	Involves testing of the first step of the summarizer module where the audio is extracted from the video uploaded	March 22nd,2022	Varun Vora
Pre-processing and Summary Generation	Involves testing whether the transcript generated is in the correct format for summary generation	April 2nd,2022	Varun Vora
Video Chapter Generation	Involves testing of the accuracy of the division of the video into multiple chapters.	April 8th,2022	Manav Shah

Test Approach: The test approach is described in Table 7.2.

Table 7.2: Test Approach

Modules	**Test Approach**	**Remarks**
User Interface Module, Summarizer Module and Video Indexer Module	The application has been tested in a methodical manner, beginning with functional testing and progressing to non-functional testing. The strategy is based on the agile approach, which consists of a four-part cycle that begins with unit and component testing. Quality of components and code would be the emphasis of the first phase. Functional testing, focused on business-driven test cases, would be the second phase of the cycle, with the goal of determining whether or not client demands have been met. Usability, user acceptability, and collaborative testing are all used in the third phase. It concentrates on incorporating comments from the previous two stages and iterating tests to improve product confidence. The last stage of the cycle encompasses non-functional testing such as performance and load testing to guarantee the project's robustness and viability under stress. Aside from the cycle plan, white box and black box testing were performed to verify that a bug-free, well-integrated system was produced.	

Test Regulatory / Mandate Criteria:
There are no strict regulations that the system is required to follow, however the summary generated should follow proper semantics of the English language and the chapters generated should be related to the topics taught in the lecture.

Test Pass / Fail Criteria:
The performance and result produced by each component of the system would determine the pass/-fail criterion for each test item(feature). If the actual result provided matches the predicted standard outcomes, a feature passes the test.

Test Entry / Exit Criteria:

In the event of an anomaly, when the actual output differs significantly from the predicted result, testing must be halted. Testing would only restart if a practical solution or an equally adequate alternative could be discovered. An example of such a circumstance would be when the generated summary is an empty string or the number of chapters created is either too low or too large.

Test Schedule:

The test schedule is shown in Table 7.3:

Table 7.3: Test Schedule

Task Name	Start	Finish	Effort
Test Planning	27th Feb	1st March	
Review Requirements documents	27th Feb	29th Feb	2 days
Create initial test estimates	29th Feb	1st March	1 day
Staff and train new test resources	1st March	2nd March	1 day
User Interface Testing	1st March	7th March	5 days
Module Testing	2nd March	10th March	7 days
Database Testing	10th March	12th March	3 days
Functional testing – Iteration 1	13th March	16th March	4 days
Functional testing – Iteration 2	16th March	18th March	2 days
Integration testing	19th March	26th March	7 days
System testing	26th March	2nd April	7 days
UAT (user acceptance)	7th April	10th April	3 days
Resolution of final defects and final build testing	11th April	13th April	2 days

Test Deliverables:

After the successful execution of the proposed test plan, the available system will allow users to upload youtube links of lectures they want to summarise. The tutor will also be allowed to upload video lectures which can be stored for the students to refer to. A concise summary can also be created with automated video chapters being embedded into the video for the user to seamlessly transition to different topics taught in the lecture.

7.2 Test Cases

This section involves various kinds of Testing, viz. Unit, Functionality, UI, Database User Acceptance testing.

The Table 7.4 represents the Unit Test Cases.

Table 7.4: Unit Test Cases

Test Case ID	**Test Scenario**	**Expected Results**	**Actual Results**	**Pass/Fail**
UT01	Login	The user should be able to login to the system if his credentials in the system exist	The credentials are matched with the credentials in the database and the user is allowed to login	Pass
UT02	Speech to text	The audio file should be extracted and transcripts for the same should be generated	The audio is extracted and a subsequent transcript is generated	Pass
UT03	Punctuator	The generated transcript, if not punctuated, should be punctuated accurately.	The transcripts are punctuated successfully	Pass
UT04	Summarizer	The punctuated transcripts are passed through the summarizer to generate summaries	The summaries of a specific ratio (to the total length of the transcript) is generated.	Pass

The Table 7.5 represents the Functionality Test Cases.

Table 7.5: Functionality Test Cases

Test Case ID	**Test Scenario**	**Expected Results**	**Actual Results**	**Pass/Fail**
FT01	Youtube Link Upload	The user should be able to upload the youtube link of the lecture he wants to summarise.	The user is successfully able to add the YouTube link and is visible on the dashboard.	Pass
FT02	Video Lecture Upload	The user should be able to upload the video lecture they want to summarise.	The user is successfully able to upload the video and play it on the dashboard.	Pass
FT03	Summary Generation	The user on clicking the Generate Summary button should be able to view the summary of the video.	The user on clicking of the Generate Summary button is able to view the summary for the lecture.	Pass
FT04	Video Chapter Generation	The user on clicking the Index Video button should be able to create bookmarks for the video.	The user is able to generate and view the multiple chapters of the video and is able to navigate to various chapters seamlessly.	Pass
FT05	Searching and Sorting Lectures	The user should be able to search for lectures using keywords or sort them according to the lecture date.	The user was not able to search lectures and sort them.	Fail

The Table 7.6 represents the User Interface Test Cases.

Table 7.6: User Interface Test Cases

Test Case ID	**Test Scenario**	**Expected Results**	**Actual Results**	**Pass/Fail**
UI01	Registration Page	The user should be able to create their account using credentials in the adequate format for the system if they do not have an existing account.	New users are successfully able to register themselves to the system.	Pass
UI02	Login Page	An existing user should be able to login using the credentials created at registration time.	The user is able to successfully login to the system.	Pass
UI03	Dashboard Features	The user type student should only be able to upload youtube links while the user type tutor is allowed to upload video lectures as well.	Students are only allowed to upload youtube links while the tutor is allowed to upload both: YouTube links and Video lectures.	Pass
UI04	Logout	The user should be able to logout of the system and should be redirected to the login page.	The user successfully logs out of the system and is redirected to the login page.	Pass

The Table 7.7 represents the Database Test Cases.

Table 7.7: Database Test Cases

Test Case ID	**Test Scenario**	**Expected Results**	**Actual Results**	**Pass/Fail**
DT01	Storing credentials	The user information should be securely stored in the database.	The information upon clicking of registration button is successfully stored.	Pass
DT02	Verifying Credentials	When the user tries to login, the credentials should be matched after fetching user information from the database.	The user login data is successfully fetched from the database and verified against the input credentials.	Pass
DT03	Storing Youtube Links	The youtube links uploaded by the user should be stored in the database.	The links are stored successfully and are further used for summarization purposes.	Pass
DT04	Storing Video Lectures	The video lectures uploaded should be stored on cloud for easy access.	The video lectures are currently being stored on the local system.	Fail
DT05	Storing Results	The summaries and video chapters generated once should be available in future without repeated processing.	Once generated, the summaries and the video chapters are stored in the database.	Pass

The Table 7.8 represents the User Acceptance Test Cases.

Table 7.8: User Acceptance Test Cases

Test Case ID	**Test Scenario**	**Expected Results**	**Actual Results**	**Pass/Fail**
UA01	User wants to summarise a youtube lecture	The user should be able to get a summary for the uploaded link	The user is able to get a summary for the uploaded link	Pass
UA02	User wants to generate a summary for video lectures	The user should be able to get a summary for the uploaded lecture	The user is able to get a summary for the uploaded lecture	Pass
UA03	User wants to generate video chapters for the uploaded lectures	The user should be able to generate video chapters for the uploaded lecture	The user is able to generate video chapters for the uploaded lecture	Pass

7.3 Testing Methods Used

Testing techniques are the tactics and processes used to test a product to ensure it is fit for purpose. Testing techniques frequently include ensuring that the product fits its specifications, has no negative side effects when utilised outside of its design restrictions, and will fail-safely in the worst-case scenario. Software testing approaches relate to the various techniques and procedures for ensuring that a software application has been thoroughly tested. Software testing methodologies cover unit testing single modules, system integration testing, and specialised types of testing such as security and performance testing.
Some of the testing methods used are:

1. Unit testing: Unit testing is a software engineering technique for testing the smallest testable pieces of program which may be called units. Used by Software developers and QA employees, unit testing's primary goal is to separate written code and confirm its working by isolated testing.
   It is a crucial phase in the development process that may assist in uncovering early defects in code that would be difficult to identify later.

2. User Interface Testing: UI Testing is used to test the interface of any software that a user interacts with. This generally involves testing visual components to ensure that they meet required functionality and performance. UI testing guarantees a bugfree user interface.

3. Database Testing: Any system or application would be incomplete without a database. Everything is based on data. Nothing is possible if there is no data. Instability in the database might lead the systems to act strangely. Either a database crashes or data is stored in a disorganised manner in a database; in either situation, the data becomes unusable. As a result, database testing assists us in identifying such vulnerabilities in a database system, allowing us to safeguard a database from becoming unstable.

4. Whitebox Testing: White Box Testing involves examining the product's architecture, code and overall structure to validate input-output flow, usability, and security. One of the key objectives of white box testing is to cover as much source code as feasible. Code coverage is a measure that indicates how much of an application's code is covered by unit tests. It also minimises the amount of time it takes for testers and developers to communicate, allowing for continual improvement of code and development procedures.

5. Blackbox Testing: Black-box testing is done without any prior knowledge of a system's internal workings. The system's user interface engaged by giving inputs and evaluating outputs in a black-box test without any understanding of where the inputs are processed.

6. User Acceptance Testing: The final stage of any software development or change request life-cycle before go-live is User Acceptance Testing (UAT). Actual users evaluate the programme in real-world scenarios to see if it achieves what it was supposed to do, verifying updates and determining conformance to their company's business needs. Of course, functional specialists in charge of the technical aspect of software development are crucial in structuring UAT cycles and interpreting the findings. When it comes to a good UAT test, however, business users are truly the stars of the show.

7. Integration Testing: Integration testing is done to ensure seamless combination of all components and ensure that they work properly together.

8. Regression Testing: Regression testing is used to ensure that a problem in the system is resolved without violating other perfectly working functionalities. Regression testing is done to guarantee that any kind of repairs don't result new flaws that might be obscure.

7.4 Experimental Results

In this section, we test or rather experiment the system with different variables, in our case different video lectures.

Video Lecture 1 (Topic - AIML):

The URL for the AIML video lecture is:
https://www.youtube.com/watch?v=iPUWwpocc1c

On clicking the ”Generate Summary” button, we obtain the following summary:

So let's kick things off and take a look at AI, so AI is really to do with the ability of computers and machines to perform tasks without explicitly programming them, otherwise known as the ability for computers and machines to think by themselves. A on general AI typically refers to the ability for a computer or a machine to be able to handle a wide variety of tasks. The ability for AI and machines to be able to do a broad range of tasks similar to humans is what we typically refer to as general AI. Narrow AI, on the other hand, is the ability for a machine to handle a really simple or a really narrow range of tasks. So I'm going to be painting a bunch of visual imagery to help you remember some of these topics, so the first one in terms of breaking out general and narrow, AI or the ability to remember general and our AI is just picture. A really narrow or really skinny journal in your mind, so that way you know that there's two different types of AI general and narrow now on to the next topic: machine learning. So, if taking a look at AI as being broken up into journal and narrow, but how does machine learning fit into this well machine learning is the application of narrow AI to specific tasks. Now, when we typically talk about machine learning, we often compare it to traditional programming, so in traditional programming we supply data plus rules or conditional logic, and we get answers now in machine learning. We can then pass new data to get new answers, so this is a bit of a change in the paradigm of how computer scientists and machine learning engineers are building programs these days. So what are some typical machine learning tasks? Well, we broadly break out machine learning into three key categories: these are supervised, learning, unsupervised, learning and semi-supervised learning. First, so supervised learning can be broadly broken out into two key categories. You could take that data and pass it through to a classification algorithm to help but learn which types of pizzas you like. Learning well, there's two key things to think about when you think of unsupervised learning these are really clustering, so the ability to group people together so say you wanted to group together, high performing and low, performing and medium performing employees or high-value, low value, medium value, customers Or a whole bunch of other different types of data, but really it's all to do with grouping things together now, dimensionality reduction, on the other hand, is all to do with condensing the features that you've got within a machine learning model. So a lot of the time you might start out with a huge data set with a lot of columns and you're, not really sure which of those columns are important for your machine learning model. Dimensionality reduction helps you reduce the number of columns that you've got so that you can really focus on the important ones now in order to remember supervised, learning and unsupervised learning. I'D suggest you remember this initialism Christopher Robin quarter duck so that way you remember classification, regression, clustering and dimensionality reduction, so that takes care of supervised and unsupervised learning. Reinforcement learning has four key things: these are an agent and action and environment and of reward. We train reinforcement, learning models to act in a correct way in a given environment in order to learn appropriate actions, given that specific environment now the best way to remember reinforcement, learning techniques is to remember area 51. Okay, so that takes care of machine learning. So deep learning is a subset of machine learning and really it's to do with performing machine learning tasks using deep neural networks. Now, the best way to remember deep learning is to remember that deep learning is just like an onion. The best way to remember the key components of data science are to look at the crisp DM framework, so the crisp DM framework stands for the cross industry, standard process for data mining and basically it's a framework to help you along your way, to producing really good Data science projects now there's six key steps in the data science process. So, whether or not you've got missing values, visualizing that data and taking a look at some summary statistics, we've then got data preparation, so this is all to do with getting our data ready for modeling. This is all to do with training your machine learning algorithms to perform well on a specific task. That way you remember business, understanding, data, understanding, data, preparation, modeling evaluation and deployment. Now the most important library in terms of machine learning is probably scikit-learn. So scikit-learn is been around for quite some time and gives you a whole bunch of really powerful algorithms and utilities to help use them to train your machine learning models. Now deep learning is becoming increasingly popular and there's a large number of libraries that can help you perform deep learning, some of which, which are notable, are tensorflow, keris, pi, torch and piano just to name a few, and that about wraps up AI versus MO versus do Versus DES thanks so much for tuning in guys, hopefully you found this video useful if you did be sure to give it a thumbs up and hit subscribe until next time: peace, [, Music,], On clicking the ”Index Video” button, the Video Chapters are displayed in the following way:

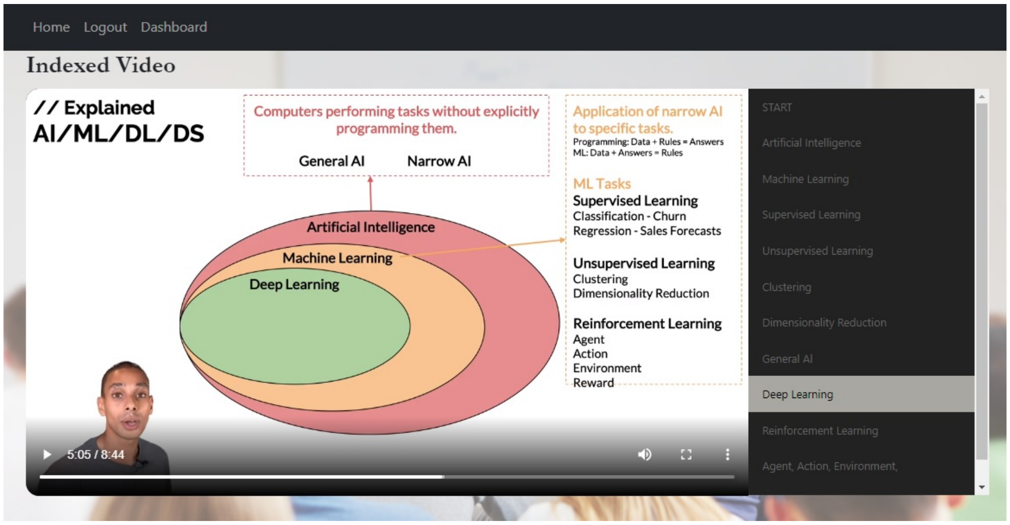

Figure 7.1: Chapters in AIML

Video Lecture 2 (Topic - Cloud Computing):

The URL for the AIML video lecture is:
https://www.youtube.com/watch?v=M988_fsOSWo

On clicking the ”Generate Summary” button, we obtain the following summary :

That’S when you hear about cloud computing, but before investing money into it, you decide to draw up the differences between on-premise and cloud-based computing to make a better decision. Cloud computing, on the other hand, allows you to pay only for how much you use with much easier and faster provisions for scaling up or down. On the other hand, cloud computing solutions are offered by cloud service providers who manage and maintain the servers, saving you both money and space. On-Premise systems offer less data security thanks to a complicated combination of physical and traditional i.t security measures, whereas cloud computing systems offer much better security and lets you avoid having to constantly monitor and manage security protocols in the event that a data loss does occur, the chance For data recovery with on-premise setups are very small. In contrast, cloud computing systems have robust disaster recovery measures in place to ensure faster and easier data recovery. On-Premises systems also require additional teams for hardware and software maintenance loading up the costs by a considerable degree cloud computing systems, on the other hand, are maintained by the cloud service providers, reducing your costs and resource allocation substantially. So now thinking that cloud computing is a better option. You decide to take a closer look at what exactly cloud computing is cloud computing refers to the delivery of on-demand computing services over the internet on a pay-as-you-go basis, in simpler words, rather than managing files and services. Cloud computing has two types of models, deployment model and service model. There are three types of deployment models: public, private and hybrid cloud. These are owned by cloud service providers. There are three major service models available: es pass and sas compared to on-premise models where you'll need to manage and maintain every component, including applications, data, virtualization and middleware cloud computing service models are hassle-free is refers to infrastructure as a service. It is a cloud service model where users get access to basic computing infrastructure. You only have to manage the data runtime middleware applications and the os, while the rest is handled by the cloud providers. Next, we have pass pass or platform as a service, provides cloud platforms and runtime environments for developing testing and managing applications. This service model enables users to deploy applications without the need to acquire, manage and maintain the related architecture. The rest of the components like runtime middleware, operating systems, servers, storage and others are handled by the cloud service providers and finally, we have sas sas or software as a service involves cloud services for hosting and managing your software applications. If you'd rather not worry about the hassles of owning any it equipment, the sas model would be the one to go with with sas the cloud service provider handles all components of the solution required by the organization time for a quiz now in which of the following Deployment models are you as the business responsible for the application data and operating system? Let us know your answer in the comments section below for a chance to win an amazon voucher coming back to cloud computing. Some of the most popular cloud computing services in the market are aws or amazon web services, microsoft, azure and google cloud platform want to learn more about them and how they differ from each other.

On clicking the "Index Video" button, the Video Chapters are displayed in the following way:

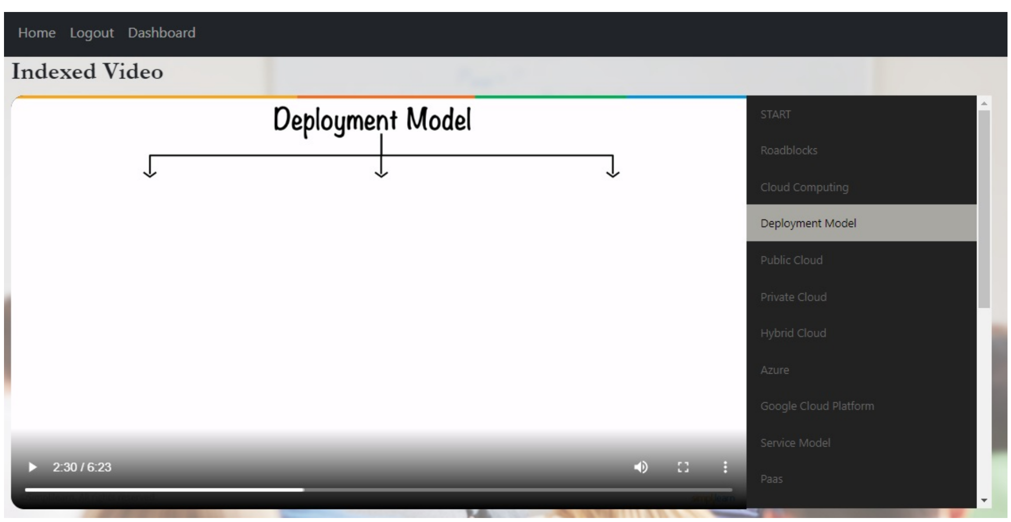

Figure 7.2: Chapters in Cloud Computing

Chapter 8

Conclusion

This report proposes a E-Lectures Summarising and Indexing system that takes a video file or URL as an input to generate and provide concise notes in the form of an Extractive Summary and Bookmarked or Indexed Video Lecture with important topics linked with their timestamps.

Video Summarization and Indexing has been proposed and implemented using many different approaches intimating a plethora of techniques for consuming content from all parts of the video. This system needs a pipeline that bifurcates into the Summarizer and Indexer modules with disparate use of methods in each moving part. IBM Watson's speech-to-text model seems to provide the most promising results with a provision to understand multiple accents of the English language. When we arrive at the summarization task, the TextRank algorithm provides satisfactory results, in terms of the length of the summary, the organization of it and absence of any major grammatical errors. In the Indexer module, regular snapshots need to be taken using python's OpenCV library and in the next phase of implementation key frames are to be extracted with the help of similarity models to eliminate redundant frames. The visual content in these key frames is to be recognized using Google Cloud Vision's OCR tool, which provides state-of-the-art results. After detecting the whiteboard content, we use SSIM as a similarity measure to detect, label and index keyframes which are important checkpoints in a video lecture. Both modules are integrated into the Application that will abstract the workings of this system and act as a bridge between students, teachers and the process happening behind the UI.

Chapter 9

Future Scope

The implemented system includes two modules which employ Unsupervised Learning, namely, Summariser and Indexer.

In case of the Summariser, there is scope to replace Extractive Summarisation with Abstractive Summarisation to better include the context of the lecture rather than just picking highly ranked sentences.

In the Video Indexer part, the quality of chapters being picked as keyframes can be improved by including some kind of entity recognition that deals with the title of the chapter better than a simple string manipulation algorithm. Also, a way to optimize the threshold for the SSIM measure could be developed.

Lastly, the improved outputs of these two modules could be used to generate a Video Summary of the lecture that could be deemed as a trailer to the lecture.

Chapter 10

References

[1] S. Garg, “Automatic Text Summarization of Video Lectures Using Subtitles,” Advances in Intelligent Systems and Computing, pp. 45–52, 2017, doi: 10.1007/978-981-10-3779-5_7.

[2] K. Davila, A. Agarwal, R. Gaborski, R. Zanibbi, and S. Ludi, “Accessmath: Indexing and retrieving video segments containing math expressions based on visual similarity,” 2013 IEEE Western New York Image Processing Workshop (WNYIPW), Nov. 2013, doi: 10.1109/wnyipw.2013.6890981.

[3] C. Xu et al., “Lecture2Note: Automatic Generation of Lecture Notes from Slide-Based Educational Videos,” 2019 IEEE International Conference on Multimedia and Expo (ICME), Jul. 2019, doi: 10.1109/icme.2019.00159.

[4] A. Vimalaksha, S. Vinay, A. Prekash, and N. S. Kumar, “Automated Summarization of Lecture Videos, ” 2018 IEEE Tenth International Conference on Technology for Education (T4E), Dec. 2018, doi: 10.1109/t4e.2018.00034.

[5] J. Zhang, Y. Zhao, M. Saleh 2, and P. Liu, “PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization,” arXiv.org, Jul. 10, 2020. https://arxiv.org/abs/1912.08777.

[6] Y.-N. Chen, Y. Huang, C.-F. Yeh, and L. Lee, “Spoken lecture summarization by random walk over a graph constructed with automatically extracted key terms,” Interspeech 2011, Aug. 2011, doi: 10.21437/interspeech.2011-374.

[7] B. Urala Kota, K. Davila, A. Stone, S. Setlur, and V. Govindaraju, “Automated Detection of Handwritten Whiteboard Content in Lecture Videos for Summarization,” 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Aug. 2018, doi: 10.1109/icfhr-2018.2018.00013.

[8] Haojin Yang and C. Meinel, “Content Based Lecture Video Retrieval Using Speech and Video Text Information,” IEEE Transactions on Learning Technologies, vol. 7, no. 2, pp. 142–154, Apr. 2014, doi: 10.1109/tlt.2014.2307305.

[9] A. Mathur, T. Saxena, and R. Krishnamurthi, “Generating Subtitles Automatically Using Audio Extraction and Speech Recognition,” IEEE Xplore, Feb. 01, 2015. https://ieeexplore.ieee.org/document/7078779 (accessed Oct. 20, 2021).

[10] Kyoung Mu LeeYasuyuki MatsushitaJames M. RehgZhanyi Hu, “Computer Vision - ACCV 2012”, Springer Berlin Heidelberg, March 2013, doi: 10.1007/978-3-642-37431-9.

[11] Sreenivasa Rao Krothapalli, Shashidhar G. Koolagudi, “Emotion Recognition Using Vocal Tract Information”, SpingerBriefs - Emotion Recognition using Speech Features, 2013, doi: 10.1007/978-1-4614-5143-3_4.

[12] Khushboo S. Thakkar, R.V. Dharaskar, M.B. Chandak, “Graph-Based Algorithms for Text Summarization”, IEEE 3rd International Conference on Emerging Trends in Engineering and Technology, 2010, doi: 10.1109/icetet.2010.104.

[13] Majid Moradi Aliabadi,Hajar Emami, Ming Dong, Yinlun Huang, “Attention-based recurrent neural network for multistep-ahead prediction of process performance”, Elsevier - Computers and Chemical Engineering, 2020, doi: 10.1016/j.compchemeng.2020.106931.

[14] S. Smys, João Manuel R. S. Tavares Valentina, Emilia Balas, Abdullah M. Iliyasu, “Computational Vision and Bio-InspiredComputing”, Springer Science and Business Media LLC, 2020, ISBN: 978-3-030-37218-7.

final-cut

ORIGINALITY REPORT

8%	4%	2%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Caledonian College of Engineering Student Paper	1%
2	stackoverflow.com Internet Source	1%
3	Submitted to Terna Engineering College Student Paper	<1%
4	Sreenivasa Rao Krothapalli, Shashidhar G. Koolagudi. "Chapter 4 Emotion Recognition Using Vocal Tract Information", Springer Science and Business Media LLC, 2013 Publication	<1%
5	people.cis.ksu.edu Internet Source	<1%
6	old.amu.ac.in Internet Source	<1%
7	wwwlmssabyacom-raj.blogspot.com Internet Source	<1%
8	Submitted to University of Greenwich Student Paper	<1%

9	Submitted to Laureate Higher Education Group Student Paper	<1 %
10	Submitted to Kingston University Student Paper	<1 %
11	www.coursehero.com Internet Source	<1 %
12	Lecture Notes in Computer Science, 2013. Publication	<1 %
13	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
14	Submitted to University of Adelaide Student Paper	<1 %
15	www.timeshighereducation.com Internet Source	<1 %
16	Submitted to Birla Institute of Technology Student Paper	<1 %
17	K S Thakkar, R V Dharaskar, M B Chandak. "Graph-Based Algorithms for Text Summarization", 2010 3rd International Conference on Emerging Trends in Engineering and Technology, 2010 Publication	<1 %
18	Lecture Notes in Computer Science, 2007. Publication	

		<1 %
19	Submitted to Monash University Student Paper	<1 %
20	Submitted to Texas A & M University, Kingville Student Paper	<1 %
21	Submitted to The University of Memphis Student Paper	<1 %
22	www.rs-online.com Internet Source	<1 %
23	Submitted to Study Group Australia Student Paper	<1 %
24	researchbank.rmit.edu.au Internet Source	<1 %
25	www.necelectronics.de Internet Source	<1 %
26	Submitted to Indian School of Business Student Paper	<1 %
27	Submitted to Universiti Sains Islam Malaysia Student Paper	<1 %
28	Submitted to University of Malaya Student Paper	<1 %
29	Submitted to Coventry University Student Paper	<1 %

	Media LLC, 2020 Publication	
38	Gal Chechik. "Large Scale Online Learning of Image Similarity through Ranking", Lecture Notes in Computer Science, 2009 Publication	<1%
39	Submitted to School of Business & Computer Science Limited Student Paper	<1%
40	dokumen.pub Internet Source	<1%
41	livrepository.liverpool.ac.uk Internet Source	<1%
42	svnlegacy.ow2.org Internet Source	<1%
43	www.sparshsecuritech.com Internet Source	<1%
44	Imran Bashir, Amrit L. Goel. "Testing Object-Oriented Software", Springer Science and Business Media LLC, 1999 Publication	<1%
45	docplayer.net Internet Source	<1%
46	www.analyticsvidhya.com Internet Source	<1%

47	K. R. Chowdhary. "Chapter 18 Information Retrieval", Springer Science and Business Media LLC, 2020 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		