**Logo

Description automatically generated**

**COLLEGE OF ARTS AND SCIENCE**

**MASTER OF COMPUTER SCIENCE**

**SPRING SEMESTER – 2024**

**COSC5369 – GRADUATE PROJECT**

**MEETING SUMMARY GENERATOR**

**Team Members:**

**Amaresh Goshika**

**Navya Sri Sunku**

**Lakshmi Padmavathi Devara**

**Supervising Professor**

**Dr. Jane Liu**

**Faculty Committee Members**

**Dr. Md Rakib Islam**

**Dr. Xingya Liu**

**Abstract**

With the rapid advancements in technology, our world is moving quickly, and data is becoming as important as oil was in the previous century. Vast amounts of data are being collected and disseminated these days, completely changing our world. The huge digital space is home to an enormous amount of data, making it increasingly necessary to design machine-learning algorithms that are appropriate for the intended task. Numerous state-of-the-art machines have been developed for text summarization and speech recognition.

Meeting Summary Generator is the process of distilling the most important information from a meeting (or sources) to produce an abridged version for a particular user (or users) and task (or tasks).

The idea of automatic summarization work is to develop techniques by which a machine can generate summarizations that successfully imitate summaries generated by human beings. The suggested system's main goal is to replace the set of pipelined instructions with neural networks fully.

So, Summarization involves extracting the most meaningful and valuable bits of information from conversations, audio/video files, etc., briefly, and concisely. It is generally done by feature-capturing the statistical, linguistic, and sentimental traits with the dialogue structure of the conversation in question.

In this project, we will use natural language processing techniques to create precise summaries of meetings while upholding the context of the entire conversation.

Table of Contents

[1. Introduction 4](#_Toc163235817)

[1.1. Introduction to Project 4](#_Toc163235818)

[1.2. Motivation 5](#_Toc163235819)

[1.3. Need for Summary Generator 5](#_Toc163235820)

[1.4. Objectives 6](#_Toc163235821)

[2. Literature Survey 6](#_Toc163235822)

[3. Proposed System 7](#_Toc163235823)

[3.1. Introduction 7](#_Toc163235824)

[3.2. Problem Statement 7](#_Toc163235825)

[3.3. Requirement Analysis 7](#_Toc163235826)

[3.4. Requirement Elicitation 8](#_Toc163235827)

[3.4.1. Functional Requirements 8](#_Toc163235828)

[3.4.2. Non-Functional Requirements 8](#_Toc163235829)

[4. UML Diagrams 8](#_Toc163235830)

[4.1. Use Case Diagram 8](#_Toc163235831)

[4.2. Class Diagram 10](#_Toc163235832)

[4.3. Sequence Diagram 11](#_Toc163235833)

[4.4. Activity Diagram 13](#_Toc163235834)

[4.5. State Chart Diagram 16](#_Toc163235835)

[4.6. Component Diagram 18](#_Toc163235836)

[4.7. Entity-relationship Diagram 18](#_Toc163235837)

[4.8. Deployment Diagram 19](#_Toc163235838)

[5. Technologies Used 19](#_Toc163235839)

[5.1. Machine Learning 19](#_Toc163235840)

[5.2. Natural Language Processing 20](#_Toc163235841)

[5.3. HTML 20](#_Toc163235842)

[5.4. CSS 20](#_Toc163235843)

[5.5. JavaScript 21](#_Toc163235844)

[5.6. Python 21](#_Toc163235845)

[5.7. Django 22](#_Toc163235846)

[5.8. Bootstrap 22](#_Toc163235847)

[6. Implementation 22](#_Toc163235848)

[7. Source Code Screenshots 24](#_Toc163235849)

[8. Test Results and Output Screenshots: 26](#_Toc163235850)

[9. Conclusion and Future Work 32](#_Toc163235851)

[10. References 33](#_Toc163235852)

# Introduction

## Introduction to Project

These days, speech-to-text systems simulate what an interpreter would do. Since each domain has its terminologies and restrictions, its effectiveness depends on the language's capacity to comprehend grammar, identify patterns, and function in that domain. When it comes to audio-to-text conversion, phraseological units or entire sentences constitute the primary aspect units, explaining the concept in general rather than a single word.

A meeting is a gathering of two or more people to make decisions or discuss company objectives and operations. Meetings are generally conducted in person in an office, however with the rise of video conferencing technologies, participants can join a meeting from anywhere.

We often think of Meetings as discussing important information with colleagues who matter to your organization most. But sharing the discussed information effectively with all other employees is also important.

Meetings are crucial aspects of an organization. It helps you keep a tab on the processes and activities an organization performs to reach its goals and objectives. It allows you to stay updated, discuss ideas, solve problems, make collective decisions, and helps in team building.

No matter how small/big your organization’s size is, meetings will always be an important aspect to evaluate and reach your targeted goals. Since meetings cannot be ignored by the management or its employees, therefore the best way is to find a balance and conduct productive meetings that help both parties.

Effective meetings are the ones that keep you engaged, provide you with the right information, and give you a direction to achieve your goals and objectives.

## Motivation

Research on audio-to-text recognition remains active in the fields of natural language processing and information retrieval. The commercial sector is using it extensively in areas encompassing word processing tools, web-based information retrieval, text database data mining, and telephone communication. Many approaches differ in the behavior of their problem formulations. Automatic text summarization is an important step for information management tasks. It solves the problem of selecting the most important portions of the text. High-quality summarization requires sophisticated NLP techniques.

This can improve accessibility to individuals who may have hearing impairments or prefer reading over listening. It could also complement other modalities such as text and images, enabling the development of multimodal applications that provide a richer user experience. This is especially useful for educational purposes, where learners may benefit from concise summaries of lectures or presentations. This can be particularly valuable in scenarios where large volumes of audio data need to be processed regularly, such as in media monitoring or customer service analysis.

## Need for Summary Generator

The main benefit of the Meeting Summarizer is that the service allows meeting participants to focus on the actual conversation rather than worry about taking notes during meetings. The Meeting Summarizer is available on demand, day, and night, 24/7 with no need for human assistance.

Another benefit of the Meeting Summarizer is that it is immune to the risk of human error. Even the most professional note-taker can miss a few sentences or mix up a few words and those mistakes, no matter how small, could cost a company time and money. The Meeting Summarizer will take detailed, accurate notes free of errors and mistakes. The Summarization engine is based on an ensemble of models and neural networks.

The process offers several benefits, including:

* A reduction in the time required to grasp lengthy pieces of text, such as long-form articles, without losing the meaning of the text.
* Reduced prejudice compared to manual, human-operated summarization.
* Improved efficacy in indexing.
* Scalable and Quick.
* Ensures all critical information is covered.

## Objectives

1. To recognize speech in meetings and convert it to text using Natural Language Processing Techniques.

2. To summarize the text by identifying key points in the meeting using Abstraction-based summarization.

3. To be able to share the produced summary with the team members.

# Literature Survey

The author has examined numerous approaches to text summarization and sentiment analysis. These techniques are then applied to determine the opinions and judgments in the content information, such as surveys regarding movies or products. The author utilizes natural language processing (NLP) in text summarizing to assess the significance of words and phrases that can be included in the final summary. This study also includes a summary of the author's previous research on sentiment analysis and content breakdown, to identify new research areas by considering the advantages and drawbacks of the flow systems and methodologies in use today.[1]

Text summarization and speech-to-text conversion are combined in the proposed study. Applications that need a synopsis of long speeches will benefit from this hybrid approach, which is helpful for documentation. In this paper, a simple and efficient voice recognition technique was used. The voice is translated into the appropriate language, which results in a written summary. This can be used for several things, such as creating lecture notes or summarizing huge document catalogs. To confirm the effectiveness of the suggested approach, a great deal of experimentation is conducted.[2]

In the field of medical data, the paper summarizes the potential of multimodal techniques for NLP problems based on audio data. They examined the potential for using natural language processing techniques in the medical field to boost medical staff productivity and streamline manual tasks associated with completing information or converting audio to text data. Additionally, a comparative examination of the current methods was carried out to determine which are the most sophisticated and dependable for creating durable multimodal audio-to-text systems and carrying out further research. In the future, this research may be used to develop a variety of Speech-to-Text models tailored to medical specialties, which will enhance speech translation jobs, lessen workloads, and increase medical professionals' time efficiency.[3]

This essay aims to provide an overview and analysis of text summarization from its inception to the present. A detailed discussion is held regarding the two main techniques, namely extractive and abstractive summarization. There are two types of summarizing techniques: linguistic and structured. An overview of the current state of research on text summarization is given in this publication. The following processes are integrated into a computerized summarization machine: feature extraction, sentence scoring, sentence ranking, sentence segmentation, tokenization, and summary extraction. In conclusion, this study compiles the most important days and relevant searches within the text summarizing zone for evaluation and evaluation for subsequent research.[4]

# Proposed System

## Introduction

Our proposed system can be considered time-efficient it can transcribe spoken words much faster than a person typing, saving time, and increasing overall efficiency. It can be more cost-effective than hiring a human transcription service. This can be beneficial for live events, meetings, or quick content creation. Also, our proposed system tends to maintain a consistent level of accuracy throughout the transcription process.

## Problem Statement

Summaries should be distributed to everyone who was in the meeting as soon as possible, so questions can be raised and any necessary changes to the summary can be made. Any changes must be made with the approval of everyone who was in attendance. Once they are agreed to, each meeting attendee signs off on the minutes to make them an official representation of what was covered at the meeting.

Having a meeting summary is for the people to remember the meeting and to keep track of the certain points that have already been discussed. It exists to record the highlights of the meeting proceedings such as the decisions that have been made (like

motions, votes, etc.), plans, identifying problems or issues, and resolving them, reporting accomplishments of the company or organization for the past month or quarter, etc.

It is also used as a track record to see who the people who participated in the meeting are and as a source of information and reference for those who were absent during the meeting to know what tasks they were assigned to do and such. However, for this

article, it will focus more on making a meeting summary than the minutes of the meeting.

## Requirement Analysis

#### Software Requirements

Operating System : Windows 64-bit, MAC

Pre-requisite Software : Python 3.0 or above versions

Frameworks Required : Bootstrap, Django

IDE : IDLE (Python GUI)

Language : HTML, CSS, JavaScript, Django, Python

#### Hardware Requirements

Hard disk space : 80 – 100 MB for installation

RAM : Minimum 2 GB for a single-core machine

and a minimum of 1GB of memory per

processor

## Requirement Elicitation

## Functional Requirements

1. Receiving and validating user credentials.

2. Users must be able to upload the audio file of their meetings.

3. Users can convert the audio file into text format.

4. Users can summarize the text with just one click.

5. The user must be capable of sending the summary to his colleagues through emails.

## Non-Functional Requirements

1. Accuracy and Precision: the system should perform its process with accuracy and precision to avoid problems.

2. Modifiability: the system should be easy to modify, any wrong should be corrected.

3. Security: the system should be secure and save user’s privacy.

4. Usability: the system should be easy to deal with and simple to understand

5. Maintainability: the maintenance group should be able to fix any problem that occurs suddenly.

6. Speed and Responsiveness: Execution of operations should be fast.

# UML Diagrams

## Use Case Diagram

The use case diagram represents the pictorial relationship between users and the modules. Using graphics, we are defining the entire flow of the project.

A diagram of a user

Description automatically generated

Figure: Use case for Account Sign-Up and Login

A diagram of a person with several circles

Description automatically generated

Figure: Use case diagram for Summarizer

A diagram of a person

Description automatically generated

Figure: Use case diagram for Admin

## Class Diagram

1. It is mainly designed for a naive user to understand the details of the project and their Relationships.
2. Improves the understandability of a non-technical user on seeing the class diagram.
3. We describe each use case to show more details about the system.

A diagram of a data flow

Description automatically generated with medium confidence

Figure: Class Diagram of ‘Meeting Summary Generator’ system

## Sequence Diagram

Sequence diagrams are a popular dynamic modeling solution because they specifically focus on lifelines or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.

A screenshot of a grid

Description automatically generated

Figure: Sequence Diagram for SignUp and Login

A screenshot of a diagram

Description automatically generated

Figure: Sequence Diagram showing the Summarization Process

## Activity Diagram

The activity diagram represents an important behavioral diagram in the UML diagram to describe dynamic aspects of the face recognition-based attendance system. An activity diagram is essentially an advanced version of a flow chart that models the flow from one activity to another activity.

A diagram of a login page

Description automatically generated

Figure: Activity Diagram for Account Creation

A diagram of a program

Description automatically generated

Figure: Activity Diagram for user profile creation

A diagram of a program

Description automatically generated

Figure: Activity Diagram for summary creation and email sending

## State Chart Diagram

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It’s a behavioral diagram and it represents the behavior using finite state transitions.

A diagram of a computer program

Description automatically generated

Figure: State chart Diagram for website

## Component Diagram

The component diagram contains components and dependencies. Components represent the physical packing of the module. Whereas dependencies between the components show how changes made to one component may affect the other components. It also shows the interfaces used by the components to communicate with each other.

A diagram of a block diagram

Description automatically generated

## Entity-relationship Diagram

The entity relationship diagram shows us the association of relationships between the entities. They form a one-to-one relation between the required methods of the project.

A diagram of a workflow

Description automatically generated

## Deployment Diagram

A diagram of a computer network

Description automatically generatedThe deployment diagram contains nodes and connections. A node usually represents a piece of hardware in the system.

# 5. Technologies Used

## 5.1. Machine Learning

Machine learning (ML) is the study of computer algorithms that improve automatically through experience and using data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as ”training data”, to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory, and application domains to the field of machine learning.

The discipline of machine learning employs various approaches to teach computers to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithm it uses to determine correct answers. For example, to train a system for the task of digital character recognition, the MNIST dataset of handwritten digits has often been used.

## 5.2. Natural Language Processing

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, and how to program computers to process and analyze large amounts of natural language data. The result is a computer capable of ”understanding” the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

Challenges in natural language processing frequently involve speech recognition, natural language understanding, and natural language generation.

## 5.3. HTML

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. The inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

## 5.4. CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enabling multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enables the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

## 5.5. JavaScript

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object orientation, and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. Most of the websites use it for client-side page behavior, and all major web browsers have a dedicated JavaScript engine to execute it.

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but they are now core components of other runtime systems, such as Node.js and Deno. These systems are used to build servers and are also integrated into frameworks, such as Electron and Cordova, for creating a variety of applications.

## 5.6. Python

Python is an interpreted, high-level, general-purpose programming language. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

Python features a comprehensive standard library and is referred to as ”batteries included”. Python interpreters are available for many operating systems. C-Python, the reference implementation of Python, is open-source software and has a community-based development model. Python and C-Python are managed by the non-profit Python Software Foundation. Python is a multi-paradigm programming language.

Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including meta-programming and meta-objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

## 5.7. Django

Django is a Python-based free and open-source web framework that follows the model-template-views (MTV) architectural pattern. It is maintained by the Django Software Foundation (DSF). The primary goal is to ease the creation of complex, database-driven websites.

The framework emphasizes reusability and "pluggability” of components, less code, low coupling, rapid development, and the principle of don’t repeat yourself. Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update, and delete interface that is generated dynamically through introspection and configured via admin models.

## 5.8. Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. Bootstrap is among the most starred projects on GitHub, with more than 142,000 stars, behind free CodeCamp (almost 312,000 stars) and marginally behind the Vue.js framework.

# Implementation

* + 1. Introduction

This project work is implemented using HTML, CSS, JavaScript, Python, and Django. This system is built using the Django web application framework. It simplifies the development process of complex, database-driven web applications. Its well-designed framework includes three major parts: model, view, and template.

Our ’Meeting Summary Generator’ site consists of an Accounts section, file handling section, summarization section, and sending mails section.

* + 1. Installing Python

Django is a Python web framework, thus requiring Python to be installed on your machine.

To install Python on your machine, go to https://python.org/downloads/. The website should offer you a download button for the latest Python version. Download the executable installer and run it. Check the boxes next to “Install launchers for all users (recommended)” then click “Install Now”.

After installation, open the command prompt and check that the Python version matches the version you installed by executing: ”...\>py –version.”

* + 1. About pip

pip is a package manager for Python and is included by default with the Python installer. It helps to install and uninstall Python packages (such as Django!). For the rest of the installation, we’ll use pip to install Python packages from the command line.

* + 1. Setting up a virtual environment

It is best practice to provide a dedicated environment for each Django project you create. There are many options to manage environments and packages within the Python ecosystem, some of which are recommended in the Python documentation. Python itself comes with venv for managing environments which we will use for this guide.

* + 1. Installing Django

Django can be installed easily using pip within your virtual environment.

In the command prompt, ensure your virtual environment is active, and execute the following command:

py -m pip install Django.

This will download and install the latest Django release. After the installation has been completed, you can verify your Django installation by executing ” django-admin –version ” in the command prompt.

* + 1. Creating the Users app

Now that the environment – a “project” – is set up, we are set to start doing work.

Each application that we write in Django consists of a Python package that follows a certain convention. Django comes with a utility that automatically generates the basic directory structure of an app, so we can focus on writing code rather than creating directories.

Django apps can live anywhere on your Python path. we will create a posts app in the same directory as the manage.py file so that it can be imported as its top-level module, rather than a sub-module of Summarizer.

* + 1. Writing templates

We can include a separate Template (HTML) in the Django project. A template is more than just a simple HTML file that allows you to insert Django code inside it in addition to serving as a means of separating the HTML. We need to create a folder that will hold our templates. Create a folder named templates in the same directory as the manage.py files. We can now add our HTML files to that template folder.

* + 1. Static files

All your graphics and CSS are in static files. Their content is consistent for each user and is not contingent on the circumstances of the request. Django will automatically locate any "static" directories within the folders of your apps. Then, its contents can be used as static files.

* + 1. PyAudio

A cross-platform audio I/O library called PortAudio has bindings for Python users because of PyAudio. It allows you to use Python to play and record audio on a variety of platforms, including GNU/Linux, Microsoft Windows, and Apple macOS. PyAudio is often used in conjunction with other libraries like Speech Recognition to build voice recognition systems and various audio-processing applications.

* + 1. Speech Recognition

Speech Recognition is a library that allows developers to integrate speech recognition into their applications easily. Recognizing speech requires audio input, and Speech Recognition makes retrieving this input easy. Instead of having to build scripts for accessing microphones and processing audio files from scratch, Speech Recognition will have you up and running in just a few minutes. It supports both Python 2 and Python 3 and can be used to transcribe audio files or recognize speech in real-time. It also includes a variety of features such as text-to-speech and keyword spotting.

* + 1. Librosa

Librosa is a powerful Python library that offers a wide range of tools and functionalities for handling audio files. It helps in loading audio files, extracting the characteristics of the music, and visualizing audio data. With the help of librosa, working with audio in Python has become straightforward.

# Source Code Screenshots

A computer screen shot of text

Description automatically generated

Audio to Text Convertor

A computer screen shot of a program code

Description automatically generated

Generated Text Summarizer

A computer screen with colorful text

Description automatically generated

View Summarized Text

A computer screen shot of text

Description automatically generated

Audio Upload Function

A screen shot of a computer program

Description automatically generated

Sending email function

A screen shot of a computer

Description automatically generated

URL Paths and Patterns

# Test Results and Output Screenshots:

* 1. Test Setup

Registered username and password with an audio file in WAV format and a list of receiver email addresses.

* 1. Test Procedure

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Step  No. | Action | | | Input | | | Expected Output |
| 1 | Open http://127.0.0.1:8000/  in browser | | | None | | | Welcome page |
| 2 | Click | on | Login | None | | | Login - Login page |
| 3 | Enter username and  password | | | Username  word | and | pass- | Valid - Home page,  invalid - login page |
| 4 | Click | on | Upload | None | | | Upload page |
| 5 | Enter audio name, description, audio file | | | audio file and its de-  scription | | | Recognize page |
| 6 | Click | on | Recognize | None | | | Summarize page |
| 7 | Click | on | Summarize | None | | | Send emails page |
| 8 | Click | on | edit | a list of receiver emails | | | receiver emails update  page |
| 9 | Click | on | send mails | None | | | Mails sent page |
| 10 | Click | on | Home | None | | | Home page |

* 1. Test Results

This is the Landing Page of our proposed system.

**A screenshot of a computer screen

Description automatically generated**

Once you click on Sing up, this window will pop up to create an account

A screenshot of a computer

Description automatically generated

After successful completion of Sign Up, you will be navigated to the Login Page.

A screenshot of a computer

Description automatically generated

Now the user is landed into the Home Page of our proposed system.

**A screenshot of a computer

Description automatically generated**

Here, We have included “Record Audio” option, to record your meeting conversations.

A screenshot of a computer

Description automatically generated

Then, the user could upload the recorded audio in the Upload Page by choosing appropriate file type(WAV).

A screenshot of a computer

Description automatically generated

Our Proposed system also display the selected audio file as preview, before you click upload button.

A screenshot of a computer

Description automatically generated

The user has access to adjust the playback according to their requirement.

A blue rectangle with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

Once you choose required options, the page loading for the audio file to recognize the speech and convert into summarized text.

A screenshot of a computer

Description automatically generated

After successful recognition, the Summarized Text will be generated as show below:

A screenshot of a computer

Description automatically generated

Also, the user can send the Summarized Text to this Team Members through email.

A screenshot of a computer

Description automatically generated

# Conclusion and Future Work

Our experiment results depict there is a significant benefit of making extensive use of the website we provided to constrain the problem of summarizing meeting conversations. The system can provide a platform that fulfills our objectives. Therefore, we are, for example, able to log in to our Summarizer, upload the meeting audio, and summarize the meeting. The System proposed in this report works using the most popular software languages and most effective web development tools and frameworks. Thus, making it even more efficient to work with. Though this model is implemented in a local server, its domain can be extended to a higher level.

This can be extended by deploying it on a real-time server and made available to the public as all other sites. This model-based approach will also be able to make effective use of spatial context which will enable the system to detect the spam content and keep only the reliable information. Hence, this model has a great scope and potential in the future and will be a win situation when implemented.

The success of this research project is significant given the broad use of audiobooks in literacy and library programs across the United States. Teachers and school librarians may also use these findings as a rationale for adding audiobooks to the list of reading strategies used successfully with people who are struggling with hearing impairments or prefer reading over listening.

# References

[1] K.P. Vijayakumar, Hemanth Singh, Animesh Mohanty ”Real-Time Speech-To-Text/Text-To-Speech Converter with Automatic Text Summarizer using Natural Language Generation and Abstract Meaning Representation”, Volume-9 Issue-4, April 2020, Page no:2361-236.

[2] A. Vinnarasu, Deepa V. Jose, “Speech-to-text conversion and summarization for effective understanding and documentation”, Vol. 9, No. 5, October 2019, pp. 3642~3648.

[3] Oleh Basystiuk, Nataliia Melnykova, “Multimodal Approaches for Natural Language Processing in Medical Data”, December 2022, Conference: IDDM 2022 Informatics & Data-Driven MedicineAt: Lyon, FranceVolume: Vol-3302.

[4] Keerthana P, “Automatic Text Summarization using Deep Learning”, Volume: 7, Issue: 4, April 2021, Journal DOI: 10.36713/epra2013, SJIF Impact Factor 2021: 8.047, ISI Value: 1.188.

[5] Natural Language Processing Techniques Applied in Information Retrieval-Analysis and Implementation in Python, TulikaNarang, International Journal of Innovations Advancement in Computer Science IJIACS ISSN 2347 – 8616 Volume 5, Issue 4 April 2016.

[6] 7 ways Audio books benefit students who struggle with reading. (n.d.).Retrieved March 09, 2021, from: https://learningally.org/Solutions-for School/7-Ways-Audio Books-Benefit-Students Who Struggle with-Reading.

[7] Anurag Shandilya, Kripabandhu Ghosh, Saptarshi Ghosh “Fairness of Extractive Text Summarization”, ACM 2018.

[8] Lili Wan “Extractive Algorithm of English Text Summarization for English Teaching” IEEE 2018.

[9] <https://www.ibm.com/topics/natural-language-processing>

[10]<https://aws.amazon.com/what-is/nlp/#:~:text=from%20call%20analytics.-,How%20does%20NLP%20work%3F,models%20to%20process%20human%20language.&text=Computational%20linguistics%20is%20the%20science,with%20computers%20and%20software%20tools>.