DECLARATION

I hereby declare that the work presented in this report entitled “TEXT TO SPEECH”, was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/ sources. I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

NAME: ANSHUL YADAV

ROLL NO: 2100290140030

NAME: DIVYA GUPTA

ROLL NO: 2100290140062

NAME: KM. PURNIMA

ROLL NO: 2100290140075

SIGNATURE

CERTIFICATE

Certified that ANSHUL YADAV (2100290140030), DIVYA GUPTA (2100290140062), KM PURNIMA (2100290140075) have carried out the project work having "TEXT TO SPEECH" for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

ANSHUL YADAV (2100290140030)
DIVYA GUPTA (2100290140062)
KM PURNIMA (2100290140075)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Sangeeta Arora (Associate Professor)
Department of Computer Applications
KIET Groupof Institutions, Ghaziabad

Signature of Internal Examiner

Signature of External Examiner

Dr. Arun Tripathi
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

Text - to - speech conversion software project is windows-based application that reads a text file to the user. The software reads a text file and associated pronunciations in its temporary database. The program then reads an entire word to the user. The software can be effectively used to help read the text document for the user so that the user does not constantly need to look at the screen and read the entire document.

Text to speech converter is a recent software project that allows even the visually challenged to read and understand various documents. The blinds cannot read a document, so this software can be an assistant to them who would read out those documents for them. It can also be a great help for those who cannot speak. The person can simply type what he/she wants to say, and the software would give a voice to them by speaking what they wanted to say. So, this software is not just an advancement towards the future development but also a boon for those who cannot speak and see. The following application can be used to convert text to audio using Tinker and python files, functions, and definitions. The main package used in this application converter is pyttsx3. Pyttsx3 is a python library used for text to speech conversions. This isthe reason which helps the machine to speak to us.

ACKNOWLEDGEMENT

Success in life is never attained single handedly. My deepest gratitude goes to my project supervisor, **Dr. Sangeeta Arora** for her guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions. Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi**, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions. Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions. Finally, my sincere thanks go to my family members and all those who have directly admin directly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

ANSHUL YADAV

DIVYA GUPTA

KM PURNIMA

TABLE OF CONTENTS

List of Figures

CHAPTER 1

INTRODUCTION

Text-to-speech and related read audio tools are being widely implemented in an attempt to assist students' reading comprehension skills.PDF to the audio system is a screen reader application designed and constructed for an effective audio communication system.

Hence, there is a need to make it more accessible to readers on-screen through audio. The text to Audio Converter project provides an alternative to access the books or articles for the blind, lazy, readers, and others. Using this text to Audio Converter the user will be able to listen to his\her favorite article and can do their daily routine.

The following application can be used to convert text to audio using Python predefined library like pyttsx3 and using some Tinker files. In the current busy routine people do not have time to take a book and spend time reading it, instead, everyone needs alternative access to read the content. If a person is traveling, he\she cannot read a book, instead of reading, they can listen to it. Reading stories or essays or any text can be arduous however an audiobook would make the task easy, by reading the text. However, an audio reading of the text is convenient and does not require much concentration as reading requires. When a person tends to read a book, it requires to invest his/her time in reading. Whereas the audiobook makes the task easy, and the user can perform their own task as well as listening to the audio.

In this project, we have implemented a simple text to audio converter using python. When we compare with the current features present in a normal audiobook converter, they convert texts into speech, and they have volume controls with single voice conversion (either male or female). Only a single choice is given to the user in case of voice modification. They provide the play and save options. It contains both the male and female voice modifying features, which helps the user to change with a single click on the button. The rate of speed of voices can also be changed (fast, normal, and slow) for better clarifications. Volume controlling features are also present.

1.1 Project Description

Text- to- speech program that lets you type in any English text and then plays it as an audio stream. Instantly convert desired text to audio. Supported language: English. The language used for the project text to speech conversion is python. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereasother languages use punctuation, and it has fewer syntactical constructions than other languages:

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP.
- Python is Interactive− You can sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented− Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language− Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python Features:

Python's features include −

- Easy-to-learn− Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- Easy-to-read− Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain− Python's source code is fairly easy-to-maintain.
- A broad standard library− Python's bulk of the library is very portable and cross- platform compatible on UNIX, Windows, and Macintosh.
- Interactive Mode− Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- Portable− Python can run on a wide variety of hardware platforms and has the same interface on allplatforms.
- Extendable− You can add low-level modules to the Python interpreter. These modules Enables programmers to add to or customize their tools to be more efficient.
- GUI Programming− Python supports GUI applications that can be created and ported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the

Window system of Unix.

Different libraries used in the project are:

Tkinter - Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit and is Python's de facto standard GUI. Tkinter is included with standard Linux, Microsoft Windows, and Mac OS Xinstalls of Python. The name Tkinter comes from Tk interface. Tkinter was written by Fredrik Lundh. Tkinter is free software released under a Python license. As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application. Python 2.7 and Python 3.1incorporate the "themed Tk" ("ttk") functionality of Tk 8.5. This allows Tk widgets to be easily themed to look like the native desktop environment in which the application is running, thereby addressing a long-standing criticism of Tk (and hence of Tkinter). There are several popular GUI library alternatives available, such as wxPython, PyQt(PySide), Pygame, Pyglet, and PyGTKCreating a GUI application using Tkinter is an easy task.

Pyttsx3 – Pyttsx is a good text to speech conversion library in python, but it was written only in python2 until now! Even some fair amount of googling didn’t help much to get tts library compatible with pyton3.There is however, one library gTTS which works perfectly in python3 but it needs internet connection to work since it relies on google to get the audio data. But Pyttsx is completely offline and works seamlessly and has multiple its engine support. The codes in this repos are slightly modified version of the pyttsx module of python 2.x and is a clone from Weston pace’s repo. The purpose of creating this repo is to help those who want to have an offline its lib for Python3 and don’t want to port it from python2 to python3 themselves. Usage

```
import pyttsx3
engine = pyttsx3.init()
engine.say(“I will speak this text”)
engine.runAndWait()
```

1.2 Project Scope

Text to speech converter is a recent software project that allows even the visually challenged to read and understand various documents. The blinds cannot read a document, so this software can be an assistant to them who would read out those documents for them. It can also be a great help for those who cannot

speak. The person can simply type what he/she wants to say, and the software would give a voice to them by speaking what they wanted to say. The user just must select the Interactive mode and then write what he wants to say in the text area and then he can easily express what he wanted to say by simply clicking the convert button. So, this software is not just an advancement towards the future development but also a boon for those who cannot speak and see. This technology can also be utilized for various purposes, e.g., car navigation, announcements in railway stations, response services in telecommunications, and e-mail reading. Thus, if we think more innovatively, we can easily get more applications out of it.

1.3 Hardware Requirements:

- Central Processing Unit (CPU) - Intel Core i5 6th gen or AMD processor equivalent.
- RAM - 8 GB minimum, 16 GB or higher is recommended.
- Operating System (OS) - Microsoft Windows 10
- Storage – 8 GB

CHAPTER 2

LITERATURE REVIEW

2.1 Literature reviews on existing systems

Large acoustic inventories must be used to produce speech close to natural quality. However, the concatenation cost space grows exponentially with the number of acoustic units in the acoustic inventory, increasing the latency of the unit selection algorithm, making algorithms unusable in real-time end-to-end systems. Even when data compression techniques are introduced, the model size is still high, representing a challenge for end-to-end systems. Thus, in this paper, we propose representing the concatenation cost space using LSTM (Long Short-Term Memory). The results show a 90% reduction in the size of the data space compared to all our previous techniques, and by an over 70% decrease in the look-up time. The proposed LSTM-based compression increases the responsiveness of the corpus-based text-to-speech systems significantly while keeping the overall speech quality at the same level. In this paper we describe a model for compressing huge concatenation cost space with an LSTM-based compression model. The proposed model reduces the data space by 90% and increases the responsiveness of the corpus-based text-to-speech system by over 70%. The capacity of the LSTM model represents the main limitation of the proposed solution. As a result, we limited the concatenation cost floating-point numbers to four-digit representations. Although this represents a lossy data compression solution, the experiments carried out show that the proposed compression does not degrade the quality of the synthesized speech. [1]

Agricultural Text-to-Speech (TTS) has attracted increasingly more attention. The application of agricultural TTS and its problems are analyzed in this paper, and the traditional framework of the TTS system and its key technologies, i.e., text analysis, rhythm generation and speech synthesis are discussed. Furthermore, two advancements in agricultural TTS, the word segmentation of analytic language and the deep learning-based end-to-end TTS system, are detailed summarized. Based on the characteristics of agriculture, some appealing research directions are pointed out: how to improve the training speed and synthesis speed of the deep learning models is still the focus; the study on the approaches of weakly-supervised learning in TTS is in fancy; and research on the real-time and high-quality speech synthesis that can be deployed in mobile devices is a key point of agricultural TTS research. Important indicators to measure the performance of the TTS include intelligibility and naturalness. With the development of deep neural network technology, the accumulation of massive textual speech data resources and the improvement of computing power, the end-

to-end system based on deep learning has become the mainstream, and the intelligibility and naturalness of synthetic speech have made breakthrough progress in being comparable to natural human speech. [2]

This paper presents CAMNet, a controllable acoustic model for efficient, expressive, high-quality TTS. CAMNet is based on deep convolutional TTS (DCTTS), a state-of-art acoustic model which is efficient and produces neutral speech. DCTTS was first adapted to generate Bark cestrum acoustic features to integrate well with the LPCNet (linear prediction coefficient) neural vocoder and to remove the reduction factor which demanded the presence of an up-sampling network before the vocoder – i.e., the CAMNet output can be directly fed into LPCNet. Next, style transfer functionality was added by means of a novel characterization of the prosodic information from the Bark cestrum acoustic features and a new approach to inject this information into the convolutional layers. In summary, CAMNet is an efficient, expressive, and controllable acoustic model which yields acoustic features that can be used by the LPCNet vocoder to produce high-quality speech. It improves upon the efficiency of DCTTS and provides additional modules to enable either style transfer or interpolation between selected expressive speech references. Controllability is achieved by means of a VAE module which generates a latent space with certain disentangled components for two speech generative factors: pitch and speaking rate. [3]

This paper presents a classroom case study where one such intervention, that of text-to-speech technology (TTST) was introduced as day-to-day classroom practice. TTST most often falls under the heading of special education or assistive technology, where its primary purpose is to support students who struggle to read or have a reading disorder that was not preventable nor was it alleviated by traditional interventions. TTST, however, when offered as a choice, allows students to deepen their understanding of reading, how TTST can be used as a reading support, what TTST can and cannot do, and what happens when certain reading skills and strategies break down. Ultimately, when provided with TTST as part of a comprehensive reading approach, students naturally integrated it into the ongoing development of metacognitive strategies, student dialogue and collaboration, spontaneous reader response, and most importantly, self-efficacy and self- advocacy. In the end, students agree that for many, TTST would be a nuisance, for some, a legitimate and equitable choice, and for a few, TTST, will be a lifelong tool. Implications for classroom practice in terms of parent, teacher, and student implementation are discussed. [4]

The presented TTS (text-to-speech) application is an auxiliary tool for language teaching. It utilizes computer-generated voices to simulate dialogs representing different grammatical problems or speech contexts.

The software is capable of producing as many examples of dialogs as required to enhance the language learning experience and thus serve curriculum representation, grammar contextualization and pronunciation at the same

time. It is designed to be used on a regular basis in the language classroom and students gladly write materials for listening comprehension tasks with it. A pilot study involving 26 students (divided into control and trial groups) practicing for their school-leaving exam, indicates that computer- generated voices are adequate to recreate audio course book materials as well. The voices used were able to involve the students as effectively as if they were listening to recorded human speech. The idea of building an entire language course around this technology is a dream for the future that deserves serious consideration; the use of simplified or full-version books is another interesting prospect, because of course they usually contain interactions between characters, but producing stimulating and representative dialogs for language learners is an important first step. [5]

In this paper, we present open-source tools that facilitates the use of controllable TTS systems in experiments, towards the democratization of TTS systems across domains. ICE-Talk is a web-based GUI that allows the use of a TTS system with controllable parameters via a text field and a clickable 2D plot. It enables the study of latent spaces for controllable TTS. A tool to design a perceptual experiment is provided and consists of three steps: pre-synthesizing samples covering the 2D plot representing controllable dimensions, including this interface inside a template question, and integrate it in a Mechanical Turk system called Turkle. This tool will enable the study and assessment of the controllability of Controllable Expressive TTS systems and how participants behave and feel with the system. [6]

Neural sequence-to-sequence text-to-speech synthesis (TTS) can produce high-quality speech directly from text or simple linguistic features such as phonemes. Unlike traditional pipeline TTS, the neural sequence-to-sequence TTS does not require manually annotated and complicated linguistic features such as part-of- speech tags and syntactic structures for system training. However, it must be carefully designed and well optimized so that it can implicitly extract useful linguistic features from the input features. In this paper we investigate under what conditions the neural sequence-to-sequence TTS can work well in Japanese and English along with comparisons with deep neural network (DNN) based pipeline TTS systems. Unlike past comparative studies, the pipeline systems also use neural autoregressive (AR) probabilistic modeling and a neural vocoder in the same way as the sequence-to-sequence systems do for a fair and deep analysis in this paper. In this paper, we investigated under what conditions sequence-to-sequence based text-to-speech (TTS) could work well given simple input such as text or phonemes in Japanese and English, along with comparing them with comparable deep neural network (DNN) based pipeline TTS systems using complex full-context labels. We empowered models of our sequence-to-sequence based TTS methods instead of enriching linguistic features to see how much enforced models could overcome the linguistic feature limitation. [7]

The rapid progress of modern AI tools for automatic speech recognition and machine translation is leading to a progressive cost reduction to produce publishable subtitles for educational videos in multiple languages. Similarly, text-to-speech technology is experiencing large improvements in terms of quality, flexibility, and capabilities. State-of-the-art systems are now capable of seamlessly dealing with multiple languages and speakers in an integrated manner, thus enabling lecturer's voice cloning in languages she/he might not even speak. This work is to report the experience gained on using such systems at the Universität Polytechnical de Valencia (UPV), mainly as a guidance for other educational organizations willing to conduct similar studies. It builds on previous work on the UPV's main repository of educational videos, Media UPV, to produce multilingual subtitles at scale and low cost. Here, a detailed account is given on how this work has been extended to also allow for massive machine dubbing of Media UPV. In a later stage, this should be followed by adapting our production pipeline to the streaming setup, to extend its applicability to live lecturing, either online or not, delivered under reasonable acoustic conditions. We think that these goals will be achieved sooner rather than later. More importantly, we are convinced that similar developments can be easily made at other educational organizations, either using in-house systems as we do, or third-party systems, if available, under reasonable cost and usage conditions. More generally, multilingual TTS technologies have a large potential for a variety of non-educational application scenarios in further fields such as the media industry, academic conferences and professional translation and interpreting. [8]

Recent trends in voice bot application development have enabled utilization of both speech-to-text and text-to-speech (TTS) generation techniques. To generate a voice response to a given speech, one needs to use a TTS engine. The recently developed TTS engines are shifting towards end-to-end approaches utilizing models such as Tacotron, Tacotron-2, WaveNet, and WaveGlow. The reason is that it enables a TTS service provider to focus on developing training and validating datasets comprising of labelled texts and recorded speeches instead of designing an entirely new model that outperforms the others which is time-consuming and costly. In this context, this work introduces the first Vietnamese FPT Open Speech Data (FOSD)-Tacotron-2-based TTS model dataset. This dataset comprises of a configuration file in 3 9 *.json format; training and validating text input files (in *.csv format); a 225,000-step checkpoint of the trained model; and several sample generated audios. The published dataset is extremely worth for serving as a model for benchmarking with other newly developed TTS models / engines. In addition, it opens an entirely new TTS research optimization problem to be addressed: How to effectively generate speech from text given: a black box TTS (trained) model and its training and validation input texts. [9]

Different technical equipment in the process of education is important and can help enough in better understanding the material that has been lectured. But they are of particular importance when we are dealing with people who have various physical disabilities. In this paper we present the idea of generating speech from written texts in Albanian, which can be used by students who have difficulties with their eye vision. But this idea could also be used by students or ordinary people, who prefer while learning the material, instead of reading, to hear the desired text in the computer. Sufficient condition in this case is the learning text to be written in a computer by using any text processor. It is evident that currently there are applications for speech generation based on written texts. But they have limited usage and are dedicated only to a few most popular languages in the world. In this context, for the Albanian language yet there are no such applications. Therefore, research in this field is reasonable and necessary for the Albanian language to ask possible options for the conversion of written texts in Albanian language into spoken Albanian. Generating speech from written texts, beside the fact that it can be implemented in the process of learning and education, also carries within the human character, which may be helpful to people with disabilities and those with difficulties in vision and talk. [10]

Interviews using a human interviewer can be used to investigate product evaluations, however they can cause interviewer biases. Speech-to-Text and Text-to-Speech (StT&TtS) techniques can potentially reduce these biases. The study compared the difference in interviewer biases when a human interviewer is used or when AI Powered StT&TtS techniques are used. 53 panelists participated in two conditions: an interview with a human interviewer and a survey using StT&TtS techniques. In each condition two sweet snacks were evaluated: a regular variant (no health claim) and a new variant (same regular product but with health claim) and panelist were asked for their preference. Food consumption behavior was also assessed. The new variant was rated higher in the human interview condition than in the StT&TtS condition ($p = 0.001$), the regular variant was not rated differently ($p = 0.748$). Within the human interviewer condition, the new variant was rated higher ($p = 0.021$) and more preferred ($p < 0.001$) than the regular variant. These effects were not seen within the StT&TtS condition ($p = 0.782$ and $p = 0.492$). Food consumption behavior ratings did not differ between conditions ($p = 0.685$). In conclusion, using a human interviewer in sensory and consumer research can cause an interviewer bias, this bias can be limited by using Speech-to-Text and Text- to-Speech techniques. [11]

In last half decade an increasing number of works published has manifested the tremendous progress in multimodal sentiment analysis. In real-life communication, people are spontaneously modulating their tone.

to accentuate specific points or to express their sentiments. This research work introduces a supervised fuzzy rule-based system for multimodal sentiment classification, that can identify the sentiment expressed in video reviews on social media platform. It has been demonstrated that multimodal sentiment analysis can be effectively performed by the joint use of linguistic and acoustic modalities. In this paper computation of the sentiment using an ingenious set of fuzzy rules has been applied to label the review into positive or negative sentiment. The confidence score from supervised Support Vector Machine (SVM) classification of text and speech cues is considered as the input variable for the fuzzy rules. In last half decade an increasing number of works published has manifested the tremendous progress in multimodal sentiment analysis. In real-life communication, people are spontaneously modulating their tone to accentuate specific points or to express their sentiments. This research work introduces a supervised fuzzy rule-based system for multimodal sentiment classification, that can identify the sentiment expressed in video reviews on social media platform. It has been demonstrated that multimodal sentiment analysis can be effectively performed by the joint use of linguistic and acoustic modalities. In this paper computation of the sentiment using an ingenious set of fuzzy rules has been applied to label the review into positive or negative sentiment. The confidence score from supervised Support Vector Machine (SVM) classification of text and speech cues is considered as the input variable for the fuzzy rules. [12]

Prosody plays an important role in improving the quality of text-to-speech synthesis (TTS) system. In this paper, features related to the linguistic and the production constraints are proposed for modeling the prosodic parameters such as duration, intonation, and intensities of the syllables. The linguistic constraints are represented by positional, contextual, and phonological features, and the production constraints are represented by articulatory features. Neural network models are explored to capture the implicit duration, F_0 and intensity knowledge using above mentioned features. The prediction performance of the proposed neural network models is evaluated using objective measures such as average prediction error (μ), standard deviation (σ) and linear correlation coefficient. The prediction accuracy of the proposed neural network models is compared with other state-of-the-art prosody models used in TTS systems. The prediction accuracy of the proposed prosody models is also verified by conducting listening tests, after integrating the proposed prosody models to the baseline TTS system. The prediction accuracy of the models is further improved by imposing the prosodic constraints from the developed models. The evaluation of quality of TTS system is carried out by integrating the proposed duration, intonation, and intensity models into the baseline TTS system. In this work, all prediction models are developed using single feedforward neural network. Hierarchical neural networks (HNN) may be explored in future for improving the accuracy of prediction. [13]

In this paper, we have explored the framework of compressed sensing (CS) and sparse representation (SR) to reduce the footprint of unit selection based speech synthesis (USS) system. In the CS based framework, footprint reduction is achieved by storing either CS measurements or signs of CS measurements, instead of storing the raw speech waveforms. For efficient reconstruction using CS measurements, the speech signal should have a sparse representation over a predefined basis/dictionary. Hence, in this work, we have also studied the effectiveness of sparse representation for compressing the speech waveform. The experimental results are demonstrated using an analytical dictionary (DCT matrix), and several learned dictionaries, derived using K-singular value decomposition (KSVD), method of optimal directions (MOD), greedy adaptive dictionary (GAD) and principal component analysis (PCA) algorithms. To further increase compression in SR based framework of footprint reduction, the significant coefficients of sparse vector are selected adaptively, based on the type of speech segment (e.g., voiced, unvoiced etc.). This work is focused on the feasibility of CS and SR to reduce the footprint of USS system. For efficient CS/SR based speech processing, the dictionary should be able to model different variations in the signal under various contexts. Hence, the proposed method employs dictionaries learned for individual phonemes and compression is achieved by storing either CS measurements, sign of CS measurements or significant coefficients of SR. The behavior of the SR obtained for different speech regions (voiced, unvoiced and transition) is also exploited further in reducing the memory requirements, without degradation in the perceivable speech quality. Experimental results demonstrate the effectiveness of these methods and confirms that the performance in terms of speech quality and storage requirements is comparable or better than the existing state-of-the-art low footprint Flite USS system. [14]

2.2 Proposed System

Text to speech is a process to convert any text into voice. Text to speech project takes words on digital devices and convert them into audio with a button click. Text to speech python project is very helpful for people who are struggling with reading.

In this project, we add a message which we want to convert into voice and click on play button to play the voice of that text message. To implement this project, we will use the basic concepts of Python, Tkinter, and play sound libraries. In this project it contains both the male and female voice modifying feature, rate ofspeed of voices can also be changed for better clarification.

CHAPTER 3

FEASIBILITY STUDY

A feasibility study is a detailed analysis that considers all the critical aspects of a proposed project in order to determine the likelihood of it succeeding. Success in business may be defined primarily by return or investment, meaning that the project will generate enough profit to justify the investment. However, many other important factors may be identified on the plus or minus side, such as community reaction and environmental impact .Before starting the project, feasibility study is carried out to measure the viable of the system. Feasibility is necessary to determine is creating a new or improved system is friendly with the cost, benefits, operation, technology and time. Following feasibility is given below:

Why Do I Need a Feasibility Study?

Feasibility studies are important for a communications service provider to determine whether your broadband project will succeed or not. It should be the first action taken when wanting to begin a new project. It is one, if not the most important factor in determining whether the project can and should move forward. Also, if you are applying for broadband loans and grants, a feasibility study is normally required.

3.1 Technical Feasibility

Technical feasibility is carried out to determine whether the project is feasible in terms of software, hardware, personnel, and expertise, to handle the completion of the project. It considers determining resources for the proposed system. As the system is developed using python, it is platform independent. Therefore, the users of the system can have average processing capabilities, running on any platform. The technology is one of the latest hence the system is also technically feasible.

3.2 Economic Feasibility

Economic feasibility defines whether the expected benefit equals or exceeds the expected costs. It is also commonly referred to as cost/benefit analysis. The procedure is to determine the benefits and the savings expected from the system and compare them with the costs. A proposed system is expected to outweigh the costs. This is a small project with no cost for development. The system is easy to understand and use. Therefore, there is no need to spend on training to use the system. Hence, the project could have economicbenefits in the future.

3.3 Legal Feasibility

In Legal Feasibility study project is analyzed in legality point of view. This includes analyzing barriers of legal implementation of project, data protection acts or social media laws, project certificate, license, copyright etc. Overall, it can be said that Legal Feasibility Study is study to know if proposed project conform legal and ethical requirements.

A feasibility study might reveal the organization's ideal location isn't zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

3.4 Schedule Feasibility

In Schedule Feasibility Study mainly timelines/deadlines are analyzed for proposed project which includes how many times teams will take to complete final project which has a great impact on the organization as purpose of project may fail if it can't be completed on time.

This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.

When these areas have all been examined, the feasibility analysis helps identify any constraints the proposed project may face, including:

- Internal Project Constraints: Technical, Technology, Budget, Resource, etc.
- Internal Corporate Constraints: Financial, Marketing, Export, etc.
- External Constraints: Logistics, Environment, Laws, and Regulations, etc.

3.5 Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives about development schedule, delivery date, corporate culture and existing business processes. To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, productibility, disposability, sustainability, affordability, and others. These parameters are required to be considered at the early stages of design if desired operational behaviors are

to be realized . A system design and development require appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

3.6 Behavioral Feasibility

Establishing the cost-effectiveness of the proposed system i.e., if the benefits do not outweigh the costs, then it is not worth going ahead. In the fast-paced world today there is a great need of online social networking facilities. Thus, the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

CHAPTER 4

SYSTEM DESIGN

4.1 Flow Chart

Flowchart is a graphical representation of an algorithm. Programmers often use it as a program-planning tool to solve a problem. It makes use of symbols which are connected among them to indicate the flow of information and processing. The process of drawing a flowchart for an algorithm is known as "flowcharting".

Basic Symbols used in Flowchart Designs-

Terminal: The oval symbol indicates Start, Stop and Halt in a program's logic flow. A pause/halt is generally used in a program logic under some error conditions. Terminal is the first and last symbols in the flowchart.

Input/Output: A parallelogram denotes any function of input/output type. Program instructions that take input from input devices and display output on output devices are indicated with parallelogram in a flowchart.

Processing: A box represents arithmetic instructions. All arithmetic processes such as adding, subtracting,multiplication and division are indicated by action or process symbol.

Decision: Diamond symbol represents a decision point. Decision based operations such as yes/no questionor true/false are indicated by diamond in flowchart.

Connectors: Whenever flowchart becomes complex or it spreads over more than one page, itis useful to use connectors to avoid any confusions. It is represented by a circle.

Flow lines: Flow lines indicate the exact sequence in which instructions are executed. Arrows represent

the direction of flow of control and relationship among different symbols of flowchart.

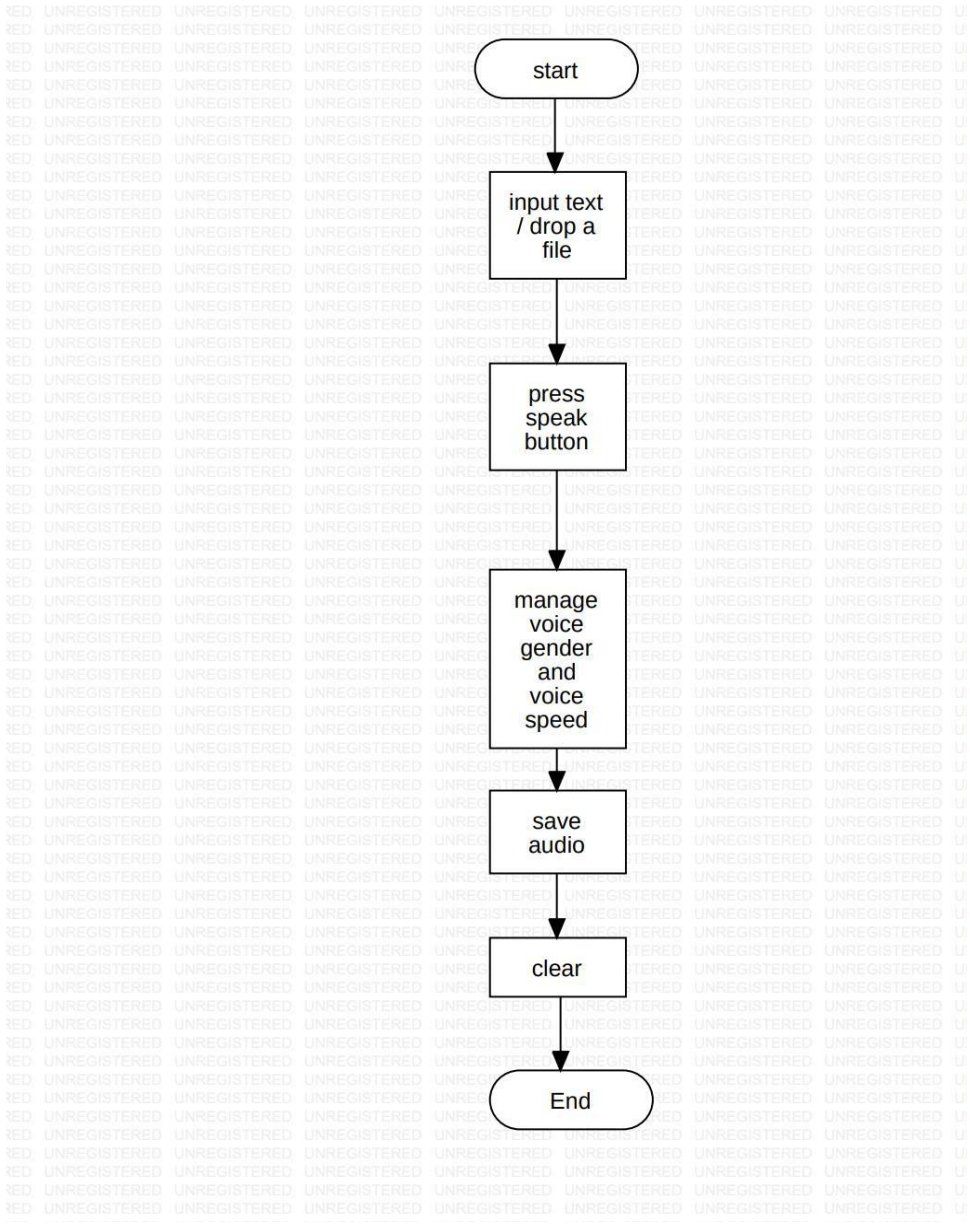

Fig 4.1 Flow Chart

First, we take input as a text then we press the speak button and can choose audio flow like normal, high, slow and also choose voice gender. Press the speak button than we listen the text into audio after that we save that audio in our directories. We can clear that text from text area.

4.1 Use Case Diagram

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of

specialized symbols and connectors. An effective use case diagram can help your team discuss and represent: Scenarios in which your system or application interacts with people, organizations, or external systems. Goals that your system or application helps those entities (known as actors) achieve.

Use case Diagram Components

To answer the question, "What is a use case diagram?" You need to first understand its building blocks. Common components include:

Actors: The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.

System: A specific sequence of actions and interactions between actors and the system. A system may alsobe referred to as a scenario.

Goals: The result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

Use case diagram symbols and notation:

The notation for a use case diagram is straightforward and doesn't involve as many types of symbols as other UML diagrams.

Use Cases: Horizontally shaped ovals that represent the different uses that a user might.

Actors: Stick figures that represent the people employing the use cases.

Associations: A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.

System boundary boxes: A box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system. For example, Psycho Killer is out side the scope of occupations in the chainsaw example found below.

Packages: A UML shape that allows you to put different elements into groups. Just as with component diagram that represent as the file folders.

.

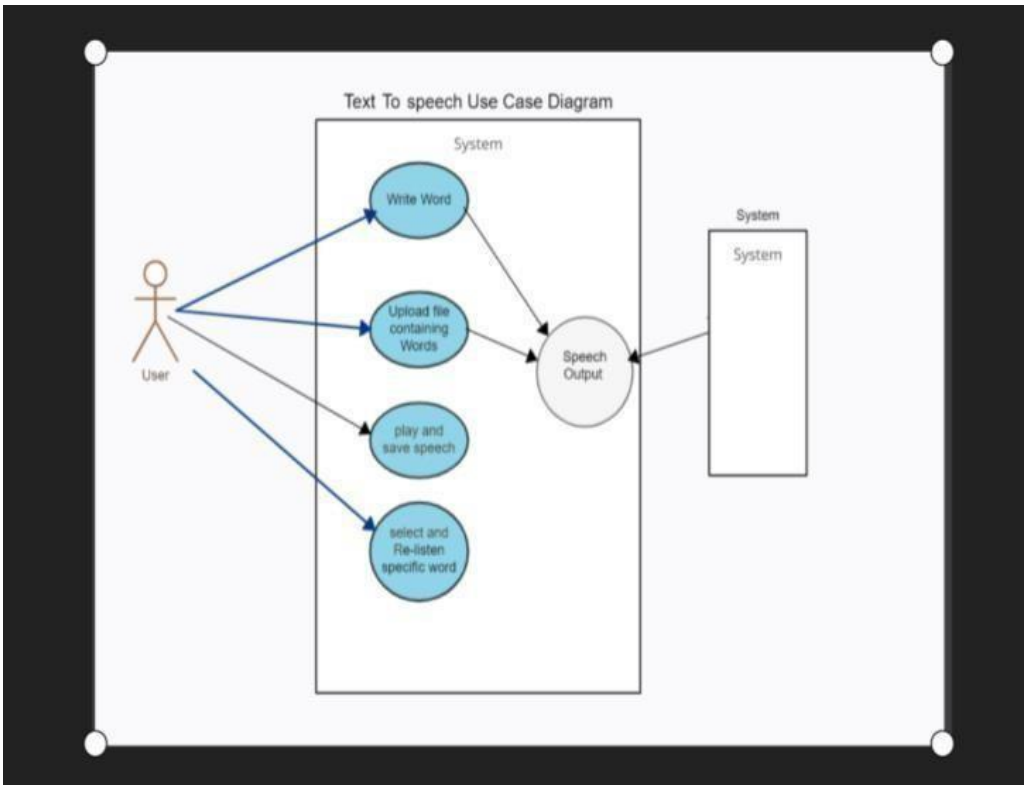

Fig 4.2 Use Case Diagram

In this diagram we represented the use case diagram where a user is represented, he can use four modules like write text, upload notepad file, convert text into audio and save that audio, clear text area. After these modules the written text and uploaded file uses speech output than it connects to the system.

4.2 Sequence Diagram

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the runtime. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching. Notations of a Sequence Diagram Lifeline. An individual participant in the sequence diagram is represented by a lifeline. It is positioned at the top of the diagram.

Actor: A role played by an entity that interacts with the subject is called as an actor. It is out of the scope of the system. It represents the role, which involves human users and external hardware or subjects. An actor may or may not represent a physical entity, but it purely depicts the role of an entity. Several distinct rolescan be played by an actor or vice versa.

Activation: It is represented by a thin rectangle on the lifeline. It describes that time period in which an operation is performed by an element, such that the top and the bottom of the rectangle is associated with theinitiation and the completion time, each respectively.

Messages: The messages depict the interaction between the objects and are represented by arrows. They are in the sequential order on the lifeline. The core of the sequence diagram is formed by messages and lifelines.

Following are types of messages enlisted below.

Call Message: It defines a particular communication between the lifelines of an interaction, which represents that the target lifeline has invoked an operation.

Return Message:

It defines a particular communication between the lifelines of interaction that represent the flow of information from the receiver of the corresponding caller message.

- **Self-Message:** It describes a communication, particularly between the lifelines of an interaction that represents a message of the same lifeline, has been invoked.
- **Recursive Message:** A self-message sent for recursive purpose is called a recursive message. In other words, it can be said that the recursive message is a special case of the self-message as it represents the recursive calls.
- **Create Message:** It describes a communication, particularly between the lifelines of an interaction describing that the target (lifeline) has been instantiated.
- **Destroy Message:** It describes a communication, particularly between the lifelines of an interaction that depicts a request to destroy the lifecycle of the target.
- **Duration Message:** It describes a communication particularly between the lifelines of an interaction, which portrays the time passage of the message while modeling a system.

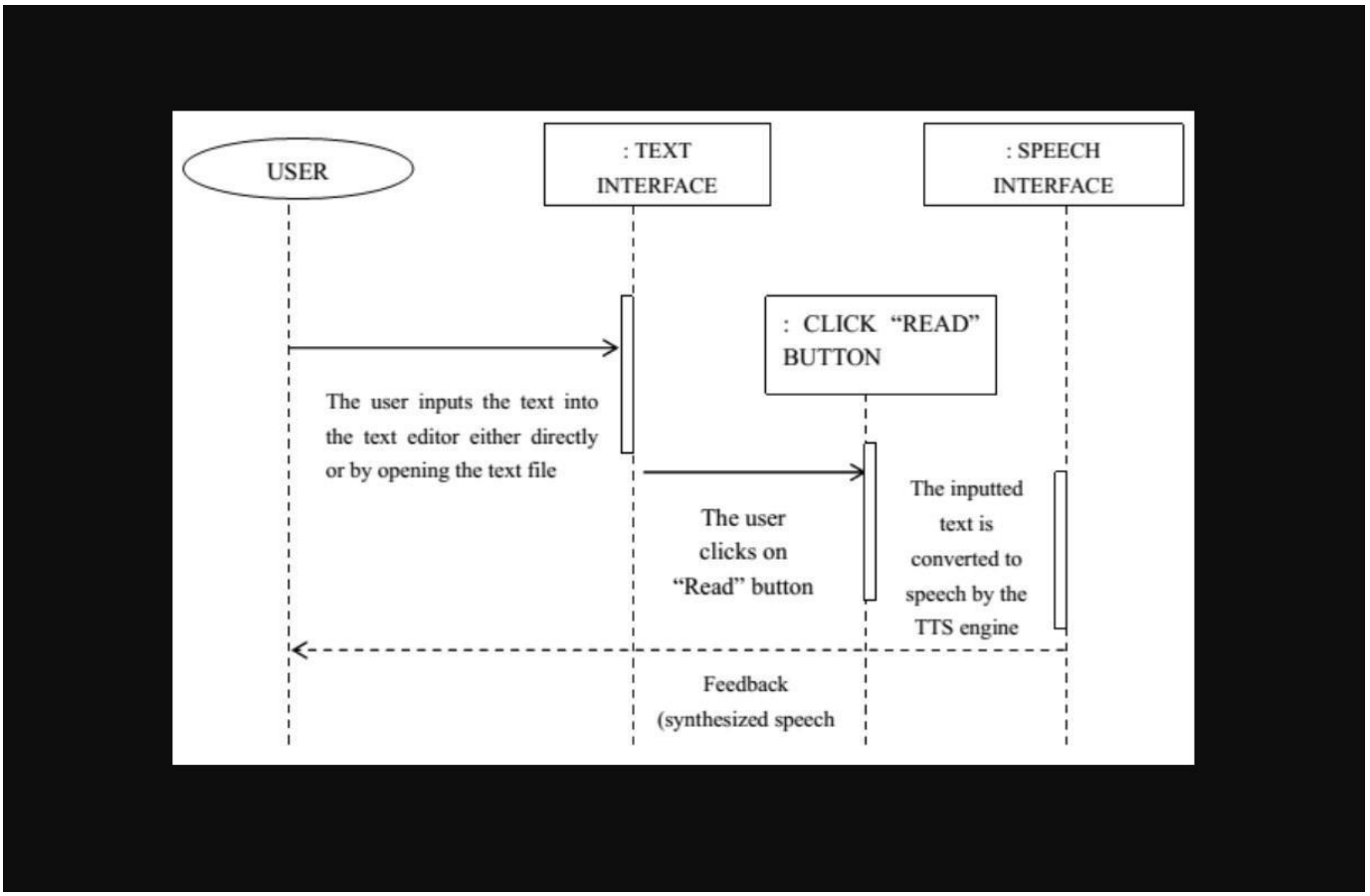

Fig 4.3 Sequence Diagram

In the sequence diagram user input the text area directly also can upload notepad file and then the click read button after that the text converted into audio speech by TTs engine.

CHAPTETR – 5

5.1 Module

Different modules that we have used in this Text to speech system are:

- Frame
- Text
- Voice Gender
- Voice Flow
- Output (Voice)
- Save Voice

Frame Module: - In the frame module we must make the frame in which all the working of the text to speech take place. In this we decide the size of the frame where will it visible when the code is run. How much space is required for all other modules also. In this frame we make places for all other module like text area, buttons for save and speak and for speed.

Text Module: - In this module we have a text area in which we can write text and can copy and paste the text for converting into the audio.

Voice Gender Module: - In this project we have two type of voice male and female. The pyttsx3 supports two voices first is female and the second is male which is provided by “sapi5” for windows.

Voice Flow Module:- In this project we have three type to voice flow

- Fast – In this audio will be audible in fast speed.
- Normal – In this audio will be audible in normal speed.
- Slow – In this audio will be audible in slow speed.

Output Module: - In this project we get the output in audio form. In this the input text converts inthe audio format.

Save Module: - After the conversion of the text into audio we can save it and listen after it whenever require.

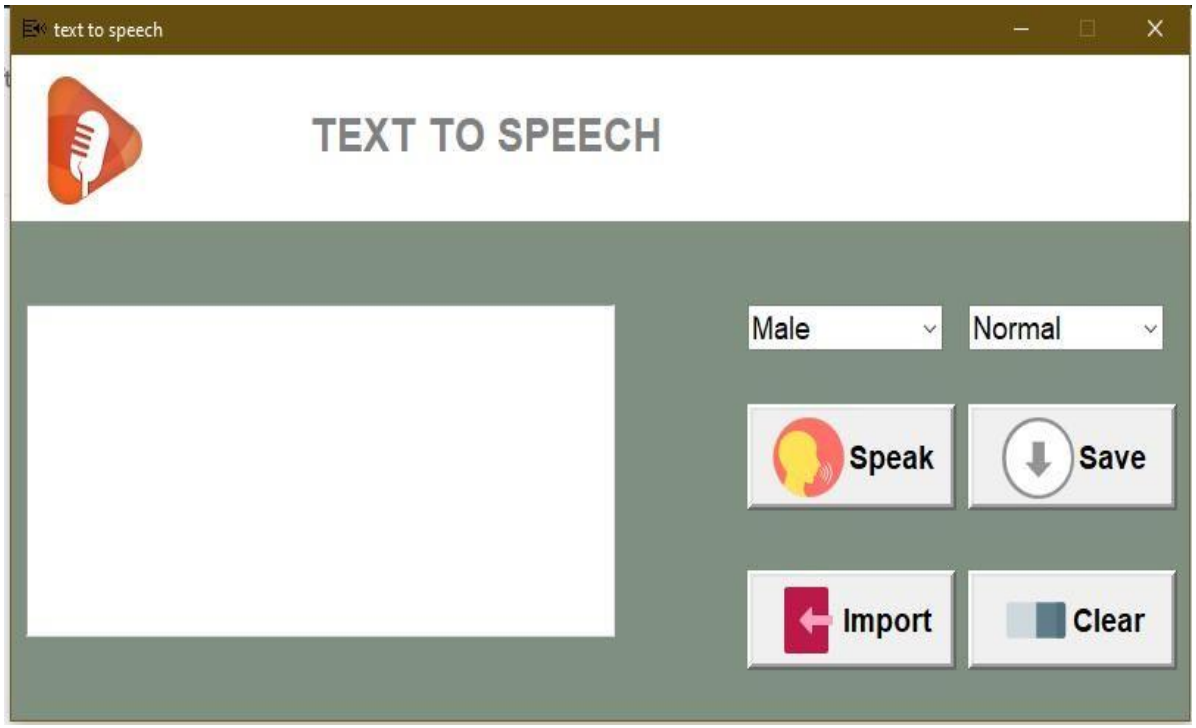

Fig 5.1 OUTLOOK

5.2 Coding

```
from importlib.resources import path
from textwrap import fill
from charset_normalizer import from_path
import pyttsx3
from cProfile import label
from cgitb import text
from email.mime import image
from logging import root
from nntplib import ArticleInfo
import tkinter as tk
from tkinter import *
from tkinter import filedialog
from tkinter.font import BOLD
from tkinter.ttk import Combobox
from turtle import left
import os
root=Tk()
root.title("text to speech")
root.geometry("800x400+100+100")

#window not resizable
root.resizable(False,False)

#speak now
engine = pyttsx3.init()
def speaknow():
    text=text_area.get(1.0, END)
    gender=gender_combobox.get()
    speed=speed_combobox.get()
    voices=engine.getProperty('voices')
```

```
    def setvoice():
      if(gender=='Male'):
          engine.setProperty('voice',voices[0].id)
          engine.say(text)
          engine.runAndWait()
      else:
          engine.setProperty('voice',voices[1].id)
          engine.say(text)
          engine.runAndWait()

    if(text):
       if(speed =='Fast'):
      engine.setProperty('rate',250)
      setvoice()
      elif(speed=='Normal'):
      engine.setProperty('rate',150)
      setvoice()
      elif(speed=='Slow'):
      engine.setProperty('rate',50)
      setvoice()

def downloadnow():
text=text_area.get(1.0, END)
gender=gender_combobox.get()
speed=speed_combobox.get()
voices=engine.getProperty('voices')
def setvoice():
if(gender =='Male'):
engine.setProperty('voice',voices[0].id)
path=filedialog.askdirectory()
os.chdir(path)
engine.save_to_file(text,'text.mp3')
engine.runAndWait()
 else:
```

```
engine.setProperty('voice',voices[1].id)
path=filedialog.askdirectory()
os.chdir(path)
engine.save_to_file(text,'text.mp3')
engine.runAndWait()

    if(text):
     if(speed == 'Fast'):
    engine.setProperty('rate',250)
    setvoice()
    elif(speed=='Normal'):
    engine.setProperty('rate',150)
    setvoice()
    elif(speed=='Slow'):
    engine.setProperty('rate',50)

    setvoice()
        def importnow():
         text=text_area.get(1.0,END)
        path = filedialog.askopenfilename()
        from_path = open(path,'r')
        read = from_path.read()
        speed = speed_combobox.get()
        voices = engine.getProperty('voices')
        text_area.insert(END,read)

#background color
root.configure(bg="#809080")

#icon
image_icon=PhotoImage(file="icon_tts.png")
root.iconphoto(False,image_icon)
```

```
#top frame
Top_frame = Frame(root,bg="white",width=800,height=100)
Top_frame.place(x=0,y=0)

#logo
Logo = PhotoImage(file="speaker logo.png")
Label(Top_frame,image=Logo, bg="white").place(x=10,y=5)

#####
text_area = Text(root,font="Robote 20",bg="white",relief=GROOVE,wrap=WORD)
text_area.place(x=10,y=150,width=400,height=200)

gender_combobox = Combobox(root,values=['Male','Female'],font="arial 14",
state='r',width=10)
gender_combobox.place(x=500,y=150)
gender_combobox.set('Male')

speed_combobox = Combobox(root,values=['Fast','Normal','Slow'],font="arial 14",
state='r',width=10)
speed_combobox.place(x=650,y=150)
speed_combobox.set('Normal')

#button speak
image_iconSpeak=PhotoImage(file="speak.png")
btn =
Button(root,text="Speak",borderwidth=4,compound=LEFT,image=image_iconSpeak,width=13
0,font="arial 14 bold",command=speaknow)
btn.place(x=500,y=210)

#audio save button
image_iconSave=PhotoImage(file="download.png")
btn =
Button(root,text="Save",borderwidth=4,compound=LEFT,image=image_iconSave,width=130,
```

```
font="arial 14 bold",command=downloadnow)
btn.place(x=650,y=210)

#Import file
image_iconImport=PhotoImage(file="import-img.png")
btn =
Button(root,text="Import",borderwidth=4,compound=LEFT,image=image_iconImport,width=1
30, font="arial 14 bold", command=importnow)
btn.place(x=500,y=310)

# clear text area
image_iconClear=PhotoImage(file="clear.png")
btn =
Button(root,text="Clear",borderwidth=4,compound=LEFT,image=image_iconClear,width=130,
font="arial 14 bold", command=lambda:text_area.delete(0.0,tk.END))
btn.place(x=650,y=310)

Label(Top_frame,text="TEXT TO SPEECH",font="ArticleInfo 20 bold",bg="white",
fg="#808080").place(x=200,y=30)
root.mainloop()
```

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

Text - to - speech conversion software project is windows-based application that reads a text file to the user. The software reads a text file or entered text and convert it in audio. The program then reads an entire word to the user. The software can be effectively used to help read the text, or entered text for the user so that the user does not constantly need to look at the screen and read the entire text file or entered text.

Text to speech converter is a recent software project that allows even the visually challenged to read and understand various documents. The blinds cannot read a document, so this software can be an assistant to them who would read out those documents for them. It can also be a great help for those who cannot speak. The person can simply type what he/she wants to say, and the software would give a voice to them by speaking what they wanted to say. The user just has to select the Interactive mode and then write what he wants to say in the text area and then he can easily express what he wanted to say by simply clicking the convert button. So, this software is not just an advancement towards the future development but also a boon for those who cannot speak and see. This technology can also be utilized for various purposes, e.g., car navigation, announcements in railway stations, response services in telecommunications, and e-mail reading. Thus, if we think more innovatively, we can easily get more applications out of it.

CHAPTER 7

REFERENCES

[1] An LSTM-based model for the compression of acoustic inventories for corpus-based text-to-speech synthesis systems (2022)
https://www.sciencedirect.com/science/article/pii/S0045790622002208

[2] Advance research in agricultural text-to-speech: the word segmentation of analytic language and the deep learning-based end-to-end system, (2021)
https://www.sciencedirect.com/science/article/pii/S0168169920331136

[3] CAMNet: A controllable acoustic model for efficient, expressive, high-quality text-to-speech **Applied Acoustics** 9 October 2021, Volume 186
https://www.sciencedirect.com/science/article/pii/S0003682X21005338

[4] The Future of Text-to-Speech Technology: How Long before it's Just One More Thing we do When Teaching Reading? https://www.sciencedirect.com/science/article/pii/S1877042812055413

[5] Duenna—An experimental language teaching application (2016)
https://www.sciencedirect.com/science/article/pii/S2352711016300231

[6] ICE-Talk 2: Interface for Controllable Expressive TTS with perceptual assessment tool (2021)https://www.sciencedirect.com/science/article/pii/S2665963821000038

[7] Investigation of learning abilities on linguistic features in sequence-to-sequence text-to-speech synthesis (2021) https://www.sciencedirect.com/science/article/pii/S0885230820301169

[8] Towards cross-lingual voice cloning in higher education (2021)
https://www.sciencedirect.com/science/article/pii/S095219762100261

[9] The First Vietnamese FOSD-Tacotron-2-based Text-to-Speech Model Dataset
https://www.sciencedirect.com/science/article/pii/S2352340920306697 (2020)

[10] Learning opportunities through generating speech from written texts (2010)https://www.sciencedirect.com/science/article/pii/S1877042810007263

[11] Can AI Powered Speech-to-Text and Text-to-Speech techniques limit the interviewer bias in sensory and consumer research? (2022)
https://www.sciencedirect.com/science/article/pii/S2772569322000147

[12] Inferring Sentiments from Supervised Classification of Text and Speech cues using Fuzzy Rules (2020) https://www.sciencedirect.com/science/article/pii/S1877050920308140

[13] Prosody modeling for syllable based text-to-speech synthesis using feedforward neural networks (2015) https://www.sciencedirect.com/science/article/pii/S0925231215010395

[14] Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation (2018) https://www.sciencedirect.com/science/article/pii/S0885230817302334