# PROJECT REPORT

**"TapVision"**

**Combines "Tap" (interaction method) with "Vision" (for accessibility).**

****

Carried Out at  
CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING  
ELECTRONIC CITY, BANGALORE  
  
UNDER THE SUPERVISION OF  
Aishwarya Pandey  
C-DAC Bangalore  
  
Submitted By:

Aditya Kumar (240850125005)

Abhishek Pal (240850125003)

Khushi Srivastava (240850125045)

Hrishikesh Kanade (240850125038)

Varad Kodgire (240850125086)

PG DIPLOMA IN BIG DATA & DATA ANALYTICS  
C-DAC, BANGALORE

# Candidate’s Declaration

We hereby certify that the work being presented in the report entitled "TapVision" – Combines "Tap" (interaction method) with "Vision" (for accessibility)." in partial fulfillment of the requirements for the award of PG Diploma in Big data and Data Analytics and submitted to the department of Big data and Data Analytics(DBDA), C-DAC Bangalore, is an authentic record of our work carried out under the supervision of Aishwarya Pandey , C-DAC Bangalore.  
  
The matter presented in the report has not been submitted by me for the award of any degree of this or any other Institute/University.  
  
  
Aditya Kumar (240850125005)

Abhishek Pal(240850125003)

Khushi Srivastava (240850125045)

Hrishikesh Kanade(240850125038)

Varad Kodgire(240850125086)

Counter Signed by:  
Aishwarya Pandey

# ACKNOWLEDGMENT

I take this opportunity to express my gratitude to all those who have supported and guided me during the completion of this project. I am sincerely thankful to Aishwarya Pandey whose knowledge and expertise provided invaluable insights into the project. His guidance throughout the development of the "TapVision" – Combines "Tap" (interaction method) with "Vision" (for accessibility)." has been instrumental in achieving this outcome.  
  
I also acknowledge my peers and faculty members for their support and encouragement. Finally, I am grateful to C-DAC Bangalore for providing the necessary resources and a conducive environment for learning and implementation.  
  
  
Aditya Kumar (240850125005)

Abhishek Pal (240850125003)

Khushi Srivastava (240850125045)

Hrishikesh Kanade (240850125038)

Varad Kodgire (240850125086)

# ABSTRACT

Text to Speech (TTS) technology is an advanced field in artificial intelligence and speech synthesis, enabling the conversion of textual information into audible speech. This report explores the various methodologies, technologies, and applications of TTS systems. We discuss the significance of natural language processing (NLP), speech synthesis techniques, and the impact of deep learning in enhancing the quality of TTS outputs.

The study also evaluates various TTS engines, their use in assistive technologies, and their growing role in human-computer interaction. Our implementation focuses on developing a TTS system using Python and speech synthesis Jibraries to achieve accurate and natural-sounding speech output. The report concludes with future prospects and areas of improvement in TTS technology

# TABLE OF CONTENTS

**Introduction** ..................................................................................................................   
 1.1 Overview of Text to Speech Technology ...............................................................   
 1.2 Objectives of the Project .........................................................................................

**Literature Survey** .........................................................................................................   
 2.1 Historical Development of TTS Systems ...............................................................   
 2.2 Current Trends in Speech Synthesis ......................................................................

**Software Requirement Specification** .........................................................................   
 3.1 Functional Requirements ........................................................................................   
 3.2 Non-Functional Requirements ................................................................................

**Architecture** ..................................................................................................................   
 4.1 System Architecture Overview ...............................................................................   
 4.2 Components of the TTS System .............................................................................

**System Design** ............................................................................................................   
 5.1 Flowchart of the TTS System .................................................................................   
 5.2 Data Flow Diagram ................................................................................................

**Implementation** ..........................................................................................................   
 6.1 Tools and Technologies Used ...............................................................................   
 6.2 Challenges Faced During Implementation ...........................................................

**Conclusion** ..................................................................................................................   
 7.1 Summary of the Project .........................................................................................   
 7.2 Future Scope ..........................................................................................................

**References** ..................................................................................................................

­­­­

## ****CHAPTER 1: INTRODUCTION****

### ****1.1 Overview****

Text to Speech (TTS) technology is a form of speech synthesis that converts written text into spoken words. It is widely used in various applications, including assistive technologies, automated customer service, and voice-enabled devices. The primary goal of TTS systems is to produce natural-sounding speech that is easily understandable by humans.

### ****1.2 Objectives****

The main objectives of this project are:

* To design and implement a Text to Speech system that is efficient and user-friendly.
* To explore and apply modern NLP techniques for text processing and speech synthesis.
* To evaluate the performance of the system in terms of speech quality and accuracy.

## ****CHAPTER 2: LITERATURE SURVEY****

**2.1 Historical Development of TTS Systems.**

Text-to-Speech (TTS) systems have evolved significantly over the years, starting in the 1950s with rudimentary mechanical devices like the "Voder" and early phonetic systems. In the 1970s, computer-based systems like DECtalk introduced more automated speech synthesis, while the 1980s saw the rise of commercial applications, such as Kurzweil's Reading Machine for the visually impaired. By the 1990s, statistical models like hidden Markov models (HMMs) improved naturalness, and the advent of neural networks in the 2000s led to more lifelike voices. The 2010s brought groundbreaking developments like Google's WaveNet and Tacotron 2, advancing TTS to human-like quality with natural cadence and intonation. Today, deep learning enables real-time, personalized, and highly natural speech synthesis for a variety of applications, including voice assistants and accessibility tools.

**2.2 Current Trends in Speech Synthesis**

Current trends in speech synthesis are driven by advancements in deep learning and artificial intelligence, leading to highly natural and expressive voices. Neural network-based models like WaveNet and Tacotron have improved speech fluidity and emotion, while voice cloning technology allows for personalized, real-time replication of specific voices. Multilingual support has advanced, enabling seamless language switching, and systems are now capable of adding emotions and tonal variations to speech for more dynamic interactions. Real-time synthesis is becoming standard, with applications in live translation and voice assistants. Additionally, TTS is enhancing accessibility for individuals with disabilities and is being integrated with AI systems for more context-aware, conversational interactions. These trends indicate a future of more human-like, versatile, and inclusive speech technology.

## ****CHAPTER 3: SOFTWARE REQUIREMENT SPECIFICATION****

### ****3.1 Functional Requirements****

* The system should accept text input from the user.
* The system should convert the input text into speech.
* The system should allow the user to select different voices and languages.

### ****3.2 Hardware Requirements****

 **CPU**: (Intel i7/i9 or AMD Ryzen 7/9)

 **GPU**: (NVIDIA RTX 30/40 series) with at least 8-16GB of VRAM

 **RAM**: Minimum of 16GB, preferably 32GB

 **Storage**: Solid-state drive (SSD) with at least 500GB for faster data access.

 **Audio Processing Hardware**: Professional sound card for high-quality audio playback.

 **Internet Connection**: Fast, stable internet for cloud-based services or API calls.

 **Peripheral Devices**: High-quality speakers or headphones for testing synthesized speech output; touchscreen devices for user interaction in embedded systems.

## ****CHAPTER 4: ARCHITECTURE****

### ****4.1 System Overview****

The TTS system consists of three main components:

* **Text Processing Module:** The **Text Processing Module** in a TTS system is responsible for preparing the input text for speech synthesis. Its main functions include tasks like **text normalization,** which standardizes and cleans the input (e.g., converting numbers, abbreviations, or symbols into words), and **tokenization.**
* **Speech Synthesis Module:** The **Speech Synthesis Module** in a TTS system is responsible for converting the processed text into speech. It takes the standardized text, often in the form of phonemes or linguistic features, and uses various techniques, like **natural language processing (NLP), prosody generation** and **audio generation** to produce the final speech output.
* **User Interface Module:** The **User Interface Module** in a TTS system is the component that allows users to interact with the system. It provides the interface through which users input text and receive synthesized speech as output.

### ****4.2 Components of the TTS System****

* **Text Normalization:** Converts text into a standardized format.
* **Phoneme Conversion:** Converts text into phonetic representations.
* **Waveform Generation:** Generates the final speech output.

### ****4.3 Data Flow****

The Text-to-Speech (TTS) system follows a structured data flow to ensure efficient text processing, speech synthesis, and output generation. The process involves several key stages, from user interaction to speech output.

**1. User Interaction:**

* The user accesses the TTS interface to input text and select voice parameters (e.g., voice type, speed, pitch).
* The user initiates the speech synthesis process through the interface.

**2. Text Processing:**

* The system validates the input text for any special characters or formatting issues.
* The text is processed by the **Text Processing Module**, where normalization (e.g., converting numbers to words) and tokenization occur.
* The processed text is passed to the **Speech Synthesis Module**.

**3. Speech Synthesis:**

* The **Speech Synthesis Module** converts the text into phonemes and applies prosody.
* The generated speech is rendered as an audio file (e.g., WAV or MP3).

**4. Output Process:**

* The speech is played through the audio output system, which sends it to speakers or other audio devices.
* A log entry is created to track the synthesis process, including details like user preferences and speech output generated.

Top of Form

Bottom of Form

## ****CHAPTER 5: SYSTEM DESIGN****

The TTS system is designed using a modular architecture to ensure efficient text processing, speech synthesis, and user interaction. The system consists of several key components, including the user interface, text processing module, speech synthesis module, and audio output system.

**1. Architecture Overview:**

The system follows a three-tier architecture:

* **Presentation Layer**: Provides a user-friendly interface for text input, voice selection, and speech output controls.
* **Processing Layer**: Contains the text processing module and speech synthesis module that handle text normalization, phoneme conversion, and waveform generation.
* **Data Layer**: Manages interactions with storage systems, handling speech output data and user preferences.

**2. System Components:**

* **User Interface (Frontend)**:
  + Built using StreamLit for easy navigation.
  + Allows users to input text and coverts to speech.
* **Text Processing Module (Backend)**:
  + Standardizes and processes input text through text normalization and tokenization.
  + Converts text into phonetic representations for accurate synthesis.
* **Speech Synthesis Module (Backend)**:
  + Converts phonetic representations into speech using natural language processing (NLP) techniques.
* **Audio Output System**:
  + Outputs the synthesized speech through speakers or audio interfaces, with support for different audio formats (e.g., WAV, MP3).

**3. Security & Performance Considerations:**

* **Data Integrity**: Ensures accurate text-to-speech conversion by using robust linguistic models.
* **Accessibility**: The system is designed to be accessible, supporting features like user-friendly feedback.

## ****CHAPTER 6: IMPLEMENTATION****

**Technology Stack:**

* **Frontend:** StreamLit for UI development.
* **Backend:** Python, NLP,Transformers.

**Key Features:**

 **Natural-Sounding Speech**: Ensures the generated speech is clear, fluent, and sounds as close to human speech as possible.

 **Multi-Language Support**: Allows conversion of text to speech in various languages, ensuring accessibility for a global audience.

 **Text Formatting Sensitivity**: Ensures proper pronunciation and pauses by recognizing punctuation and formatting (like commas, periods, and question marks).

 **Integration with APIs**: Can be easily integrated into applications or websites, offering dynamic text-to-speech conversion.

 **Real-Time Processing**: Converts text to speech instantly, providing an immediate response for real-time applications like virtual assistants.

 **Voice Clarity and Tone Control**: Ensures the speech is clear with adjustable tone and pitch to match the content's emotional tone.

 **Error Handling & Notifications**: Provides error feedback or notifications in case the text is not recognized or the system encounters issues.

 **Offline Capability**: Some implementations may offer offline text-to-speech functionality, making it available even without an internet connection.

## ****CHAPTER 7: CONCLUSION****

In this **Text-to-Speech** project, we successfully integrated multiple functionalities that enable users to interact with and process text through various methods such as speech synthesis, summarization, translation, sentiment analysis, and file content extraction. Using **Streamlit** for the user interface, the application is designed to be intuitive, accessible, and user-friendly.

The key highlight of this project is the combination of **voice commands** and **text-to-speech** (TTS) conversion, where users can easily give voice instructions to summarize, translate, or analyze text, enhancing the interactivity and accessibility of the system. The **pyttsx3** library for local text-to-speech conversion provides offline access giving users a personalized experience. Additionally, **gTTS** (Google Text-to-Speech) is used to generate downloadable audio files for further use or online access.

The **text extraction** feature supports various file formats such as **PDF**, **DOCX**, and **TXT**, allowing users to upload different types of documents and have their content converted into speech or processed for other tasks like sentiment analysis or translation. This is made possible by leveraging advanced models from **Hugging Face** (such as **T5** for summarization and **MarianMT** for translation) to ensure high-quality processing of text.

With its customizable and scalable design, the application can be extended with additional features such as multi-language support, more advanced NLP tools, and better voice command capabilities, making it a versatile solution for a variety of use cases.

### ****Future Scope****

The **Text-to-Speech** project has the potential for extensive future improvements and additions, especially as advancements in Natural Language Processing (NLP) and Speech Synthesis continue to evolve. Here are several key areas where the project can be expanded:

1.  **Multilingual Support**: Extend the system to support more languages and regional accents for broader global usage.
2.  **Real-Time Speech Generation**: Implement real-time speech generation for dynamic content such as news or live text.
3.  **Integration with IoT Devices**: Enable text-to-speech functionality with smart home devices or assistive technologies for enhanced accessibility.
4.  **Emotional Intelligence in Speech**: Incorporate sentiment analysis to make speech delivery reflect emotions and context for a more natural sound.
5.  **Improved Accuracy with AI**: Enhance voice recognition and pronunciation with deeper neural networks, allowing more accurate conversion.
6.  **Cross-Platform Support**: Develop mobile or desktop applications for a more versatile and portable solution.
7.  **Custom Voice Models**: Allow users to create personalized voices, integrating advanced deep learning models to produce unique voices.

## ****CHAPTER 8: REFERENCES****

* **MDPI and ACS Style**
  + Patwardhan, N.; Marrone, S.; Sansone, C. Transformers in the Real World: A Survey on NLP Applications. *Information* **2023**, *14*, 242. <https://doi.org/10.3390/info14040242>
* **AMA Style**
  + Rahali A, Akhloufi MA. End-to-End Transformer-Based Models in Textual-Based NLP. *AI*. 2023; 4(1):54-110. <https://doi.org/10.3390/ai4010004>
* **Navya Pant** "Speech to text and text to speech recognition systems- Areview." IOSR Journal of Computer Engineering (IOSR-JCE) 20.2 (2018): 36-43.
* J. P. Olive, M. Y. Liberman; Text to speech—An overview. J. Acoust. Soc. Am. 1 November 1985; 78 (S1): S6. <https://doi.org/10.1121/1.2022951>
* A. Anil, P. Verma, A. O. K and P. S. Kumar, "Revolutionizing Summarization: Unleashing the Power of Transformer-Based Innovations for Efficiency and Impactful Applications," 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 2024, pp. 1-6, doi: 10.1109/ADICS58448.2024.10533483. keywords: {Deep learning;Technological innovation;Navigation;Scalability;Focusing;Transformers;Information age;T5 Model;Streamlit Framework;Natural Lan