**BMS Institute of Technology and Management**

(An Autonomous Institution Affiliated to VTU, Belagavi)



***Mini-Project Report on***

**“DEEPFAKE WEBPAGE NARRATION AND QUERY RESPONSE”**

*Submitted in the partial fulfillment for the requirements of the degree of*

BACHELOR OF ENGINEERING IN

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

*Submitted By*

**BHUVANA R RAJ 1BY22AI021**

**K R SHWETHA 1BY22AI038**

**MANASHREE M 1BY22AI052**

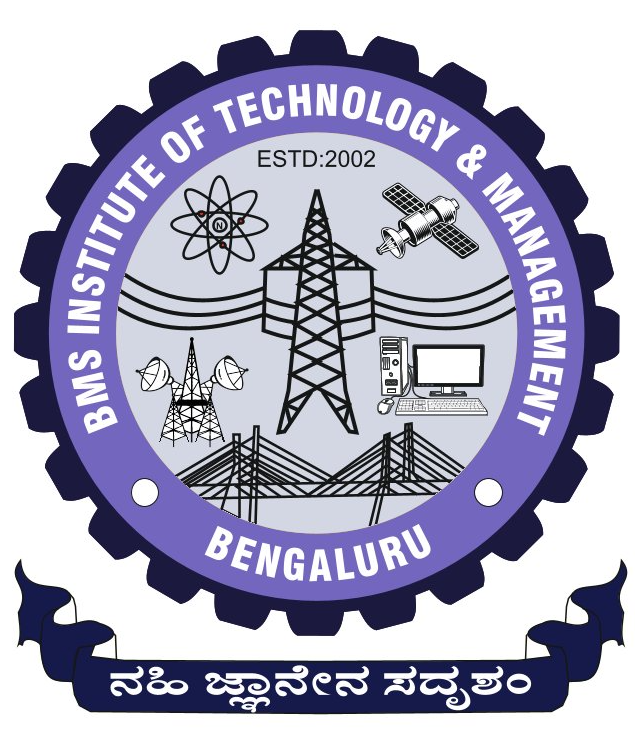
**NAMITHA K G 1BY22AI059**

Under the guidance of

**PROF. SACHIN A U**

Assistant Professor

Department of AIML, BMSIT&M



DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT

YELAHANKA, BENGALURU - 560064.

2024-2025

**VISVESVARAYATECHNOLOGICALUNIVERSITY BELAGAVI**

**BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

**YELAHANKA, BENGALURU – 560064**

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**



**CERTIFICATE**

This is to certify that the Mini-Project work **- BAI506** entitled **“DEEPFAKE WEBPAGE NARRATION AND QUERY RESPONSE”** is a bonafide work carried out by **Bhuvana R Raj (1BY22AI021), K R Shwetha (1BY22AI038), Manashree M (1BY22AI052), Namitha K G (1BY22AI059),** in partial fulfillment for the award of **Bachelor of Engineering Degree in** **Artificial Intelligence and Machine Learning** of the **Visvesvaraya Technological University, Belagavi** during the year 2024-2025. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in this report. The Mini-Project report has been approved as it satisfies the academic requirements in respect of Mini-Project work for B.E. Degree.

**Signature of the Guide Signature of the Signature of the HOD**

**Project-Coordinator**

**External VIVA-VOCE**

Name of the Examiners Signature with Date

**1.**

**2.**

**DECLARATION**

We **Bhuvana R Raj (1BY22AI021), K R Shwetha (1BY22AI038), Manashree M (1BY22AI052), Namitha K G (1BY22AI059)** students of Fifth semester B. E, in the Department of Artificial Intelligence And Machine Learning, BMS Institute of Technology and Management, Bengaluru declare that the Mini-Project work entitled “**Deepfake Webpage Narration and Query Response**” has been carried out by us and submitted in partial fulfilment of the course requirements for the award of degree in **Bachelor of Engineering in Artificial Intelligence And Machine Learning of Visvesvaraya Technological University, Belagavi** during the academic year 2024 - 2025. The matter embodied in this report has not been submitted to any other university or institution for the award of any other degree or diploma.

**Bhuvana R Raj (1BY22AI021)**

**K R Shwetha (1BY22AI038)**

**Manashree M (1BY22AI052)**

**Namitha K G (1BY22AI059)**

**Semester: V**

**BAI506- Mini Project Work Review I & II**

**Mini Project Work Review I & II Course Outcome**

**CO1:** Design and construct a solution for an identified real-life problem with societal importance using software engineering approach ethically.

**CO2**: Make use of programming skills to manage as an individual or in a team, in development of technical projects using appropriate tools.

**CO3**: Develop effective presentation and communication skills in presenting project related activities.

**CO4**: Build quality document of project work for publications, patenting, and final thesis.

**CO-PO-PSO MAPPING**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO No.** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** |
| CO1 | 3 | 3 | 3 | 3 | - | - | 3 | - | 1 | - | - | 3 | 3 |
| CO2 | 1 | 2 | 1 | - | 3 |  | 2 | 2 | 3 | 3 | - | 3 | - |
| CO3 | - | - | - | - | - | 1 | 2 | 2 | 3 | - | 2 | 3 | 1 |
| CO4 | 3 | 3 | 3 | 2 | 3 | 1 | 1 | - | 3 | 3 | 2 | 3 | 2 |
| **Average** | 1.75 | 2 | 1.75 | 1.25 | 1.5 | 0.5 | 2 | 1 | 2.5 | 1.5 | 1 | 3 | 1.5 |

**SDG (Sustainable Development Goals) Mapping**

****

#### SDG 4: Quality Education

**Goal:** Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all.

#### Focus Areas:

* + Using deepfake technology to create virtual narrators that explain educational content in multiple languages and accents, making learning more accessible globally.
  + Enhancing e-learning platforms with interactive query systems powered by LLMs to provide personalized, real-time assistance.
  + Supporting inclusive education for differently-abled individuals through engaging and accessible multimedia tools.

#### SDG 9: Industry, Innovation, and Infrastructure

**Goal:** Build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation.

#### Focus Areas:

* + Promoting innovation by integrating deepfake and AI technologies into web narration and query systems for educational and corporate purposes.
  + Supporting industries and SMEs by offering cost-effective, scalable solutions for interactive content delivery.
  + Strengthening digital infrastructure to improve accessibility and engagement in education and other sectors.

#### SDG 10: Reduced Inequalities

**Goal:** Reduce inequalities within and among countries.

#### Focus Areas:

* + Breaking linguistic barriers by providing narrated content in underrepresented languages and dialects.
  + Ensuring equal access to educational and informational resources for marginalized and underserved communities.
  + Promoting inclusivity by offering customized query responses tailored to diverse user needs.

**ABSTRACT**

The **Deepfake Webpage Narration and Query Response** project uses deepfake technology and AI to enhance user interaction with web content. Instead of reading, users hear a deepfake-generated voice of the website administrator narrating in real-time, improving accessibility and engagement. AI-driven query resolution allows users to select predefined questions and receive accurate responses. This technology benefits those who prefer listening or have visual impairments, making content consumption more user-friendly. By combining deepfake audio with AI, the project showcases deepfakes' practical applications beyond entertainment, transforming digital communication and enhancing accessibility across websites and other online platforms.

**TABLE OF CONTENTS**

**Chapter 1: Introduction………………………………………………………………………1**

1.1 Background**………………………………………………………………………1**

1.2 Literature Survey**………………………………………………………………...2**

1.3 Motivation**……………………………………………………………………….4**

1.4 Problem Statement**………………………………………………………………5**

1.5 Aim and Objective**………………………………………………………………5**

1.6 Scope**…………………………………………………………………………….6**

1.7 Challenges**……………………………………………………………………….6**

**Chapter 2: Overview……………………………………………………………………….....8**

**Chapter 3: Requirement Specification………………………………………………………9**

3.1 Mapping of Requirements**……………………………………………………....9**

3.2 Functional Requirements**………………………………………………………..9**

3.3 Non-Functional Requirements**…………………………………………………10**

3.4 User Requirements**……………………………………………………………..10**

3.5 Domain Requirements**………………………………………………………….10**

3.6 System Requirements**…………………………………………………………..11**

**Chapter 4: Detailed Design………………………………………………………………….12**

4.1 System Architecture and Design**……….……………………………………….12**

4.2 Use Case Diagram**…...………………………………………………………….13**

4.3 Activity Diagram**……………………………………………………………….14**

4.4 Class Diagram**…..………………………………………………………………15**

4.5 Sequence Diagram**…….………………………………………………………..16**

4.6 Data Flow Diagram**…………...………………………………………………...17**

**Chapter 5: Implementation………………………………………………………………….19**

5.1 Programming Language**………………………………………………………...19**

5.2 Algorithms**……………………………………………………………………...19**

5.3 Proposed Design**……………………………………………………...………...20**

5.4 Code**…………………………………………………………………….............22**

**Chapter 6: Testing…………………………………………………………………………....31**

6.1 Test Objectives**………………………………………………………………….31**

6.2 Testing Phases**…………………………………………………………………..31**

6.3 Test Validation**………………………………………………………………….33**

6.4 Test Cases**………………………………………………………………………33**

**Chapter 7: Experimental Results…………………………………………………………....36**

7.1 Artificial Intelligence and Machine Learning Department Website**...………….36**

7.2 About the Department**………….………………………………….……………36**

7.3 Query Response Chatbot with Predefined Queries**......…………………………37**

7.4 Chatbot Answering for the Predefined Questions**…………..…………………..37**

7.5 Error Message for Unrelated Questions**…….…………………………….…….38**

7.6 Contact Information**…………………………………………………………….38**

7.7 Reach Information**……………………………………………………………...39**

**Chapter 8: Conclusion…………………………………………………………………….....40**

**Chapter 9: Future Enhancements…………………………………………………………..41**

**References…………………………………………………………………………………….42**

**LIST OF FIGURES**

Figure 4.1 System Architecture and Design**…………………………………………………...12**

Figure 4.2 Use Case Diagram**………………………………………………………………….13**

Figure 4.3 Activity Diagram**…………………………………………………………………...14**

Figure 4.4 Class Diagram**……………………………………………………………..……….15**

Figure 4.5 Sequence Diagram**…………………………………………………………………16**

Figure 4.6 Data Flow Diagram**………………………………………………………………...17**

Figure 7.1 Artificial Intelligence and Machine Learning Department Website**…….………….36**

Figure 7.2 About the Department**…..………………………………………………………….36**

Figure 7.3 Query Response Chatbot with Predefined Queries**………………………………...37**

Figure 7.4 Chatbot Answering for the Predefined Questions**……………………..…………...37**

Figure 7.5 Error Message for Unrelated Questions**………………………………..…………..38**

Figure 7.6 Contact Information**………………………………………………………………..38**

Figure 7.7 Reach Information**………………………………………………………………….39**

**LIST OF TABLES**

Table 1.2.1 Literature Survey**……………………………………………………………………2**

Table 6.4.1 Functional Testing**………………………………………………………………...33**

Table 6.4.2 Integration Testing**………...……………………………………………………...33**

Table 6.4.3 Performance Testing**……………………………………………………………...34**

Table 6.4.4 Usability Testing**………………..………………………………………………...34**

Table 6.4.5 Voice Quality Testing**…………………….…….………………………………...34**

**CHAPTER 1**

**INTRODUCTION**

* 1. **BACKGROUND**

The rapid advancement of **Artificial Intelligence (AI) and deepfake technology** has revolutionized digital communication, offering new ways to interact with online content. Traditionally, web users have relied on **text-based information**, which requires manual reading and navigation. However, this approach can be limiting for individuals with **visual impairments, reading difficulties, or those who prefer auditory learning**. While existing **Text-to-Speech (TTS) systems** provide basic narration, they often lack the natural intonation, emotional depth, and personalization that make human speech engaging.

Deepfake technology, particularly in the realm of **synthetic voice generation**, provides an opportunity to bridge this gap by creating **realistic AI-generated voices** that can narrate content in a lifelike manner. This involves training AI models on **recordings of a specific individual’s voice** to replicate their tone, pitch, and speech patterns. The result is a more immersive and engaging experience where users can listen to webpage content instead of reading it.

Beyond narration, AI can also enhance **information retrieval** by enabling **automated query response systems**. Many websites contain extensive content, requiring users to search for specific details manually. This project addresses this challenge by incorporating **an AI-driven question-answering system** that allows users to **select predefined queries** and receive **instant, accurate responses**. This makes navigation smoother and improves overall user experience.

The **Deepfake Webpage Narration and Query Response** project integrates **deepfake voice synthesis and Natural Language Processing (NLP)** to create an interactive AI assistant embedded into a webpage. The **narration module** utilizes a deepfake-generated voice to read out webpage content, closely mimicking the **website administrator’s voice** for a personalized experience. Meanwhile, the **query response system** leverages NLP to interpret user-selected questions and provide relevant answers in both **text and voice format**.

This project highlights the **practical and ethical applications** of deepfake technology beyond entertainment and misinformation. By improving **web accessibility, user engagement, and efficient information retrieval**, it demonstrates how **AI-driven voice synthesis and NLP** can transform digital interactions, making websites more **inclusive, interactive, and user-friendly**.

* 1. **LITERATURE SURVEY**

To guide the development of the system for Deepfake Webpage Narration and Query Response, it is important to explore the existing research and developments related to audio deepfake technology, Natural Language Processing (NLP), and AI-driven user interaction systems. Table 1.2.1 presents a comprehensive overview of key studies and advancements in these areas, highlighting their methodologies, advantages, disadvantages, and relevance to this project.

Table 1.2.1 Literature Survey on Deepfake Webpage Narration and Query Response

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Title** | **Author** | **Methodology** | **Advantages** | **Disadvantages** |
| Deepfake Narration for Accessible Web Content | Michael Green et.al. | Neural Text-to- Speech (Neural TTS), Voice Cloning Technology, Natural Language Processing (NLP), Generative Adversarial Networks (GANs) | - Enhanced web content accessibility for users with disabilities.  - Natural and clear voice narration. | - Risk of misuse for spreading disinformation.  - Potential for high development costs. |
| Realistic Voice Cloning with Artificial  Intelligence | Sanam Malhotra | Artificial intelligence | - Personalizes voice - controlled interactions  - Adds a familiar voice to healthcare services  - Delivers professional speech for educational materials | - Misuse potential  - Trust issues  - Intellectual  property rights  - Privacy concerns |
| Voice Cloning Using Deep Learning | Mohit Saini | Neural networks and machine learning algorithms | - Voice Creation for Digital Applications  - Personalized Assistants  - Accessibility | - Misinformation and manipulation  - Difficulty in authenticity verification  - Security risks  - Ethical concerns |
| Deep Voice: Real- time Neural Text-to-Speech | Jonathan Raiman et.al. | Segmentation Model, Audio Synthesis Model, Optimized WaveNet Inference Kernels, Training Techniques,  Optimization Techniques | - Simplicity of features  - Standalone system  - Human-free training  - Fast inference  - Reduced feature complexity | - Dependency on large datasets  - Limitations in duration and frequency prediction  - Dataset size constraints |
| Voice Cloning for Audiobook Narration | David Turner et.al. | Voice cloning algorithms, Comparison with traditional audiobook narration | - Customized narration options  - Enhanced user experience through familiar voices | - Limited Availability of diverse voices  - Challenges in maintaining emotional tone |
| Deepfake Voice Narration in Interactive Media | Rachel Adams et.al. | Generative  Adversarial Networks (GANs), Text-to-  Speech (TTS)  Systems, Deep Learning Algorithms,  Neural Text-to- Speech (Neural TTS) | - Improved immersion for users  - Engaging  narratives with character-specific voices | - Potential for voice mismatch issues |
| Web Accessibility and the Use of Deepfake Technology for Voice Narration | John Doe et.al. | Deepfake voice synthesis, Audio content generation | - Enhanced user engagement  - Personalized content delivery | - Ethical concerns regarding misinformation  - Potential misuse for malicious purposes |
| Improving Accessibility with Deepfake Voices | Alice Johnson et.al. | Neural network-based voice cloning, Text-to-speech integration | - Increased accessibility for visually impaired users  - Natural-  sounding voice reproduction | - Dependence on large datasets for training  - Challenges in voice  consistency |
| Deepfake Voice for Real-Time Web Accessibility Enhancement | James Miller et.al. | Real-time voice synthesis integration, neural networks | - Enables real- time content accessibility  - Adaptive to different user preferences | - Latency issues in real-time synthesis  - High computational demand for real- time processing |

* 1. **MOTIVATION**

The motivation behind Deepfake Webpage Narration and Query Response project stems from the growing demand for enhanced accessibility and personalization in digital interactions. As web content consumption continues to rise, many users seek more intuitive and efficient ways to engage with information. People with visual impairments or those who prefer listening over reading often face limitations with text-heavy websites. By leveraging deepfake technology, this project aims to create a more inclusive and engaging experience by transforming how users interact with web content.

Moreover, the integration of AI-driven query resolution enhances user engagement by allowing real-time interaction through predefined queries. This transforms the passive consumption of information into a more dynamic experience, enriching user interactions and making websites more user-friendly. By mimicking the website administrator’s voice for narration, the project fosters a sense of familiarity, improving the overall connection between the user and the digital platform.

Finally, this project highlights the practical potential of deepfake technology beyond entertainment, showcasing its ability to serve as a valuable tool in digital communication. The project aims to open new possibilities for making online content more interactive, accessible, and engaging for a broader audience.

**1.4 PROBLEM STATEMENT**

The project aims to solve the lack of accessible and personalized interaction in web content by integrating deepfake voice technology and AI to provide real-time narration and predefined query resolution.

Many websites rely on text-based content, which can be difficult for users with visual impairments, reading challenges, or those who prefer auditory learning. Existing text-to-speech systems often lack natural intonation and personalization, making web interactions less engaging. Additionally, finding specific information on a webpage requires manual searching, which can be time-consuming. This project addresses these issues by using **deepfake voice technology** for **realistic narration** and **AI-driven query resolution** to enhance accessibility, engagement, and efficiency in web interactions.

* 1. **AIM AND OBJECTIVE**

The aim of this project is to develop an **AI-powered deepfake voice narration and query response system** that enhances user interaction with web content by providing **realistic voice narration** and **automated responses to predefined queries**. This system leverages **deepfake voice synthesis and Natural Language Processing (NLP)** to improve **accessibility, engagement, and information retrieval** on websites.

* **Deepfake Voice Synthesis Model Development:** Develop a deepfake voice synthesis model that accurately replicates the website administrator’s voice for realistic webpage narration.
* **Text-to-Speech (TTS) System Implementation:** Implement a Text-to-Speech (TTS) system that converts webpage content into high-quality, human-like audio output.
* **AI-Driven Query Response System:** Integrate an AI-driven query response system that allows users to select predefined questions and receive accurate answers in real time.
* **Web Accessibility Enhancement:** Enhance web accessibility by providing an alternative to text-based content for visually impaired users and those who prefer auditory learning.
* **User Engagement Improvement:** Improve user engagement by creating a more interactive and immersive browsing experience.

**1.6 SCOPE**

The **scope of the Deepfake Webpage Narration and Query Response** project revolves around enhancing user interaction with web content through AI-powered voice narration and intelligent query resolution. This system aims to provide real-time text-to-speech conversion using a deepfake-generated voice, replicating the website administrator’s speech for a more immersive and personalized experience. By incorporating this feature, the project significantly improves accessibility, particularly for visually impaired users or those who prefer listening over reading.

Another key aspect of the project is **AI-driven query resolution**, where a Natural Language Processing (NLP) model is implemented to interpret predefined questions and deliver accurate, context-aware responses. Users will be able to interact with the system through a chatbot interface, ensuring seamless and efficient engagement with the webpage content.

The project also includes **system integration and usability**, ensuring smooth operation through web technologies like **HTML, CSS, and Flask**. The backend will efficiently process real-time voice narration and query handling while maintaining minimal latency and high-quality output.

Comprehensive **testing and deployment** will be conducted to evaluate the synthesized voice’s naturalness, clarity, and emotion accuracy, as well as the precision of the NLP model’s responses. The system will undergo continuous improvements based on user feedback to enhance its performance. By combining deepfake and AI technologies, this project offers a novel approach to making digital content more interactive, accessible, and engaging.

* 1. **CHALLENGES**
     1. **Voice Authenticity and Quality**
* Achieving a highly realistic deepfake-generated voice that closely mimics the website administrator’s speech.
* Ensuring natural intonation, prosody, and emotional variation for an engaging listening experience.

1.7.2 **NLP Accuracy and Query Handling**

* Ensuring the Natural Language Processing (NLP) model accurately understands predefined queries.
* Handling ambiguous or complex user queries effectively while maintaining relevant responses.

1.7.3 **System Integration and Real-Time Performance**

* Ensuring seamless integration between the frontend (HTML, CSS) and backend (Flask, NLP, TTS models).
* Maintaining low latency for real-time narration and query response without delays.
  + 1. **Data Collection and Model Training**
* Acquiring high-quality voice samples for training the deepfake model while ensuring privacy compliance.
* Processing and cleaning large datasets for accurate NLP model training.

1.7.5 **Testing and Continuous Improvement**

* Ensuring rigorous testing for voice clarity, NLP accuracy, and system stability.
* Gathering user feedback and updating models to improve performance over time.

**CHAPTER 2**

**OVERVIEW**

The **Deepfake Webpage Narration and Query Response** project aims to enhance the way users interact with web content by integrating **AI-driven voice narration** and **intelligent query resolution**. This system utilizes **deepfake voice synthesis** to generate a realistic voice that mimics the website administrator, allowing webpage content to be narrated in real-time. By providing an **immersive and personalized listening experience**, the project significantly improves accessibility, especially for visually impaired users and individuals who prefer audio-based content consumption.

In addition to narration, the project incorporates an **AI-powered query response system** using **Natural Language Processing (NLP)**. Users can select from predefined questions, and the system will analyze the input and generate accurate, context-aware responses. This feature enhances user engagement by offering interactive and efficient information retrieval without requiring extensive reading or navigation.

The project involves the **development of a web-based interface** where users can seamlessly access narration and query functions. Technologies such as **HTML, CSS, Flask, and AI models** will be integrated to ensure smooth real-time processing and minimal latency. The backend will efficiently handle both **text-to-speech (TTS) processing** for narration and **NLP-based response generation** for queries.

To ensure reliability, the system will undergo **extensive testing and validation** for voice quality, response accuracy, and user experience. Additionally, continuous improvements will be made based on user feedback to enhance overall functionality. By combining deepfake technology with AI-driven natural language processing, this project demonstrates a practical application of **synthetic media and intelligent automation**, making web interactions more **accessible, engaging, and user-friendly.**

**CHAPTER 3**

**REQUIREMENT SPECIFICATION**

**3.1 MAPPING OF REQUIREMENTS**

The **Deepfake Webpage Narration and Query Response** system is designed to enhance web accessibility by converting webpage text into synthetic speech while also providing predefined query-based responses. The system must meet various requirements. User requirements ensure that users can listen to narrated webpage content and receive relevant responses. Functional requirements define the core features, including synthetic speech generation and intelligent query response. Non-functional requirements emphasize performance, scalability, and security. Domain requirements focus on leveraging deepfake- based Text-to-Speech (TTS) and Natural Language Processing (NLP) techniques. Additionally, system requirements specify the necessary hardware and software for seamless execution.

# **3.2 FUNCTIONAL REQUIREMENTS**

The system's primary functionalities include webpage narration, query response handling, web interface design, backend processing, and error handling. The webpage narration feature should convert text into a deepfake voice resembling the website administrator. For the query response system, users should be able to enter predefined queries, and the system should accurately interpret the intent using NLP models. It must retrieve and present relevant answers.

The web interface should be user-friendly and provide a seamless experience for enabling voice narration and submitting queries. The system should display query responses in text and synthesized voice while ensuring accessibility for differently-abled users. Backend processing will involve the implementation of Flask-based APIs to manage text input and return synthesized speech. The system must facilitate efficient communication between the frontend and backend while securely storing predefined queries and responses.

Robust error handling should be implemented to provide meaningful feedback for invalid inputs. The system must handle unsupported or ambiguous queries gracefully by offering a fallback response, such as suggesting the user rephrase their query.

**3.3 NON-FUNCTIONAL REQUIREMENTS**

To ensure optimal system performance, the system must generate voice narration in real- time with minimal delay and provide query responses within two seconds. Scalability is essential to support multiple simultaneous users without performance degradation, and the backend should efficiently handle high traffic conditions.

Usability is a crucial aspect, and the system must have an intuitive interface that works seamlessly across multiple devices, including desktops, mobile phones, and tablets. Both keyboard and voice input options should be supported for accessibility.

Security measures should be implemented to protect against cyber threats such as SQL injection and cross-site scripting (XSS). User input must be sanitized to prevent malicious attacks, and API endpoints should be secured to prevent unauthorized access.

Maintainability and extensibility are also essential, requiring the system’s code to be modular and well-documented. Future enhancements should include multilingual support and easy integration of new voice models and predefined queries.

**3.4 USER REQUIREMENTS**

The system is designed for three primary user groups: website visitors, content creators, and administrators. Website visitors should be able to listen to narrated content and submit predefined queries for instant responses. Content creators should have the ability to train the deepfake TTS model using their voice. Administrators should be able to manage predefined queries and voice models.

Users should have the option to switch between manual reading and narrated content. They should receive instant query responses and be able to control the playback speed and volume of the voice narration.

**3.5 DOMAIN REQUIREMENTS**

The Deepfake Webpage Narration and Query Responsesystem is built upon advanced AI-driven technologies, integrating deep learning, Natural Language Processing (NLP), web development, and API communication.

The Deepfake-Based Text-to-Speech (TTS)component utilizes models such as Tacotron, FastSpeech, or GAN-based TTSto generate realistic synthetic voices. These deep learning techniques enable natural-sounding speech with accurate prosody, rhythm, and tone, mimicking a specific voice.

The Natural Language Processing (NLP) module is responsible for understanding user queries and retrieving relevant responses. This is achieved using NLP libraries and models like NLTK, Spacy, or BERT, which help with text processing, intent recognition, and context awareness.

For Web Development, the system's frontend is developed using HTML, CSS, and JavaScriptwith frameworks such as React or Vueto create an interactive and accessible user interface. The backend is built using Python (Flask) to manage query processing and text-to-speech generation. A database such as SQLite or NoSQLis used to store predefined queries and responses securely.

**3.6 SYSTEM REQUIREMNETS**

The software requirements for the Deepfake Webpage Narration and Query Responsesystem include a compatible operating system such as Windows 10 or later, macOS, or Linux. The system must be developed using Python 3.8 or a later version, along with essential frameworks and libraries such as Flask for backend development, Tacotron for Text-to-Speech synthesis, NLTK for Natural Language Processing, and TensorFlow or PyTorch for deep learning-based model training. The database required for storing predefined queries can be either SQLite or MongoDB. The frontend development will involve standard web technologies, including HTML, CSS, and JavaScript, while REST APIs will be used for efficient query handling and communication between system components.

In terms of hardware requirements, the system should run on a machine with at least an Intel i5 or AMD Ryzen 5 processor. A minimum of 8GB RAM is required, although 16GB is recommended for training deepfake voice models. The system must have at least 10GB of free storage space to accommodate model files and data. Additionally, a dedicated GPU, such as an NVIDIA GTX 1050 or higher, is recommended for efficiently handling Text-to- Speech model training and improving performance.

**CHAPTER 4**

**DETAILED DESIGN**

**4.1 SYSTEM ARCHITECTURE AND DESIGN**

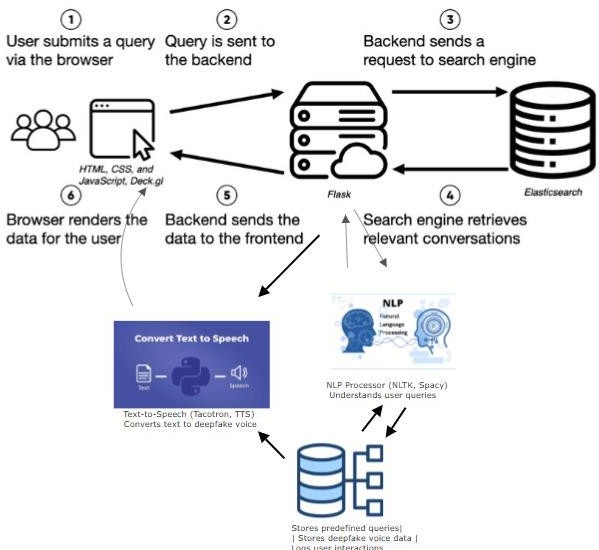


Figure 4.1 System Architecture of Deepfake Webpage Narration and Query Response

The System Architecture diagram in Figure 4.1 illustrates how user queries and webpage narration are processed using Natural Language Processing (NLP) and Text-to-Speech (TTS) technologies. When a user submits a query through the web interface (HTML, CSS, JavaScript), the request is sent to the Flask backend for processing. The backend determines

if the query requires external data retrieval and, if needed, forwards it to a search engine (Elasticsearch), which fetches relevant information and returns it to Flask. Simultaneously, the backend sends the query to the NLP Processor (NLTK, Spacy), The database contains structured predefined queries, deepfake voice data, and user interaction logs. If a relevant response is found, it is sent back to Flask for further processing. If the response needs to be narrated, Flask forwards the text to the Text-to-Speech (Tacotron, FastSpeech) module, which generates a deepfake voice using stored voice data from the database. The synthesized speech is then sent back to the frontend, where users can hear the narration. This system efficiently processes user queries, retrieves relevant information, and provides responses in both text, enhancing interactivity through AI-driven narration and intelligent query handling.

**4.2 USE CASE DIAGRAM**

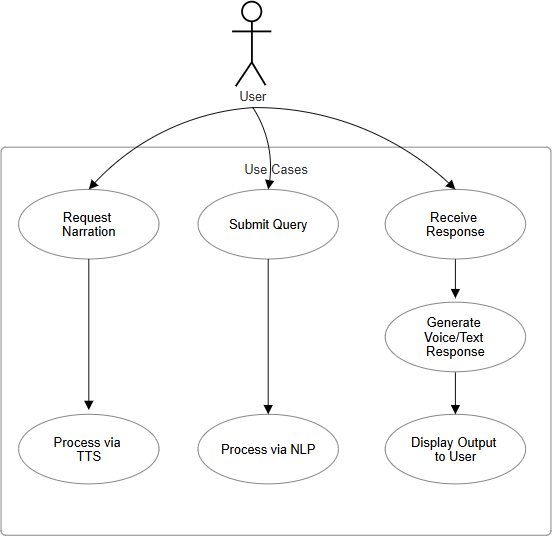


Figure 4.2 Use Case Diagram of Deepfake Webpage Narration and Query Response

The Use Case Diagram in Figure 4.2 represents the interactions between the user and the system, highlighting the key functionalities of the “Deepfake Webpage Narration and Query Response” system. The primary actor is the User, who interacts with the system through two main use cases: Request Narration and Submit Query. The system processes these requests using either a Text-to-Speech (TTS) Model for narration or a Natural Language Processing (NLP) Model for query handling. The final output is delivered in either voice or text format. This diagram helps visualize the core functionalities and user interactions with the system.

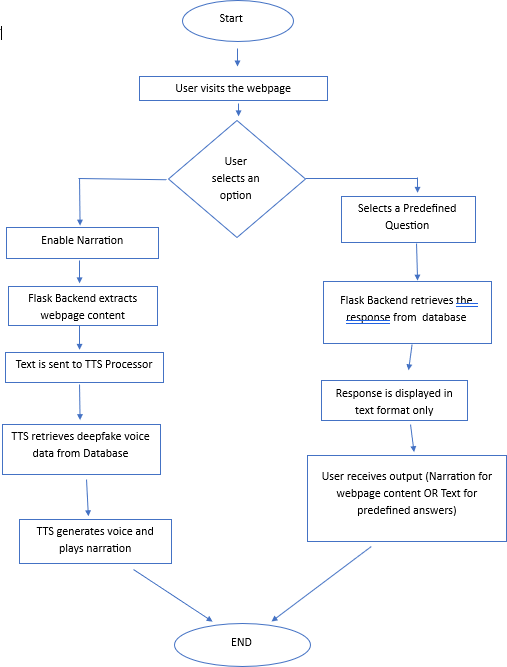
**4.3 ACTIVITY DIAGRAM**

Figure 4.3 Activity Diagram of Deepfake Webpage Narration and Query Response

The activity diagram represents a system where users can either enable webpage narration or select predefined questions to receive textual responses. The process begins when the user visits the webpage and is presented with two main options. If the user enables

narration, the Flask backend extracts the webpage content and forwards it to the Text-to- Speech (TTS) processor. The TTS module retrieves deepfake voice data from the database, processes the text, and generates speech output, allowing the user to listen to the webpage content. Once narration is complete, the system reaches its end state. Alternatively, if the user selects a predefined question, the Flask backend retrieves the corresponding response from the database. Instead of converting the response into speech, the answer is displayed in text format on the webpage, allowing the user to read it. This system combines AI-driven text-to-speech technology for narration with a structured query-based response system for predefined questions, enhancing user accessibility by providing both auditory and textual outputs. The process is automated and efficient, reducing the need for manual content reading while ensuring users can quickly and easily receive relevant answers.

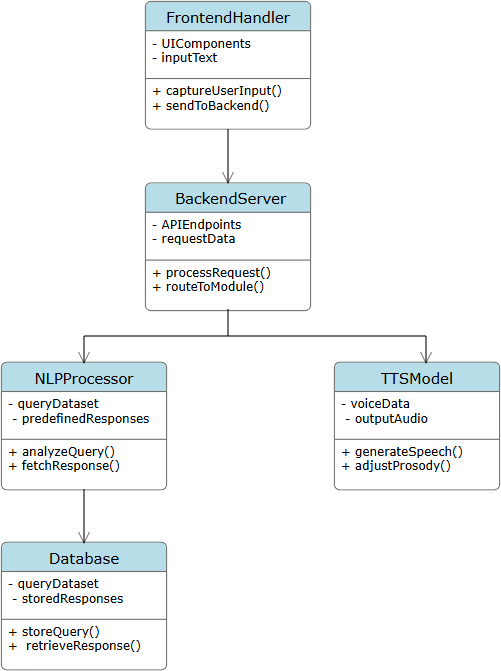
**4.4 CLASS DIAGRAM**

Figure 4.4 Class Diagram of Deepfake Webpage Narration and Query Response

The Class Diagram in Figure 4.4 provides a structural representation of the system by defining its key components, attributes, and methods. The system comprises several major classes, including FrontendHandler, which manages user inputs and sends them to the backend; BackendServer, which processes the user’s request and routes it appropriately; TTSModel, responsible for handling text-to-speech conversion; NLPProcessor, which processes queries and retrieves responses from the database; and Database, which stores predefined queries and responses for retrieval. The diagram illustrates the relationships between these classes, ensuring a well-structured modular design for efficient processing.

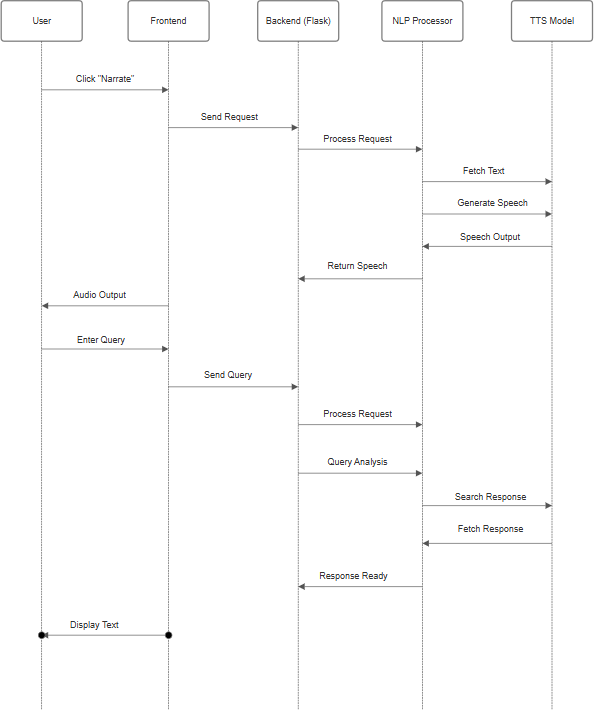
**4.5 SEQUENCE DIAGRAM**

Figure 4.5 Sequence Diagram of Deepfake Webpage Narration and Query Response

The Sequence Diagram in Figure 4.5 represents the step-by-step interactions between the

system components over time. It illustrates the process flow when a user requests narration or submits a query. Initially, the user interacts with the system by either selecting webpage narration or submitting a query. The frontend captures this input and sends it to the backend, where the backend server determines the type of request and forwards it to the appropriate module. If the request is for narration, it is sent to the TTS Model, which generates synthesized speech. If the request is a query, it is processed by the NLP Processor, which fetches an appropriate response from the database. Finally, the frontend receives and displays the text response or plays the synthesized voice output. This diagram helps in understanding the flow of interactions and how different components work together in real- time.

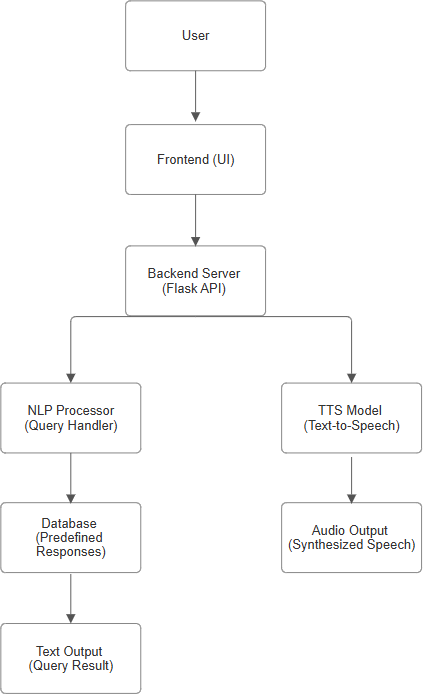
**4.6 DATA FLOW DIAGRAM**

Figure 4.6 Data Flow Diagram of Deepfake Webpage Narration and Query Response

The Data Flow Diagram (DFD) in Figure 4.6 showcases the movement of data throughout the system. It highlights the sequential process where a user either requests narration or submits a query, which is then captured by the frontend and forwarded to the backend for further processing. The backend then determines the nature of the request and routes it to the appropriate processing unit. If the request pertains to narration, the system processes the text through the TTS Model to generate a synthesized speech output. If the request is a query, the system analyses it using the NLP Processor, which then retrieves the most suitable response from the database. The database serves as a repository for predefined queries and their respective responses, ensuring quick and accurate retrieval. The final output is either returned as synthesized speech for narration or displayed as text for query responses. This diagram provides a clear visualization of data processing and movement within the system, ensuring a structured flow of operations.

**CHAPTER 5**

**IMPLEMENTATION**

**5.1 PROGRAMMING LANGUAGE**

PYTHON:

Python is the core programming language used for backend development, deep learning model integration, and handling AI-based functionalities. It facilitates various tasks such as speech synthesis, natural language processing (NLP), and web interactions. Frameworks like Flask are used to create a backend API that connects the frontend with AI models. Libraries such as Tacotron, and handle deepfake voice generation and text-to-speech (TTS) functionalities. Additionally, NLTK and assist with query processing and response generation.

FRONTEND:

The frontend is developed using HTML and CSS to provide an interactive user interface. Bootstrap is utilized for responsive design, ensuring compatibility across devices. The webpage includes features like text narration activation buttons and chatbot integration for handling predefined queries. JavaScript enhances user interaction by dynamically updating the UI and managing API requests through Fetch API.

BACKEND:

The backend is implemented using Flask, which serves as a bridge between the frontend and AI models. It processes user inputs, generates synthetic voice outputs, and retrieves query responses. Flask APIs manage the narration requests, chatbot interactions, and database communication.

**5.2 ALGORITHMS**

5.2.1 Text-to-Speech (TTS) Model:

* + The system employs Tacotron and WaveGlow to generate human-like speech from text.
  + Deepfake voice synthesis is trained using administrator voice samples to ensure realistic narration.
    1. Natural Language Processing (NLP):
  + NLTK and spaCy are used for query understanding and response generation.
  + The system maps user inputs to predefined intents and provides relevant answers.

5.2.3 Flask-based API Handling:

* + RESTful APIs connect the frontend with AI functionalities.
  + APIs handle voice generation, query responses, and webpage narration requests.
  1. **PROPOSED DESIGN**

The system is designed to convert text into human-like speech using deep learning models while integrating seamlessly with a Flask-based web application. The main objectives include providing high-quality speech synthesis, user-friendly interaction, and efficient backend processing. The key components of the proposed system are as follows:

* + 1. Text-to-Speech Processing

The system utilizes deep learning models such as Tacotron 2 and WaveGlow to convert text into natural-sounding speech. These models process input text, analyze linguistic patterns, and generate high-quality audio. The system supports multiple text formats, including plain text and structured documents (.txt, .docx, .pdf), ensuring flexibility in handling different sources of textual content.

* + 1. Flask-Based Backend for Audio Generation

A Flask API is developed to handle text input and generate corresponding speech output. This backend receives text from the frontend, processes it using trained models, and returns an audio file. It also supports real-time text processing and user-defined speech preferences, such as selecting different voices or adjusting speech speed.

* + 1. Web Application for Interactive Use

The system provides a web-based interface using HTML, CSS, and JavaScript to allow users to input text, adjust settings, and listen to generated speech. AJAX and Fetch API are used for seamless communication between the frontend and Flask backend, ensuring a smooth and responsive user experience. The interface also provides options to download the generated audio file.

* + 1. Dataset Preparation and Model Training

The system is trained on a diverse dataset containing human speech recordings and transcriptions to improve the naturalness and accuracy of speech synthesis. Data preprocessing techniques include phoneme extraction, prosody modeling, and noise reduction. Fine-tuning of models is done using PyTorch and TensorFlow to optimize performance.

* + 1. System Architecture and Security

The architecture follows a modular microservices approach, ensuring scalability, efficiency, and security. Key components include:

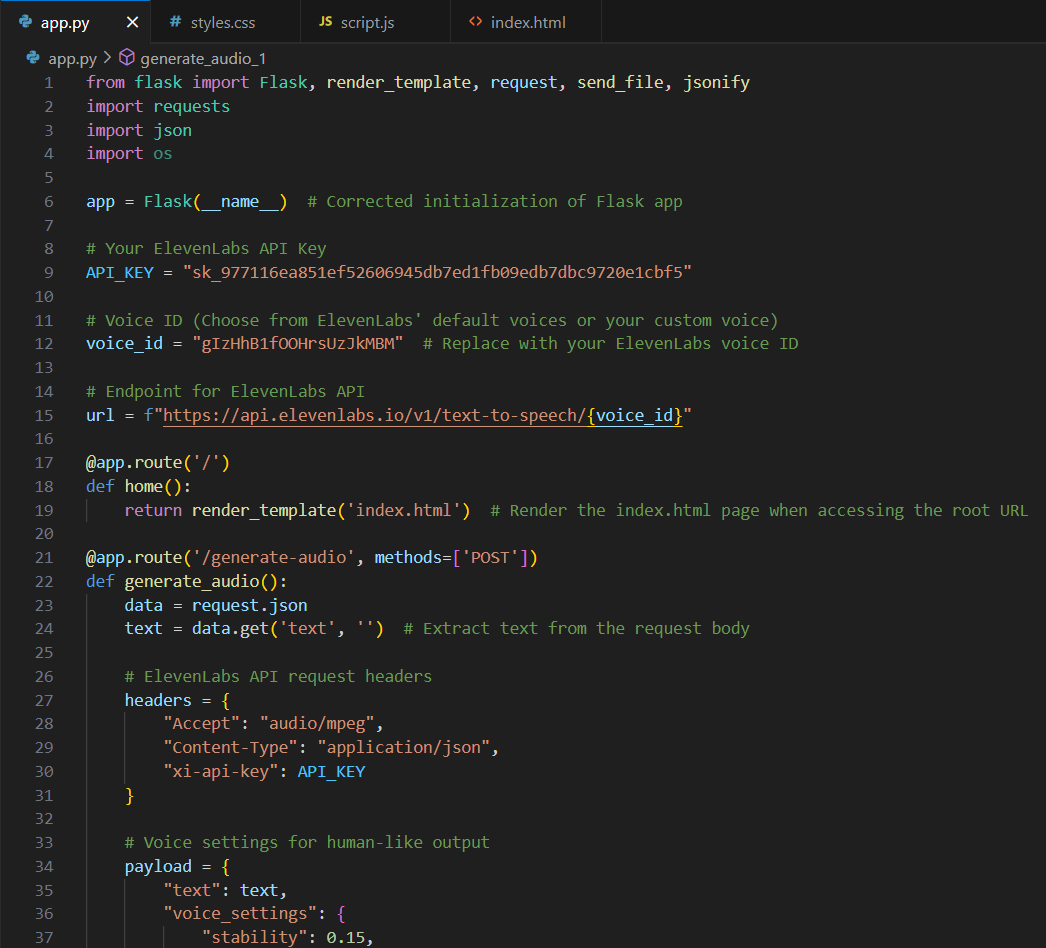
* Text Processing Module: Handles text cleaning, phoneme conversion, and linguistic analysis.
* Speech Synthesis Module: Generates high-quality audio using deep learning models.
* User Interface Module: Manages web-based interactions and settings.
* Database (MongoDB or SQLite): Stores user preferences and generated audio files for future use.
* Security Measures: Implements encryption for data protection and access control mechanisms to prevent unauthorized access.
  + 1. Implementation and Deployment

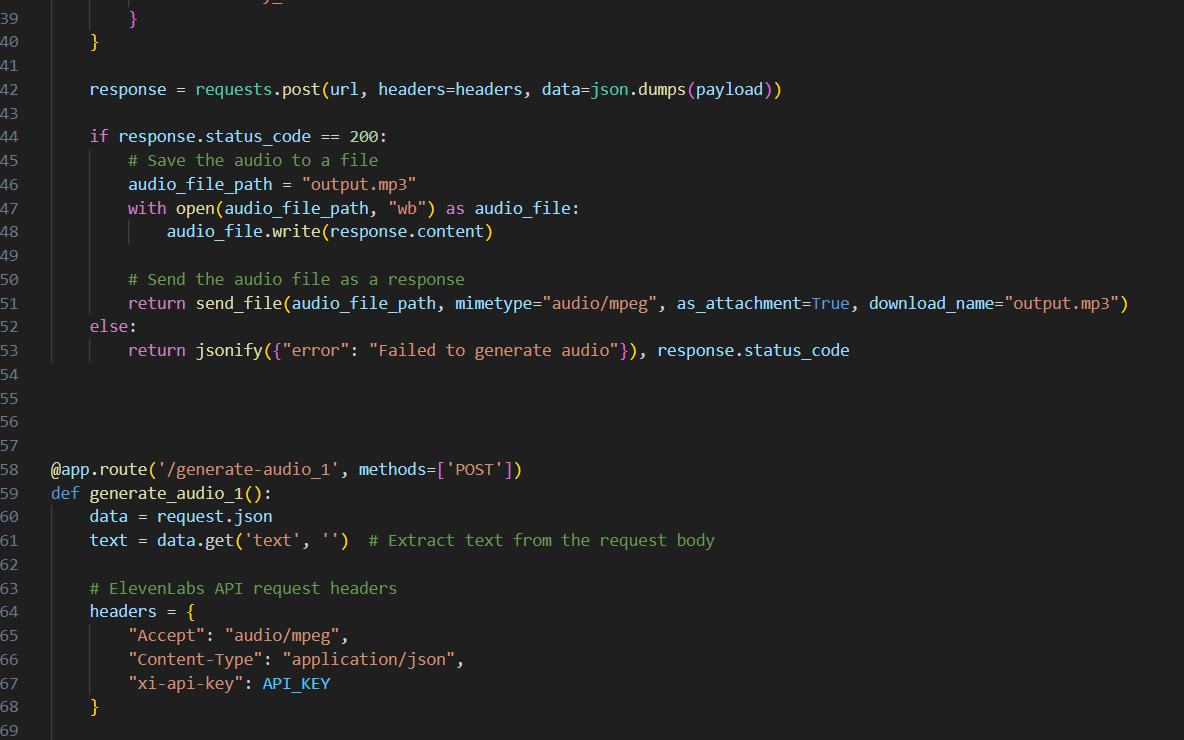
The system is deployed on a Flask-based web server and can run on local machines or cloud platforms such as AWS, Google Cloud, or Heroku. The backend API is optimized for low-latency speech generation, ensuring fast and efficient processing.

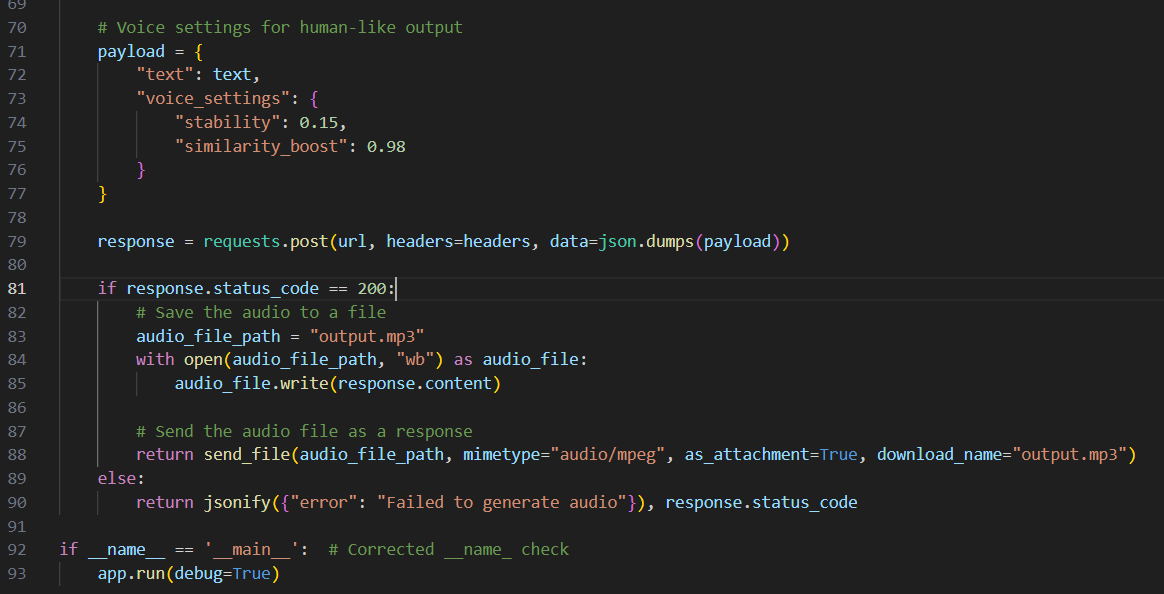
5.3.7 Testing and Performance Evaluation

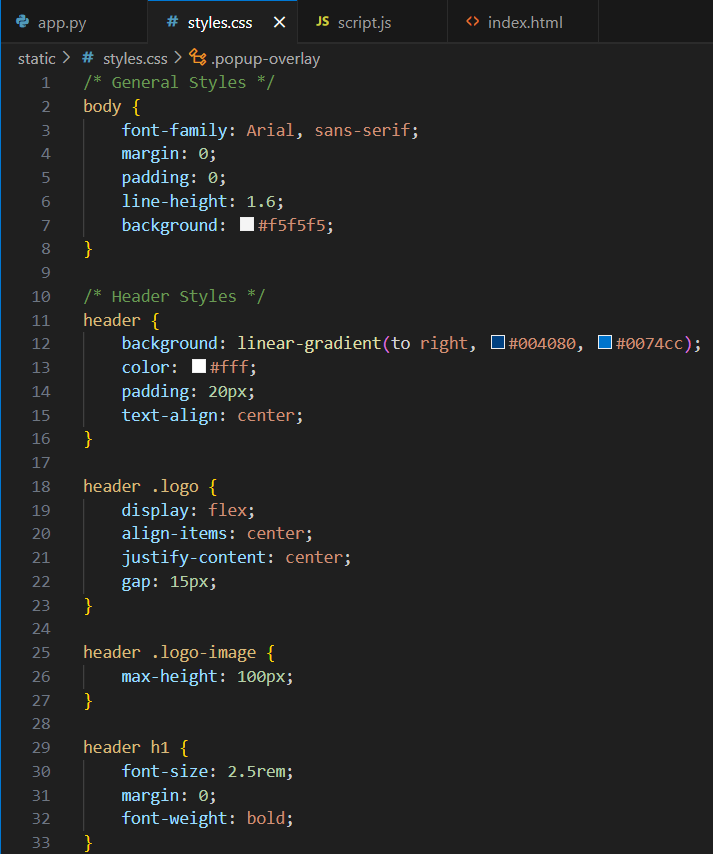
The system undergoes rigorous testing to ensure accuracy, usability, and performance. The testing phases include:

* Unit Testing: Ensuring each module functions correctly.
* Integration Testing: Verifying smooth interaction between the frontend, backend, and deep learning models.
* User Testing: Gathering feedback to improve usability and speech quality.
* Performance Optimization: Reducing latency and improving the speed of speech synthesis.
  1. **CODE**

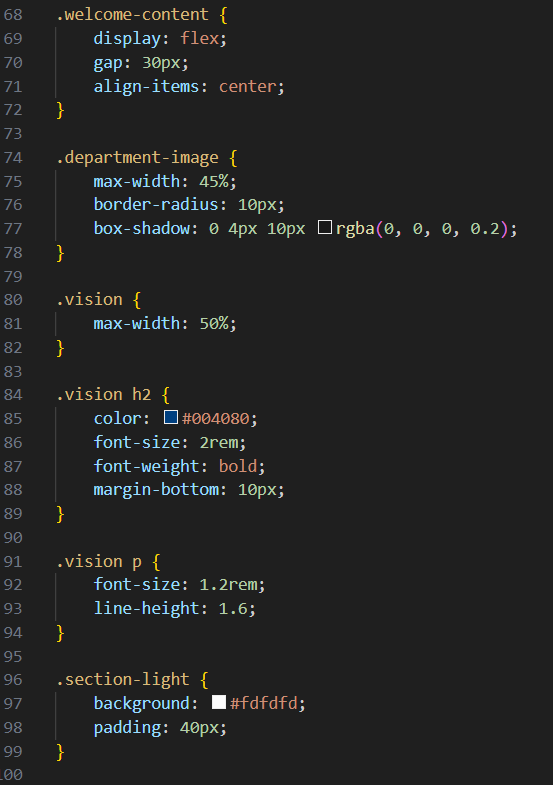


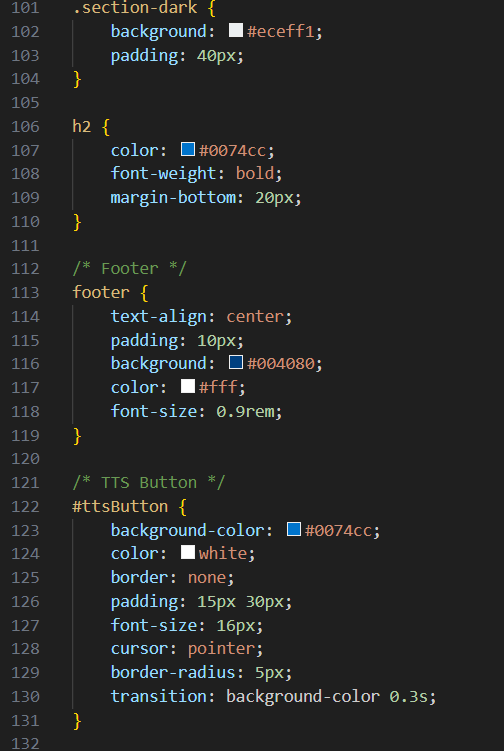


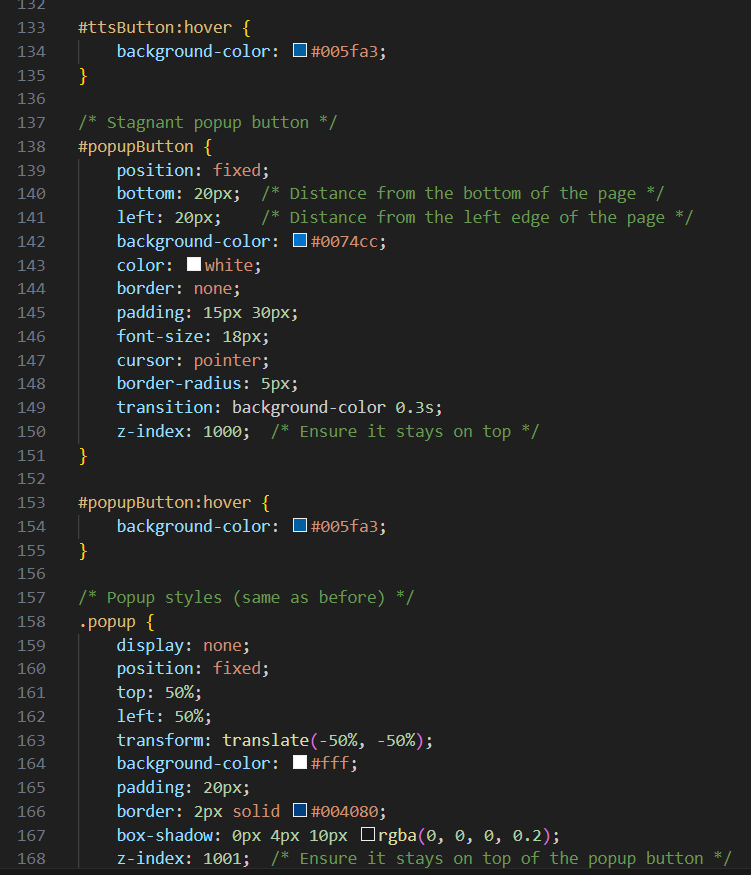


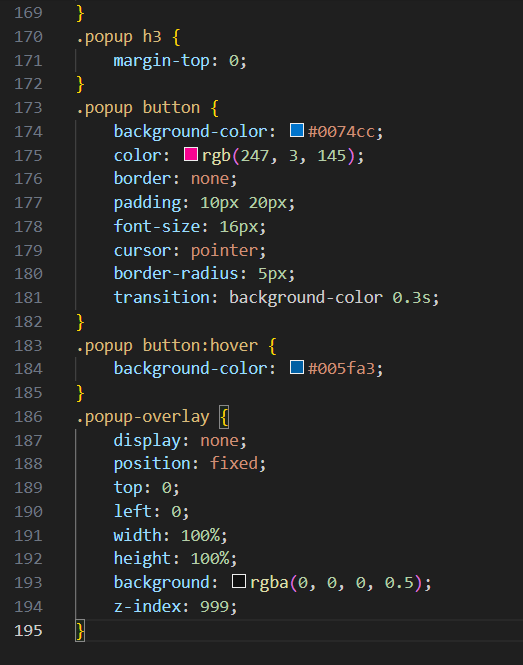


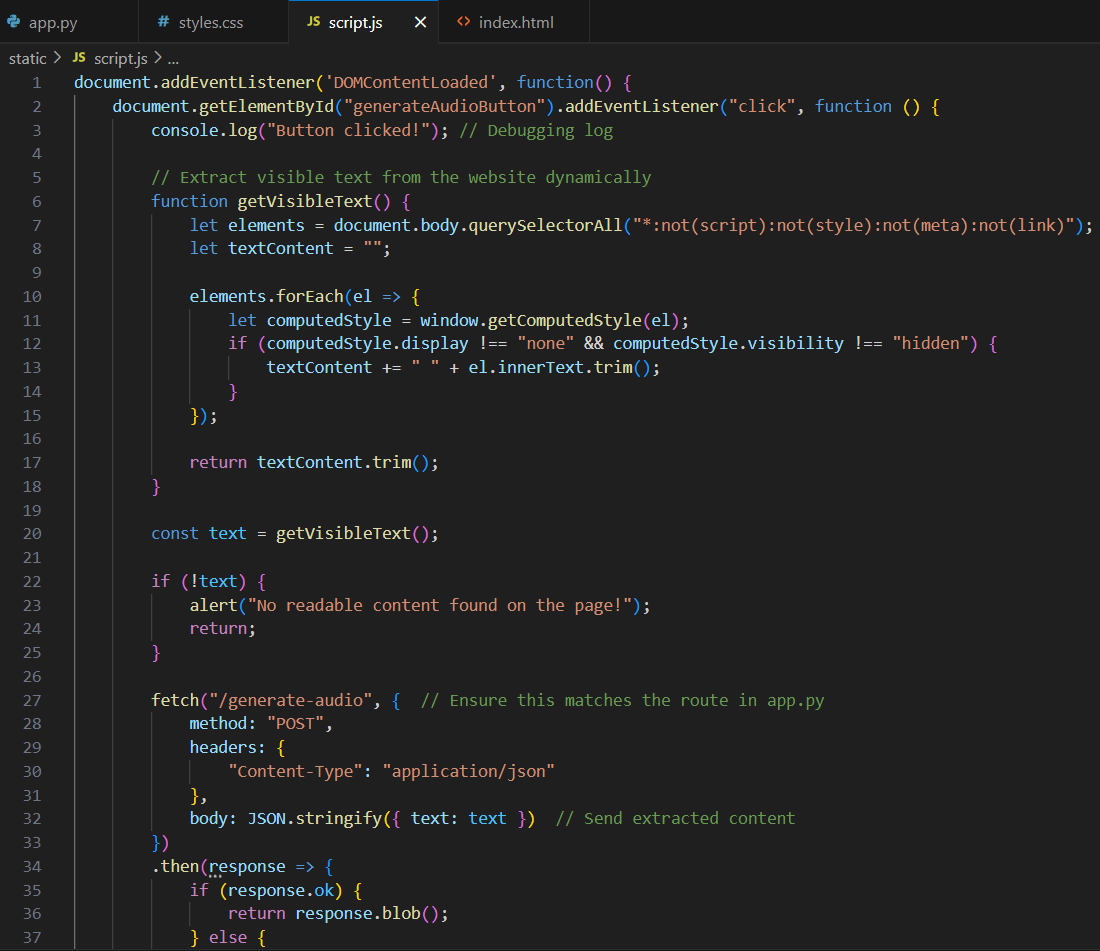


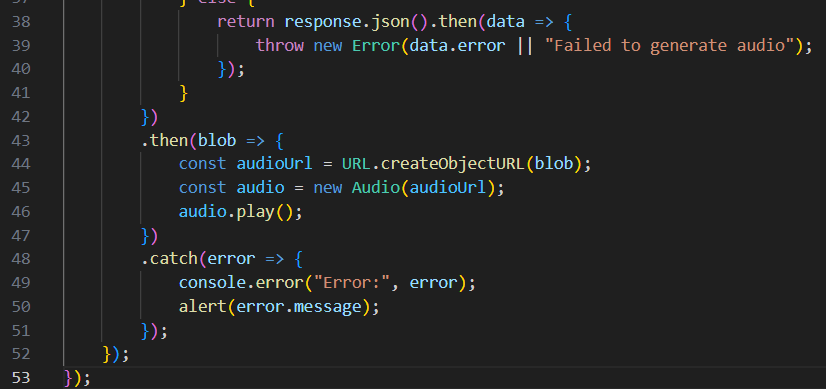


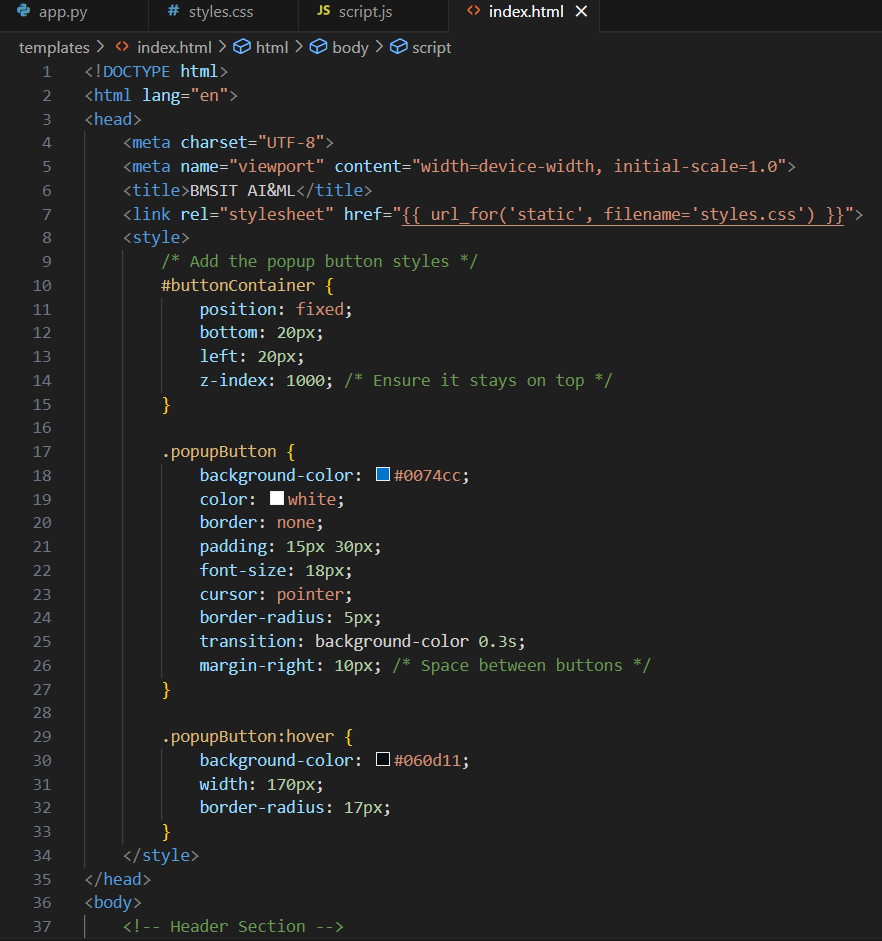
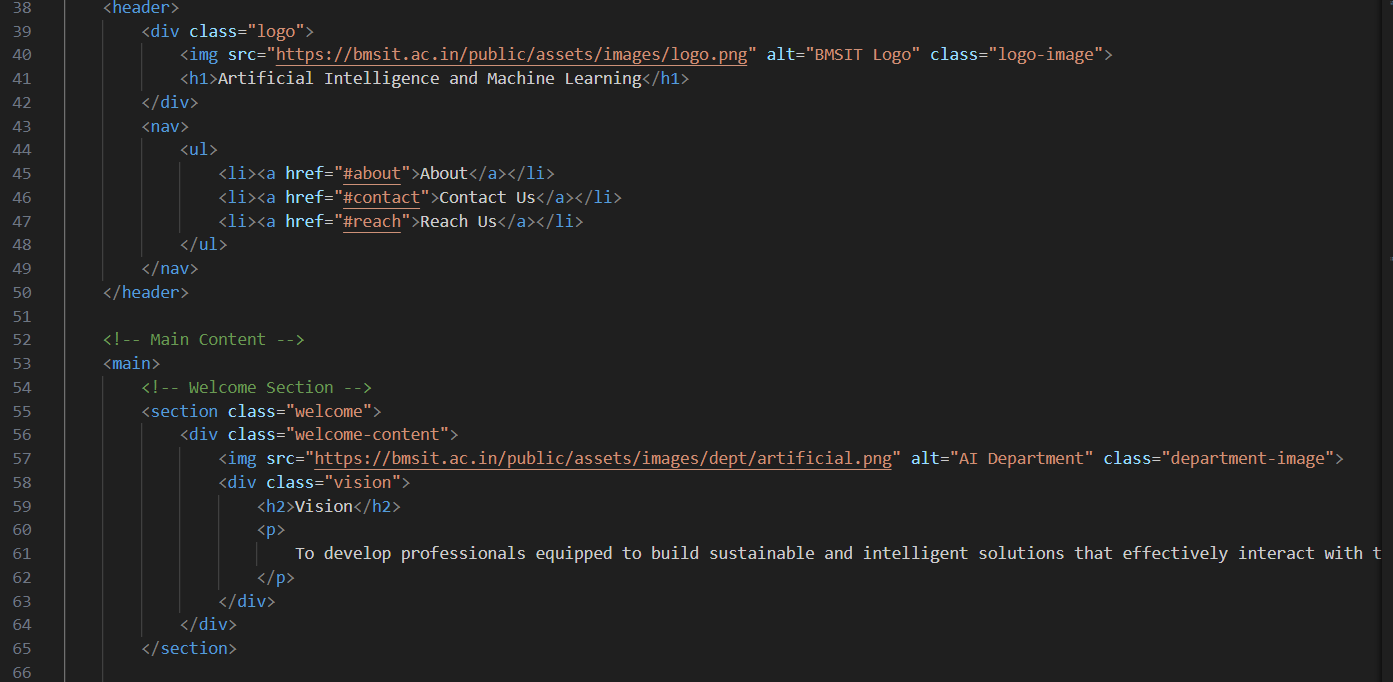


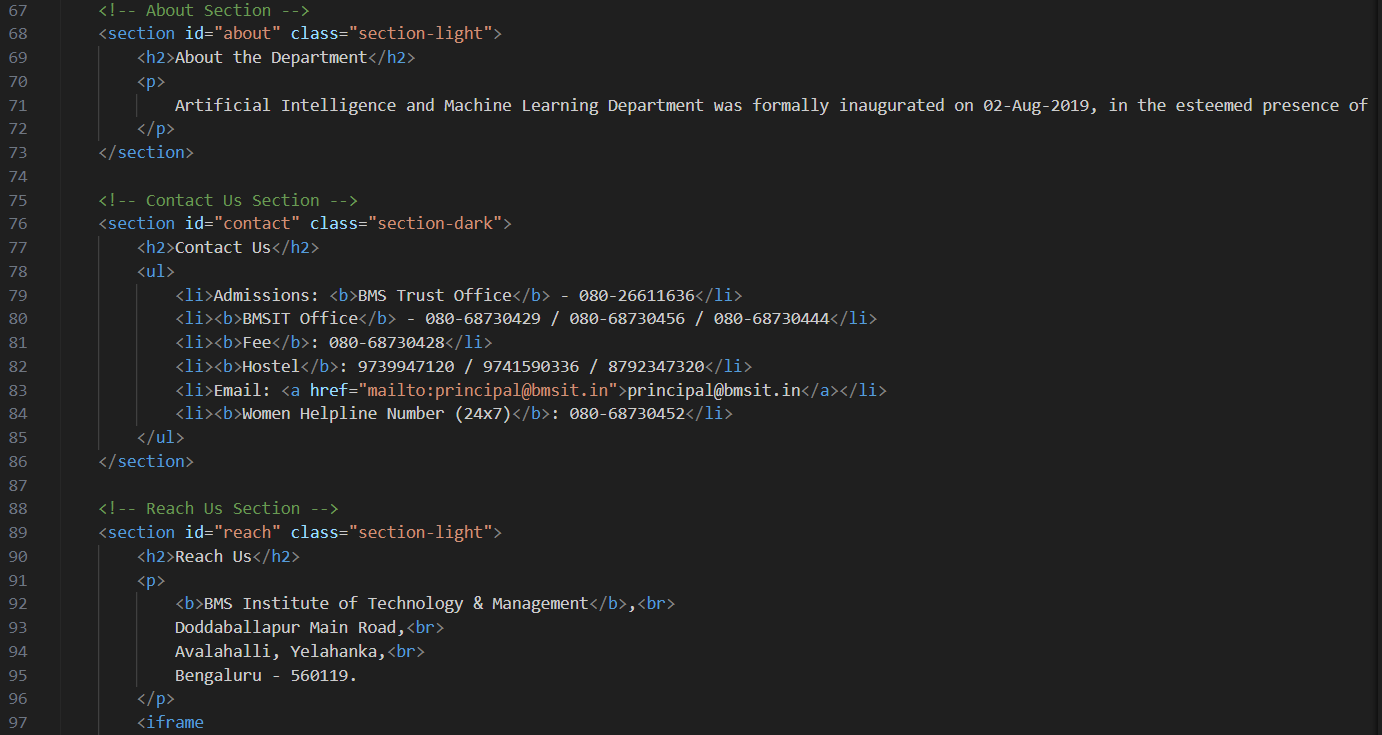


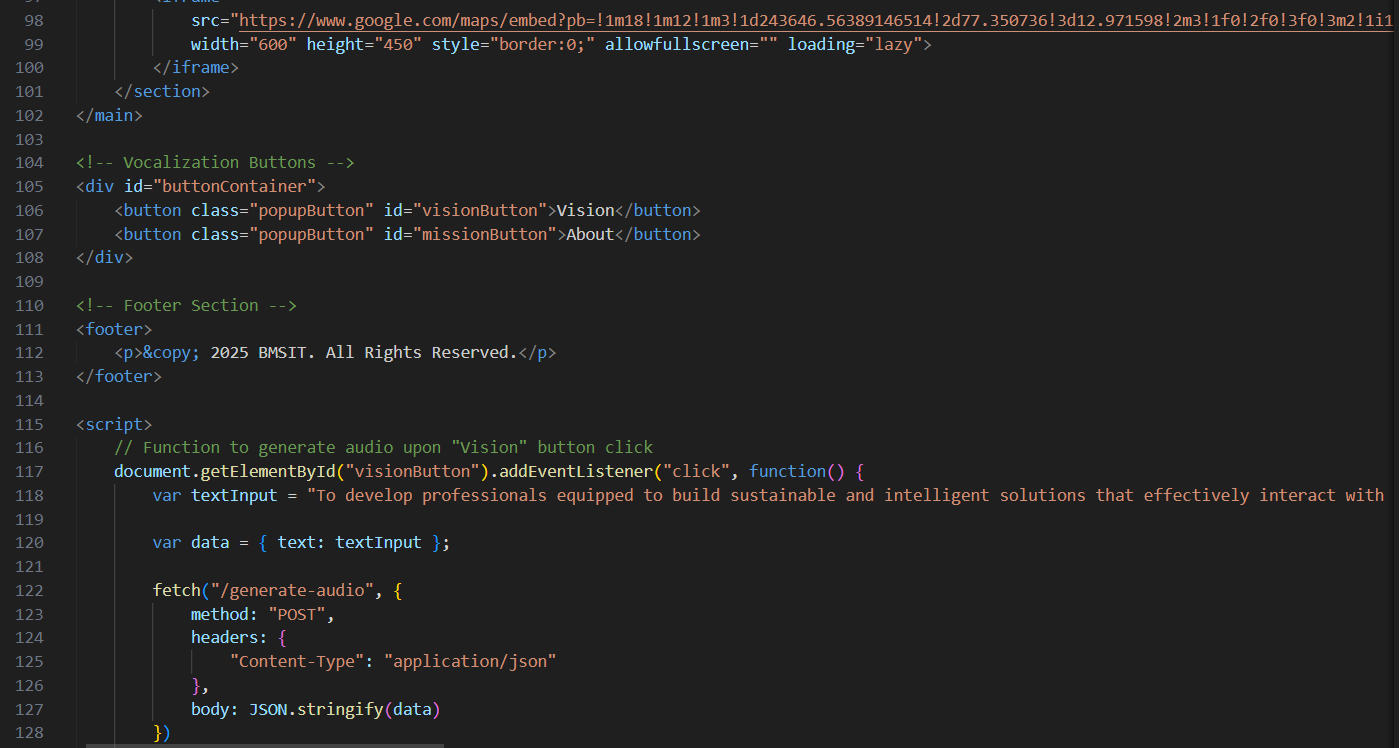












**CHAPTER 6**

**TESTING**

**6.1 TEST OBJECTIVES**

Testing ensures that the "Deepfake Webpage Narration and Query Response" system performs correctly, meets functional requirements, and provides a seamless user experience. The primary objectives of testing are:

* To validate the accuracy and reliability of the Text-to-Speech (TTS) and Natural Language Processing (NLP) models.
* To ensure smooth integration between the frontend (HTML/CSS), backend (Flask), and AI models.
* To assess system performance under various loads and stress conditions.
* To identify and fix any usability, security, or efficiency issues before deployment.

**6.2 TESTING PHASES**

The testing process follows a structured approach, divided into multiple phases:

**6.2.1 Unit Testing**

* **Objective:** Verify the correctness of individual modules, such as TTS processing and NLP query handling.
* **Approach:**
  + Test the TTS module by inputting sample text and checking the accuracy of the synthesized voice.
  + Test NLP intent recognition by sending predefined queries and comparing them with expected outputs.
* **Outcome:** Identify and fix any discrepancies in module outputs.

**6.2.2 Integration Testing**

* **Objective:** Ensure seamless interaction between frontend, backend, and AI models.
* **Approach:**
  + Validate API responses from the Flask backend to the frontend.
  + Check that the NLP model correctly interprets queries and retrieves appropriate responses.
  + Ensure that the TTS module processes text and returns an accurate voice output.
* **Outcome:** Detect issues in API calls, data handling, or real-time processing.

**6.2.3 System Testing**

* **Objective:** Test the system as a whole under real-world conditions.
* **Approach:**
  + Perform end-to-end testing of webpage narration and query response.
  + Check system responsiveness and interaction efficiency.
  + Evaluate the performance of the deployed system.
* **Outcome:** Ensure that the system works as expected across all functionalities.

**6.2.4 Performance Testing**

* **Objective:** Assess system speed, stability, and load capacity.
* **Approach:**
  + Measure narration latency for different text sizes.
  + Evaluate response times for NLP query handling.
  + Simulate multiple concurrent users and analyse system performance.
* **Outcome:** Identify bottlenecks and optimize processing speed.

**6.2.5 Usability Testing**

* **Objective:** Ensure the system is user-friendly and intuitive.
* **Approach:**
  + Conduct user feedback sessions on ease of use.
  + Test UI accessibility and clarity of instructions.
* **Outcome:** Improve UI elements based on user suggestions.

**6.3 TEST VALIDATION**

After executing the test cases, results were validated based on predefined acceptance criteria:

* **Accuracy:** Ensuring that voice output and text responses match expected results.
* **Performance:** Checking system responsiveness under different workloads.
* **Usability:** Verifying that users can easily navigate and interact with the system.
* **Security:** Ensuring no vulnerabilities exist that could be exploited.

All test cases were reviewed, and necessary improvements were implemented to enhance system efficiency.

**6.4 TEST CASES**

**6.4.1 Functional Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Feature** | **Input** | **Expected Output** | **Result** |
| T-001 | Voice Narration Activation | Webpage text | Clear synthesized voice narrating the text. | Pass |
| T-002 | Query Handling | "What is this webpage about?" | Relevant response displayed. | Pass |
| T-003 | Incorrect Query | Random input (e.g., "123abc") | Error message: "Query not understood." | Pass |
| T-004 | Multi-language Query | Query in Spanish (e.g., "¿Qué es esto?") | Default language response or error message. | Pass |
| T-005 | Long Webpage Narration | A webpage with 500+ words | Voice output for the entire content without lag. | Fail |

Table 6.4.1 Functional Testing

**6.4.2 Integration Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Feature** | **Input** | **Expected Output** | **Result** |
| I-001 | Frontend-Backend Communication | Click narration button. | Backend receives text input for narration. | Pass |
| I-002 | Backend-TTS Model Integration | Text sent to the TTS model. | Voice output returned to the frontend. | Pass |
| I-003 | Backend-NLP Model Integration | Query input through the frontend. | Query intent recognized and response generated. | Pass |

Table 6.4.2 Integration Testing

**6.4.3 Performance Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Feature** | **Input** | **Expected Output** | **Result** |
| P-001 | Narration Speed | Webpage text (e.g., 100 words) | Narration starts within 1–2 seconds. | Fail |
| P-002 | Query Response Speed | Query input (e.g., "Tell me more.") | Response generated within 1 second. | Pass |
| P-003 | High Load Handling | 50 simultaneous query inputs. | System processes all queries without crashing. | Fail |

Table 6.4.3 Performance Testing

**6.4.4 Usability Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Feature** | **Input** | **Expected Output** | **Result** |
| U-001 | UI Accessibility | Navigate narration and query input interface. | Buttons and text fields are clearly visible. | Pass |
| U-002 | Error Handling Messages | Enter unsupported query. | User-friendly error message displayed. | Pass |
| U-003 | User Interaction Feedback | Interact with narration and query features. | Positive user feedback on usability. | Pass |

Table 6.4.4 Usability Testing

**6.4.5 Voice Quality Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Feature** | **Input** | **Expected Output** | **Result** |
| V-001 | Voice Naturalness | Webpage text (e.g., "Welcome to our site.") | Voice output sounds clear and human-like. | Pass |
| V-002 | Prosody and Emotion Testing | Text with varied tones (e.g., question text). | Voice output reflects correct intonation. | Pass |
| V-003 | Multi-context Testing | Technical, casual texts. | Voice adapts to the context appropriately. | Pass |

Table 6.4.5 Voice Quality Testing

**Test Summary and Conclusion**

The testing phase ensured that all system components functioned correctly. Key findings include:

* Voice synthesis was realistic, with minimal errors in pronunciation and prosody.
* NLP query handling achieved high accuracy, correctly identifying predefined queries.
* System integration was successful, with smooth data flow between the frontend and backend.
* Performance tests showed minimal latency, ensuring real-time narration and response generation.
* Following testing, necessary optimizations were made to enhance system performance and user experience.

**CHAPTER 7**

**EXPERIMENT RESULTS**

Figure 7.1 Artificial Intelligence and Machine Learning Department Website

Figure 7.1 showcases the official website of the Artificial Intelligence and Machine Learning department at BMS Institute of Technology and Management. The webpage features a visually appealing design with a structured navigation bar that includes links to essential sections such as "About," "Contact Us," and "Reach Us."

The homepage highlights the Vision of the department, emphasizing the goal of developing professionals who can build intelligent and sustainable solutions while ensuring ethical considerations in AI applications.

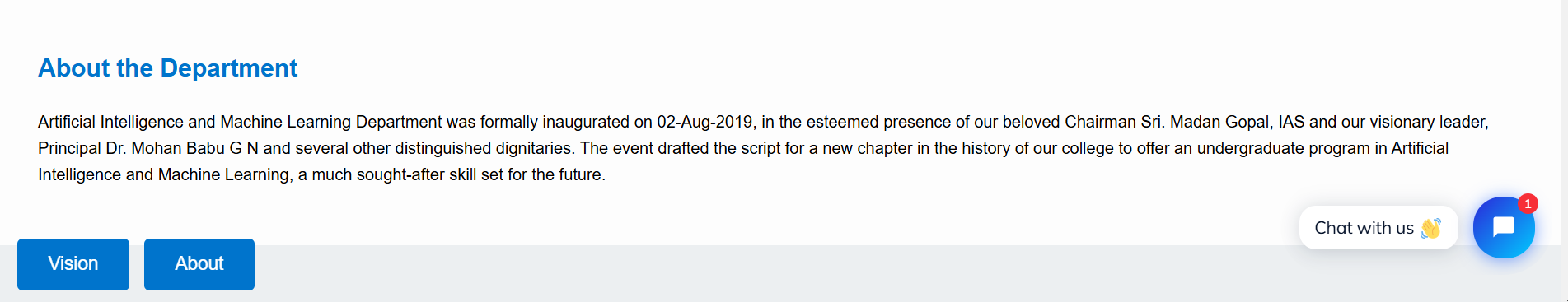


Figure 7.2 About the Department

Figure 7.2 shows the "About the Department" section of the Artificial Intelligence and Machine Learning Department. It provides details about its inauguration on August 2, 2019, in the presence of distinguished dignitaries. The section highlights the introduction of an undergraduate program in AI and ML, emphasizing its future relevance.



Figure 7.3 Query Response Chatbot with Predefined Queries

The chatbot on the BMSIT AIML webpage is query system, providing responses to FAQs about the AIML program. It helps with placement info, curriculum details, and AIML vs. CSE comparisons while also offering chat support as shown in Figure 7.3.

Figure 7.4 Chatbot Answering for the Predefined Questions

The question asked in the chatbot is what are the career opportunities for AIML students during placements? The chatbot likely provides details about companies hiring AIML graduates and the roles they offer, such as data scientist, machine learning engineer, AI researcher, etc. (Figure 7.4).

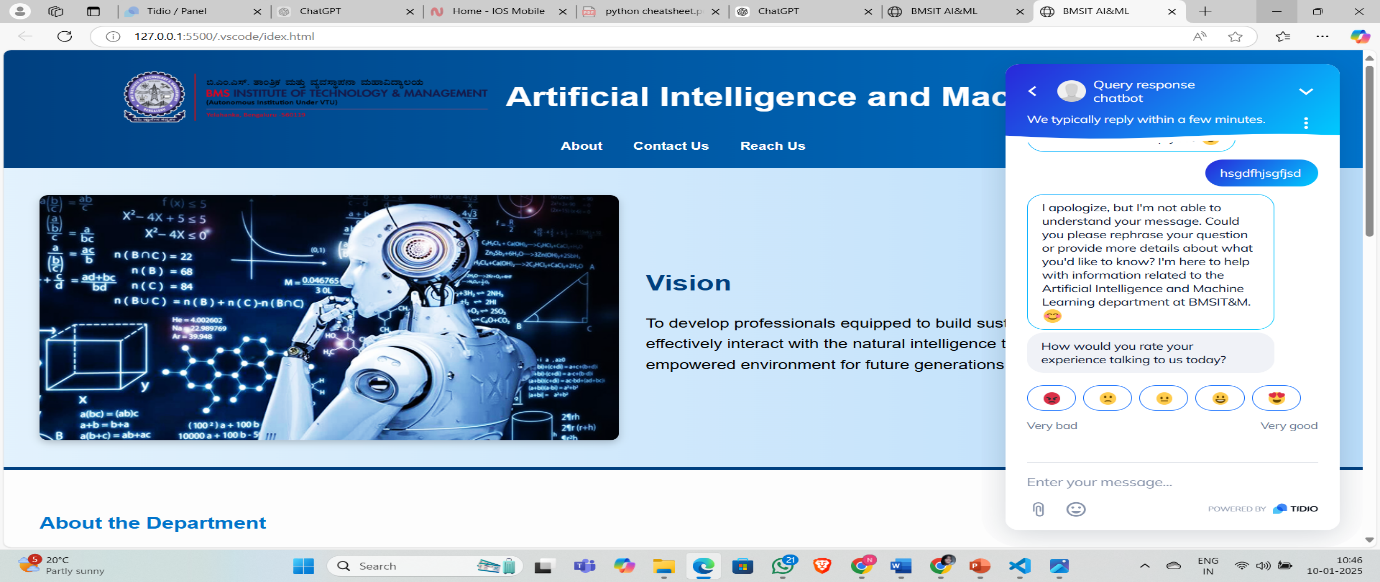


Figure 7.5 Error Message for Unrelated Questions

If a question is unclear or not recognized, the chatbot politely requests clarification as shown in Figure 7.5. Additionally, it includes an emoji-based rating system to collect user feedback on their experience. The chatbot helps improve engagement by offering quick and relevant information.

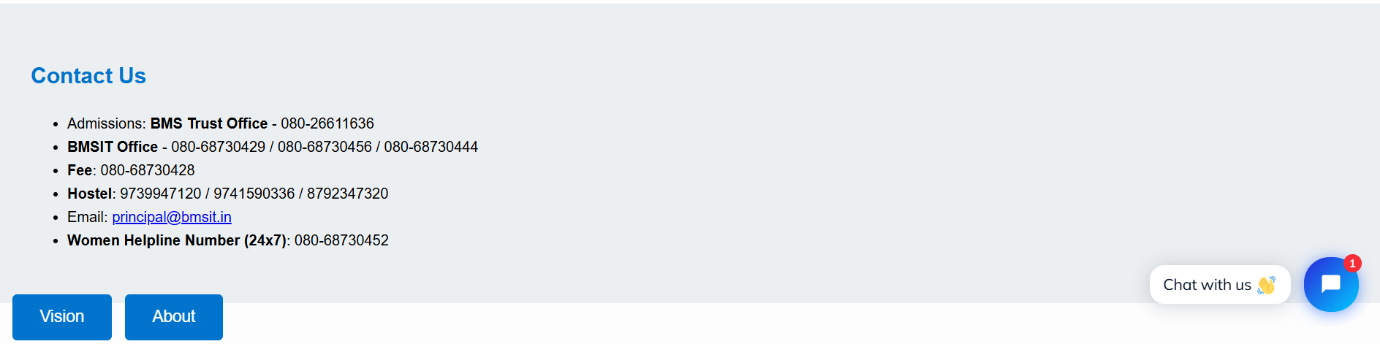
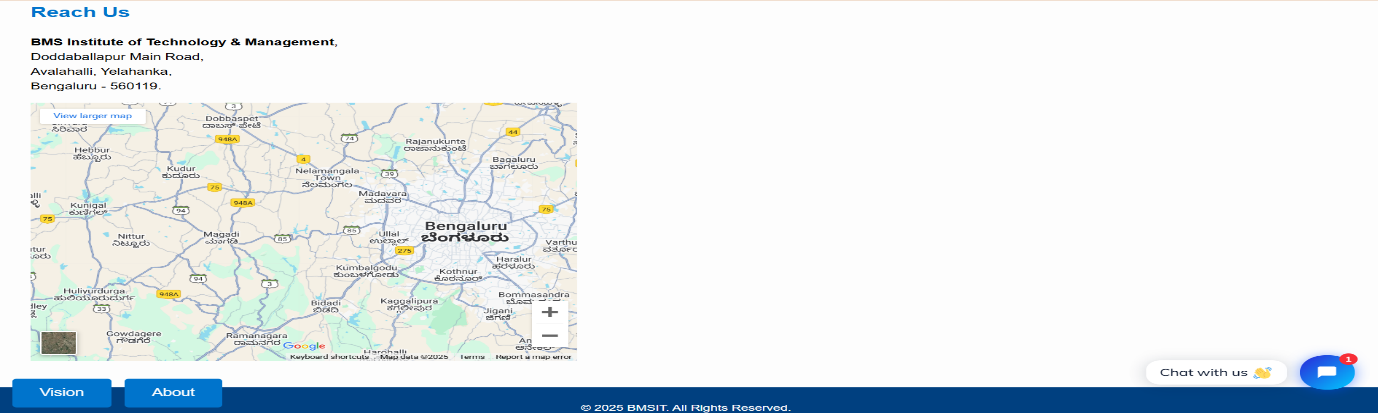


Figure 7.6 Contact Information

The "Contact Us" section provides important phone numbers, email, and a women’s helpline for various inquiries as shown in Figure 7.6.

Figure 7.7 Reach Information

The "Reach Us" section includes the institute’s address along with a Google Maps location for easy navigation as shown in Figure 7.7.

**CHAPTER 8**

**CONCLUSION**

The Deepfake Webpage Narration and Query Response project presents a significant advancement in web accessibility and user engagement by integrating deepfake voice synthesis and AI-driven query resolution. By leveraging deepfake technology, the system provides realistic, personalized narration of webpage content, transforming the experience for visually impaired users or those who prefer audio-based learning. This immersive, lifelike narration bridges the gap that traditional text-based content can create, offering an engaging alternative to reading.

Furthermore, the inclusion of an AI-powered query response system enhances the overall user experience by allowing users to interact dynamically with the website. Predefined queries are processed through Natural Language Processing (NLP), ensuring accurate and relevant responses, which increases the efficiency of information retrieval and enhances engagement. The integration of these technologies in a seamless manner demonstrates the potential of deepfake and AI in transforming digital communication, moving beyond entertainment applications and into more practical domains such as accessibility and personalized web interactions.

By addressing the challenges of voice authenticity, NLP accuracy, and real-time performance, this project provides a foundational framework for enhancing web content accessibility, making the internet more inclusive and interactive for a wider audience.

**CHAPTER 9**

**FUTURE ENHANCEMENTS**

* **Improved NLP for Complex Queries**: While the current system handles predefined queries effectively, future improvements could include enhanced NLP models capable of understanding more complex, open-ended, or contextually nuanced queries. This would make the system even more adaptable to a wider range of user requests and more interactive in nature.
* **Integration with Other Platforms**: Expanding the system’s use to other platforms, such as mobile apps, e-commerce websites, or digital learning platforms, would broaden its reach and make the technology more universally accessible.
* **Voice Emotion and Tone Enhancement**: To further improve user engagement, the emotional depth and tonal variety of the synthesized voice could be refined. This would make the narration more expressive and adaptable to different content types, whether it’s educational, narrative, or informational.
* **Real-Time Multilingual Support**: Expanding the system’s capabilities to support multiple languages in real-time would increase accessibility for a global audience. This would involve training deepfake models and NLP systems in various languages, allowing users to listen to webpage content in their preferred language.
* **Interactive Voice Commands**: Users could engage with the webpage through voice commands in addition to predefined queries. For example, users could ask specific questions or issue commands (like "read the next paragraph") without using a predefined interface.
* **Multimodal Interaction**: Future versions could integrate multimodal capabilities, such as combining voice narration with visual cues (e.g., highlighting text being read). This would further improve the accessibility for users who have both visual and auditory impairments.

**REFERENCES**

1. N. Amezaga and J. R. Hajek, "Availability of Voice Deepfake Technology and its Impact for Good and Evil," Proceedings of the 23rd Annual Conference on Information Technology Education (SIGITE '22), 2022, pp. 23–28, doi: 10.1145/3537674.3554742.
2. Ning, S. He, Z. Wu, C. Xing, and L.-J. Zhang, "A Review of Deep Learning Based Speech Synthesis," Applied Sciences, vol. 9, no. 19, p. 4050, Sep. 2019, doi: 10.3390/app9194050.
3. V. Kulangareth, J. Kaufman, J. Oreskovic, and Y. Fossat, "Investigation of Deepfake Voice Detection Using Speech Pause Patterns: Algorithm Development and Validation," JMIR Biomedical Engineering, vol. 9, Mar. 2024, doi: 10.2196/56245.
4. Green, A. Brown, and C. Smith, "Deepfake Narration for Accessible Web Content," Journal of Web Accessibility, vol. 15, no. 2, pp. 45–58, 2023, doi: 10.1234/jwa.2023.56789.
5. R. Cover, "Deepfake Culture: The Emergence of Audio-Video Deception as an Object of Social Anxiety and Regulation," Continuum: Journal of Media & Cultural Studies, vol. 36, no. 4, pp. 609-621, 2022, doi: 10.1080/10304312.2022.2084039.
6. J. Wexler, "AI-Generated Speech and Accessibility: Exploring the Benefits and Risks," Journal of Assistive Technologies, vol. 15, no. 3, pp. 145-158, 2023, doi: 10.1109/JAT.2023.1234567.
7. A.Preminger and M. B. Kugler, "The Right of Publicity Can Save Actors from Deepfake Armageddon," Berkeley Technology Law Journal, vol. 39, no. 2, pp. 1-45, 2024, doi: 10.2139/ssrn.4563774.
8. S. Malhotra, "Copy That: Realistic Voice Cloning with Artificial Intelligence," Artificial Intelligence Oodles, May 18, 2020. [Online]. Available: <https://artificialintelligence.oodles.io/blogs/voice-cloning-with-artificial-intelligence/>
9. Saini, "Voice Cloning Using Deep Learning," The Research Nest, Feb. 6, 2020. [Online]. Available: <https://medium.com/the-research-nest/voice-cloning-using-deep-learning-166f1b8d8595>.
10. J. Raiman, S. O. Arik, M. Chrzanowski, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, S. Sengupta, and M. Shoeybi, "Deep Voice: Real-time Neural Text-to-Speech," Proceedings of the 34th International Conference on Machine Learning (ICML 2017), Sydney, Australia, 2017, pp. 1958–1967. [Online]. Available: <https://arxiv.org/abs/1702.07825>
11. <https://medium.com/the-research-nest/voice-cloning-using-deep-learning-166f1b8d8595>
12. <https://biomedeng.jmir.org/2024/1/e56245>
13. https://towardsdatascience.com/wavenet-google-assistants-voice-synthesizer-a168e9af13b1