

MINI PROJECT REPORT

on

VOICE BIOMETRICS

Submitted by

RAHUL BIJU | MGP21UCS121

KRISHNA SAGAR P | MGP21UCS093

SWATHY MAHESH | MGP21UCS138

To APJ Abdul Kalam Technological University in partial fulfilment of the requirements for
the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering



SAINTGITS
LEARN.GROW.EXCEL

Department of Computer Science and Engineering

Saintgits College of Engineering (Autonomous)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

June 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SAINTGITS COLLEGE OF ENGINEERING (Autonomous)



2023-2024

CERTIFICATE

Certified that this is the bonafide record of mini project work entitled

VOICE BIOMETRICS

Submitted by

RAHUL BIJU | MGP21UCS121

KRISHNA SAGAR P | MGP21UCS093

SWATHY MAHESH | MGP21UCS138

Under the guidance

Of

Er. Gayathri J L

In partial fulfilment of the requirements for award of the degree of Bachelor of Technology in Computer Science and Engineering under the APJ Abdul Kalam Technological University during the year 2023-2024.

HEAD OF DEPARTMENT

PROJECT COORDINATOR

PROJECT GUIDE

Dr. Arun Madhu

Dr. Reni K Cherian

Er. Gayathri J L

Er. Sania Thomas

DECLARATION

We undersigned hereby declare that the project report ‘VOICE BIOMETRICS’, submitted for partial fulfilment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of Er. Gayathri J L. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:

Name of the students

Signature

Date:

ACKNOWLEDGEMENT

We express my gratitude to Dr. Sudha T, Principal of Saintgits College of Engineering, for providing an excellent ambiance that laid a potentially strong foundation for this work.

We express my heartfelt thanks to **Dr. Arun Madhu**, Head of the Department of Computer Science and Engineering, Saintgits College of Engineering who has been a constant support in every step of our seminar and the source of strength in completing this mini project.

We express my sincere thanks to **Er. Gayathri J L**, Computer Science and Engineering Department for providing all the facilities, valuable and timely suggestions and constant supervision for the successful completion of our mini project.

We are highly indebted to project coordinators, **Dr. Reni K Cherian** and **Er. Sania Thomas** and all the other faculties of the department for their valuable guidance and instant help and for being with us. We extend our heartfelt thanks to our parents, friends and well-wishers for their support and timely help.

Last but not the least we thank Almighty God for helping us in successfully completing this mini project.

ABSTRACT

Voice biometrics, an innovative technology in the realm of biometric authentication, presents a compelling solution for secure and seamless identity verification. This project report provides a comprehensive overview of voice biometrics systems, exploring their design, implementation, and performance evaluation.

Through the analysis of key features such as pitch, frequency, and speech patterns, our project investigates the effectiveness of voice biometrics in accurately identifying individuals. We discuss the development process, including data collection, feature extraction, and algorithm selection, as well as the integration of voice biometrics into real-world applications across various industries.

Furthermore, this report addresses challenges encountered during implementation, such as environmental noise and speaker variability, and proposes strategies for optimization. By presenting experimental results and case studies, we demonstrate the potential of voice biometrics to enhance security and user experience in authentication systems. Ultimately, this project contributes to the advancement of voice biometrics research and its practical implications for identity verification in modern society.

TABLE OF CONTENTS

INDEX	PAGE NO:
ABSTRACT-----	V
LIST OF FIGURES-----	VIII
1. INTRODUCTION-----	1
2. LITERATURE REVIEW-----	2
3. REQUIREMENT ANALYSIS-----	5
3.1 Feasibility Study -----	5
3.2 Software Requirement-----	5
3.2.1 Voice Capture -----	5
3.2.2 IBM Watson Speech to Text API -----	6
3.2.3 Voice Biometric Algorithms -----	6
3.2.4 Enrollment and Verification Modules -----	6
3.2.5 File System -----	6
3.2.6 Development and Testing Tools-----	6
3.3 Hardware Requirement -----	7
3.3.1 Internet Connection -----	7
3.3.2 Web Hosting -----	7
3.3.3 Environmental Control-----	7
3.3.4 Development Environment -----	7
3.3.5 Applications -----	7
3.3.6 Advantages -----	8
3.3.7 Disadvantages-----	8
3.4 General Requirement -----	9
3.4.1 Accessing the voice biometric system -----	9
3.4.2 Interacting with the system -----	9
3.4.3 Entering the Voice Samples from Users -----	9
3.4.4 Logging in Using the Voice Samples -----	9
4. DESIGN-----	10
4.1 Key Features-----	11
4.2 Architecture and technologies-----	11
4.3 Local Host Setup for Retrieval of Audio samples -----	12
4.4 Prompt generation in Voice page-----	12
4.5 User Interface Design-----	13
5. DEVELOPMENT-----	16

6. TESTING & MAINTANENCE	23
6.1 Testing Process	23
6.2 Performance Testing	23
6.3 User Acceptance Testing	24
6.4 Test planning	25
6.5 Security Test	25
7. CONCLUSION	28
REFERENCES	30
APPENDIX	31

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
4.1	Voice Biometrics Use Case Diagram	10
4.2	Prompt generation in Voice Page	13
4.3	Home Page	14
4.4	Enroll Page	15
4.5	Authentication Page	15

CHAPTER 1

INTRODUCTION

In today's rapidly evolving technological landscape, safeguarding our devices and personal information has become paramount. However, the traditional methods of authentication such as passwords or PINs can often be cumbersome and prone to security risks. Voice biometrics offers a seamless and intuitive alternative to traditional authentication methods. By simply using unique voice patterns, users can reliably authenticate their identities to access secure systems and sensitive information. This innovative technology not only eliminates the need for remembering complex passwords but also provides a convenient option for individuals with busy lifestyles. Beyond its convenience, voice biometrics offers unparalleled security. With sophisticated algorithms analysing various vocal characteristics, including pitch, tone, and cadence, it creates a highly secure authentication process that is exceptionally difficult to replicate or spoof.

1.1 Project Objective

Our goal is to implement a robust voice biometrics system for secure user authentication. Leveraging advanced signal processing and machine learning, it seeks to identify unique vocal characteristics reliably. The project also explores practical applications in access control, finance, and telecommunications, contributing advancements to voice biometrics for real-world use.

1.2 Project Scope

In our fast-paced digital world, protecting data is paramount. Voice biometrics offers a seamless, secure alternative to cumbersome passwords. With unique voice patterns, users can effortlessly authenticate their identities, ensuring access to sensitive information. Additionally, voice biometrics enhances accessibility for individuals with disabilities and finds versatile applications, from website logins to high-security environments.

1.3 Project overview

This project focuses on developing a robust voice biometrics system for secure user authentication. Using advanced signal processing and machine learning, it aims to accurately identify individuals based on unique vocal characteristics. The project involves data collection, algorithm development, and practical applications in domains like access control and finance. Through experimentation, it aims to enhance the usability and effectiveness of voice biometrics in real-world scenarios.

CHAPTER 2

LITERATURE REVIEW

Researchers have investigated various methods for voice biometrics, striving to improve accuracy and dependability under diverse conditions. Markowitz [6] discusses various models and methods used in voice biometrics, including the use of Mel-Frequency Cepstral Coefficients (MFCCs) for feature extraction and pattern recognition algorithms for matching voice samples. The primary method involves capturing voice samples from users, extracting unique voice features using techniques like MFCCs, and then comparing these features to stored voice prints using pattern recognition algorithms. The paper details the technical process of voice biometrics, including signal processing techniques and the use of statistical models for voice feature extraction and matching. It concludes that voice biometrics offers a promising and user-friendly approach to secure authentication, with advantages such as non-intrusiveness and ease of use. However, it also acknowledges challenges such as variability in voice due to health conditions, background noise interference, and the risk of spoofing. The overall result is a positive outlook on the future of voice biometrics, emphasizing the need for ongoing research and development to address the existing challenges and enhance the robustness and accuracy of the technology.

González-Rodríguez, Toledano, and Ortega-García [5] discuss various models and methods used in voice biometrics, including the use of Mel-Frequency Cepstral Coefficients (MFCCs) for feature extraction and pattern recognition algorithms for matching voice samples. The primary method involves capturing voice samples from users, extracting unique voice features using techniques like MFCCs, and then comparing these features to stored voice prints using pattern recognition algorithms. The paper details the technical process of voice biometrics, including signal processing techniques and the use of statistical models such as Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM) for voice feature extraction and matching. It concludes that voice biometrics offers a promising and user-friendly approach to secure authentication, with advantages such as non-intrusiveness and ease of use. However, it also acknowledges challenges such as variability in voice due to health conditions, background noise interference, and the risk of spoofing. The overall result is a positive outlook on the future of voice biometrics, emphasizing the need for ongoing research and development to address the existing challenges and enhance the robustness and accuracy of the technology.

Scheffer, N., Ferrer, L., Lawson, A., Lei, Y. and McLaren, M., 2013. [1] discuss various models and methods used in voice biometrics, including the use of i-Vectors for compact representation of speaker and session variability, Probabilistic Linear Discriminant Analysis (PLDA) for improved speaker verification, and Deep Neural Networks (DNNs) for feature extraction and classification tasks. The primary method involves capturing voice samples from users, extracting unique voice features using techniques like Mel-Frequency Cepstral Coefficients (MFCCs) and other advanced features, and then comparing these features to stored voice prints using pattern recognition algorithms. The paper details the technical process of voice biometrics, including signal processing techniques and the use of statistical models such as i-Vectors and PLDA to create robust speaker models that handle variability in voice data. It concludes that recent advancements in voice biometrics, particularly the use of i-Vectors, PLDA, and DNNs, significantly improve the robustness and accuracy of speaker verification systems. The authors highlight the effectiveness of these models in handling diverse and challenging conditions, such as background noise and different recording environments. The overall result is a positive outlook on the future of voice biometrics, emphasizing the need for ongoing research and development to address existing challenges and enhance the robustness and accuracy of the technology.

Aronowitz, H., Hoory, R., Pelecanos, J. and Nahamoo, D. (2011.) [4] discuss various models and methods used in voice biometrics for user authentication, including the use of Gaussian Mixture Models (GMM) and Universal Background Models (UBM) for speaker recognition, and Maximum A Posteriori (MAP) adaptation to refine models for specific users. The primary method involves capturing voice samples from users, extracting unique voice features using techniques like Mel-Frequency Cepstral Coefficients (MFCCs), and then comparing these features to stored voice prints using advanced modeling techniques. The paper details the technical process of voice biometrics, including signal processing techniques and the use of statistical models such as i-Vectors to provide a compact representation of speaker and session variability. It concludes that the incorporation of i-Vectors and advanced modelling techniques significantly improves the robustness and accuracy of speaker verification systems. The authors emphasize the enhanced performance of these systems in challenging conditions, such as background noise and different recording environments. The overall result is a positive outlook on the future of voice biometrics for secure user authentication, highlighting the need for ongoing research and development to address existing challenges and enhance the robustness and accuracy of the technology.

Dhillon, I., Rupp, J., Vankina, A., & Slater, R. (2021). [3] discuss various models and methods used in voice biometrics, including the use of Mel-Frequency Cepstral Coefficients (MFCCs) for feature

extraction and pattern recognition algorithms for matching voice samples. The primary method involves capturing voice samples from users, extracting unique voice features using techniques like MFCCs, and then comparing these features to stored voice prints using pattern recognition algorithms. The paper details the technical process of voice biometrics, including signal processing techniques and the use of statistical models for voice feature extraction and matching. It concludes that voice biometrics offers a promising and user-friendly approach to secure authentication, with advantages such as non-intrusiveness and ease of use. However, it also acknowledges challenges such as variability in voice due to health conditions, background noise interference, and the risk of spoofing. The overall result is a positive outlook on the future of voice biometrics, emphasizing the need for ongoing research and development to address the existing challenges and enhance the robustness and accuracy of the technology. Kuznetsov, A., Oleshko, I., Chernov, K., Bagmut, M. and Smirnova, T., 2019, [2] present a study on biometric authentication employing Convolutional Neural Networks (CNNs). The paper discusses the utilization of CNNs for biometric authentication tasks, likely focusing on image recognition applications. CNNs are trained on biometric data, such as facial images or fingerprint images, to learn discriminative features for individual identification. The study may detail the technical implementation of CNNs in biometric authentication, including data pre-processing, network architecture design, and training methodologies. Although the specific conclusion results are unavailable without access to the full paper, it is reasonable to expect a discussion on the effectiveness of CNNs for biometric authentication. This discussion might cover aspects such as accuracy, efficiency, and security compared to traditional methods. Furthermore, potential avenues for future research and enhancements in CNN-based biometric authentication systems may be highlighted.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 Feasibility Study

Market Demand:

The project addresses market demand by offering a voice biometrics system focused on enhancing security through real-time fraud detection and prevention. By analysing voice patterns for user enrollment and authentication, the system ensures accurate identification, reducing fraudulent activities and building user trust. This innovative solution meets the growing need for secure authentication methods, positioning it competitively in the market.

Operational Feasibility:

The voice biometrics system prioritizes ease of use with an intuitive interface for seamless user access. Streamlined voice recording and authentication processes ensure a hassle-free experience, enhancing operational feasibility.

Technology Stack:

On the backend, it uses Flask, a lightweight Python web framework, to handle HTTP requests and manage server-side logic. The frontend comprises HTML, CSS, and JavaScript, creating a responsive and interactive user interface. The project employs several Python libraries for its core functionality: `speech_recognition` and IBM Watson's `SpeechToTextV1` for speech-to-text conversion, `python_speech_features` for extracting MFCCs.

Legal Feasibility:

Adherence to data protection regulations, including implementing user consent mechanisms for voice data processing, and assessing legal liabilities such as data breaches, are vital considerations.

3.2 Software Requirement

3.2.1 Voice Capture

Audio Input: The system captures audio input through microphones or other recording devices. This raw audio data is often in WAV or other standard audio file formats.

3.2.2 IBM Watson Speech to Text API

The IBM Watson Speech to Text API is an advanced cloud-based solution that transforms spoken audio into text with exceptional accuracy. It employs sophisticated machine learning techniques to recognize speech in multiple languages and dialects. Within this platform, the API plays a crucial role in converting user-recorded voice samples into written text, a fundamental step in both enrollment and authentication procedures. This process involves capturing and processing the user's voice data, which is then submitted to the Speech to Text service for transcription.

3.2.3 Voice Biometric Algorithms

Speaker Recognition Engines: These algorithms analyse voice features to create a unique voice print or model for each user.

Machine Learning Frameworks: To implement and train models for speaker verification/identification. Common frameworks include scikit-learn.

3.2.4 Enrollment and Verification Modules

Enrollment Software: To register users' voiceprints in the system by capturing and processing their voice samples.

Verification Software: To compare a new voice sample with stored voiceprints and determine if there is a match.

3.2.5 File System

In the voice biometrics application, the file system is utilized to store audio samples and models for each user in an organized manner. Each user has a dedicated directory within a parent "Users" directory where their audio samples are saved as WAV files, typically named with unique identifiers. Additionally, a separate "Models" directory stores the Gaussian Mixture Models (GMMs) for each user, with filenames including the username for easy association. This structure allows efficient storage and retrieval of audio samples and user models, facilitating accurate speaker authentication based on the stored speech patterns.

3.2.6 Development and Testing Tools

Development Environments: VS Code is leveraged for its versatility in integrating the frontend, backend, and machine learning components. It supports various extensions that facilitate seamless development across HTML, CSS, JavaScript for the frontend, Flask for the backend, and Python for machine learning. Additionally, Google Colab is employed for developing and testing machine

learning code due to its ease of use, collaborative features, and access to powerful computational resources, making it ideal for handling intensive machine learning tasks.

Testing Frameworks: Tools for unit testing, integration testing, and performance testing of the voice biometric system.

3.3 Hardware Requirement

3.3.1 Internet Connection

A stable and reliable internet connection is crucial for developing and deploying Voice Biometrics. It enables communication with the necessary APIs and ensures uninterrupted connectivity with users.

3.3.2 Web Hosting

Selecting a reliable web hosting service is crucial for the successful deployment of the voice biometrics project. It's essential to prioritize factors such as uptime reliability, scalability, performance, security, and responsive customer support. Once the hosting provider is chosen, deploying the application involves configuring the server environment and deploying the application code (e.g., Flask app).

3.3.3 Environmental Control

Acoustic Treatment: In environments with significant background noise, acoustic panels, soundproofing, and echo-reducing materials can enhance voice capture quality.

Temperature and Humidity Control: Ensuring that hardware operates within optimal temperature and humidity ranges to maintain performance and longevity.

3.3.4 Development Environment

A computer or laptop is necessary for developing and testing Voice Biometrics. It should meet the minimum requirements for the chosen development environment and programming language. The version and compatibility of the computer should be sufficient to support the development and deployment of Voice Biometrics.

3.3.5 Applications

1. Security Authentication: Access Control: Securing access to physical locations or sensitive areas by verifying individuals through their voice.
2. Multi-Factor Authentication (MFA): Combining voice biometrics with other authentication methods (e.g., passwords, fingerprints) for added security in sensitive applications.

3. Smartphones and Tablets: Enhancing security features by allowing users to unlock their devices or authorize transactions using their voice.

3.3.6 Advantages

1. Convenience and User Experience
 - Hands-Free Authentication: Users can be authenticated without needing to physically interact with a device, making it convenient in situations where hands are occupied.
 - Natural Interaction: Speaking is a natural and easy way for users to interact with systems, reducing the need to remember passwords or carry additional tokens.
2. Cost-Effective Implementation
 - No Special Hardware Required: Many voice biometrics systems can be implemented using existing devices with in built microphones.
3. Scalability
 - Remote Authentication: Voice biometrics can be used for remote authentication, making it ideal for telecommunication, online services, and remote workforce management.
 - Cross-Platform Compatibility: Can be integrated across various platforms and devices, providing a seamless user experience.
4. Accessibility
 - Supports Diverse User Needs: Beneficial for users with disabilities who may have difficulty using traditional authentication methods like typing passwords or using fingerprint sensors.

3.3.6 Disadvantages

1. Environmental Sensitivity:
 - Background Noise: Voice biometrics can be affected by background noise, making it difficult to accurately capture and analyse voice prints in noisy environments.
 - Acoustic Variations: Different environments (e.g., indoor vs. outdoor, different room acoustics) can affect the quality of voice recordings and the accuracy of the biometric system.
2. Physical and Health Variability
 - Voice Changes: An individual's voice can change due to illness (e.g., cold or flu), aging, stress, or fatigue, which can affect the accuracy of voice recognition.

- Temporary Conditions: Temporary changes in voice due to conditions like laryngitis can lead to authentication failures.

3. Performance in Diverse Populations

- Accent and Dialect Variations: Differences in accents, dialects, and speaking styles can affect the accuracy of voice biometrics, potentially leading to biased outcomes against certain groups.
- Language Variability: Systems may struggle with languages or dialects that were not well-represented in the training data.

3.4 General Requirement

3.4.1 Accessing the voice biometric system

- The user should be able to access the voice biometric system whenever they need for easy login using their voice.
- The login feature made available to the user must be properly working and should have all its expected functionalities.

3.4.2 Interacting with the system

- The user should be able to interact using the options given in the menu.
- When selecting an option, the system should perform the expected response or action.
- Moving between different menu sections are possible. For this purpose, proper navigation buttons are to be made available.

3.4.3 Entering the Voice Samples from Users

- A new user is to input their voice samples to train the model.
- The model is trained using these voice samples from the new user.
- The user has successfully enrolled.

3.4.4 Logging in Using the Voice Samples

- The voice biometrics allows the user for easy log in using his voice.
- An existing user can input his voice sample and log into the website.

CHAPTER 4

DESIGN

Voice biometrics leverages the unique characteristics of an individual's voice for identity verification, providing enhanced security. This report outlines the key features, architecture, and implementation of a voice biometrics system designed to identify and authenticate users based on their voice patterns.

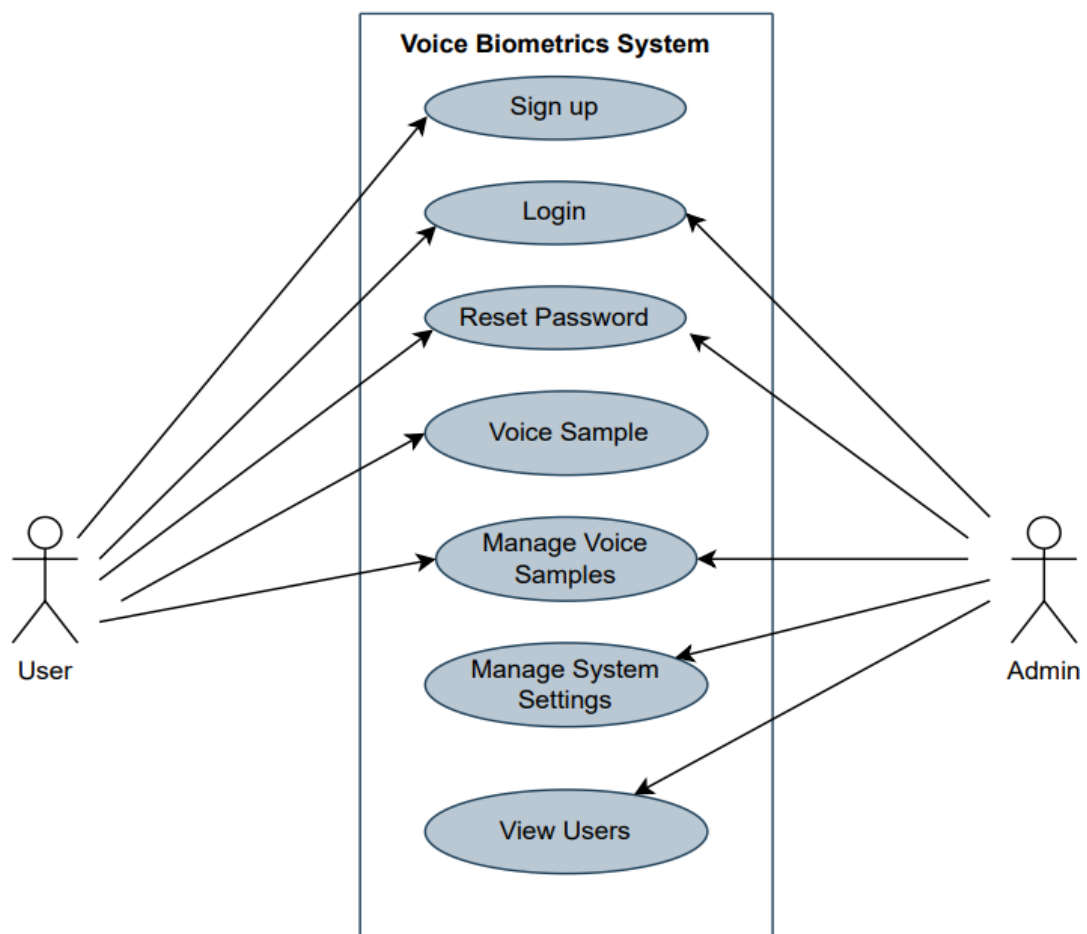


Figure 4.1: Voice Biometrics use case diagram

4.1 Key features

The Voice Biometrics model offers the following features:

1. Voice Enrollment: New users can create an account and enroll their voice samples. This process involves recording and submitting voice samples to generate a unique voice print that will be stored in the database for future authentication.

2. Voice Authentication: Existing users can authenticate themselves by recording a new voice sample. The system compares the new sample against the stored voice prints to verify the user's identity.
3. Speech Recognition: Utilizes IBM Watson Speech to Text service to transcribe spoken words into text. This transcription is used to match against expected phrases for both enrollment and authentication processes.
4. Multi-factor Authentication: Integration with other authentication methods, such as passwords or PINs to provide multi-factor authentication for enhanced security.

4.2 Architecture and technologies

The architecture of Voice Biometrics is designed to be modular, scalable, and secure. The main components include the client interface, server-side application logic, voice processing, and storage mechanisms. Here's a detailed breakdown:

- Client Interface:

Frontend: HTML, CSS, JavaScript, and Flask templates for rendering dynamic content.

Interactions: Users interact with the application via a web interface, where they can enroll or authenticate using voice samples.

- Server-Side Application:

Backend Framework: Flask, a lightweight WSGI web application framework in Python.

Endpoints: Several routes (/ , /home, /enroll, /auth, /voice, /biometrics, /verify) handle different functionalities like user enrollment, authentication, voice recording, and biometrics processing.

- Voice Processing:

Speech Recognition: IBM Watson Speech to Text API is used to convert spoken words into text.

Voice Activity Detection: SpeechRecognition library is used to adjust for ambient noise and detect voice activity.

Feature Extraction: Python Speech Features library to extract MFCC and delta features from audio signals.

Fuzzy Matching: Fuzzy Wuzzy library to compare the recognized text with the expected text using Levenshtein Distance.

Random Words Generation: Random Words library to generate random words for voice prompts.

- **Model Training and Verification:**

Gaussian Mixture Model (GMM): Scikit-learn's GaussianMixture class is used to train models on the extracted voice features.

Storage: Pickle is used to serialize and store trained GMM models for each user.

- **Storage:**

File System: User directories are created to store audio files and models.

Models Directory: Stores the serialized GMM models.

4.3 Local Host Setup for Retrieval of Audio samples

For a successful local host setup in a voice biometrics system, it's essential to manage audio sample storage and ensure the use of appropriate file formats. Proper configuration of these elements is crucial for accurate and efficient voice recognition and authentication processes.

- **Audio Storage**

Create a structured directory to store audio samples securely. Each user's audio data should be stored in a dedicated folder, organized by username, to ensure easy access and management. This hierarchical storage system aids in efficiently retrieving and processing audio samples during enrollment and authentication phases.

- **File Format**

Ensure that audio samples are stored in a consistent and high-quality format, preferably WAV, which is widely supported and maintains audio fidelity. The WAV format is ideal for voice biometrics as it preserves the necessary details in the audio signal, ensuring accurate feature extraction and subsequent voice model training and authentication.

4.4 Prompt generation in voice page

The Random Words library in Python is a simple tool that generates random English words. In Voice Biometrics, this library is employed to create random word prompts during the voice authentication process. These prompts serve as the text that users are instructed to recite during voice authentication. By using randomly generated words, Voice Biometrics enhances security and prevents predictable patterns, ensuring that each authentication attempt requires the user to speak different, unpredictable words. This randomness adds an additional layer of challenge for potential unauthorized access attempts and strengthens the overall security of the voice biometric authentication system.

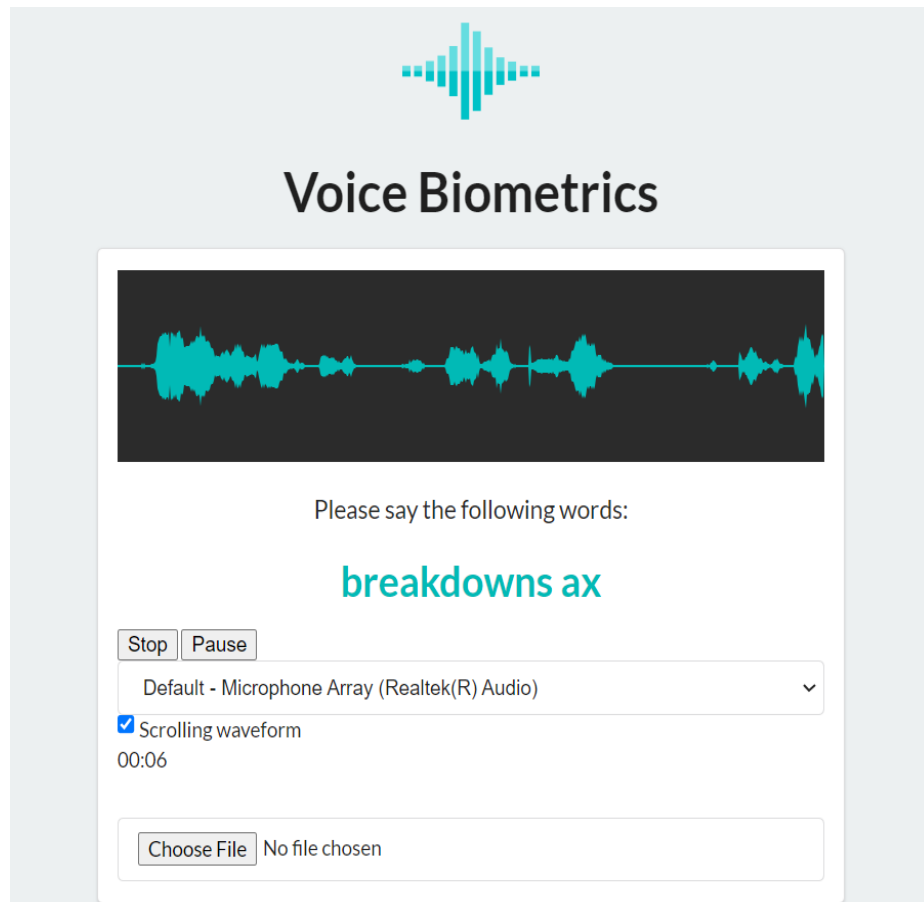


Figure 4.2: Prompt generation in voice page

4.5 User Interface Design

The UI design of Voice Biometrics is crafted to provide a user-friendly graphical user interface (GUI) for both enrollment and authentication processes. The design focuses on simplicity, clarity, and ease of use to ensure that users can navigate through the application with minimal effort. Here's a brief overview of the key components and pages of the UI:

- Homepage: Offers two primary options - "Enroll a new user" and "Authenticate an existing user".
- Enrollment Page: Allows new users to create an account and enroll their voice samples.
- Authentication Page: Enables existing users to authenticate themselves using their voice.
- Voice Biometrics Page: To record the user's voice sample for analysis and comparison with stored voice prints.

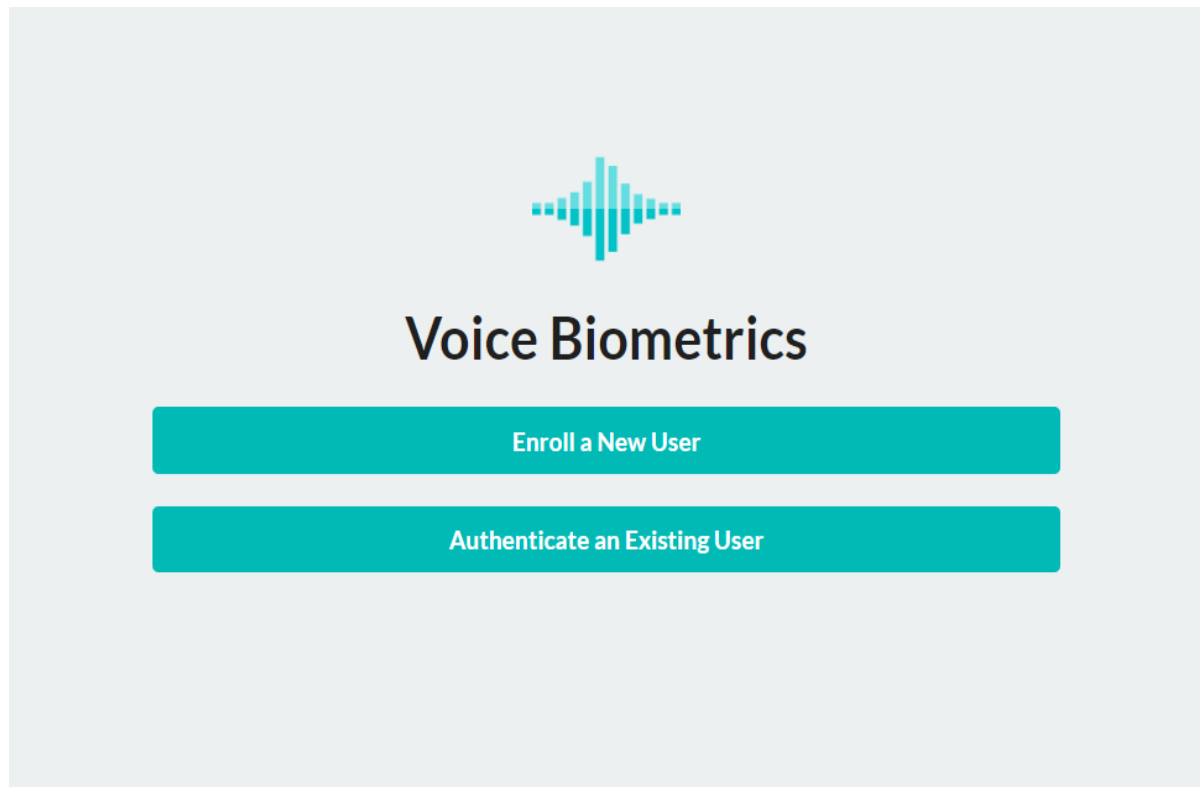


Figure 4.3: Home Page

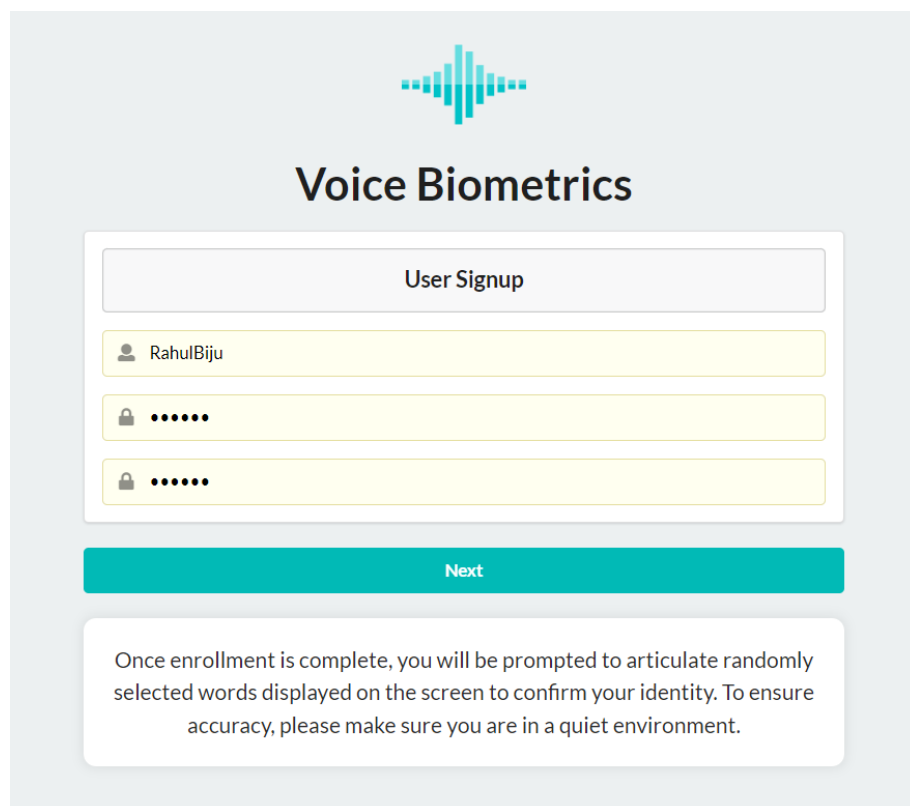


The image displays the 'User Signup' form within the 'Voice Biometrics' application. The form is centered on a light gray background. At the top of the form area is the 'Voice Biometrics' logo and title. Below this, a white box contains the title 'User Signup'. The form includes three input fields: the first is for a username, containing the text 'RahulBiju'; the second and third are for passwords, each represented by a lock icon followed by six dots. A teal 'Next' button is positioned below the input fields. At the bottom of the form, a white box contains instructional text: 'Once enrollment is complete, you will be prompted to articulate randomly selected words displayed on the screen to confirm your identity. To ensure accuracy, please make sure you are in a quiet environment.'


Figure 4.4: Enroll Page



Voice Biometrics

User Signin

 RahulBiju



Next

Figure 4.4: Authentication Page

CHAPTER 5

DEVELOPMENT

In the development phase, we're transforming abstract ideas and requirements into a practical and user-friendly voice recognition system.

The development process can be divided into the following stages

- **Requirement Analysis**

During this stage, the project team gathers and analyzes the requirements essential for the development of the voice biometric system. This process involves collecting voice samples, capturing user information, creating and confirming voice prints, and establishing feedback mechanisms to ensure system usability.

- **Design**

In the design phase, a comprehensive plan is created detailing the system architecture, user interface, and machine learning model. This stage focuses on translating conceptual ideas and requirements into a functional and user-friendly system design.

- **Development**

The development phase involves implementing the voice biometric system using a machine learning model capable of identifying and predicting users based on their voice. Integration with an interactive user interface, including login and sign-up functionalities, is also carried out during this stage.

- **Testing**

During the testing phase, the system undergoes various types of testing to ensure accurate and secure user authentication based on voice samples. This includes testing for functional correctness, performance, security measures, and overall usability of the system.

- **Deployment**

The deployment phase involves deploying the developed system to ensure scalability, reliability, and availability to users. This stage ensures that the system can effectively serve its intended purpose in a production environment.

- **Monitoring and Maintenance**

Continuously monitoring the system's performance is crucial post-deployment. This involves identifying and addressing any issues or enhancements that arise to maintain optimal system

functionality. Regular maintenance activities help ensure the long-term reliability and usability of the voice biometric system.

5.1 Requirement Analysis

Thorough requirement analysis was conducted to identify the specific needs and expectations of the target users for the voice biometrics system. Insights were collected from potential users to determine the essential features and functionalities that the system should offer. The goal is to create a voice biometrics system that enhances user security by accurately identifying and authenticating users based on their unique voice characteristics. The system should provide seamless integration with existing user interfaces for login and sign-up functionalities. Additionally, the system should be resilient against spoofing attacks, ensuring the integrity and security of the authentication process.

5.2 Technology selection

- Visual Studio Code (VS Code): VS Code is a lightweight and versatile code editor that supports Python development through extensions. It offers a wide range of extensions for Python development, including features like linting, debugging, and code snippets.
- Flask: The Flask web framework is utilized to create the web application, define routes, and handle HTTP requests and responses.
- Scikit-Learn (sklearn): Scikit-Learn is a machine learning library for Python. In this code, the GaussianMixture class from sklearn.mixture is used for Gaussian Mixture Models (GMMs).
- Watson Developer Cloud: The SpeechToTextV1 class from the watson_developer_cloud package is used to integrate IBM Watson Speech to Text service for converting speech to text.
- HTML, CSS and JavaScript: HTML structures the content of web pages, CSS styles the appearance, and JavaScript adds interactivity and dynamic behavior.

By integrating HTML, CSS, JavaScript, Flask backend, and the machine learning capabilities of Scikit-Learn, it is possible to create a voice biometrics system. This system can capture audio samples from users, process them to extract unique voice features, and use these features to authenticate and identify the users.

5.3 Voice Biometrics Setup and Configuration

- Obtaining and Storing Watson API Credentials

To use IBM Watson's Speech to Text service, first obtain API credentials from the IBM Cloud dashboard. Store the API key and URL securely in a configuration file to ensure they can be accessed by the application without exposing sensitive information. This setup allows the application to interact with Watson's services for accurate voice recognition.

- Configuring the System for Audio Processing

Ensure the system is configured to handle audio input by installing necessary libraries like `speech_recognition` and `python_speech_features`. These tools will enable the system to preprocess audio, adjust for noise, and extract features like Mel-Frequency Cepstral Coefficients (MFCC), which are crucial for effective voice recognition.

- Setting Up Voice Enrollment and Authentication

Set up the enrollment process where users register their voice by submitting a sample, which is stored for future reference. For authentication, the system compares the user's voice sample with the stored models using Gaussian Mixture Models (GMM). This process ensures accurate verification based on the user's unique vocal characteristics.

- Customizing System Settings

Customize system settings to meet specific requirements, such as adjusting the duration for background noise adjustment and fine-tuning the voice recognition parameters. These configurations can enhance the system's performance and accuracy, catering to different user environments and needs.

- File System Configuration

Organize the file system to store user-specific voice data and models efficiently. Create separate directories for each user to keep their data organized, and ensure that the system can recreate these directories as needed for a clean slate during new enrollment.

- Frontend and Backend Integration

Integrate the frontend with the backend using Flask to create a seamless user experience. The frontend handles user interactions and input, while the backend processes these inputs, performs voice recognition, and manages data storage. This integration ensures that the system operates smoothly and provides real-time feedback to users.

5.4 Basic Functionality Implementation and Integration of Core Features

- User Enrollment

User enrollment is the initial step where users register their voice with the system by providing a username and password. A directory specific to the user is created to store their voice data, and they are prompted to record their voice. This voice sample is stored for future authentication.

- User Authentication

User authentication involves verifying a user's identity using their voice. The user submits a voice sample, which the system compares against pre-stored voice models using Gaussian Mixture Models (GMM). Based on this comparison, the system verifies the user's identity.

- Audio Upload and Processing

This step handles audio data and extracts necessary features for voice recognition. The system ensures the audio file is valid and adjusts for background noise. It then extracts features like Mel-Frequency Cepstral Coefficients (MFCC) for accurate voice model comparison.

- Feedback Mechanism

Providing immediate feedback is crucial for a smooth user experience. The system offers success or error messages for each action, guiding users on how to correct errors, such as re-recording their voice or adjusting microphone settings.

- Help

To assist users, the system includes support contact information. These resources help users understand the system and resolve any issues they may encounter.

- Speech-to-Text Mechanism

The speech-to-text mechanism in Voice Biometrics uses IBM Watson's Speech to Text service to convert spoken language into text. When a user recites a phrase, the audio signal is captured and processed by Watson's advanced machine learning algorithms, which analyse the acoustic signal and convert it into text based on learned patterns and linguistic rules. This transcribed text is then used for further processing, such as authentication or verification within the application.

- Text-Dependent Voice Biometric System

A text-dependent voice biometric system enhances security by generating random phrases for users to recite during both enrollment and authentication. The system extracts MFCC and delta features from

these samples to train a Gaussian Mixture Model (GMM), creating a robust voice print. During authentication, a new random phrase is generated, and the user's recited sample is compared against the stored voice print using the GMM, ensuring high security and accuracy by varying the phrases and consistently analyzing speaker-specific features.

5.5 Feature Representation of the Speech Signal

Voice Biometrics enhances speaker recognition by combining Mel-Frequency Cepstral Coefficients (MFCC) and their corresponding delta features into a single feature vector. This combined vector incorporates both the static and dynamic properties of the speech signal, which improves the system's ability to recognize and differentiate between speakers.

Steps to Create Combined Features

- Extract MFCC Features:

MFCC Extraction: This process involves converting the audio signal into a set of coefficients that represent the short-term power spectrum of the sound. These coefficients are derived using transformations like the Fourier Transform and Mel-scale filtering.

- Normalize MFCC Features:

Normalization: The extracted MFCC features are scaled to have zero mean and unit variance. This step reduces the impact of varying loudness and ensures the features are more robust against differences in recording conditions.

- Calculate Delta Features:

Delta Calculation: Delta features are computed as the first-order differences of the MFCC features. They capture the rate of change of the MFCC coefficients, providing information about the dynamics and temporal evolution of the speech signal.

- Combine MFCC and Delta Features:

Horizontal Stacking: The MFCC features and their corresponding delta features are combined into a single feature vector. Each frame in the audio signal is thus represented by both its static spectral characteristics (MFCC) and its dynamic properties (delta features).

Static Features (MFCC) capture the spectral properties of the speech signal. They are useful for identifying the phonetic content and inherent characteristics of the speaker's voice.

Dynamic Features (Delta) provide information about how the spectral properties change over time, capturing the temporal dynamics of the speech signal.

5.6 Error Handling

In the realm of voice biometrics, effective error handling is pivotal to ensure the system's resilience and reliability. Throughout the system's operation, numerous potential errors can arise, spanning data pre-processing, feature extraction, model training, and authentication phases. Robust error handling mechanisms must be in place to detect and manage these issues adeptly. For instance, during feature extraction, errors might stem from factors like incompatible audio formats, data omissions, or unexpected runtime behaviours. In response, employing appropriate exception handling techniques, such as try-except constructs, is imperative to capture and address exceptions promptly, thereby preventing system crashes and ensuring user experience continuity. Furthermore, employing comprehensive logging mechanisms facilitates the recording of detailed error information, aiding in post hoc troubleshooting and issue resolution. Additionally, the implementation of input validation routines and sanity checks serves to uphold data integrity and validity at each processing stage, thereby pre-emptively mitigating potential errors before they escalate. By adopting such robust error handling strategies, the voice biometrics system can maintain its resilience and dependability, delivering consistent performance across various operational scenarios.

5.7 Deployment and Testing

The system was deployed on a local development server to facilitate testing and development phases. Extensive testing methodologies were employed to validate the system's functionality, security, and performance. This included functional testing to verify accurate enrollment and authentication processes, as well as stress testing to assess the system's resilience under varying loads. Additionally, non-functional testing was conducted to evaluate aspects such as system responsiveness, data privacy, and error handling. User interaction with the system was streamlined to facilitate seamless enrollment and authentication, ensuring a frictionless experience for users while upholding stringent security standards.

CHAPTER 6

TESTING & MAINTANENCE

This report aims to delineate the testing and maintenance procedures implemented for a voice biometrics system geared towards enhancing security and user authentication. The voice biometrics system represents an innovative solution designed to verify individual identities based on unique vocal characteristics, thereby bolstering data security and user privacy. By elucidating the key aspects of testing, including functional testing, performance testing, and user acceptance testing, this report underscores the rigorous measures undertaken to validate the system's functionality, reliability, and user satisfaction. Additionally, the report underscores the significance of ongoing maintenance to uphold the system's optimal performance and efficacy.

6.1 Testing Process

Functional testing plays a crucial role in ensuring that the voice biometrics system meets its specified functional requirements. The following steps were executed as part of the functional testing process:

- **Testing Command Execution:** Various commands and functionalities like enrolment, authentication, and system configuration of the voice biometrics system were tested to ensure seamless execution.
- **Verification of Accuracy:** The accuracy of the system's responses to user inputs was thoroughly verified.
- **Handling of Scenarios:** The system's capability to handle diverse scenarios, including unexpected inputs or errors, was rigorously tested.
- **Uptime Verification:** Continuous monitoring was performed to ensure the system maintains optimal uptime.

6.2 Performance Testing

Performance testing for the voice biometrics system focuses on evaluating its responsiveness and stability under varying workload conditions. The following measures were undertaken during performance testing:

- **Response Time Evaluation:** The system's response time was assessed by simulating different levels of audio processing workload and monitoring the time taken to process each audio sample.
- **Concurrency Handling:** The system's ability to handle multiple concurrent authentication requests without experiencing significant performance degradation was evaluated. This involved assessing how well the system maintains responsiveness under high demand.
- **Resource Utilization Monitoring:** The utilization of system resources, such as CPU and memory, was monitored during performance testing to identify any potential resource constraints or scalability issues. This helps ensure that the system can efficiently handle increasing workload demands.
- **Stress Testing:** Stress testing was conducted to determine the system's performance limits by subjecting it to extreme usage scenarios, such as a sudden surge in authentication requests or prolonged periods of high workload. This helps assess the system's robustness and identify potential failure points under stress.

6.3 User Acceptance Testing

User acceptance testing for the voice biometrics system aims to gather feedback from real users to ensure that the system meets their expectations and effectively fulfils its intended purpose. The following actions were undertaken during user acceptance testing:

- **Engagement of Diverse User Group:** A diverse group of users, including individuals with varying demographic characteristics and voice profiles, was engaged to interact with the voice biometrics system. This ensured that the system's performance and usability were evaluated across different user segments.
- **Feedback Collection:** Feedback from users was collected through various channels, such as surveys, interviews, and direct interaction with the system. Users were encouraged to provide feedback on the system's usability, accuracy, and overall user experience.
- **Alignment with User Needs:** The voice biometrics system was validated to ensure that it aligns with users' needs and expectations in terms of providing secure and convenient authentication. Feedback from users was used to validate that the system effectively meets users' requirements and adds value to their authentication experience.

6.4 Test Planning

Before commencing the testing process, a comprehensive test plan was formulated to define the goals, scope, and necessary resources for testing the voice biometrics system. The test plan covered multiple facets, including functional testing, performance testing, security testing, and user acceptance testing. This structured approach ensured thorough evaluation of the system's functionality, performance, security measures, and user satisfaction.

6.5 Security Testing

Security testing played a critical role in ensuring the robustness of the voice biometrics system. This phase of testing focused on identifying and mitigating potential security vulnerabilities that could pose risks to the confidentiality, integrity, or availability of user data and system resources. Measures such as input validation, encryption, access controls, and authentication mechanisms were rigorously tested to uphold industry-standard security protocols and safeguard sensitive user information.

Testing Results

During the testing phase, several issues were identified that affected the optimal performance and reliability of the voice biometrics system:

a. Functional Issues:

Voice Sample Recording: Some users reported difficulties in recording their voice samples due to issues with the integration between the frontend and backend systems. Specifically, the voice samples were not being successfully passed from the frontend interface to the backend processing module. This issue hindered the enrolment process and affected the overall functionality of the system.

b. Performance Issues:

Latency in Voice Processing: Under high loads or during peak usage periods, the system experienced delays in processing voice samples, resulting in slower authentication and identification processes. Some other issues that were encountered were that:

1. **User Authentication Errors:** Some users reported difficulties in completing the sign-in process due to authentication errors or inconsistencies in user verification. This led to inconvenience and frustration among users attempting to access the system.

2. When accessing YouTube videos directly there occurs a considerable amount of delay. To solve this, we have implemented the response in such a case to be the YouTube playlist instead of directly accessing the video itself.

Testing and maintenance are crucial aspects of ensuring the effectiveness and reliability of the voice biometrics system. Through comprehensive testing, including functional, performance, and security testing, any potential issues or vulnerabilities can be identified and addressed before the system is deployed for actual use. Additionally, ongoing maintenance activities, such as monitoring system performance, updating security protocols, and addressing user feedback, are essential for ensuring the system's continued effectiveness and user satisfaction. With a robust testing and maintenance strategy in place, the voice biometrics system can provide secure and convenient authentication for users while maintaining high levels of reliability and performance.

Maintenance

Maintenance is paramount to ensuring the sustained performance and effectiveness of the voice biometric system. Regular monitoring and analysis of system performance metrics, including accuracy rates and user feedback, are essential to identify areas for improvement. Routine updates should be conducted to incorporate new features and enhancements, adapting the system to changes in user behaviour and technological advancements. Prompt resolution of reported bugs and issues is crucial to maintaining system reliability and user satisfaction. Continuous improvement efforts should focus on refining algorithms, expanding feature sets, and optimizing performance to meet evolving user needs. Compliance with regulatory requirements and industry standards must be maintained, with periodic reviews and updates to security protocols and encryption mechanisms. Additionally, staying abreast of changes in hardware and software components ensures compatibility and seamless operation of the system. By implementing these maintenance activities, the voice biometric system can continue to deliver reliable and accurate authentication services to users.

1. Bug Fixes:

During the testing phase, various bugs and issues were uncovered within the voice biometric system. These ranged from inconsistencies in voice sample recording to challenges in backend integration for processing collected data. The development team swiftly responded to these issues by implementing bug fixes and optimizations to ensure seamless functionality. Additionally, efforts were made to establish robust monitoring and feedback mechanisms to continually track the system's performance and gather user input. This proactive approach enabled the team to

promptly address any emerging issues and maintain the system's reliability and effectiveness over time. Regular monitoring and feedback collection remain essential to detect and resolve potential issues and uphold the system's integrity and performance.

2. Performance Optimization

Following the results of performance testing, adjustments were implemented to elevate the responsiveness and scalability of the voice biometric system. To sustain peak performance levels, ongoing monitoring and optimization endeavours were initiated. These optimizations encompassed refinements in code structure, adoption of efficient caching techniques, and fine-tuning of database operations. Continuous scrutiny and profiling of the system's performance were conducted to pinpoint any potential bottlenecks and guarantee the efficient utilization of resources.

3. Error Monitoring and Logging

Following the outcomes of performance testing, enhancements were integrated to enhance the responsiveness and scalability of the voice authentication system. To uphold optimal performance, continuous monitoring and optimization efforts were initiated. These optimizations included refining the code structure, implementing efficient caching mechanisms, and fine-tuning database operations. A continuous assessment of the system's performance was conducted to identify and address any potential bottlenecks, ensuring the efficient utilization of resources.

CHAPTER 7

CONCLUSION

The development and implementation of a voice biometrics authentication system signifies a transformative leap in the domain of online security and user authentication. Through meticulous research, strategic planning, and seamless execution, we have embarked on a journey to revolutionize how users access and safeguard their online accounts, transactions, and sensitive information.

Our exploration commenced with an exhaustive literature review, which illuminated the evolving landscape of biometric security and the increasing adoption of voice recognition technology. We recognized the growing reliance on biometric authentication methods as a more secure and user-friendly alternative to traditional credentials. Voice biometrics, in particular, emerged as a promising avenue due to its inherent uniqueness, ease of use, and potential for widespread adoption.

With insights gleaned from our literature review, we meticulously outlined the software requirements essential to bring our voice biometrics authentication system to fruition. From selecting appropriate programming languages and frameworks to identifying voice recognition APIs and machine learning algorithms, each decision was made with the aim of creating a robust and reliable authentication solution.

In addition to its broader implications for online security, our voice biometrics authentication system holds particular significance for individuals with disabilities, including those who are blind or visually impaired. By leveraging voice recognition technology as the primary means of authentication, we strive to create a more inclusive and accessible online environment for all users, regardless of their physical abilities.

Furthermore, in today's fast-paced world, where individuals lead increasingly busy lives, managing numerous passwords for various online accounts can become burdensome and time-consuming. Our system offers a seamless and password less authentication experience, alleviating the burden of password management and streamlining the authentication process for users with busy lifestyles.

With a clear understanding of our project's objectives and requirements, we explored the diverse applications and advantages of our voice biometrics authentication system. From securing user logins and transactions to enabling password less access and protecting sensitive information, the potential applications are vast and far-reaching. The system offers enhanced security, user convenience, and reduced risk of identity theft by eliminating the need for traditional passwords.

However, it is essential to acknowledge the potential challenges and limitations associated with our project. Despite advancements in voice recognition technology, inherent constraints such as susceptibility to noise interference and the need for continuous updates and maintenance must be addressed diligently to ensure accuracy and reliability. Additionally, concerns regarding privacy and data security must be addressed to instil trust and confidence among users.

In conclusion, our voice biometrics authentication system represents a significant advancement in online security and user authentication. By harnessing the power of voice recognition technology and integrating it into our website, we aim to redefine the standards of online authentication and usher in a new era of secure, user-centric digital experiences. As we continue to refine and optimize our system, we remain committed to delivering unparalleled security, convenience, and peace of mind to users worldwide.

REFERENCES

- [1] Scheffer, N., Ferrer, L., Lawson, A., Lei, Y. and McLaren, M., 2013, November. Recent developments in voice biometrics: Robustness and high accuracy. In 2013 IEEE international conference on technologies for homeland security (HST) (pp. 447-452). IEEE.
- [2] Kuznetsov, A., Oleshko, I., Chernov, K., Bagmut, M. and Smirnova, T., 2019, April. Biometric authentication using convolutional neural networks. In *Conference on Mathematical Control Theory* (pp. 85-98). Cham: Springer International Publishing.
- [3] Dhillon, I., Rupp, J., Vankina, A. and Slater, R., 2021. Real-Time Voice Biometric Speaker Verification. *SMU Data Science Review*, 5(2), p.11.
- [4] Aronowitz, H., Hoory, R., Pelecanos, J. and Nahamoo, D., 2011. New developments in voice biometrics for user authentication. In Twelfth Annual Conference of the International Speech Communication Association.
- [5] González-Rodríguez, J., Toledano, D.T. and Ortega-García, J., 2008. Voice biometrics. In *Handbook of biometrics* (pp. 151-170). Boston, MA: Springer US.
- [6] Markowitz, J.A., 2000. Voice biometrics. *Communications of the ACM*, 43(9), pp.66-73.
- [7] Aronowitz, H., 2012, June. Voice biometrics for user authentication. In *Afeka-AVIOS Speech Processing Conference 2012* (Vol. 29).
- [8] Gupta, P., Singh, S., Prajapati, G.P. and Patil, H.A., 2022. Voice privacy in biometrics. In *Biomedical Signal and Image Processing with Artificial Intelligence* (pp. 1-29). Cham: Springer International Publishing.
- [9] Shah, H.N.M., Ab Rashid, M.Z., Abdollah, M.F., Kamarudin, M.N., Chow, K.L. and Kamis, Z., 2014. Biometric voice recognition in security system. *Indian journal of Science and Technology*, pp.104-112.
- [10] Sadkhan, S.B., Al-Shukur, B.K. and Mattar, A.K., 2017, March. Biometric voice authentication auto-evaluation system. In 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT) (pp. 174-179). IEEE.
- [11] AL-Shakarchy, N.D., Obayes, H.K. and Abdullah, Z.N., 2023. Person identification based on voice biometric using deep neural network. *International Journal of Information Technology*, 15(2), pp.789-795.
- [12] Ibrahim, W., Candra, H. and Isyanto, H., 2022. Voice recognition security reliability analysis using deep learning convolutional neural network algorithm. *Journal of Electrical Technology UMY*, 6(1), pp.1-11.

APPENDIX

9.1 PROGRAM

Backend

Importing Required Libraries and Modules

```
import pickle
import datetime
import os
import shutil
import matplotlib.pyplot as plt
import numpy
import scipy.cluster
import scipy.io.wavfile
import speech_recognition
from fuzzywuzzy import fuzz
from python_speech_features import mfcc
from random_words import RandomWords
from sklearn import preprocessing
from sklearn.mixture import GaussianMixture
from watson_developer_cloud import SpeechToTextV1
import config
from flask import Flask, render_template, request, jsonify, url_for, redirect,
abort, session, json
```

Flask Application Initialization

```
app = Flask(__name__)
PORT = 8080
# Global Variables
random_words = []
random_string = ""
username = ""
user_directory = "Users/"
filename = ""
filename_wav = ""
```

IBM Watson Speech to Text Initialization

```
speech_to_text = SpeechToTextV1(
    iam_apikey=config.APIKEY,
```

```
url=config.URL
)
```

Enroll Route Definition

```
@app.route('/enroll', methods=["GET", "POST"])
def enroll():
    global username
    global user_directory

    if request.method == 'POST':
        data = request.get_json()
        username = data['username']
        password = data['password']
        repassword = data['repassword']
        user_directory = os.path.join("Users", username)

        if not os.path.exists(user_directory):
            os.makedirs(user_directory)
        else:
            shutil.rmtree(user_directory, ignore_errors=False, onerror=None)
            os.makedirs(user_directory)

        return redirect(url_for('voice'))
    else:
        return render_template('enroll.html')
```

Authentication Route Definition

```
@app.route('/auth', methods=['POST', 'GET'])
def auth():
    global username
    global user_directory
    global filename

    user_exist = False

    if request.method == 'POST':
        data = request.get_json()
        user_directory = 'Models/'
        username = data['username']
        password = data['password']

        for file in os.listdir(user_directory):
            filename = os.fsdecode(file)
            if filename.startswith(username):
                user_exist = True
```

```

        break

    if user_exist:
        return "User exist"
    else:
        return "Doesn't exist"
else:
    return render_template('auth.html')

```

Voice Activity Detection Route Definition

```

@app.route('/vad', methods=['GET', 'POST'])
def vad():
    if request.method == 'POST':
        global random_words

        f = open('./static/audio/background_noise.wav', 'wb')
        f.write(request.data)
        f.close()

        background_noise = speech_recognition.AudioFile(
            './static/audio/background_noise.wav')
        with background_noise as source:
            speech_recognition.Recognizer().adjust_for_ambient_noise(source,
duration=5)

        random_words = RandomWords().random_words(count=5)
        return " ".join(random_words)
    else:
        background_noise = speech_recognition.AudioFile(
            './static/audio/background_noise.wav')
        with background_noise as source:
            speech_recognition.Recognizer().adjust_for_ambient_noise(source,
duration=5)

        random_words = RandomWords().random_words(count=5)
        return " ".join(random_words)

```

Voice Page Route Definition

```

@app.route('/voice', methods=['GET', 'POST'])
def voice():
    global user_directory
    global filename_wav
    global random_words
    global username

```



```

if request.method == 'POST':
    if 'file' not in request.files:
        return "fail"

    file = request.files['file']

    if file.filename == '':
        return "fail"

    if file:
        try:
            user_dir_path = os.path.join('Users', username)
            os.makedirs(user_dir_path, exist_ok=True)
            filename_wav = os.path.join(user_dir_path, f"{'-'.join(random_words)}.wav")
            file.save(filename_wav)
            file_size = os.path.getsize(filename_wav)

            if file_size == 0:
                return "fail"

            with open(filename_wav, 'rb') as audio_file:
                header = audio_file.read(44)
                if header[:4] != b'RIFF' or header[8:12] != b'WAVE':
                    return "fail"

        except Exception as save_error:
            return "fail"

        try:
            with open(filename_wav, 'rb') as audio_file:
                result = speech_to_text.recognize(audio=audio_file,
content_type='audio/wav').get_result()
                recognised_words =
result['results'][0]['alternatives'][0]['transcript']
                random_words_str = " ".join(random_words).strip().lower()

            if fuzz.ratio(random_words_str, recognised_words) < 5:
                os.remove(filename_wav)
                return "fail"
            else:
                return redirect(url_for('biometrics'))
        except Exception as e:
            return "fail"

    random_words = RandomWords().random_words(count=2)
    return render_template('voice.html', random_words=" ".join(random_words))

```

Biometrics Route Definition

```
@app.route('/biometrics', methods=['GET', 'POST'])
def biometrics():
    global user_directory

    if request.method == 'POST':
        pass
    else:
        directory = os.fsencode(user_directory)
        features = numpy.asarray(())

        for file in os.listdir(directory):
            filename_wav = os.fsdecode(file)
            if filename_wav.endswith(".wav"):
                (rate, signal) = scipy.io.wavfile.read(os.path.join(user_directory,
filename_wav))
                extracted_features = extract_features(rate, signal)

                if features.size == 0:
                    features = extracted_features
                else:
                    features = numpy.vstack((features, extracted_features))

        gmm = GaussianMixture(n_components=16, max_iter=200, covariance_type='diag',
n_init=3)
        gmm.fit(features)

        pickle.dump(gmm, open("Models/" + str(username) + ".gmm", "wb"),
protocol=None)
        features = numpy.asarray(())

        return "User has been successfully enrolled ...!!"
```

Verify Route Definition

```
@app.route("/verify", methods=['GET'])
def verify():
    global username
    global filename
    global user_directory
    global filename_wav

    (rate, signal) = scipy.io.wavfile.read(filename_wav)
    extracted_features = extract_features(rate, signal)

    gmm_models = [os.path.join(user_directory, user) for user in
os.listdir(user_directory) if user.endswith('.gmm')]
```

```

models = [pickle.load(open(user, 'rb')) for user in gmm_models]
user_list = [user.split("/")[-1].split(".gmm")[0] for user in gmm_models]
log_likelihood = numpy.zeros(len(models))

for i in range(len(models)):
    gmm = models[i]
    scores = numpy.array(gmm.score(extracted_features))
    log_likelihood[i] = scores.sum()

identified_user = numpy.argmax(log_likelihood)

auth_message = ""
if user_list[identified_user] == username:
    auth_message = "success"
else:
    auth_message = "fail"

return auth_message

```

Helper Functions for Feature Extraction

```

def calculate_delta(array):
    Helper Functions for Feature Extraction

    rows, cols = array.shape
    deltas = numpy.zeros((rows, 20))
    N = 2
    for i in range(rows):
        index = []
        j = 1
        while j <= N:
            if i-j < 0:
                first = 0
            else:
                first = i-j
            if i+j > rows - 1:
                second = rows - 1
            else:
                second = i+j
            index.append((second, first))
            j += 1
        deltas[i] = (array[index[0][0]]-array[index[0][1]] + (2 *
(array[index[1][0]]-array[index[1][1]]))) / 10
    return deltas

def extract_features(rate, signal):
    mfcc_feat = mfcc(signal, rate, winlen=0.020, preemph=0.95, numcep=20, nfft=1024,
ceplifter=15, highfreq=6000, nfilt=55, appendEnergy=False)

```

```

mfcc_feat = preprocessing.scale(mfcc_feat)
delta_feat = calculate_delta(mfcc_feat)
combined_features = numpy.hstack((mfcc_feat, delta_feat))
return combined_features

```

Main Program Execution

```

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=PORT, debug=True)

```

Frontend

main.js

```

document.querySelector('#enrollOptionButton').addEventListener('click', function () {
{
    console.log("The enroll button works!");
    window.location.href = '/enroll';
});

document.querySelector('#authOptionButton').addEventListener('click', function () {
{
    console.log("The auth button works!");
    window.location.href = '/auth';
});
});

```

enroll.js

```

var usernamePointer = document.querySelector('#usernamePointer');
var passwordPointer = document.querySelector('#passwordPointer');
var repasswordPointer = document.querySelector('#repasswordPointer');

usernamePointer.style.display = 'none';
passwordPointer.style.display = 'none';
repasswordPointer.style.display = 'none';

window.onload = function (event) {

    // Simulate login click when user presses Enter/Return key
    document.querySelector('#mainForm').addEventListener('keydown', function (event) {
        if (event.keyCode === 13) {
            document.querySelector('#nextButton').click();
        }
    });
});

```

```

function isValidUsername(username) {
    var pattern = /^[a-zA-Z0-9.\-_@$*!]{3,30}$/;
    return pattern.test(username);
}

function validateCredentialsFormat(loginCreds) {
    var passedCheck = true;
    var passwordField = document.querySelector('#passwordField');
    var userField = document.querySelector('#userField');
    if (loginCreds.password.length < 6) {

        passwordPointer.style.display = '';

        passwordField.classList.add('error');
        passedCheck = false;
    } else {
        passwordField.classList.remove('error');
        passwordPointer.style.display = 'none';
    }
    if (loginCreds.repassword != loginCreds.password) {

        repasswordPointer.style.display = '';

        repasswordField.classList.add('error');
        passedCheck = false;
    }
    else {
        repasswordPointer.style.display = 'none';
        repasswordField.classList.remove('error');
    }

    if (!isValidUsername(loginCreds.username)) {
        usernamePointer.style.display = '';

        userField.classList.add('error');
        passedCheck = false;
    }
    else {
        usernamePointer.style.display = 'none';
        userField.classList.remove('error');
    }

    // console.log("passCheck : ", passedCheck)
    return passedCheck;
}

```

```

document.querySelector('#nextButton').addEventListener('click', function () {
    var us = document.querySelector('input[name="username"]').value;
    var pass = document.querySelector('input[name="password"]').value;
    var repass = document.querySelector('input[name="repassword"]').value;

    var loginCreds = {
        username: us,
        password: pass,
        repassword: repass
    };

    if (validateCredentialsFormat(loginCreds)) {
        console.log("Valid Credentials have been entered ...\n Proceeding to sending
data");

        var xhr = new XMLHttpRequest();

        xhr.onreadystatechange = function () {
            if (xhr.readyState == XMLHttpRequest.DONE && xhr.status == 200) {
                console.log("(xhr.readyState == XMLHttpRequest.DONE && xhr.status ==
200)")
                window.location.href = '/voice';
            }
        }

        xhr.open("POST", "/enroll", true);
        xhr.setRequestHeader("Content-type", "application/json");

        xhr.send(JSON.stringify(loginCreds));

        console.log("Your http message has been sent.");
    }
    else {
        console.log("Invalid credentials have been entered ...\nPlease try again
...");
    }

    console.log("username : ", us)
    console.log("password : ", pass)
    console.log("password : ", repass)

    console.log("You clicked the login Next button");
});
};

```

auth.js

```

var usernamePointer = document.querySelector('#usernamePointer');
var passwordPointer = document.querySelector('#passwordPointer');

usernamePointer.style.display = 'none';
passwordPointer.style.display = 'none';

window.onload = function (event) {

    // Simulate login click when user presses Enter/Return key
    document.querySelector('#mainForm').addEventListener('keydown', function (event) {
        if (event.keyCode === 13) {
            document.querySelector('#nextButton').click();
        }
    });

    document.querySelector('#nextButton').addEventListener('click', function () {
        var us = document.querySelector('input[name="username"]').value;
        var pass = document.querySelector('input[name="password"]').value;

        var loginCreds = {
            username: us,
            password: pass
        };

        var xhr = new XMLHttpRequest();

        xhr.onreadystatechange = function () {
            if (xhr.readyState === XMLHttpRequest.DONE && xhr.status === 200) {
                console.log("Response : ", xhr.response);

                if (xhr.response === "Doesn't exist") {
                    usernamePointer.style.display = '';
                }
                else{
                    window.location.href = '/voice';
                }
            }
        }

        // xhr.open("GET", "/voice", true);
        // xhr.setRequestHeader("Content-type", "application/json");

        // xhr.send();

        // console.log("The enroll button works!");

        xhr.open("POST", "/auth", true);
        xhr.setRequestHeader("Content-type", "application/json");
    });
}

```

```

    xhr.send(JSON.stringify(loginCreds));

    console.log("Your http message has been sent.");

    console.log("You clicked the login Next button");
  });
};
//
// function hideElement(elSelector) {
//   document.querySelector(elSelector).style.display = 'none';
// }
//
// function showElement(elSelector) {
//   document.querySelector(elSelector).style.display = '';
// }

```

voice.js

```

import WaveSurfer from './wavesurfer.esm.js';
import RecordPlugin from './wavesurfer.record.esm.js';

let wavesurfer, record;
let scrollingWaveform = false;

const initWaveSurfer = () => {
  console.log('Initializing WaveSurfer instance...');
  if (wavesurfer) {
    wavesurfer.destroy();
  }
  wavesurfer = WaveSurfer.create({
    container: '#waveform',
    waveColor: '#01BAB6',
    progressColor: '#019B95',
  });
  console.log('WaveSurfer instance created.');

  record = wavesurfer.registerPlugin(RecordPlugin.create({ scrollingWaveform,
renderRecordedAudio: false }));
  console.log('Record plugin registered.');

  record.on('record-end', handleRecordEnd);
  record.on('record-progress', updateProgress);

  resetControls();
};

```



```

const handleRecordEnd = (blob) => {
  console.log('Recording ended.');
```

```

  const container = document.querySelector('#recordings');
  const recordedUrl = URL.createObjectURL(blob);

  const recordedWaveSurfer = WaveSurfer.create({
    container,
    waveColor: 'rgb(200, 100, 0)',
    progressColor: 'rgb(100, 50, 0)',
    url: recordedUrl,
  });

  const playButton = createButton('Play', () => recordedWaveSurfer.playPause());
  recordedWaveSurfer.on('pause', () => (playButton.textContent = 'Play'));
  recordedWaveSurfer.on('play', () => (playButton.textContent = 'Pause'));

  const downloadLink = createDownloadLink(recordedUrl, blob.type);
  container.append(playButton, downloadLink);

  // Send the recorded blob to the server
  uploadRecording(blob);
};

const handleFileUpload = (event) => {
  const file = event.target.files[0];
  if (file && file.type === 'audio/wav') {
    const fileUrl = URL.createObjectURL(file);
    const container = document.querySelector('#recordings');

    const uploadedWaveSurfer = WaveSurfer.create({
      container,
      waveColor: 'rgb(200, 100, 0)',
      progressColor: 'rgb(100, 50, 0)',
      url: fileUrl,
    });

    const playButton = createButton('Play', () => uploadedWaveSurfer.playPause());
    uploadedWaveSurfer.on('pause', () => (playButton.textContent = 'Play'));
    uploadedWaveSurfer.on('play', () => (playButton.textContent = 'Pause'));

    const downloadLink = createDownloadLink(fileUrl, file.type);
    container.append(playButton, downloadLink);

    // Send the uploaded file to the server
    uploadRecording(file);
  } else {
    alert('Please upload a valid WAV file.');
```

```

    }
  };

const createButton = (text, onClick) => {
  const button = document.createElement('button');
  button.textContent = text;
  button.onclick = onClick;
  return button;
};

const createDownloadLink = (url, type) => {
  const link = document.createElement('a');
  link.href = url;
  link.download = `recording.${type.split(';')[0].split('/')[1]} || 'webm'`;
  link.textContent = 'Download recording';
  return link;
};

const updateProgress = (time) => {
  const formattedTime = [
    Math.floor((time % 3600000) / 60000),
    Math.floor((time % 60000) / 1000),
  ].map(v => (v < 10 ? '0' + v : v)).join(':');
  document.querySelector('#progress').textContent = formattedTime;
};

const resetControls = () => {
  const pauseButton = document.querySelector('#pause');
  const recButton = document.querySelector('#record');
  pauseButton.style.display = 'none';
  recButton.textContent = 'Record';
};

const setupMicOptions = () => {
  const micSelect = document.querySelector('#mic-select');
  RecordPlugin.getAvailableAudioDevices().then(devices => {
    console.log('Available audio devices:', devices);
    devices.forEach(device => {
      const option = document.createElement('option');
      option.value = device.deviceId;
      option.text = device.label || device.deviceId;
      micSelect.appendChild(option);
    });
  }).catch(err => {
    console.error('Error fetching audio devices:', err);
  });
};

```

```

const handleRecording = () => {
  const recButton = document.querySelector('#record');
  const pauseButton = document.querySelector('#pause');
  const micSelect = document.querySelector('#mic-select');

  recButton.onclick = () => {
    if (record.isRecording() || record.isPaused()) {
      record.stopRecording();
      resetControls();
      console.log('Recording stopped.');
```

return;

```
    }

    recButton.disabled = true;
    const deviceId = micSelect.value;
    record.startRecording({ deviceId }).then(() => {
      recButton.textContent = 'Stop';
      recButton.disabled = false;
      pauseButton.style.display = 'inline';
      console.log('Recording started.');
```

}).catch(err => {

```
      console.error('Error starting recording:', err);
      recButton.disabled = false;
    });
  });

  pauseButton.onclick = () => {
    if (record.isPaused()) {
      record.resumeRecording();
      pauseButton.textContent = 'Pause';
      console.log('Recording resumed.');
```

} else {

```
      record.pauseRecording();
      pauseButton.textContent = 'Resume';
      console.log('Recording paused.');
```

}

```
    }
  });
};

document.querySelector('input[type="checkbox"]').onclick = (e) => {
  scrollingWaveform = e.target.checked;
  console.log('Scrolling waveform:', scrollingWaveform);
  initWaveSurfer();
};

const uploadRecording = (blob) => {
  const url = 'http://127.0.0.1:8080/voice';
  const formData = new FormData();

```

```

formData.append('file', blob, 'recording.wav');

fetch(url, {
  method: 'POST',
  body: formData
}).then(response => response.text())
  .then(data => {
    if (data === "fail") {
      alert('Error: The words you have spoken aren\'t entirely correct. Please try
again.');
```

```

    } else {
      alert('Transcription successful.');
```

```

    }
  }).catch(error => {
    alert('An error occurred while uploading the file.');
```

```

    console.error('Upload error:', error);
  });
});

document.querySelector('#file-input').addEventListener('change', handleFileUpload);

document.addEventListener('DOMContentLoaded', () => {
  initWaveSurfer();
  setupMicOptions();
  handleRecording();
});

```