INTERNSHIP PROJECT REPORT

VOICE ASSISTANT-BASED ATTENDANCE MONITORING SYSTEM

PROJECT - II

Submitted by

ANUPAM KANOONGO (TEAM LEAD)
SUMAN M
SHAHANA NARGEES

To

INFOTACT SOLUTIONS

(Electronics City Phase 1, Electronic City, Bengaluru, Karnataka 560100)

FEBRUARY 2025

ABSTRACT

VOICE-BASED ATTENDANCE SYSTEM USING AI

Traditional attendance systems often rely on manual methods or biometric verification, which can be time-consuming and prone to errors. This project proposes an AI-powered Voice-Based Attendance System that automates attendance marking by recognizing individuals through their voice. The system utilizes Librosa for feature extraction, SpeechRecognition for capturing audio, and Scikit-learn/TensorFlow for voice classification. The recognized voice is matched with stored profiles, and attendance is logged into a MongoDB/MySQL database in real-time. A Flask/Django web interface provides accessibility and management capabilities. This approach enhances efficiency, reduces human intervention, and ensures a seamless attendance process. Future improvements include deep learning-based accuracy enhancement, mobile integration, and multi-factor authentication for security.

TABLE OF CONTENTS

1.INTRODUCTION

1.1 PROBLEM STATEMENT CONTEXT

Traditional attendance methods like manual roll calls and biometric systems are time-consuming, error-prone, and require physical contact. The Voice Assistant-based Attendance Monitoring System automates attendance using speech recognition, allowing users to mark attendance by speaking their names. It ensures real-time feedback, secure database storage, and voice-based reporting, reducing errors and administrative workload. With error handling and retry mechanisms, the system enhances accuracy, making attendance tracking efficient, contactless, and scalable for classrooms and workplaces.

1.2 OBJECTIVES

- Develop a speech recognition-based system to mark attendance using voice input, eliminating manual methods.
- Reduce errors and administrative workload by ensuring real-time recognition and automatic data logging.
- Store attendance records in an SQL/NoSQL database for easy retrieval, reporting, and analysis.
- Provide real-time feedback, error handling, and voice-based reporting for a seamless and interactive attendance system.

2. TECHNOLOGIES USED

1. **PYTHON** – Used for backend development and integrating voice processing, recognition, and database operations.

2. **LIBROSA -** Librosa extracts key audio features like MFCCs for voice analysis from a wave file dataset.

3. **SPEECH RECOGNITION -** SpeechRecognition captures and transcribes voice input. Together, librosa and SR enable accurate voice-based attendance marking.

4. **MONGO DB / MY SQL -** Used for storing and managing attendance records securely. MongoDB (NoSQL) enables flexible data storage, while MySQL (SQL) supports structured data retrieval and reporting.

5. **SCIKIT LEARN -** Scikit-learn is used for training machine learning models like SVM, Random Forest, or KNN to classify and recognize voices. It processes extracted voice features (MFCCs) and matches them with stored voice profiles for identity verification.

6. **TENSORFLOW -** TensorFlow enables deep learning models like CNNs, RNNs, or LSTMs for more accurate speaker identification. It learns complex voice patterns, improving recognition accuracy even in noisy environments.

7. **FLASK / DJANGO -** Provides a web-based interface for user interaction, attendance management, and real-time feedback. Flask is lightweight for simple applications, while Django offers scalability and security.

3. FEATURES AND FUNCTIONALITIES

3.1 VOICE RECOGNITION FOR ATTENDANCE

- The system captures and processes voice input using **SpeechRecognition** to identify and authenticate users.
- Users can speak their names, and the system verifies their identity by matching voice features with stored profiles.

3.2 AUTOMATIC ATTENDANCE LOGGING

- Attendance is marked instantly upon successful voice recognition, with timestamps recorded in the database.
- Data is securely stored in **MongoDB/MySQL**, allowing easy retrieval and report generation.

3.3 DATABASE INTEGRATION

- All attendance records are stored securely in **SQL or NoSQL databases** for structured data management.
- The system supports querying and reporting, enabling administrators to analyze attendance trends efficiently.

3.4 VOICE COMMANDS FOR REPORTING

- Users can request attendance reports using voice commands for **hands-free interaction.**
- The system processes commands and provides insights such as **total attendance and missed sessions.**

3.5 ERROR HANDLING AND RETRY MECHANISM

- If a voice input is **unclear or unrecognized**, the system prompts the user for another attempt.
- Advanced machine learning techniques improve **recognition accuracy, reducing misinterpretations** over time.

4. DATABASE SETUP

1. USER DATABASE FOR VOICE PROFILES

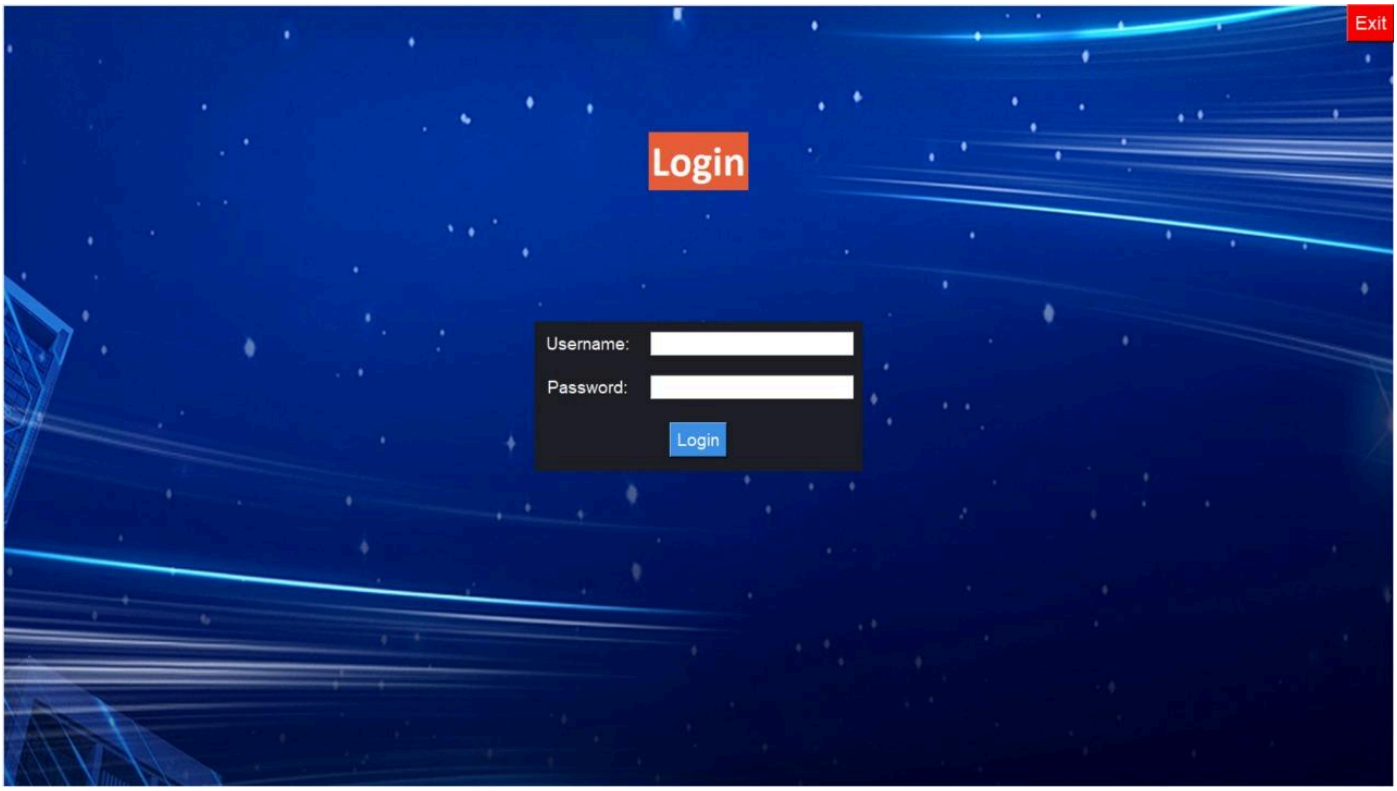
- The database securely stores user information, including voice profiles, for **authentication and attendance tracking.**
- A dedicated **Users** table contains essential details such as **User_Id, Name, and Voice_Features**, where voice data is stored as extracted feature vectors (MFCCs).
- To ensure data security, user details are encrypted, and access control measures are implemented.
- Each user's voice input is compared against stored voice features for authentication, ensuring accurate and secure attendance marking.

2. CONVERSION HISTORY TABLE FOR VISUALIZATION

- A separate **Attendance** table logs each user's attendance details in real-time.
- This table includes columns such as **User_Id, Name, Timestamp, and Status** to track attendance history efficiently.
- The system automatically updates attendance records upon successful voice recognition, eliminating manual intervention.
- Administrators can query and analyze attendance trends, enabling efficient record management and reporting.

5. USER INTERFACE & DATABASE

LOGIN SCREEN



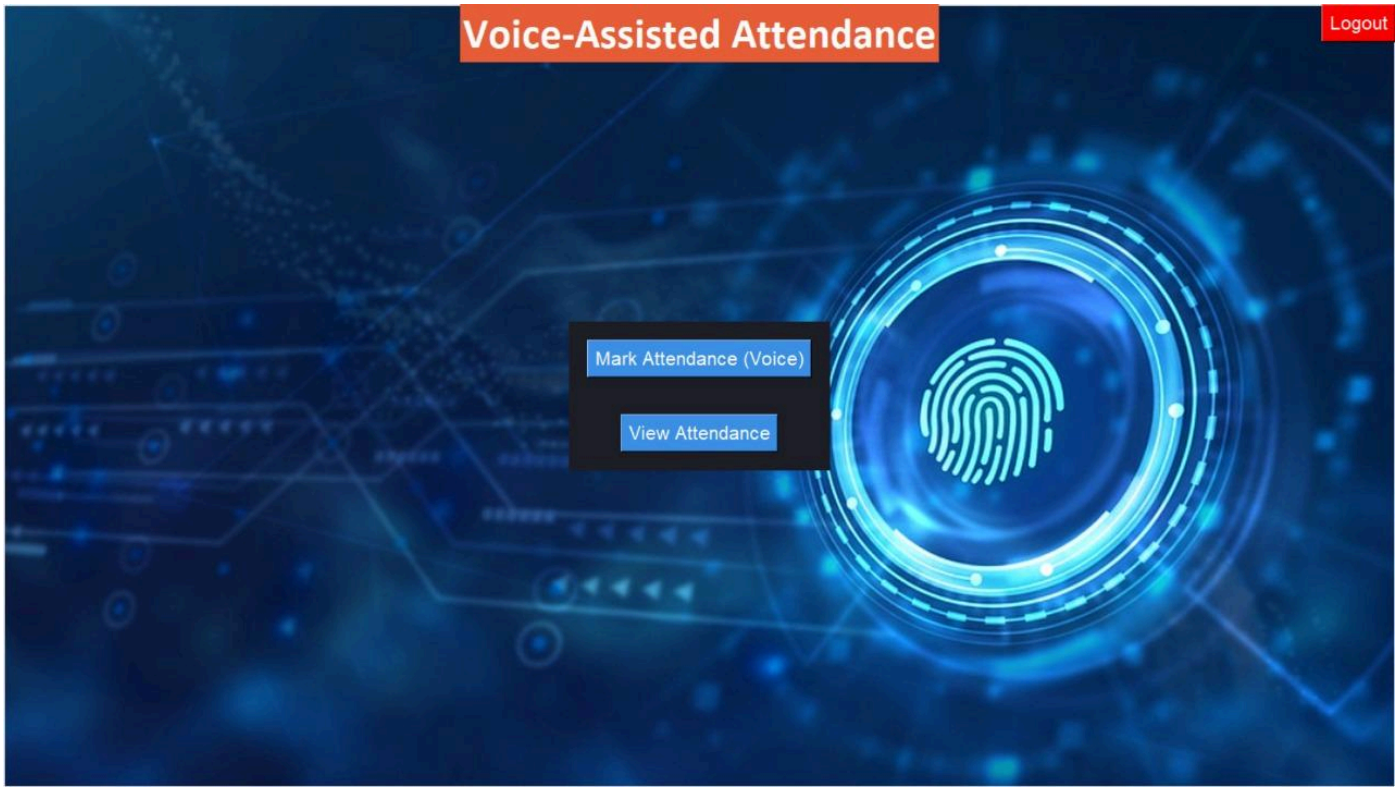
The login interface ensures **secure access** by authenticating users with their credentials. It is designed for **efficiency and ease of use**, enabling authorized users to manage their attendance seamlessly.

To enhance user interaction, the interface provides one key buttons:

1. **Username & Password Authentication:** Users must enter their credentials to access the system securely.
2. **Login Button:** Upon entering valid credentials, users can log in and navigate to the main dashboard.
3. **Exit Button:** Allows users to safely close the application when needed.

This **intuitive and responsive interface** ensures a **smooth, secure, and user-friendly login experience** for efficient attendance management.

MAIN INTERFACE



The **main interface** of the **Voice-Assisted Attendance System** is is designed with a **modern, secure, and user-friendly layout** to provide an efficient way of recording attendance using **voice recognition technology**. This **automated system** ensures accuracy, reduces manual effort, and enhances security through biometric authentication.

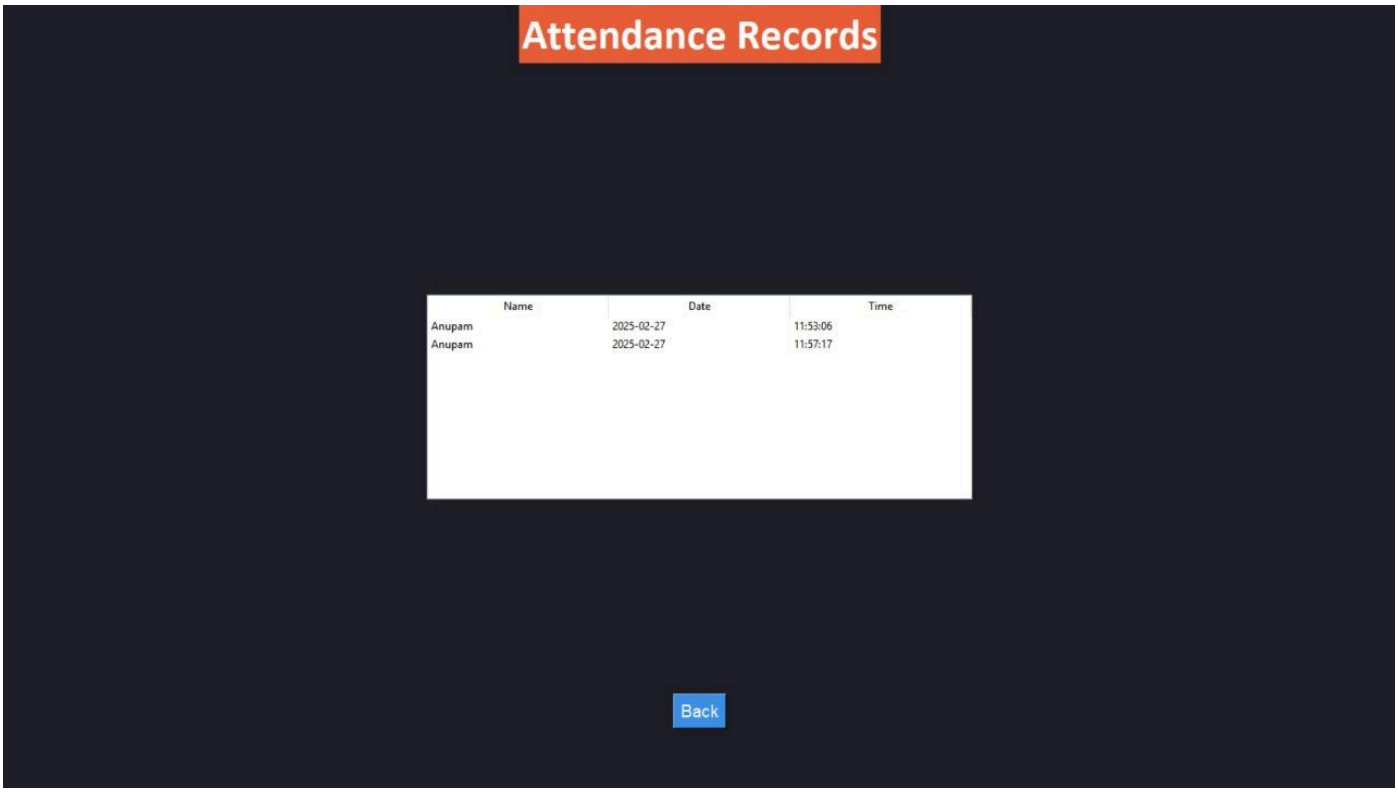
Key Features:

1. **Mark Attendance (Voice):** Users can **record attendance through voice input**, utilizing **speech recognition technology** for authentication.The system listens to the user's voice input. It compares the voice sample with the stored biometric data. If authenticated, attendance is successfully recorded in the system.

2. **View Attendance:** Provides a **detailed log** of all attendance records, allowing users to **track their history** efficiently.Displays **attendance history** in a structured format. Allows users to **filter by date and time**. Ensures **transparency** for both students and employees.

3. **Logout Button:** Ensures **secure exit** from the system, protecting user data. An Exit button at the top-right corner closes the application. **Ensures security** by allowing users to log out of their session once attendance is marked. **Prevents unauthorized access** to personal attendance records. Helps maintain **data privacy and protection**.

ATTENDANCE RECORD



The **User Interface (UI)** of the attendance system is designed with the following key elements:

1. **FRONT END:**
    - The interface displays attendance records in a **table format** with columns for **Name, Date, and Time**.
    - A **Back button** allows users to return to the previous screen.
    - The UI is designed using frameworks with **Django, or Flask** possibly being used for rendering dynamic content.
2. **BACK END:**
    - **Data Collection:** Attendance is recorded through a biometric scanner, RFID card, facial recognition, or manual input.
    - **Database Storage:** Each record (Name, Date, Time) is stored in a database such as **MySQL, PostgreSQL, or MongoDB**.
    - **Fetching & Displaying Data:** A backend system retrieves attendance records and sends them to the frontend.
3. **WORK FLOW:**
    - **User Marks Attendance** : Data is stored in the database.
    - **Backend Fetches Records** : Retrieves stored data.
    - **Frontend Displays Data** : Attendance logs appear in the UI.

7. CONCLUSION

The **Attendance Record System** provides an efficient and reliable method for tracking attendance in real time, integrating a user-friendly interface with a secure backend to ensure accurate record-keeping and easy data access. By automating data storage and retrieval, it minimizes errors and eliminates the inefficiencies of manual tracking. The system can be further enhanced with advanced technologies like facial recognition, RFID, or biometric authentication for improved accuracy and security. Overall, it streamlines attendance monitoring, reduces administrative workload, and offers valuable insights for organizations or institutions, with future potential for data analytics, cloud integration, and mobile accessibility to enhance scalability and usability.

REFERENCES:

- **Python Tkinter Documentation**
- **Librosa Installation Guide – Librosa Documentation**
- **Librosa Installation Guide – Librosa Documentation**
- **Pickle Module (Serialization in Python) – Python Documentation**
- **Support Vector Classification (SVC) in Scikit-Learn – Scikit-Learn Documentation**
- **Train-Test Split for Machine Learning Models – Scikit-Learn Documentation**
- **Text-to-Speech Conversion with pyttsx3 – PyPI: pyttsx3**
- **Speech Recognition in Python (v2.1.3) – PyPI: SpeechRecognition**

THANK YOU