**Abstract**

**Introduction**

In recent years, Artificial Intelligence (AI) has transformed human-computer interaction through intelligent systems capable of responding to text and voice inputs. AI chatbots have become essential tools in customer service, education, and personal assistance, offering automated and efficient interactions. This project aims to develop an advanced AI chatbot that provides a seamless text and voice-based communication experience. Unlike cloud-based AI assistants, this chatbot operates entirely offline, ensuring privacy and independence from external servers. It leverages advanced natural language processing (NLP), text-to-speech (TTS), and speech-to-text (STT) technologies to offer realistic and engaging interactions. The chatbot is powered by a local AI model, uses a robust speech recognition engine, and generates human-like voice responses through Piper TTS. Designed for use on both desktops and mobile devices, the system features a minimalistic yet responsive web interface, ensuring accessibility across different platforms.

**Users of the System**

This AI chatbot caters to a diverse set of users, including:

* **General Users**: Individuals seeking an AI companion for casual conversations and assistance.
* **Developers**: Programmers and researchers requiring an offline AI chatbot for integration into applications.
* **Students**: Those in need of an AI-powered study aid for learning and research.
* **Business Users**: Organizations looking for automated customer support and task management solutions.

**Modules of the Project**

The project consists of the following key modules:

1. **User Interface Module**: A web-based frontend that allows users to interact via text and voice.
2. **Speech Recognition Module**: Converts voice input into text using Vosk for offline speech recognition.
3. **AI Processing Module**: Utilizes a locally hosted AI model (KoboldCpp) for intelligent responses.
4. **Text-to-Speech Module**: Generates human-like voice responses using Piper TTS.
5. **Database Module**: Stores conversation history using MySQL to enhance contextual responses.
6. **Backend Module**: Manages API requests, processes interactions, and handles data storage.

**Current Systems in the Market**

Several AI chatbot systems are available in the market, including:

* **Google Assistant, Alexa, and Siri**: Cloud-based AI assistants requiring an internet connection.
* **OpenAI’s ChatGPT**: A powerful AI chatbot that operates via API access.
* **Rasa & Dialogflow**: Open-source and cloud-based chatbot development platforms.
* **Cortana & Bixby**: AI assistants integrated into Windows and Samsung ecosystems.

While these systems provide sophisticated AI capabilities, they depend on internet connectivity and centralized data storage. Our proposed system eliminates these limitations by offering a fully offline AI chatbot with integrated voice and text interaction.

**Proposed System**

The proposed system overcomes the challenges of cloud-based chatbots by delivering a completely offline AI chatbot that:

* Supports both text and voice interactions.
* Utilizes voice recognition for seamless communication.
* Generates realistic AI responses using a local model.
* Provides high-quality speech output with Piper TTS.
* Stores conversation history for improved contextual awareness.
* Offers a responsive web interface for access on multiple devices.

**Frontend**

The frontend is a dark-themed, aesthetically designed web application optimized for both desktop and mobile devices. It is built using:

* **React.js with Vite** for fast UI development.
* **Tailwind CSS** for a modern and responsive layout.
* **Lucide Icons** to enhance user experience with intuitive visuals.
* **Framer Motion** for smooth animations and transitions.

Users can type messages or use the microphone button for voice input, and responses are displayed dynamically with animated elements.

**Backend**

The backend is responsible for managing user interactions, AI processing, and database operations. It includes:

* **Flask (Python)** for handling API requests and processing responses.
* **MySQL** for storing conversation history.
* **KoboldCpp** for AI response generation.
* **Vosk** for offline speech-to-text processing.
* **Piper TTS** for high-quality text-to-speech synthesis.
* **Subprocess** for executing external voice synthesis tasks.

**Tools Used**

The following tools and technologies were used to build this system:

* **Frontend**: React.js, Tailwind CSS, Framer Motion, Lucide Icons
* **Backend**: Flask, MySQL, KoboldCpp
* **Speech Recognition**: Vosk
* **Text-to-Speech**: Piper TTS
* **Database**: MySQL
* **Other Dependencies**: Torch, PyAudio, SoundFile, Pydub

By integrating these tools, the project delivers a highly functional, offline, and user-friendly AI chatbot that excels in text and voice-based interactions.