**Abstract**

**Introduction**

**Artificial intelligence (AI) has significantly transformed the way humans interact with technology. AI-powered chatbots have become an essential part of various industries, providing automation and efficiency in customer service, virtual assistance, and smart interactions. Most AI chatbots rely on cloud-based processing, requiring an active internet connection and often leading to concerns regarding data privacy and latency.**

**This project aims to develop an advanced AI chatbot that operates entirely offline while offering both text and voice interaction capabilities. The chatbot runs on a local machine, allowing users to access it through a web application on their mobile devices. The system integrates key AI technologies such as natural language processing (NLP), speech-to-text (STT), and text-to-speech (TTS) to enable seamless, human-like interactions.**

**A locally hosted AI model generates intelligent responses, a speech recognition engine captures voice input, and an advanced TTS model ensures natural voice output. This project provides a secure and responsive chatbot experience without reliance on cloud services, ensuring greater control over data privacy and security. By leveraging state-of-the-art tools and frameworks, this chatbot delivers a user-friendly, efficient, and aesthetically appealing interface designed for an optimal conversational experience.**

**Users of the System**

**The AI chatbot is designed for a wide range of users, including:**

* **General Users: Individuals looking for an AI assistant for casual conversations, reminders, or general inquiries.**
* **Developers: Programmers and researchers who want an offline AI chatbot for integration into other applications.**
* **Students: Users who need an AI-powered study companion or learning assistant.**
* **Business Users: Organizations looking to deploy an AI chatbot for customer support and task automation.**

**Modules of the Project**

**The project consists of several key modules:**

1. **User Interface Module: A web-based frontend allowing 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: Uses a locally hosted AI model (KoboldCpp) to generate responses.**
4. **Text-to-Speech Module: Converts AI-generated text into speech using the Coqui TTS VITS model.**
5. **Database Module: Stores conversation history using MySQL for contextual interactions.**
6. **Backend Module: Manages API requests, connects various components, and ensures smooth communication between the frontend and AI model.**

**Current Systems in the Market**

**There are several chatbot solutions available in the market, including:**

* **Google Assistant, Alexa, Siri: Cloud-based AI assistants that require an internet connection.**
* **OpenAI’s ChatGPT: An advanced AI chatbot with powerful language understanding but requires API access.**
* **Rasa & Dialogflow: Open-source and cloud-based chatbot frameworks for businesses.**
* **Cortana & Bixby: AI assistants built into specific ecosystems like Windows and Samsung devices.**

**While these solutions provide robust AI interactions, they are primarily cloud-based, limiting offline usability and customizability. Our system addresses this by offering an offline AI chatbot with text and voice interaction capabilities.**

**Proposed System**

**The proposed system overcomes the limitations of cloud-based AI assistants by providing an entirely offline chatbot experience. It allows users to:**

* **Send text input and receive AI-generated responses.**
* **Use voice input to interact with the chatbot.**
* **Hear AI responses through realistic text-to-speech synthesis.**
* **Store and retrieve previous conversation history for contextual interactions.**
* **Access the chatbot through a web application on mobile devices while the main processing runs on a local machine.**

**Frontend of the System**

**The frontend of the system is a minimalistic and aesthetically designed web application that mimics an iOS-like user experience. It is built using:**

* **React.js with Vite for fast, responsive UI development.**
* **Tailwind CSS for a sleek and modern design with fluid animations.**
* **Shadcn UI for enhanced UI components.**
* **Lucide Icons to provide intuitive visual elements.**

**Users can send messages using a text input box or initiate voice recognition via a microphone button. The chatbot's responses are displayed in a chat format with smooth animations.**

**Backend of the System**

**The backend serves as the core processing unit of the chatbot. It is responsible for:**

* **Handling text and voice inputs from the frontend.**
* **Generating AI responses using the locally hosted KoboldCpp model.**
* **Managing text-to-speech synthesis using Coqui TTS (VITS model).**
* **Communicating with the MySQL database to store and retrieve past interactions.**
* **Running a Flask server to provide RESTful API endpoints for seamless frontend-backend communication.**

**Tools Used**

**The following tools and technologies were used to develop the project:**

* **Frontend: React.js, Vite, Tailwind CSS, Shadcn UI, Lucide Icons**
* **Backend: Flask (Python), MySQL**
* **AI & NLP: KoboldCpp for AI-generated responses**
* **Speech-to-Text: Vosk for offline voice recognition**
* **Text-to-Speech: Coqui TTS (VITS model)**
* **Database: MySQL for storing conversation history**
* **Other Libraries: Torch, PyAudio, SoundFile, Pydub (for audio processing)**

**The integration of these tools ensures an efficient, offline, and user-friendly AI chatbot capable of handling both text and voice interactions seamlessly.**