

# ■ Ganesh Bappa Chatbot – Technical Documentation

## System Overview

The Ganesh Bappa Chatbot is an AI-powered conversational assistant that allows users to interact via text or voice. It provides responses in a natural Ganesh Bappa voice and uses React (with TailwindCSS) for the frontend and Flask (Python) for the backend API.

## System Architecture

Frontend (React + TailwindCSS) communicates with Flask backend via REST API. Flask handles text/voice inputs, integrates Speech Recognition, Gemini API, and Edge-TTS for responses, returning both text and audio output.

## Tools & Technologies

**Frontend:** React.js, TailwindCSS, Recorder.js (MediaRecorder API), Axios/Fetch

**Backend:** Flask, SpeechRecognition, Pydub, FFmpeg, Edge-TTS, Asyncio, Glob, Subprocess

**APIs & Models:** Google Speech Recognition API, Gemini API (LLM), Edge-TTS Neural Voice (en-IN-PrabhatNeural)

## API Endpoints

**/handle** (POST) – Takes user text query → returns AI response

**/process\_audio** (POST) – Takes audio input → STT → Gemini → TTS → returns text+audio

**/text\_to\_speech** (POST) – Converts text to mp3

**/get\_audio/<filename>** (GET) – Fetches generated audio

## Workflow

**Text Query:** User → React → Flask → Gemini → Flask → React

**Voice Query:** User → Recorder.js → Flask → STT → Gemini → TTS → Flask → React

## Security & Error Handling

- Max upload size: 16MB
- Temporary audio stored in uploads/ and auto-cleaned
- Handles errors in speech recognition, API calls, conversion failures

- Friendly fallback response when failure occurs

## **Future Enhancements**

- Multi-language support (Hindi, Marathi, Sanskrit)
- Emotion-aware responses
- Database integration for logs
- WebSocket streaming
- PWA deployment
- Docker containerization