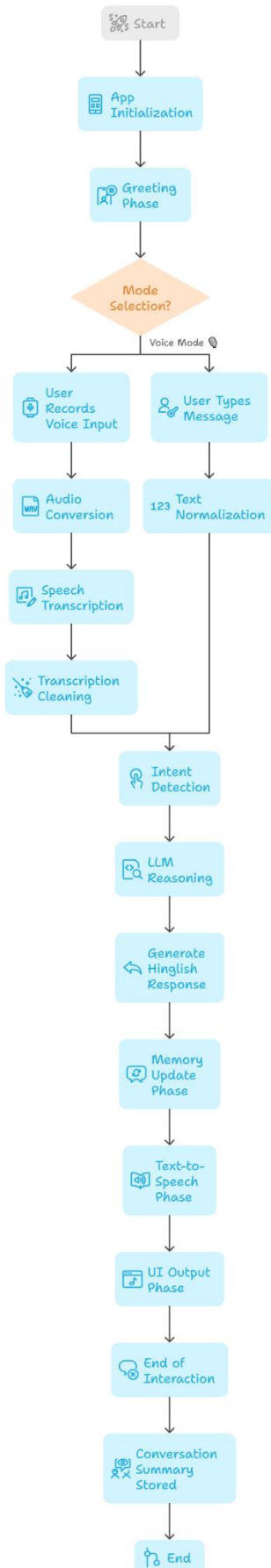


Miss Riverwood Voice Agent

Miss Riverwood Voice Agent Flowchart



1. Start

The process begins when the user launches the Miss Riverwood Voice Agent application. It prepares the environment for interaction and restores any **previous session memory** if available.

2. App Initialization

The app loads the Streamlit interface, microphone recorder, project memory, and offline models like Vosk (STT), Ollama (LLM), and gTTS (TTS). This makes the system ready for **both voice and text inputs**.

3. Greeting Phase

Miss Riverwood **greet**s the user with a friendly message like “Namaste! Main Miss Riverwood hoon — aapke daily construction updates ki saathi,” setting a conversational and approachable tone.

4. Mode Selection

The app checks the user’s preferred input mode — **Voice Mode** or **Text Mode** — and directs the process accordingly.

5. User Records Voice Input / User Types Message

If the user speaks, the voice is recorded for **transcription**. If the user types, the text is captured directly for understanding.

6. Audio Conversion / Text Normalization

Recorded audio is converted into a standard **WAV format** (16kHz), while typed text is **cleaned** and **normalized** to remove unnecessary spaces or characters.

7. Speech Transcription

In voice mode, the converted audio is processed using the **Vosk speech-to-text model**, producing accurate **Hindi** or **Hinglish** text.

8. Transcription Cleaning

The **transcribed** text is cleaned and formatted for the next stage of processing.

9. Intent Detection

The system analyzes the query using **RapidFuzz** to detect what the user is asking — such as **project progress**, **materials**, **delays**, **weather**, or **site team updates**.

10. LLM Reasoning

The cleaned query, along with memory context, is passed to **Ollama (Llama 3.2)** for reasoning. The AI understands the context and prepares a meaningful response.

11. Generate Hinglish Response

The LLM produces a natural, **human-like Hinglish answer** — for example, “Ji Sir, aaj ka kaam 60% complete ho gaya hai, aur kal concreting start hogi.”

12. Memory Update Phase

The new conversation is saved in **session memory**, allowing Miss Riverwood to remember context and respond consistently in future turns.

13. Text-to-Speech Phase

The generated Hinglish reply is converted to voice using **gTTS**, giving the assistant a natural **female tone**.

14. End of Interaction The conversation round finishes. The system resets the **input state**, stores a brief **session summary**, and prepares itself to receive the user’s next message or close the interaction

3. Tech Explanation

1. Streamlit – Frontend UI framework for building the interactive app.
2. Vosk – Offline Speech-to-Text (STT) engine (English + Hindi).
3. RapidFuzz – Fuzzy logic for intent detection and keyword matching.
4. gTTS (Google Text-to-Speech) – Converts AI reply text into natural audio.
5. Ollama + Llama3.2:1b – Local LLM for generating contextual replies (no internet).
6. Python – Core language integrating all AI modules.
7. Pydub / wave / io – Audio conversion, preprocessing, and handling.
8. Requests + JSON – For API communication between Streamlit and Ollama.

Technologies Mentioned in PDF but Not Used

1. OpenAI API / Cloud GPT models
2. Whisper (STT)
3. Hugging Face Transformers / LangChain
4. Azure / Firebase Cloud Integration
5. SpeechRecognition / PyAudio / DeepSpeech

Why Not Used / Issues Faced

- Cloud LLMs (OpenAI, Hugging Face) → Needed paid API keys & internet; violated “local-only” requirement.
- Whisper / DeepSpeech → High RAM usage & slow real-time transcription on local CPU.
- LangChain → Overkill for small local memory; slowed response time.
- Azure / Firebase → Unnecessary for offline setup; increased complexity.
- SpeechRecognition / PyAudio → Required mic device setup & dependencies; unstable in Streamlit cloud.

Key Fixes & Improvements

- Switched to Vosk → lightweight, fully offline STT.
- Integrated Ollama local LLM → fast, privacy-safe replies.
- Used RapidFuzz → handled flexible Hinglish queries.
- Optimized session state & rerun logic → fixed repeated execution issues.
- Added custom CSS theme → modern dark UI look.

3. How This Project Meets the Company's Requirements

1. Conversational Voice Flow: The final version of Miss Riverwood follows a complete professional voice assistant loop — it listens, understands, reasons, and speaks back naturally. The system fully supports both voice and text inputs, processes them through Vosk (STT), Ollama LLM, Memory, and gTTS (TTS), and then outputs the reply through the Streamlit UI. This mirrors the design of modern conversational AI systems while remaining lightweight and efficient.

2. Natural Hinglish Interaction: Miss Riverwood communicates in a friendly and natural Hinglish tone that reflects how real users in India speak. The model combines Hindi and English fluently, ensuring the responses sound human and relatable rather than robotic. This bilingual communication style directly matches Riverwood's goal of creating a "local, customer-friendly" voice agent.

3. Personalization: The assistant uses a built-in memory system that stores recent chats, user inputs, and simulated project details. This enables Miss Riverwood to maintain context across conversations and give more meaningful, consistent responses. The ability to "remember" what the user said earlier aligns perfectly with Riverwood's requirement for a context-aware, personalized virtual assistant.

4. Fast Prototype Ability: The project demonstrates rapid development with a fully functional end-to-end prototype built using open-source and offline tools. It does not rely on cloud infrastructure or paid APIs, ensuring zero recurring costs. This reflects practical system thinking — integrating all major components (STT, LLM, Memory, TTS, and UI) in one efficient pipeline that's smoothly on a local machine.

5. Realistic Project Simulation:

Miss Riverwood goes beyond just conversation — it simulates real construction site updates using editable project memory. Details like progress percentage, material status, weather conditions, and safety updates make the system directly relevant to Riverwood's operational environment. This ensures the assistant not only talks smartly but also provides useful, domain-specific information.

6. Professional and Engaging UI:

The custom dark-themed interface designed with Streamlit provides a clean, branded look consistent with Riverwood's visual identity. The UI is user-friendly, responsive, and features a voice recorder, text input, and playback — ensuring a professional, complete experience.

7. Storage: Currently, Miss Riverwood uses a small project memory dictionary stored inside the code.

It contains example data like progress percentage, materials, and weather, which the model uses to form answers.

Once the company provides live data, I can directly connect the same logic to a real database or API, so the assistant will speak actual, real-time construction updates instead of sample ones.

The design already supports this, just have to change the data source.

8. Time: Currently, the app takes around 4 to 5 seconds to respond — which is normal because everything runs locally. It first converts speech to text, sends it to the LLM for reasoning, and then generates Hinglish speech using gTTS. In future versions, I'll replace gTTS with a local voice model and add real-time streaming for both speech and text, so the responses come instantly, just like Alexa or Siri.

Future Advancements for Miss Riverwood



On-Device STT

Replace Vosk/gTTS with Whisper Tiny model running locally for lower latency and better Hinglish support.

Stream audio input/output for instant replies (Siri-like response loop).

Realtime Streaming STT + TTS



Expanded Multilingual Support

Auto-detect and speak in Marathi, Gujarati, or Tamil besides Hindi and English.

Use expressive voice models with emotion tags (happy, calm, energetic).

Emotional TTS



Face-to-Avatar Integration

Add 3D avatar that lip-syncs TTS audio for more interactive experience.

Connect voice agent to project dashboards → fetch live weather, progress, CRM data, and emails.

Tool Use & Automation



Analytics + User Behaviour Insights

Track common queries, STT accuracy, and LLM response time for model improvement.

Provide core features even without internet using compressed local models.

Offline Mode Enhancement

