RiverWood AI Voice Agent – Technical Report

1. Introduction

RiverWood Projects LLP requires a personalized AI assistant for providing **construction project updates**, casual interactions, and quick information to clients. The AI assistant, **Miss Riverdale**, is a **bilingual desktop application** capable of interacting in **Hindi, Hinglish, and English**.

The prototype provides:

- Real-time project progress updates
- Field-specific status information (e.g., plumbing, flooring, painting)
- Polite conversational capabilities
- Session memory for continuous interaction across sessions

---

2. System Architecture

2.1 Frontend (User Interface)

- **Tkinter**: Provides a lightweight GUI for desktop usage.
- **ScrolledText widget**: For the chat area to display multi-line messages.
- **Input and Buttons**: Users can type messages or use voice input.
-

Voice Integration: Microphone button triggers speech recognition.

- **Status Indicators**: Displays online/offline status of AI (API connectivity).

2.2 Backend (Core Logic)

- **Python Modules**:
  - ai_core.py – Handles intent detection, language detection, session memory, AI response generation.
  - construction.py – Stores project data: completed tasks, pending tasks, current in-progress task, and progress percentages.
  - voice_utils.py – Handles text-to-speech output for responses.
- **OpenAI GPT-4o-mini API**: Generates conversational replies for general queries or complex interactions outside predefined construction responses.

2.3 Session Management

- Memory stored locally in **memory.json**.
- Tracks:
  - User messages
  -

Bot responses

- Last accessed project ID for personalized updates

- Ensures continuity when the app is restarted.

2.4 Data Flow

User Input Language & Intent Detection [Construction Query] Fetch Project Data Format & Respond [Fun/General Query] GPT API or Local Response Respond [Restricted Topic] Polite Decline Respond

---

3. Functional Modules

3.1 Language Detection

- **Hindi**: Detected via Devanagari Unicode ranges.

- **Hinglish**: Detects common Hindi words written in English script.

- **English**: Default when neither Hindi nor Hinglish detected.

3.2 Intent Classification

- **Construction**: Keywords like update, plumbing, painting, flooring.
- **Fun/General**: Greetings, jokes, casual conversation.
- **Restricted**: Sensitive topics (politics, religion, conflict).

3.3 Construction Data Handling

- Project data stored in construction.py:

"RW00125": { "name": "Amit", "progress": 80, "current": {"task": "Painting", "percent": 10}, "completed": ["Foundation", "Walls", "Roof", "Plumbing"], "pending": ["Fixtures"], "status": "Slight Delay" }

- Queries like plumbing or painting fetch **specific field progress**.
- Responses are **formatted cleanly** without markdown artifacts, asterisks, or emoji issues for text-to-speech compatibility.
- Supports **Hinglish, Hindi, and English** dynamically.

3.4 Session Memory Logic

- Each user's interaction stored in memory.

- When a user queries a construction update without providing a project ID, the bot retrieves the **last project accessed in memory**.
- Ensures multi-turn conversations maintain **contextual awareness**.

3.5 Voice Integration

- **SpeechRecognition**: Converts user speech into text.
- **pyttsx3**: Converts bot text replies to speech.
- **UI Timer**: Shows recording duration during voice input.

---

4. Infrastructure & Cost Estimation

4.1 Local Desktop Application

- Runs fully locally with Python & Tkinter.
- Minimal disk usage (<50MB including dependencies).
- No dedicated server required.

4.2 OpenAI API

- GPT-4o-mini for dynamic conversational replies.
- Token usage ~0.0003 USD per 1k tokens.
- For 50 queries/day → ~5k tokens → $0.002/day/user.
- Monthly cost for ~30 users < **$2**.

4.3 Total Cost

- Minimal: Local environment + OpenAI API consumption.
- Prototype deployment comfortably **under $60/month**.

---

5. Security & Privacy

- **API Key** stored in .env file, never hardcoded.
- **No sensitive user data** is uploaded beyond session memory and optional API calls.
- Local JSON memory ensures **offline resilience**.

---

6. Conclusion

The **RiverWood AI Voice Agent** is a lightweight, bilingual AI assistant tailored for the construction industry. It:

- Provides **real-time project updates**
- Maintains **session context** across interactions
- Responds in **user's preferred language**
- Handles **field-specific queries**
- Is cost-efficient, simple to deploy, and easy to maintain.