Vaani – Voice-First Banking Assistant

Round 3 Submission – Technical Walkthrough

Hackathon Theme: AI Voice Assistant for Financial Operations

Date: 23 Nov 2025

Product Vision	Vaani brings safe, natural, and multilingual banking to Bharat through a voice-firs
Target Users	Tier■2/3 towns, rural India, and on-the-go mobile users who prefer speaking over navig
Core Outcomes	Reduce support costs, improve First Contact Resolution, and convert 'account ownersh

1. Technology Stack

Frontend (Web)

- React 19 + Vite; React Router; Axios; CSS Modules.
- Voice: Web Speech API for speech-to-text and text-to-speech (browser-native).
- Environment: VITE_BACKEND_URL (API) and VITE_AI_BACKEND_URL (AI).

Backend API

- FastAPI + SQLAlchemy 2.0; Pydantic v2; JWT auth (python-jose).
- SQLite for dev; PostgreSQL ready for production.
- Voice biometrics via Resemblyzer; device binding; reminder and beneficiary services.

AI Backend

- LangGraph orchestration with a Hybrid Supervisor routing intents to specialists.
- LLM: Ollama locally (Qwen 2.5 7B primary, Llama 3.2 3B fast); optional OpenAI.
- RAG: ChromaDB vector store; HuggingFace sentence-transformers (all-MiniLM-L6-v2).
- Observability: Structured logs; optional LangSmith tracing.

Local Ports	**Frontend: 5173 | Backend API: 8000 | AI Backend: 8001 | Ollama: 11434**

2. System Architecture

Three microservices work in tandem: Frontend (SPA), Backend API (banking operations), and AI Backend (agentic orchestration). The AI service calls tools that in turn interact with the Backend API and database. Documents are ingested into ChromaDB for grounded answers.

End-to-End Flows

- Balance inquiry: User asks → IntentRouter → Banking Agent → tool get_account_balance → Backend API/DB → friendly reply.
- Loan/investment FAQs: User asks in English/Hindi → RAG Supervisor → specialist (loan/investment) → vector retrieval → answer with citations.
- Hello UPI payments: Wake phrase or UPI mode → UPI Agent → recipient resolution → PIN entry (manual) → initiate payment → audit trail.

Agents	**Greeting, Banking, UPI, RAG Supervisor (Loan/Investment/Support), Feedback.**
Supervisor	Deterministic routing + shared conversation state; predictable outputs for UI and logs.
Voice	Browser-native TTS/STT today; Azure TTS pluggable.

3. Data Model and Storage

The application separates transactional data from knowledge data. Banking data sits in SQL; product knowledge is embedded into a vector store.

Relational (Operational)

- Users, Accounts, Transactions, Reminders, Beneficiaries, DeviceBindings, Sessions, Branches, Cards.
- SQLite for development; migrate to PostgreSQL for production (connection pooling, read replicas).
- Transactions carry channel (UPI/NEFT/IMPS/etc.), reference IDs, and balance-after to simplify statements.

Vector (Knowledge/RAG)

- ChromaDB with four collections: loan_products, loan_products_hindi, investment_schemes, investment_schemes_hindi.
- Semantic chunking preserves sections, tables and FAQs; rich metadata enables precise retrieval (loan_type, scheme_type, language, section).
- Embeddings via sentence-transformers/all-MiniLM-L6-v2 (multilingual, CPU■friendly).

Statements	Generated via transaction history; agent can return structured "statement_data"
UPI Ledger	UPI payments recorded as debit/credit pairs with a unique reference like UPI-YYYYMM

4. AI/ML/Automation Components

Hybrid Supervisor & Agents

- Supervisor builds conversation state, classifies intent, and dispatches to specialists.
- Banking Agent executes tool-augmented operations (balance, transactions, transfers, reminders).
- UPI Agent runs a guided, multi-step flow with consent and PIN verification (manual entry).
- RAG Supervisor routes to Loan/Investment/Support specialists for grounded answers.

Retrieval-Augmented Generation (RAG)

- PDF ingestion → semantic chunking → embeddings → ChromaDB; runtime similarity search fetches precise context.
- Bilingual collections support English and Hindi; code-mixed queries handled via prompts + metadata filtering.

Automation & Observability

- Structured logs for agent decisions and tool calls; optional LangSmith traces for deep debugging.
- Graceful failure messages and fallbacks to maintain user confidence.

5. Security and Compliance

Designed as compliance-by-design with Indian banking norms and RBI guardrails in mind.

Identity & Sessions

- JWT-based sessions with expiry; device binding (trusted/suspicious/revoked).
- Voice biometrics via embeddings; adaptive thresholds and AI-assisted scoring path available.
- Passwords hashed (bcrypt). No raw voice audio retained — only embeddings/signatures where required.

UPI & Payments

- PIN is never spoken or stored; must be entered manually via secure keypad.
- Explicit first-time UPI consent; clear confirmation of amount, recipient, and source account before PIN.
- Full audit trail with reference IDs; velocity limits and anomaly detection planned for production.

Data Minimisation & Privacy

- RAG accesses only relevant document chunks at query time (purpose limitation).
- Environment variables for secrets; strict input validation on all APIs.
- Production hardening: TLS everywhere, Postgres with encryption-at-rest/HSM for PIN, rate limiting and WAF.

6. Scalability and Performance

Horizontal scaling

- Frontend is static and CDN-friendly.
- Backend API is stateless (JWT) and horizontally scalable behind a load balancer.
- AI Backend scales out by running multiple replicas; can switch from local Ollama to cloud LLMs.

Caching & Throughput

- RAG context cache (TTL) reduces repeated retrieval costs; Redis recommended for shared cache in prod.
- Keep last-N messages only; use fast model for voice turn-taking; stream tokens for perceived latency gains.

Target SLAs (dev baselines)

- Intent classification: 100–200 ms; standard replies: sub■second to ~2 s; RAG answers: under ~3 s on developer hardware.
- UPI flow end-to-end: typically under 5 s including user confirmation and PIN entry.

Annexure – Evaluation Fit

This solution addresses Round■1 criteria comprehensively: problem-solution fit for Bharat, core banking coverage, security-first UX, bilingual support (English/Hindi), agentic AI design with RAG, observability, and a practical pilot plan with measurable outcomes.

Pilot Scope (3 months)	**1,000 users executing Balance, Transactions, and Hello UPI payments; collect Hi**
Success Metrics	CSAT uplift, FCR improvement, Rs. cost-to-serve reduction, task completion times, and

Vaani – Banking that speaks your language.