Personalized AI Chat & Voice Agent – System Architecture Document

This document explains how the AI Chat and Voice Agent delivers personalized responses based on user data, how multiple backend services collaborate, and how security and privacy are preserved across the system.

1. High-Level Overview

The system consists of a frontend chat and voice interface, an AI orchestration layer, context and intent services, data stores (MongoDB and Supabase), and optional text-to-speech services. The agent never responds generically; each response is contextual, personalized, and scoped to the authenticated user.

2. Chat & Voice Interaction Flow

- User initiates interaction via text input or voice input in their native language.
- Voice input is converted to text using the browser’s Speech Recognition API.
- The frontend sends the text, language, and authentication token to the AI API.
- The AI controller processes the request and builds a personalized context.
- The AI model generates a response aligned with the user’s financial data and intent.
- Optional TTS converts the response to audio in a supported language.

3. Personalization Logic

Personalization is achieved by combining intent detection with user-specific contextual data. The system identifies what the user is asking (for example: earnings, savings, habits, or reminders) and enriches the AI prompt with relevant historical and real-time data belonging only to that user.

4. Role of MongoDB

MongoDB acts as the primary transactional and conversational data store. It stores user profiles, financial entries, conversation history, detected intents, and AI responses. Each record is scoped by user ID, ensuring strict data isolation.

5. Role of Supabase

Supabase is used for secure object storage and optional authentication services. Generated audio files (TTS output), user-uploaded documents, and media assets are stored in Supabase buckets with access rules tied to the authenticated user.

6. AI Orchestration & Agent Communication

The AI agent operates as an orchestrator rather than a single monolithic model. It communicates with intent detection utilities, context builders, and data services before invoking the language model. Each component has a single responsibility, making the system modular and auditable.

7. Security Model

- JWT-based authentication ensures only authenticated users can access AI services.
- User context is fetched server-side and never exposed directly to the client.
- AI prompts are built per-request and are never shared across users.
- Supabase storage rules prevent cross-user access to media files.

8. Privacy Guarantees

The system enforces privacy by design. AI responses are generated in real time and are not used to train external models. Sensitive financial data remains within the controlled backend environment and is never logged or exposed to third parties.

9. Fault Tolerance & Reliability

Text-based AI responses always succeed even if voice or TTS services fail. Failures in auxiliary services do not impact core chat functionality, ensuring a reliable user experience under all conditions.

10. Conclusion

This architecture ensures that every AI response is personalized, secure, and privacy-preserving. By separating concerns across services and enforcing strict data boundaries, the system delivers trustworthy AI-driven assistance tailored to each individual user.