

Voice-Driven Multi-Agent Financial Assistant

Overview

This project is a modular, voice-enabled financial assistant built using **multi-agent architecture**, combining **voice interface**, **data scraping**, **retrieval**, **LLM reasoning**, and **text-to-speech output**. It allows users to query financial information using **natural speech**, and the system returns summarized insights using **real-time data** from APIs, financial news, and contextual documents.

System Architecture

The system is divided into five logical layers:

1. Input Layer

- **User Voice Input:** Captures user queries via microphone.
- **Whisper STT (Speech-to-Text):**
 - Converts voice to text.
 - Model used: OpenAI's Whisper.

2. User Interface Layer

- **Streamlit UI:**
 - Main frontend for both voice/text input and output display.
 - Acts as a lightweight orchestrator to simulate agent coordination.

3. Agent Layer

This layer consists of modular agents with clearly defined roles:

- **API Agent (yFinance):**
 - Fetches real-time stock data (prices, volume, changes, etc.).
 - Lightweight and fast API interaction agent.

- **Scraper Agent (News/HTML):**
 - Scrapes relevant financial news headlines or article summaries.
 - Parses HTML and extracts latest market events.
- **Retriever Agent (FAISS RAG):**
 - Vector-based retrieval agent using **FAISS**.
 - Performs similarity search on stored financial context (e.g., PDFs, docs, reports).
- **Language Agent (OpenAI LLM):**
 - Core reasoning unit.
 - Aggregates results from other agents and generates coherent financial summaries using GPT-style LLM.
 - Adds reasoning, trend insights, and context interpretation.

4. Data Layer

- **Data + Context Store:**
 - Stores static contextual documents (PDFs, notes, historical data).
 - Used by the Retriever Agent to supply relevant information during summarization.
 - Enables **RAG (Retrieval-Augmented Generation)**.

5. Output Layer

- **LLM-based Market Summary:**
 - Generated by the Language Agent.
 - Includes data-driven insight: trends, movements, sentiment.
- **pyttsx3 TTS (Text-to-Speech):**

- Converts the LLM's text response into audio.
- **Spoken Output:**
 - Final voice response played to the user.

Tech Stack Summary

Component	Technology
UI	Streamlit
Voice Input	Whisper (OpenAI STT)
API Data Fetch	yFinance
Scraper Agent	BeautifulSoup, Requests
Retriever Agent	FAISS, LangChain RAG
Reasoning Agent	OpenAI GPT-4
TTS Output	pyttsx3

Orchestration Strategy

- **No separate orchestrator** service is used.
- Streamlit UI simulates coordination by calling agents in sequence.
- Logs and modular functions maintain flow clarity and debug ability.

Sample Flow

1. **User speaks:** “What’s the latest news on Tesla stock?”
2. **Whisper** transcribes to text.

3. **Streamlit UI** parses and routes:

- yFinance Agent gets current stock price.
- Scraper Agent fetches latest Tesla headlines.
- Retriever Agent fetches past financial reports.

4. **Language Agent** combines insights:

- Generates a summary of Tesla's current market position.

5. **TTS** converts it to speech.

6. **User hears** the answer.