Kyle Agent – Initial Setup Prompt for Cursor

Objective:

Develop an AI-powered assistant, Kyle, inspired by *Jarvis from Iron Man*, with:

- **Screen awareness** (capture & process user's screen)
- **Audio analysis** (recognize guitar notes & speech)
- **Long-term memory** (store & retrieve contextual interactions)
- **Advanced conversational abilities** using OpenAI models
- **A simple UI** to indicate when the agent is listening and speaking

---

Project Setup:

1. Create a structured project directory:

```
plaintext
CopyEdit
/kyle-agent
  ├── backend/ (Python - handles AI processing)
  ├── frontend/ (React/Electron - UI)
  ├── docs/ (Documentation)
  ├── .env (API keys & config)
  ├── README.md
```

2. Install dependencies:

- **Python Libraries:**

```
sh
CopyEdit
pip install openai whisper pydub opencv-python numpy torch
```

- **Node.js Libraries:**

```
sh
CopyEdit
npm install express mongoose tailwindcss electron
```

- Install **FFmpeg** for audio processing

Task Breakdown for Cursor:

Backend Development (Python)

✅ Set up **FastAPI or Flask** for backend API

✅ Implement **Whisper API** for audio transcription

✅ Integrate **OpenCV** for screen analysis

✅ Add **long-term memory** using Weaviate/Pinecone/ChromaDB

✅ Create API endpoints:

- `/listen` → Process audio input
- `/screen` → Analyze screen content
- `/respond` → Generate AI responses

Frontend UI (React/Electron)

✅ Build a **simple UI** showing when Kyle is listening/speaking

✅ Connect UI to backend API

✅ Implement **state management** for UI responsiveness

Testing & Optimization

✅ Ensure **audio & screen capture** work correctly

✅ Optimize **API calls** to reduce cost & improve performance

✅ Set up **Docker** for containerized deployment

Cursor's Immediate Task:

🚀 **Generate an initial project scaffold with a working backend API and a basic UI.**

- Start by **creating the FastAPI backend** and an Electron-based UI
- Ensure basic API calls work for **speech recognition** and **screen analysis**
- Provide clear logs & error handling

Additional Notes for Cursor:

- Use **OpenAI GPT-4 Turbo** for conversational responses
- Store API keys in a `.env` file
- Write **clean & modular code** with proper documentation

Expected Output (First Version)

✅ A functional **backend API** that listens and processes audio

✅ A **UI prototype** that shows when Kyle is listening/speaking

✅ Logs confirming that audio & screen capture are working

This prompt ensures Cursor **creates a solid foundation** for Kyle. Let me know if you want any refinements! 🚀