# Umemiya AI Agent

This project is a complete implementation of the Umemiya AI Agent, a sophisticated conversational AI system with a React-based frontend and a FastAPI backend.

**Author:** Sayed Allam

## Features

* **FastAPI Backend**: A robust backend with API endpoints for health checks, status, voice selection, and conversation handling.
* **React Frontend**: A user-friendly interface for interacting with the AI agent.
* **Real-time Audio Streaming**: WebSocket communication for real-time audio streaming between the frontend and backend.
* **9-Trait Personality System**: A personality system that allows users to customize the AI’s personality.
* **Text-to-Speech**: Multiple voice options for the AI’s responses.

## Project Structure

/umemiya\_ai\_agent  
|-- /backend  
| |-- requirements.txt  
| |-- main.py  
|-- /frontend  
| |-- package.json  
| |-- /public  
| | |-- index.html  
| | |-- manifest.json  
| |-- /src  
| |-- App.css  
| |-- App.js  
| |-- index.css  
| |-- index.js  
| |-- reportWebVitals.js  
|-- README.md

## Getting Started

### Prerequisites

* Python 3.7+
* Node.js and npm

### Backend Setup

1. **Navigate to the backend directory:**

* cd umemiya\_ai\_agent/backend

1. **Install the required dependencies:**

* pip install -r requirements.txt

1. **Run the FastAPI server:**

* uvicorn main:app --reload

### Frontend Setup

1. **Navigate to the frontend directory:**

* cd umemiya\_ai\_agent/frontend

1. **Install the required dependencies:**

* npm install

1. **Start the React development server:**

* npm start

## API Endpoints

* GET /api/health: Health check endpoint.
* GET /api/status: Get the status of the AI agent.
* GET /api/voices: Get the available voices.
* POST /api/conversation: Handle conversations with the AI.
* WS /ws: WebSocket for real-time audio streaming.