```
kai_agent/
│
├── 📁 agent/ (Das "Gehirn" und die "Hände" des Agenten)
│   ├── __init__.py
│   ├── agent_core.py       # Definiert die Kernlogik, den System-Prompt und die ReAct-Schleife.
│   └── 📁 tools/           # Die Fähigkeiten (Werkzeuge) des Agenten.
│   ├── __init__.py
│   ├── coding_tools.py     # Werkzeuge zum Lesen, Schreiben und Auflisten von Dateien.
│   ├── execution_tools.py  # Werkzeug zur sicheren Code-Ausführung in der Docker-Sandbox.
│   ├── governance_tools.py # Werkzeug für die menschliche Genehmigung ("Human-in-the-Loop").
│   ├── memory_tools.py     # Werkzeuge für das Langzeitgedächtnis (ChromaDB).
│   └── telephony_tools.py  # Werkzeug zum Initiieren von echten Telefonanrufen via Twilio.
│
├── 📁 agent_memory/ (Persistenter Speicher für das Langzeitgedächtnis)
│   └── ...                 # ChromaDB-Dateien, die automatisch hier erstellt werden.
│
├── 📁 agent_workspace/ (Der sichere Arbeitsbereich des Agenten)
│   └── ...                 # Dateien, die der Agent während seiner Arbeit erstellt (z.B. hallo.py).
│
├── 📁 sandbox/ (Konfiguration für die isolierte Code-Ausführungsumgebung)
│   ├── Dockerfile          # Bauplan für den Sandbox-Container, in dem Code ausgeführt wird.
│   └── requirements.txt    # Python-Pakete, die *innerhalb* der Sandbox verfügbar sind.
│
├── 📁 telephony_server/ (Der dedizierte Server für Echtzeit-Gespräche)
│   ├── __init__.py
│   ├── audio_processor.py  # Handhabt die STT -> LLM -> TTS Pipeline.
│   ├── Dockerfile          # Bauplan, um den Telefonie-Server zu containerisieren.
│   ├── requirements.txt    # Python-Pakete, die nur für den Telefonie-Server benötigt werden.
│   └── server.py           # Der Flask/WebSocket-Server, der mit Twilio kommuniziert.
│
├── 📁 tests/ (Framework für automatisierte Tests)
│   ├── __init__.py
│   └── test_agent.py       # Enthält Unit- und LLM-Qualitätstests (z.B. mit DeepEval).
│
├──.env                     # Ihre private Datei mit allen API-Schlüsseln und Geheimnissen.
├──.env.example             # Eine Vorlage, die zeigt, welche Schlüssel benötigt werden.
├── docker-compose.yml      # Orchestriert das Starten aller Dienste (Agent + Telefonie-Server) mit einem Befehl.
├── Dockerfile              # Bauplan, um die Haupt-Agenten-Anwendung zu containerisieren.
├── main.py                 # Der Haupt-Einstiegspunkt, um den Agenten zu starten und ihm eine Aufgabe zu geben.
├── README.md               # Die vollständige A-Z-Anleitung zur Einrichtung, zum Testen und zur Bereitstellung.
└── requirements.txt        # Python-Pakete, die für die Haupt-Agenten-Anwendung benötigt werden.
```