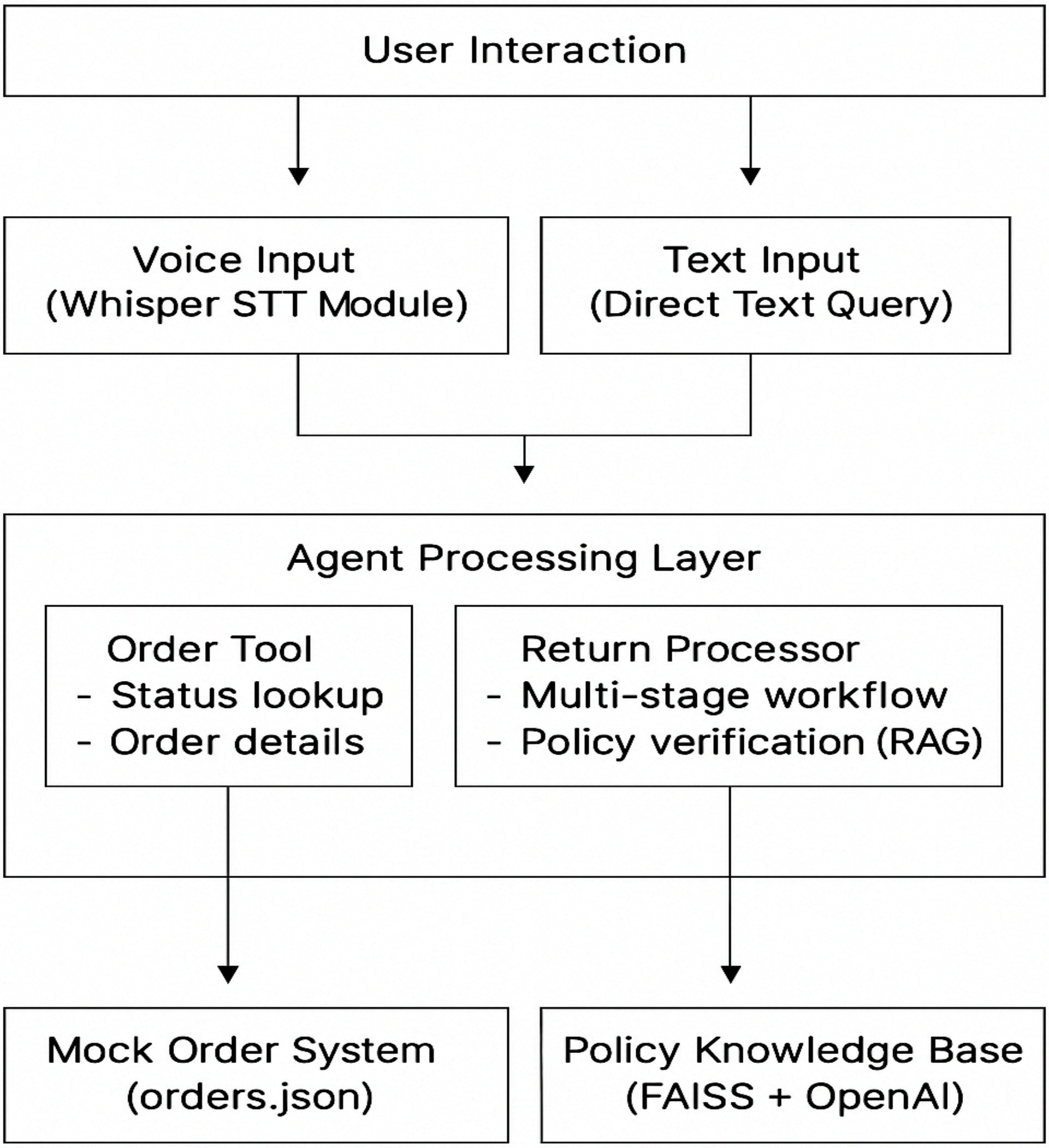
Technical Document of AWS GenAI Assessment

System Architecture 2
Implementation Approach 2
  Project Structure 3
  Core Components 3
  Key Workflows 3
    Pictorial Form 4
    Order Tracking 4
    Policy Verification (RAG) 4
    Return Request Processing 5
Screenshots of UI 5
  1- Voice Enabled (Tool 1: Order_tracker) 6
  2- Chat Enabled (Tool 2: Return Request Processor) Stage 1: What do you want to return ? 6
Tools and Technologies 8

mrhassands@gmail.com
WA/Cell # 0336-7974084

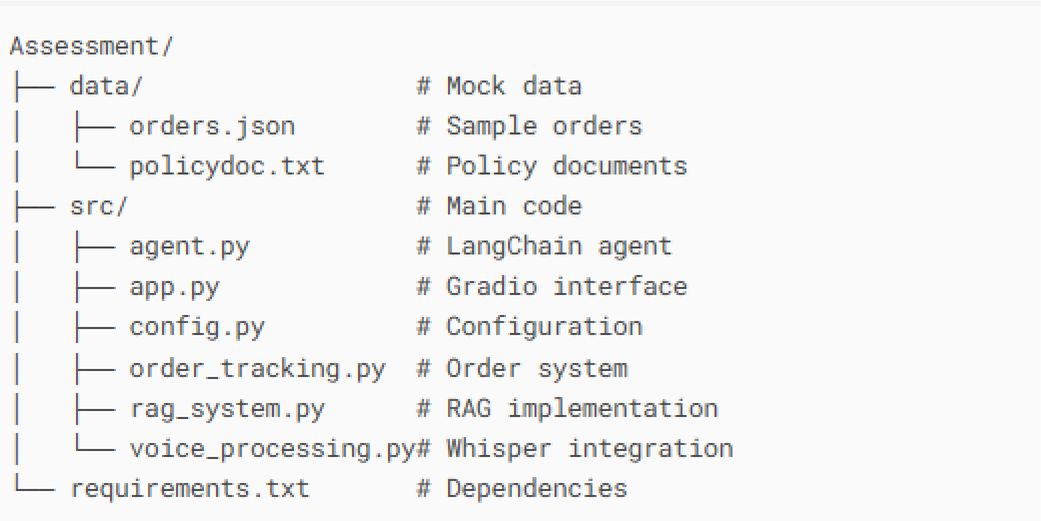
System Architecture



mrhassands@gmail.com
WA/Cell # 0336-7974084

Implementation Approach

Project Structure



```
Assessment/
├── data/                 # Mock data
│   ├── orders.json       # Sample orders
│   └── policydoc.txt     # Policy documents
├── src/                  # Main code
│   ├── agent.py          # LangChain agent
│   ├── app.py            # Gradio interface
│   ├── config.py         # Configuration
│   ├── order_tracking.py # Order system
│   ├── rag_system.py     # RAG implementation
│   └── voice_processing.py# Whisper integration
└── requirements.txt      # Dependencies
```

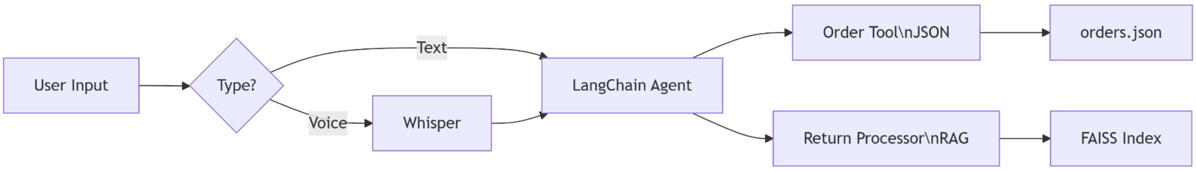
Core Components

Component	Technology	Purpose
Voice Processing	Whisper	Convert customer voice queries to text
Agent Framework	LangChain	Route queries to appropriate tools and manage conversation workflow
Order Tracking	JSON Database	Mock order system with lookup capabilities
Return Processing	Multi-Step Agent	Guided return workflow with Policy verification (RAG)
Policy Retrieval (RAG)	RAG (FAISS)	Vector search for realtime policy document
User Interface	Gradio	Unified interface for voice, text inputs

Key Workflows

mrhassands@gmail.com
WA/Cell # 0336-7974084

Pictorial Form



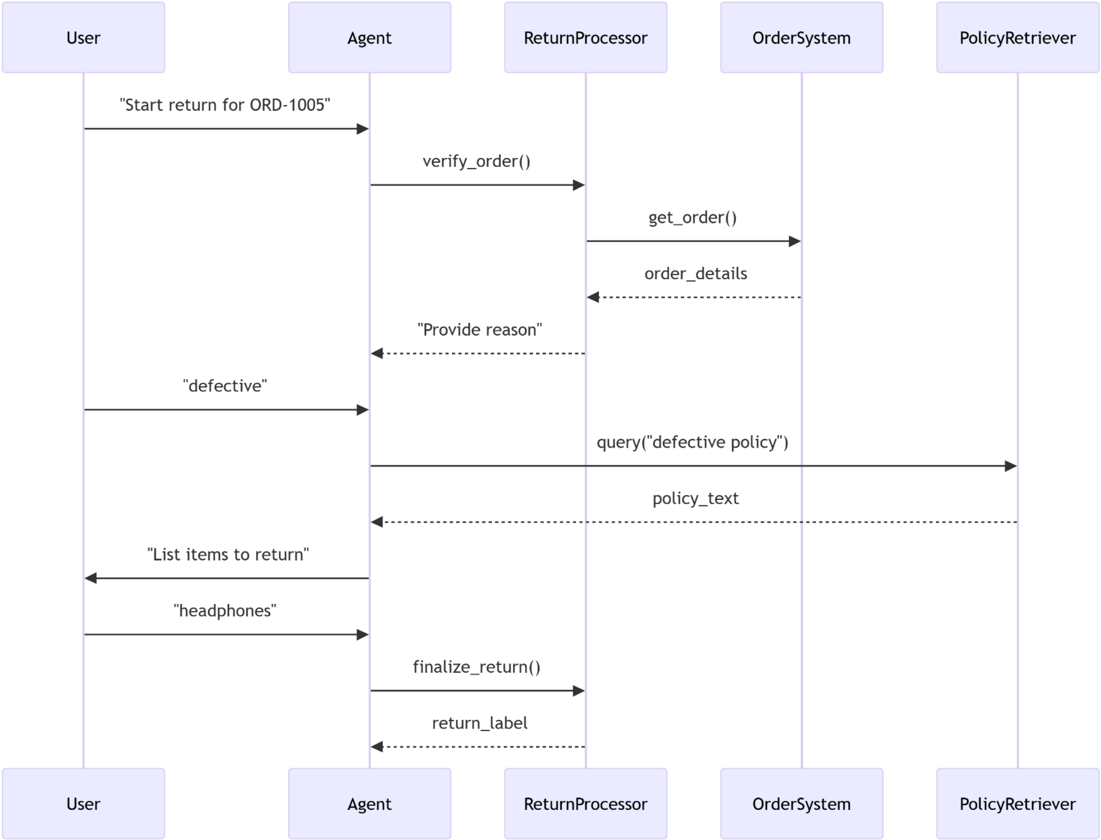
Order Tracking

1. User provides order ID/email
2. Agent invokes get_order_status tool
3. System queries orders.json mock database
4. Returns: Status + tracking number + item list

Policy Verification (RAG)

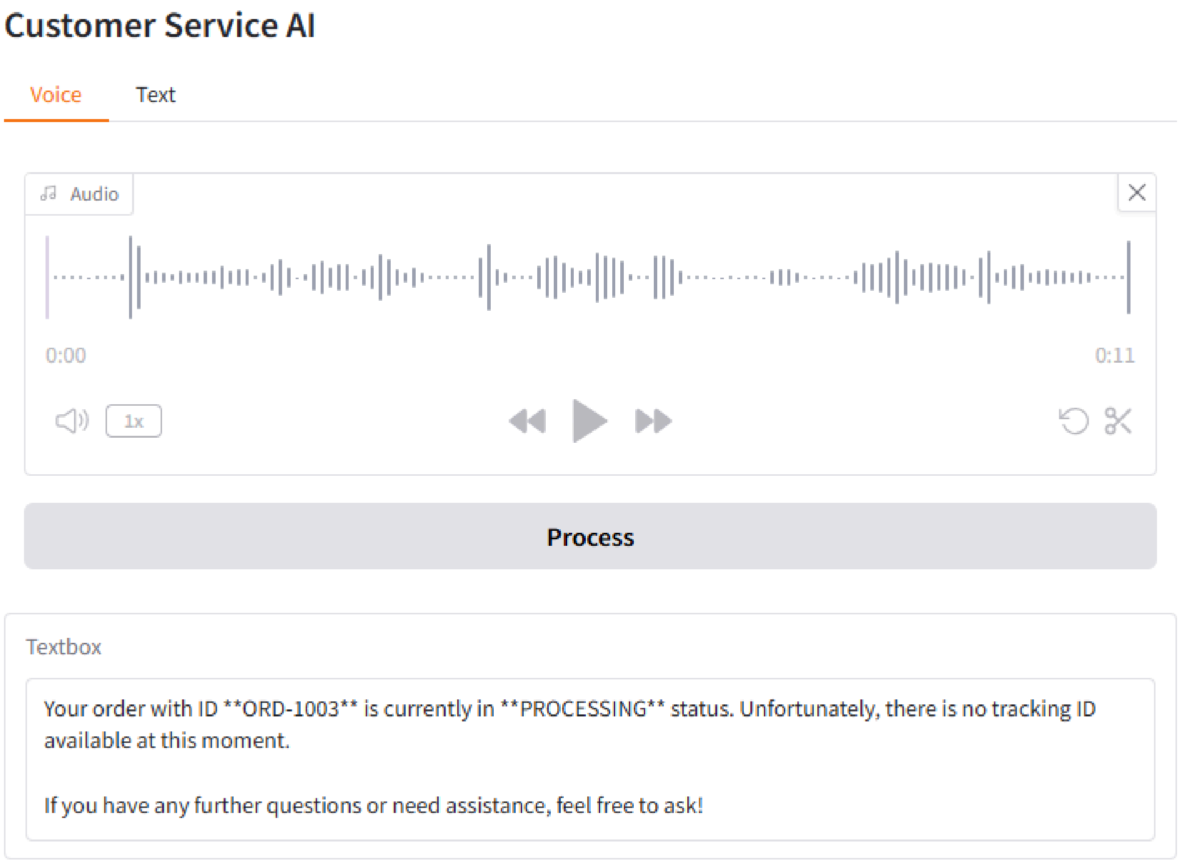
1. Documents loaded from policydoc.txt
2. Text split into chunks (500 chars, 100 overlap)
3. Embedded using text-embedding-3-small
4. FAISS index enables semantic search
5. RetrievalQA chain combines search + LLM synthesis

Return Request Processing



Screenshots of UI

1- Voice Enabled (Tool 1: Order_tracker)

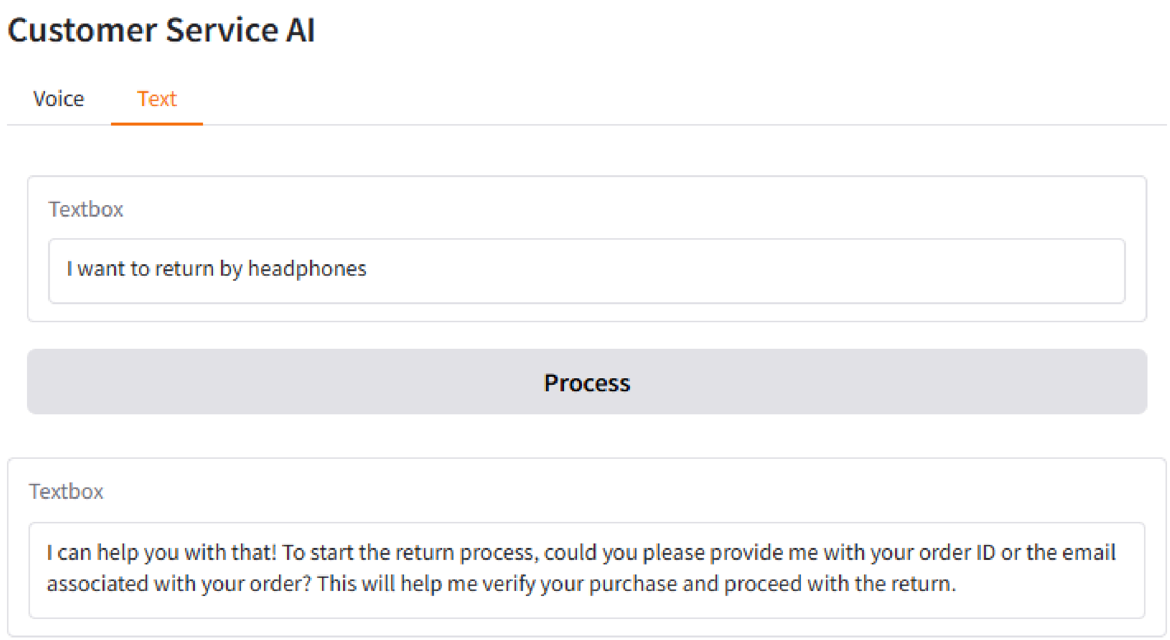


Your order with ID **ORD-1003** is currently in **PROCESSING** status. Unfortunately, there is no tracking ID available at this moment.

If you have any further questions or need assistance, feel free to ask!

2- Chat Enabled (Tool 2: Return Request Processor)

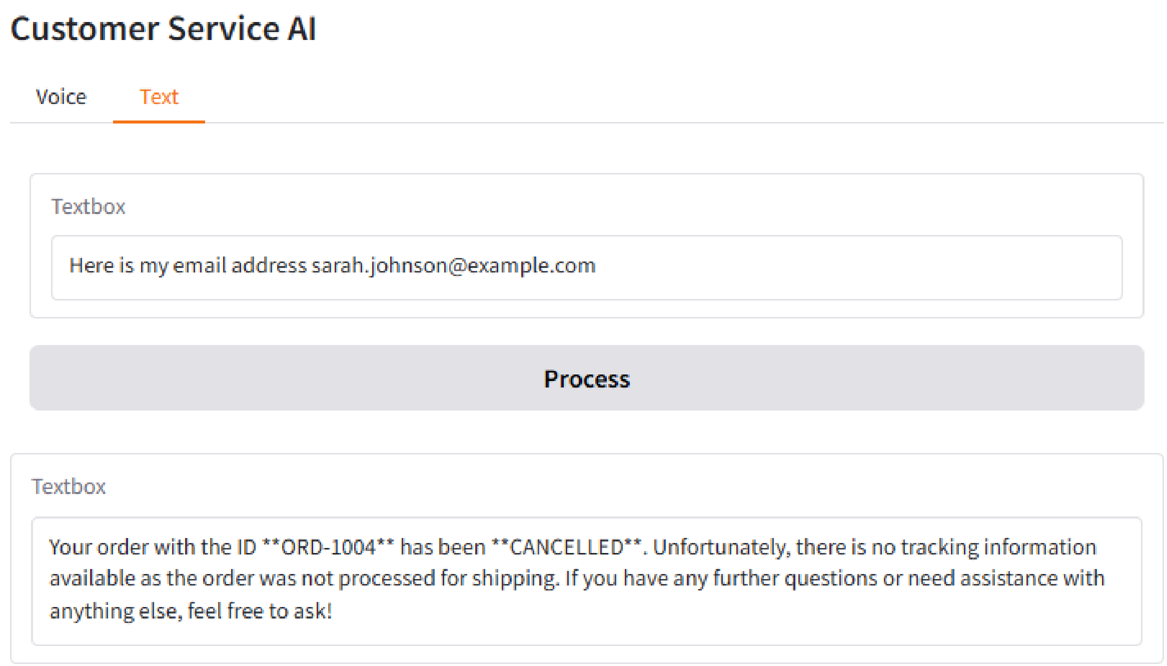
Stage 1: What do you want to return ?



I want to return by headphones

I can help you with that! To start the return process, could you please provide me with your order ID or the email associated with your order? This will help me verify your purchase and proceed with the return.

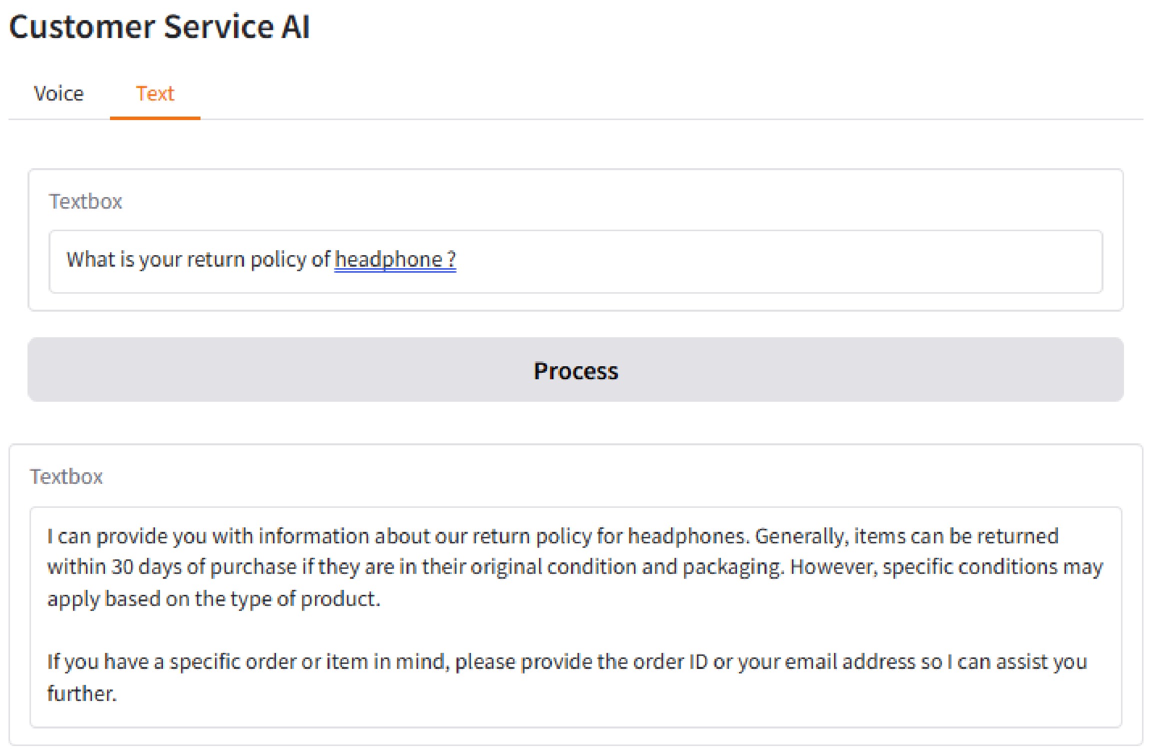
Stage 2:



Here is my email address sarah.johnson@example.com

Your order with the ID **ORD-1004** has been **CANCELLED**. Unfortunately, there is no tracking information available as the order was not processed for shipping. If you have any further questions or need assistance with anything else, feel free to ask!

Stage 3: Return Policy (RAG Utilization)



What is your return policy of headphone ?

I can provide you with information about our return policy for headphones. Generally, items can be returned within 30 days of purchase if they are in their original condition and packaging. However, specific conditions may apply based on the type of product.

If you have a specific order or item in mind, please provide the order ID or your email address so I can assist you further.

Tools and Technologies

Core Stack

1. Python 3.9+
2. Whisper (OpenAI) – Speech-to-text
3. LangChain – Agent framework
4. FAISS (Meta) – Vector similarity search
5. OpenAI Embeddings (`text-embedding-3-small`)
6. Gradio – Web UI

Data & Backend

7. JSON – Mock order database
8. FFmpeg – Audio processing

Libraries

9. python-dotenv – Environment variables
10. langchain-openai – OpenAI integrations
11. langchain-community – Tool integrations