YOI Media Chatbot-Documentation

Introduction

The **YOI Media AI Chatbot** is an intelligent, branded, website-integrated assistant designed to enhance user engagement, provide instant information about YOI Media's services, streamline appointment bookings, and offer a modern digital experience for visitors.

The chat bot is a combination of :

-React (frontend widget)

-FastAPI (backend API server)

-Groq LLaMA 3.1 (LLM engine)

-Custom RAG logic (knowledge base + few-shot examples)

-Fully branded UI with animations

The system is optimized for performance, accuracy, and brand consistency.

Objectives

The chatbot was designed to achieve the following:

✓ Improve user engagement

✓ Provide accurate answers about YOI Media's services

✓ Reduce manual query handling

✓ Offer guided options like booking appointments

✓ Maintain a premium, brand-consistent experience

✓ Deliver fast, reliable responses using Groq LLaMA

Frontend Architecture

Stack:

- React
- TypeScript
- CSS animations
- Fully standalone widget

Core Component: YOIChatBot.tsx

Responsibilities:

- •Render UI
- •Handle open/close
- •Send user messages
- •Show typing animation
- •Display booking UI
- •Manage conversation state
- •Auto-scroll management

Backend Architecture

Stack:

- •Python
- •FastAPI
- •Groq API

Main Files:

File	Purpose
main.py	API server, /chat endpoint
response_handler.py	LLM + KB + few-shot logic
knowledge_base.json	Verified company data
few_shots.json	Example question/answers
system_prompt.txt	Tone + behavior rules

Response Logic (Pipeline)

This is the core of accuracy

Step 1 → KB Lookup

If query matches any KB entry → return directly
Example:

- “Where is YOI Media located?”
- “What services do you provide?”

Step 2 → Few-shot Match

If query matches a predefined example → return mapped answer
Example:

- “Who founded YOI Media?”
- “What is YOI Media?”

Step 3 → LLM Response

Fallback:

- LLaMA 3.1 model
- Strict tone rules
- Uses embedded KB context
- Temperature = 0 for accuracy

Tone Guidelines

Tone is defined in system_prompt.txt.

The bot MUST**:**

- Be confident
- Be concise
- Avoid unnecessary marketing hype
- Not guess
- Offer help without pushing

Deployment

BACKEND:

- Host on VPS (Hostinger)
- Run using: uvicorn app.main:app --host 0.0.0.0 --port 8080

Maintenance Guide

To update bot knowledge : “ knowledge_base.json”

To correct or add example answers : “ few_shots.json”

To update visuals : “YOIChatStyles.css”

To update greeting or behavior: “system_prompt.txt”

To add a new feature: “YOIChatBot.tsx”

Troubleshooting

Issue	Cause	Solution
“Server error”	Backend offline	Restart FastAPI

Issue	Cause	Solution
No LLM responses	API key missing	Add GROQ_API_KEY
Wrong answers	KB missing data	Update knowledge_base.json
Messages not scrolling	useRef missing	Re-add auto-scroll hook
UI misaligned	CSS conflict	Reset YOIChatStyles.css

Cost Efficiency & Usage Analysis

LLM Provider Used Groq Cloud (LLaMA 3.1 Instant 8B)

“According to Groq’s Free Tier for the llama-3.1-8b-instant model supports up to **14,400 requests per day** or roughly **250,000 tokens per day** (Groq docs). As our chatbot uses ~100 tokens per interaction, this translates into ~2,500 interactions per day under conservative usage. The Free Tier is listed as **no expiry**, meaning YOI Media can run the chatbot continuously with **zero LLM cost**. If limits are exceeded, , the API returns a 429 error until the daily reset.”