Vivan Developer Documentation

1. Introduction

Vivan is a context-aware chatbot designed to help recruiters and users explore the resume, project experience, and background of Venkata Surya Teja Susarla. It uses a combination of Flask, LangChain, and OpenAI technologies to enable conversational queries over unstructured documents like HTML and DOCX files.

The project is optimized for deployment using Docker and Google Cloud Run, and includes a modern WebSocket-based frontend for real-time chat experience.

2. Key Features

- Real-time WebSocket communication using Flask-SocketIO and flask-sock
- Document ingestion from HTML and DOCX files
- LangChain-powered LLM query processing using a MapReduce pipeline
- Custom prompt templates to ensure personality, tone, and context
- Vector similarity search using OpenAI embeddings and FAISS
- Deployed to Google Cloud Run with Dockerized configuration
- UI available both as a native Flask page and via personal website widget

3. Architecture Overview

- User interacts via chat.html or embedded widget on suryasusarla.info
- Message sent to backend WebSocket endpoint (/ws/stomp)
- Flask receives and routes it to `get_answer()` function
- LangChain loads preprocessed document chunks from FAISS vectorstore
- Uses OpenAI LLM with map-reduce chain to formulate an accurate response
- Response is streamed back to frontend WebSocket for real-time display

4. Project File Structure

- app.py: Flask app with WebSocket/STOMP handling
- chat.html: Main frontend for live chat interaction
- langchain_pipeline.py: Document loading, chunking, embeddings, LangChain QA chain

- htmlfiles/: Folder with resume and HTML/DOCX source files
- requirements.txt: Python dependencies
- Dockerfile: Optimized container build with Gunicorn and Eventlet

5. Deployment (Cloud Run)

- Dockerfile exposes port 8080 (required by Cloud Run)
- Built using python:3.11-slim and installed dependencies
- Cloud Run service created via `gcloud run deploy` with flags:
  --source=. --region=us-central1 --platform=managed --allow-unauthenticated

6. Why LangChain and Not LlamaIndex?

LangChain gives deeper control over prompt templates, chaining logic, and map-reduce document aggregation. This allowed the chatbot to maintain professionalism, handle multi-doc summarization, and customize tone depending on content origin (resume, blog, HTML, etc.).

7. Live Links and Credit

- Live Demo: https://dev.vivan.chat
- Widget: https://suryasusarla.info (bottom right)
- Medium Article: https://medium.com/p/f67e8450a5b6
- Developer: Venkata Surya Teja Susarla