Likhith Bavisetti

314-393-2075 | sailikhithbavisetti@gmail.com | linkedin.com/likhithbavisetti | github.com/Likhith1030 | St Louis, MO

Professional Summary

Graduate Software Developer specializing in AI-powered systems, Retrieval-Augmented Generation (RAG), and agent-based architectures, with hands-on experience building end-to-end LLM applications using Python, FastAPI, and modern cloud platforms. Skilled in designing scalable data pipelines, integrating vector search and embeddings, and deploying production-ready solutions across AWS and GCP. Proven ability to develop real-world applications including voice-based AI assistants and domain-specific RAG systems, with a strong focus on performance, reliability, and practical impact.

Education

Master of Computer Science GPA: 3.93/4.0
University of Missouri — St. Louis *08/2024 — 05/2026*

Technical Skills

Languages: Python, JavaScript, SQL
Cloud & Infrastructure: AWS (Bedrock), GCP (Vertex Ai) , Docker, Terraform, Linux
AI & LLM Systems: Large Language Models (GPT-5 for reasoning and generation, GPT-4o-mini for lightweight parsing, Llama3 for data extraction pipelines, Gemini 1.5 Pro for multimodal/context-heavy tasks), Retrieval-Augmented Generation (RAG), Agent-Based AI Systems, Tool Calling, Prompt Engineering, Multi-Step Reasoning Workflows
AI Platforms & Frameworks: LangChain, CrewAI, OpenAI API, Vector Databases (ChromaDB, pgVector)
Backend & Systems: FastAPI, Data Pipelines
Engineering Practices: CI/CD, Git, Agile, Jira

Experience

Graduate Intern 01/2025 – Present
University of Missouri — St. Louis *St Louis, MO*

- Automated event data collection from **50+ community websites** using Crawl4AI package and LLama3 LLM, reducing a week of manual data gathering each month to a few hours including cleaning and processing.
- Built end-to-end data pipelines using Python, Pandas, and NumPy to clean, normalize, and structure scraped data for reliable downstream publishing and structured dataset creation.
- Utilized Unstructured API to parse multi-column PDF documents and built a processing pipeline to store structured data for downstream retrieval and analysis.
- Designed and deployed a RAG-based conversational agent trained on **UMSL Community Connect newsletters** using LangChain, ChromaDB, and OpenAI embeddings, OpenAI LLM (GPT-4o-mini), implementing retrieval workflows for context-grounded response generation to answer questions related to the content.

Projects

MediRag | *Python, FastAPI, PostgreSQL, Docker, LangChain, Vector Search (IVFFlat)*

- Built a Retrieval-Augmented Generation (RAG) application that provides estimated healthcare costs using the Medicare Physician Other Practitioners - by Geography and Service (US CMS dataset, ∼250K rows).
- Implemented NLP guardrails to extract state and disease from natural language input and map them to structured backend queries, using GPT-4o-mini for query parsing and GPT-5 for response generation.
- Designed a PostgreSQL-based system with IVFFlat indexing for scalable vector similarity search and generated embeddings to retrieve relevant cost data.
- Developed FastAPI endpoints, integrated LangChain for orchestration, and containerized the application using Docker for production readiness.

AI Hospital Call Assistant | *Python, GCP (Dialogflow CX, Vertex AI, Cloud Run, Firestore), Twilio Voice*

- Built a voice-based hospital assistant that handles incoming patient calls, detects intent from natural language speech, and manages real-time conversation flow using Dialogflow CX.
- Implemented backend tools on Cloud Run for doctor availability lookup and appointment booking, with structured data stored in Firestore to simulate real-time hospital scheduling.
- Integrated Vertex AI for intelligent fallback handling and contextual responses, enabling dynamic multi-turn conversations and improved understanding of user requests.
- Connected Twilio Voice API for telephony, enabling end-to-end call handling from speech input to automated voice responses using GCP speech services.