Anurag Jaiswal

AI Engineer

Sanepa-02, Lalitpur | anuragj614@gmail.com | +977 9742928650

linkedin.com/in/anurag-jaiswal1 | github.com/anuragj614

Overview

AI Engineer with hands-on experience building production-grade Agentic AI systems, RAG pipelines, and multi-agent orchestration using LLMs. Demonstrated ability to reduce manual workflows from hours to minutes through LLM-powered automation, with expertise in LangGraph, MCP, and human-in-the-loop system design. Passionate about building reliable, scalable AI systems that solve real-world problems.

Education

BE in Computer Engineering April 2021 – May 2025

Sagarmatha Engineering College, Sanepa-02, Lalitpur, Nepal

- **Percentage**: *78.41%*
- **Relevant Coursework:** Computer Architecture, Data Structures and Algorithms, Database Systems, Computer Network, Microprocessor, Artificial Intelligence

Experience

AI Application Developer, *Smart Contents Nepal – Kathmandu* June 2025 – Present

- Built and maintained a custom chatbot builder SaaS platform enabling clients to ingest their own data (PDFs, text, websites) and create custom AI chatbots embeddable on their websites as a widget.
- Refactored the embedding layer into a dedicated service to decouple heavy dependencies from a custom HuggingFace model, improving system efficiency and scalability through a dual-server design (one for chatbot interactions and another for embeddings).
- Developed an MCP server used by AI agents for frequently used tools, including an OCR tool with Qwen-OCR (a VisionLLM specializing in CJK languages) for complex non-selectable texts in Japanese scanned documents.
- Designed and developed an agentic document generation system for business subsidy workflows using LangGraph, featuring AsyncPostgresSaver checkpointers for fault-tolerant state persistence and a Human-in-the-Loop (HITL) interface via an integrated chatbot for user inputs and confirmations. Applied prompt engineering to improve output quality.
- Architected a multi-agent AI system featuring an orchestrator agent that intelligently routes user inputs and coordinates specialized sub-agents via tool calling, automating an end-to-end workflow and reducing manual processing time from 5–6 hours to under 10 minutes.

Personal Projects

InterviewAssist: AI-Based Job Interview System January 2026 - Ongoing

A revamped version of a college group project, rebuilt as an interview platform where users upload their CV and a target Job Description (JD). A RAG-powered agent evaluates candidate fit; eligible users proceed to an AI-driven interview while others receive CV improvement recommendations.

- Built a RAG agent that performs a CV–JD fit analysis by retrieving and comparing relevant content from the uploaded documents, determining candidate eligibility before proceeding to the interview stage.
- Implemented an interview agent that generates role-specific questions targeting skill gaps and mismatches identified between the candidate's CV and the JD, enabling a focused and adaptive interview experience.
- Designed a feedback loop for ineligible candidates, providing structured CV improvement suggestions based on the gap analysis and allowing re-evaluation after updates.
- Architected the system using a containerized service stack with Docker, leveraging Qdrant for semantic document retrieval and Redis for session and state management across multi-turn interactions.
- **Tools used**: FastAPI, OpenAI model, LangChain, LangGraph, Qdrant, Redis, PostgreSQL, Docker.

Conversational Agent February 2026 - March 2026
GitHub

A conversational AI system that enables users to query ingested documents through a ReAct-based agent with persistent multi-turn memory, built on a semantic search pipeline powered by pgvector and LangGraph.

- Built a conversational Agentic RAG system using LangGraph's ReAct loop, enabling multi-turn document Q&A and automated interview booking with persistent session memory backed by Redis.
- Implemented a FastAPI backend with pgvector for semantic document search, supporting dual chunking strategies (Recursive & Semantic) and dual embedding options (local sentence-transformers & OpenAI).
- **Tools used**: FastAPI, OpenAI model, Local Embedding model(all-MiniLM-L12-v2) from Hugging face, Langchain, Langgraph, Docker, PostgreSQL, Redis.

MCP Server February 2026
GitHub

A lightweight Model Context Protocol (MCP) server exposing tools for AI agents, deployed on Render and compatible with Claude Desktop and MCP Inspector for seamless agent integration.

- Built and deployed an MCP server using FastMCP, exposing web search capabilities via Serper API as callable tools for AI agents.
- Integrated the server with Claude Desktop and MCP Inspector, enabling plug-and-play tool usage across AI workflows with a live deployment on Render.
- **Tools used**: FastMCP, Docker, Serper API, Render.

Pawdoption Nov 2023 - Jan 2024
GitHub

- Developed the back-end of a pet adoption platform using Django Rest Framework for RESTful APIs.
- Implemented user authentication and secured endpoints using Django's built-in authentication system.
- **Tools used**: Django, Django Rest Framework, and MySQL.

Technical Skills

Programming Language: Python

Frameworks & Tools: FastAPI, Django Rest Framework, LangChain, LangGraph, FastMCP, RAG Pipelines, Ollama, PostgreSQL, SQLAlchemy, Alembic, Redis, Docker, Git, CI/CD Pipelines.

Soft Skills: Teamwork, Communication, Problem solving