Ankit Ranjan Das

+91 9572523758 | ankitdas3758@gmail.com | LinkedIn | GitHub | Portfolio

Education

Vellore Institute of Technology (VIT) — Bhopal, India
Bachelor of Technology in Computer Science — *2022 – Present*

Technical Skills

Languages: Python, C++, TypeScript, JavaScript, SQL

AI & Machine Learning: RAG Pipelines, LangGraph, Vector Databases, LLMs, Embedding Models, Hybrid Search, Cross-Encoder Reranking

Frameworks & Backend: Next.js (App Router), React, FastAPI, Node.js, Flask, REST APIs, WebSockets, Async Processing

Infrastructure & Data: PostgreSQL, pgvector, Redis, Celery, Docker, Supabase, Git

Experience

Tata Steel — **Web Developer Intern**

EdCare | *Predictive Health Diagnostics Engine*

- Developed a real-time predictive diagnostics backend using **Python (Flask)** and **Scikit-Learn**, deploying a **Gradient Boosting classifier** to analyze biological markers and deliver patient risk predictions in **under 100 ms inference latency**.
- Designed a secure **PostgreSQL data architecture** with **Row-Level Security (RLS)** and optimized **compound SQL indexes**, improving analytical query performance by ∼**40%** while enforcing strict patient **data validation pipelines**.

Projects

OmniSearch | ***Autonomous Multi-Agent Research Platform*** — Code

Next.js 14, LangGraph, FastAPI, PostgreSQL (pgvector), Redis, Celery, Docker Compose

- Architected an autonomous multi-agent research platform using **Next.js, LangGraph, and FastAPI**, orchestrating a **5-stage agent workflow (Planning → Crawling → Indexing → Retrieval → Publishing)** that aggregates insights from **20+ live web sources per query** to generate structured long-form research reports up to ∼**32k tokens**.
- Engineered an enterprise-grade hybrid RAG pipeline using **PostgreSQL pgvector (HNSW indexing) + BM25 retrieval + Reciprocal Rank Fusion**, improving semantic context precision by ∼**35%** and isolating the top-15 most relevant document chunks via a **cross-encoder reranking layer**.
- Designed a **distributed ingestion architecture with Celery and Redis workers**, containerizing **7 microservices with Docker Compose** to process JavaScript-rendered pages concurrently and maintain **sub-second WebSocket telemetry streaming** under multi-source workloads.

GitMetrix | ***AI Developer Intelligence Platform & Repository Analytics*** — Live Demo | Code

Next.js 16, Supabase, pgvector, Inngest, MoE LLM Router, React Flow, Redis

- Built an **AI-driven repository intelligence platform** using **Next.js, Supabase pgvector, and a Mixture-of-Experts LLM router**, enabling semantic Q&A across codebases by combining vector search with **PostgreSQL full-text retrieval** and reranking results via **Cohere cross-encoder models**.
- Developed an **AST-aware distributed indexing pipeline** using **Inngest background workflows**, capable of analyzing repositories containing **1,500+ files per job** with **15 concurrent GitHub fetch streams and Redis-based caching** to reduce redundant processing latency.
- Implemented an **interactive architecture visualization dashboard with React Flow**, mapping repository dependency graphs using **BFS/DFS traversal algorithms** and computing **cyclomatic complexity metrics** to enable **AI-assisted code reviews and structural code analysis**.

DocuMind | ***Intelligent Document Analysis Platform*** — Live Demo | Code

Next.js 16, RAG, Stripe, Inngest Queues, BM25

- Engineered an **AI document analysis system** combining vector embeddings and **BM25 lexical search**, integrating **Reciprocal Rank Fusion (RRF)** to improve retrieval accuracy by ∼**40%** compared to vector-only pipelines.
- Built an **asynchronous document ingestion pipeline with Inngest queues**, enabling concurrent processing of large document sets and supporting **thousands of PDF uploads per day** while maintaining consistent semantic indexing performance.
- Optimized context retrieval using **hybrid search ranking and token-efficient chunking strategies**, reducing **LLM token consumption by ∼60%** while improving answer relevance across long multi-document queries.