Vansh Jangir

+91 9649542580 | vanshjangir0001@gmail.com | Github | LinkedIn | Website

Education

IIIT Naya Raipur Raipur, Chhattisgarh
B.Tech. in Data Science and Artificial Intelligence, CGPA: 8.43/10 2022 – 2026

Coursework

• Operating Systems	• AI and ML	• Data Structures	• DBMS
• Reinforcement Learning	• Deep Learning	• Distributed Systems	• Computer Vision

Experience

Karya - SWE Intern | TypeScript, React, Python, FastApi, Redis, Postgres Feb 2025 - Present

- Developed NLQ to translate natural language prompts into SQL using database schema, increasing report generation efficiency by 2x.
- Used **Celery** and **Redis** to asynchronously execute queries and implemented Role-Based Access Control (**RBAC**) with 6 distinct roles and column level control to enforce secure and restricted access to database resources.
- Created VoIP based Audio chat platform for collecting high quality audio datasets at 48Khz.

ContractKen - SDE Intern | Python, Flask, Redis, React, TypeScript August 2024 - November 2024

- Engineered 5+ advanced formatting tools, enhancing user interaction and navigation for legal documents. Increased document review speed 30% across the platform.
- Facilitated the adoption of formatting tools by over 500 users, streamlining the contract drafting process. Reduced manual document corrections by 40% in document corrections.

Projects

Xdb: Database and storage engine in Go | github

- Programmed an **ACID-compliant** database and storage engine in **Go**, processing over 5000 transactions/second. Integrated concurrency control, ensuring data consistency across multiple transactions.
- Constructed Disk-based **B+tree** for storage, with **Copy-on-Write** mechanism reducing I/O overhead by 30%. Improved disk utilization, allowing more efficient memory management.
- Attained **O(logn)** query time for point lookups, and **O(k*logn)** for range queries. Streamlined indexing structure to support high-performance query execution at scale.

Rapid Go: An online Go game playing server | website github

- Built a real time platform for playing Go against other players and different bots (gnugo, 5-7 Kyu), with a responsive frontend using **ReactJS**, with **Go** backend, **Postgres** database.
- Distributed websocket and http server and used **Redis Pubsub** as message broker which led to incresed scalability, handling 10,000+ concurrent users.

Load Balancer | github

- Designed a load balancer in **Rust** using libc's socket api, which can handle **50k+** connections per second.
- Leveraged thread library for **multi-threading** and **Epoll** (for Linux) which reduced CPU usage by **50%**, enhancing system stability under high traffic loads and ensuring consistent performance.

Go routines in C | github

- Implemented Go style preemptive function scheduling in **C** and inline **x86 Assmebly**, using custom stacks.

Achievements

Hackathons: Hack-O-Harbour - AIML track winner (4th overall) among 40+ teams, held at IIIT Naya Raipur.

Technical Skills

Languages: C, C++, TypeScript, Python, Go, Rust, SQL.
Frameworks: ReactJS, ExpressJS, NextJS, Gin, Go-chi, Flask, HTML, CSS
Tools: GNU/Linux, Git, Docker, AWS, MongoDB, Make, CMake, Vim/Neovim, POSIX threads, Socket Programming.