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: 8.5 — *2022 – 2026*

Dundlod VidyaPeeth — Jhunjhunu, Rajasthan
CBSE 12th - 95.00% — *2015 – 2022*

Experience

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

- Developing NLQ, which translates natural language prompts to SQL query based on db schema and execute it.
- Using **Celery** to asynchronously execute query, **Redis** to cache results and SSE to stream intermediate responses.
- Implementing Role-Based Access Control (RBAC) to enforce secure and restricted access to database resources.

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: An ACID compliant database | github

- Devised 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

- Architected a real time platform for playging Go against other players and different bots.
- Crafted a responsive frontend using **ReactJS**, with **Go** and **Gin** for the backend, **Postgres** database, and **Docker** for containerization.
- Implemented **WebSocket** based connections for faster move and abort notifications along with real time chat.

Load Balancer | github

- Designed a load balancer in **Rust** using libc's socket api, which can handle **50k+** connections per second, with distributed resource allocation across threads, enhancing system responsiveness.
- Leveraged thread library for **multi-threading** and **epoll** for event driven architecture in the system. Multi-threading integration reduced latency by **4x** (no of cores).
- Implemented Epoll (for Linux) and reduced CPU usage by **50%**. Enhanced system stability under high traffic loads, ensuring consistent performance.

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.
Courses: Operating Systems, DBMS, AI/ML, Deep Learning, Reinforcement Learning, Data Structures, Distributed Systems.