

Adarsh Jha

☎ +91-8130902767 | ✉ adarshjha.dev1@gmail.com | 🌐 AdarshJha-i | 🐙 AdarshJha-1 | 📄 LeetCode/xFoo1

PROJECTS

Nakama — Anime Social Media Platform | Next.js, TypeScript, PostgreSQL, Drizzle ORM



- Built a full-stack social media platform using **Next.js**, **TypeScript**, **Drizzle ORM**, **PostgreSQL**, **React Query**, and **shadcn/ui** supporting posts, comments, likes, bookmarks, follows, notifications, profiles, and search.
- Implemented **Google OAuth** authentication using **Better Auth**, media uploads with **UploadThing**, cursor-based infinite scrolling, and optimistic UI updates for a responsive user experience.
- Designed REST APIs with Next.js Route Handlers and automated orphan media cleanup using **Vercel Cron Jobs**, improving storage management and maintainability.

Forge — Distributed Video Processing Service | Go, Redis, SQLite, FFmpeg CLI



- Built an **asynchronous video processing service** using a **producer-consumer architecture** to decouple uploads from transcoding jobs.
- Implemented **REST APIs** for video uploads, processing status, transcoded video delivery, and thumbnail retrieval.
- Designed a **Redis-backed job queue** with concurrent workers to process transcoding jobs independently from the API server.
- Integrated the **FFmpeg CLI** to generate **1080p, 720p, 480p, and 360p** video variants while preserving audio and generating thumbnails.
- Stored video metadata in **SQLite** and managed local storage for original and processed media.

Vault — In-Memory Key-Value Store | Go, Protocol Buffers



- Built a concurrent in-memory key-value store in **Go** with a custom TCP server supporting multiple client connections using goroutines and **sync.RWMutex**.
- Implemented a custom **LRU cache** using a hash map and doubly linked list for **O(1)** GET, SET, DELETE, and eviction operations.
- Designed a persistent **Write-Ahead Log (WAL)** using Protocol Buffers with CRC32 validation, segmented log rotation, graceful shutdown, and crash recovery.

Tempest — Distributed Load Testing Service | Go, Redis, SQLite



- Built a **distributed load testing service** in **Go** using a **worker-based architecture**, **Redis-backed job queues**, and **SQLite**.
- Designed a **YAML-driven execution engine** supporting configurable concurrency, weighted request scenarios, HTTP methods, headers, query parameters, and request bodies to simulate realistic workloads.
- Implemented **concurrent virtual users** using goroutines to generate thousands of HTTP requests while collecting latency, throughput, response codes, and network error metrics.
- Separated the **API server** from execution workers through asynchronous job queues, enabling independent execution and horizontal scaling.

EDUCATION

Indira Gandhi National Open University

Bachelor of Computer Applications

Jul. 2022 – 2026

New Delhi, India

TECHNICAL SKILLS

- **Languages:** Go, C, C++, JavaScript, TypeScript, SQL
- **Frameworks & Libraries:** React, Next.js, HTMX, React Query, Drizzle ORM, Better Auth
- **Backend & Databases:** REST APIs, Protocol Buffers, PostgreSQL, Redis, SQLite, MySQL, MongoDB
- **Tools:** Docker, Git, Linux, AWS (EC2, S3), Vercel, Postman

CODING PROFILE

- Solved 600+ Data Structures and Algorithms problems across **LeetCode** and **GeeksforGeeks**.