

Aryan Kathawale

[linkedin.com/in/kiritocode1](https://www.linkedin.com/in/kiritocode1) | github.com/kiritocode1

TECHNICAL SKILLS

Languages: Typescript, Javascript, HTML/CSS, Go, C, Python, SQL, Gleam, Java.

Frameworks: Next.js, Jest/Puppeteer, Angular, Vue, Jquery, React Native, Expo, Nest.js, Fastify, Hono, Express, Spring Boot.

Developer Tools: Git, Node.js, Tailwind CSS, Figma, Deno, Bun, Docker, Rabbit MQ, Kafka, GCP, AWS, Cloudflare, Jira, LLMs, Gradle.

Libraries: React, Tensorflow, Pytorch, Redux, GraphQL, HTMX, Lodash, Go-Chi, .

Database: PostgreSQL, Prisma, Drizzle Non-Relational: MongoDB, Redis, Sqlite, Geo Location Database.

Focus on : Front-end Design, UI / UX, Animation, Refactoring, Optimization, Complexity Compression, System Architecture.

EDUCATION

Sinhgad College Of Engineering

Bachelor of Engineering in Computer Science, Major in Machine Learning, 7.64 GPA

Pune, Maharashtra

Jan. 2021 – Aug 2024

EXPERIENCE

Fullstack Embedding Engineer

IotRides

Dec 2024 – Current

Canada (remote)

- Implementing Our own Inhouse GPS tracking system with custom protocol and architecture , using **Open source street maps data, Geographical Db, Java , Docker and python**, resulting in an Efficient System architecture with **10x returns**. Supervising the construction of custom **LoRaWAN Protocol**
- Working on Refactoring UI developer in **Vite , React and Redux** challenged in restructuring and simplifying the UI to better suit the need of the backend's architecture

Software Engineer II

Born Agile AI.

Jan 2024 – June 2024

Hungary (remote)

- Led the development team of 3 and implementation of the Proof of Concept and engineered ML models utilizing **Langchain, Vector Db, and Next.js**, resulting in a scalable architecture with **10x returns**. Supervising the construction of custom **RAG and Streaming API**.
- Set up authentication via OAuth and email pipeline, Streamlined and 40% faster. Used Next-Auth, React-email, and Nodemailer for SMTP requests. **Saving \$1200 per month**.

Backend engineer

Yottalligent

May 2022 – Oct 2022

Mumbai, India

- Refactored code from **GraphQL to TRPC** in production, optimized JavaScript with promises, and improved microservice compile times, rewriting some in Go by 25%. Deployed on Serverless for enhanced efficiency.
- We successfully deployed the API code to **Cloudflare workers**, resulting in a **400 % cost reduction** using the Wrangler CLI. Moving to an **Event Driven Design** Pattern instead of a Monolith.

PROJECTS

Vscode Animation Extension | Webpack, Typescript, Sass, Node, Vsx, Apc Custom UI

Jan 2025 – Jan 2025

- Implemented A vscode Extension that Brings Smooth Animations to Vscode.
- Leveraged Apc Custom UI++ library to inject custom css into vscode .
- Demonstration of Webpack Orchestration skills and software quality,.

URL shortner in deno | Deno, OAuth, Redis, Sqlite, Docker, Preact, Bash, Hono.js

Jul 2024 – Sept 2024

- Developed a Url Shortner without using any external dependencies and using the Native Deno Apis.
- Leveraged Sqlite as a KV data store and used it to store the OAuth and User Credentials.
- Rewrote react rendering logic to generate static site generation, coupling the backend tightly with frontend.
- Installed no npm modules for react , database Orm or for authentication. Making the project platform agnostic.

GIGI : Git implementation in go | *GoLang, MakeFile, Charm CLI Library*

Jan 2024 – Mar 2025

- Implemented a Software Version Management system similar to git
- Had to work on compilation problems as well as system design problems suited for different platforms like linux , windows and mac.
- Demonstration of Go and Low level System Design.

Drawy | *Next, Typescript, LLM, Streaming, Excalidraw API, Fal.AI*

Sept 2024 – Nov 2024

- Developed a real-time image-generating application that uses rough sketches by the user to generate accurate images based on the user's requirements.
- Used Fal.AI and Excalidraw to stream in images based on real-time sketches or illustrations.
- Innovative research and experimentation

Zebra Crossing | *Deno , Typescript, JSR , Hono.js*

Jan 2024 – Jul 2024

- Rate Limit Written in Typescript for Hono.JS
- Deployed on JSR and is a middleware to handle Rate Limiting.