

Krushiraj Tula

Senior Full-Stack Engineer | Backend, Distributed Systems & Applied AI

Backend, Distributed Systems & Applied AI · Open To Offers

Hyderabad, India (Remote, UTC+5:30) · krushiraj.github.io · github.com/krushiraj · linkedin.com/in/krushiraj

SKILLS

TypeScript, JavaScript, Node.js, Python, Go, Rust, React, Next.js, Tailwind CSS, shadcn/ui, TanStack Query, Zustand, Vue.js, Semantic HTML, WCAG accessibility, Microservices, REST APIs, GraphQL, Express.js, Hono, Nest.js, Django, MongoDB, Elasticsearch, PostgreSQL, SQLite, Redis, ClickHouse, Vector search, Semantic search, Embeddings, Indexing, Query optimization, MCP, Multi-agent orchestration, NLP, LLM integration, Docker, Kubernetes, CI/CD

ABOUT

Senior full-stack engineer (7+ years) and sole developer of a 230M+ document platform owned end to end, from React/Next.js front-ends to Node/TypeScript and Python services and the distributed systems behind them. Deep on backend, APIs, and reliability, with real front-end craft and a daily habit of building with and for AI (MCP servers, multi-agent orchestration, agents on real codebases). Active open-source author (78 public repos, including Homebrew taps and a GitHub CLI extension). High agency, extreme ownership, and comfortable shipping complex production features independently.

EXPERIENCE

Full-Stack Engineer · Paperpile

Jan 2020 – Present

Remote

- Sole developer of an internal academic search platform with 230M+ documents, owning evaluation, design, deployment, and production support.
- Increased verified-abstract coverage from 27% to 51% by building crawler/scrapper engines (Apify, Firecrawl, custom) over 500+ public sources.
- Engineered an author-name parser (Stanford NLP + custom trie) for identity resolution, reaching 87% accuracy vs 45–58% for open-source parsers.
- Reduced p99 search latency from ~10s to <1s by redesigning indexing and query paths across Elasticsearch, MongoDB, and SQLite.
- Designed and maintained 10+ microservices with REST APIs, third-party integrations (Google Drive, OneDrive, S3), and RPC pipelines handling millions of dynamic-payload messages.
- Scaled ingestion and data-processing pipelines to 300GB+ of text monthly, keeping indexing throughput ahead of corpus growth.
- Shipped an MCP server over search/match APIs so LLM tools can verify citations against the corpus; cut production error rate from 10% to 2% via CI/CD on Docker and Kubernetes.
- Environment: TypeScript, Node.js, Python, React, Next.js, MongoDB, Elasticsearch, PostgreSQL, Redis, ClickHouse, S3, Docker, Kubernetes, ELK.

Software Engineer · NCR Corporation

Jul 2019 – Dec 2019

Hyderabad, India

- Cut ad load latency from ~2 minutes to <30 seconds on low-memory ATM hardware.
- Owned the Content Management and Advertising module and root-caused a production defect blocking a major bank chain's ads.

Software Developer Intern · ServiceNow

Jan 2019 – Jun 2019

Hyderabad, India

- Contributed to a proprietary internal frontend framework.
- Enhanced chatbot and activity-data collection systems, including improvements to a patented activity-monitoring system.

Full-Stack Engineer · Paperpile

Jan 2020 – Present

Remote · full time

- Sole developer of the internal academic search platform, a continuously growing 230M+ document corpus in MongoDB, owned end to end from ingestion to query.
- Reduced p99 search latency from ~10s to under 1s by redesigning indexing and query paths across Elasticsearch, MongoDB, and SQLite.
- Designed and maintained 10+ microservices handling millions of RPC messages with strict retries, ordering, and idempotency.
- Scaled ingestion pipelines to 300GB+ of text monthly and raised abstract coverage from 27% to 51% via Apify/Firecrawl/custom crawlers and improved parsing.
- Built a Stanford NLP + custom-trie author-name parser reaching 87% accuracy vs 45–58% for open-source baselines on a labeled set.
- Normalized external imports (XMP, BibTeX, RIS, Zotero) into a unified schema for reliable data migration.
- Shipped an MCP server over search/match APIs so LLM tools can verify citations against the corpus.
- Cut crawler update cycle from ~2 weeks to ~2 days with AI plugins and a test-gated harness.
- Drove CI/CD (Docker, Kubernetes, Jenkins, PM2), ELK observability, deployments/rollbacks, incident response.
- Environment: TypeScript, Node.js, Python, MongoDB, Elasticsearch, PostgreSQL, SQLite, ClickHouse, React, Next.js, GraphQL, Docker, Kubernetes, ELK, MCP.

Software Engineer · NCR Corporation

Jul 2019 – Dec 2019

Hyderabad, India · full time

- Cut ad initial-load latency from ~2 minutes to under 30 seconds on constrained, low-memory ATM hardware.
- Owned the Content Management and Advertising module; root-caused a production defect blocking a major bank chain's ads.

Software Developer Intern · ServiceNow

Jan 2019 – Jun 2019

Hyderabad, India · internship

- Contributed to a proprietary internal front-end framework; fixed prop-handling issues and designed a reusable activity-history component adopted across flows.
- Enhanced chatbot and activity-data collection systems.

PROJECTS

Pepe NFT

pepenfts.vercel.app

Solidity · Next.js · Ethereum · ERC-20 · Wallet integration · Generative art

Solo-built dividend-sharing NFT project: wrote and deployed the Solidity smart contract (burn + dividend mechanics reimaged for NFTs), created the generative art, and built the Next.js mint site accepting ETH and ERC-20 PEPE tokens with wallet integration, end to end.

dSynths

x.com/dsynths

Redis · DeFi · Caching · On-chain data

Freelance performance work for a DeFi synthetic-assets trading platform: reduced average chart render time from ~1s to ~10ms using Redis caching and improved storage of historical on-chain and trade data.

Alphaboard

Next.js · PostgreSQL · Cron workers · Webhooks · Scrapers · APIs

Co-built a crypto market-intelligence prototype ingesting Twitter, ~10 news APIs, and custom scrapers via cron workers and webhooks into PostgreSQL to surface market signals; led the ingestion side using Next.js edge APIs.

posix.co.in

posix.co.in

Next.js · TypeScript · Hono · PostgreSQL · Authentication

Production platform for local retailers (Next.js App Router, TypeScript, Hono, PostgreSQL, auth), live with 4 partner businesses; designed, built, and operated solo.

Royal Lands

royal-lands.up.railway.app

Go · WebSockets · SQLite · PixiJS · TypeScript · Railway

Live online multiplayer tile-placement board game (2–4 players). Server-authoritative Go backend over WebSockets with one goroutine owning each room's state, event-sourced SQLite persistence with deterministic replay to survive redeloys, and a PixiJS/TypeScript client; shipped as a single container on Railway.

gist-pdf

github.com/krushiraj/gist-pdf

Rust · Go · TypeScript · Homebrew · APT · Cargo · Docker

Open-source CLI and web tool converting GitHub Gists to styled PDFs. Implemented in Rust/Go/TypeScript, packaged via Homebrew tap, APT, Cargo, Docker, and prebuilt binaries.

Multi-agent content harness

Agents · LLM · Evaluation harness

Autonomous multi-agent pipeline that turns a topic into a finished short-form video with tool use and a sub-agent evaluation gate that iterates until output meets a quality bar, then routes each scene to the right renderer; cut production time ~10x.

dSynths

x.com/dsynths

DeFi · Caching · Time-series data · Redis

Freelance performance work for a DeFi synthetic-assets trading platform: reduced average price-chart render time from ~1s to ~10ms via caching and rebuilt storage and retrieval of historical time-series trade data.

Alphaboard

Next.js · PostgreSQL · Cron · Webhooks · Scrapers · APIs

Cron/RSS/webhook pipeline ingesting Twitter, ~10 news APIs, and custom scrapers into PostgreSQL to surface market signals; Next.js + edge API.

trade-ml

Python · ML · Sentiment

Independent research project: Python ensemble model on news and technical indicators for swing trades; technicals dominate normal regimes, news/sentiment during sudden drops.

Pepe NFT

pepenfts.vercel.app

Solidity · Smart contracts · Generative art · Next.js

Independently wrote and deployed a Solidity smart contract with burn and dividend mechanics, created the generative art, and built the web front end end to end.

posix.co.in

posix.co.in

Next.js · Tailwind CSS · shadcn/ui · Hono · PostgreSQL · Railway

Production full-stack SaaS for local retail, pharmacy, and wholesale sellers. Next.js, Tailwind, shadcn/ui, Hono, and Postgres; deployed on Railway; live with 4 partner businesses.

mini-terminal

krushiraj.github.io/mini-terminal

Vue.js · TypeScript

Browser-based terminal UI built in Vue.js; live on GitHub Pages.