DARSHAN VAGHELA

Full Stack Developer · React · Node.js · TypeScript · AWS · 5+ Years

Ahmedabad, Gujarat, India (Open to Remote) • +91 84690 61320 • vdarshan613@gmail.com

• linkedin.com/in/darshan-vaghela-32752b1aa • github.com/DarshanVaghela284

PROFILE SUMMARY

- Full Stack Developer with **5+ years** of experience designing and shipping production web applications using **React.js, Next.js, Node.js, Nest.js, TypeScript, PostgreSQL, Docker, and AWS**. Proven end-to-end ownership for frontend UI, backend APIs, databases, integrations, CI/CD, and cloud deployments.
- Delivered enterprise full-stack platforms across healthcare AR, governance / proxy voting, and loyalty / rewards including microservices, event-driven pipelines, multi-tenant dashboards, and secure auth (JWT, RBAC, SAML).
- Proven track record of improving application **performance, scalability, and reliability** through API optimization, database indexing, build optimization, automated workflows, and production-focused engineering practices.

TECHNICAL SKILLS

Frontend	React.js, Next.js, TypeScript, JavaScript (ES6+), HTML5, CSS3, Tailwind CSS, Material UI, Ant Design, Chakra UI, Vite, Axios, TanStack Query, Redux, Zustand
Backend	Node.js, Nest.js, Express.js, REST APIs, GraphQL, Microservices, JWT, RBAC, Prisma ORM, TypeORM, Sequelize
Database	PostgreSQL, MongoDB, MySQL, Redis, Snowflake
Cloud / DevOps	AWS (S3, Lambda, SQS, Step Functions, Secrets Manager, ECS), Docker, Terraform, Kubernetes, CI/CD, GitHub Actions, GitLab CI
AI / LLM	Prompt Engineering, RAG Concepts
Testing	Jest, Vitest, Cypress, React Testing Library
Tools	Git, GitHub, Bitbucket, JIRA, Slack, Postman, VS Code
AI Tools	Cursor, Claude Code, GitHub Copilot, Codex

WORK EXPERIENCE

Full Stack Developer

April 2021 – Present

ApexSutra | Ahmedabad, India

Responsibilities

- Built and maintained production full-stack applications using React.js, Next.js, Node.js, Nest.js, TypeScript, and PostgreSQL.
- Owned end-to-end delivery of enterprise platforms — from requirement discussion and architecture decisions through development, testing, deployment, and production support.
- Designed scalable API and data flows (REST, event-driven sync, multi-tenant dashboards) to support growing users, roles, and third-party integrations.
- Developed responsive UIs and reusable components; designed REST APIs with JWT / SAML auth and role-based access control.
- Handled Docker deployments, CI/CD pipelines, and AWS services (S3, Lambda, SQS, Secrets Manager) for reliable cloud releases.
- Collaborated with product and design in Agile/Scrum; reviewed code and delivered features on schedule.

Key Achievements

- Reduced frontend build time by **50%** by migrating CRA to Vite, speeding up local development and production builds.
- Improved API response times by **70%** through query optimization and indexing on high-traffic endpoints.
- Reduced production defects by **50%** through stronger validation, error handling, and API reliability improvements.

- Delivered end-to-end enterprise applications, contributing across requirements, development, testing, deployment, and production support.
- Designed and built scalable REST APIs and event-driven integrations for multi-tenant platforms and third-party systems.

PROJECTS

Paystack — Automated AR Management System

React.js · TypeScript · Vite · Node.js · Express.js · PostgreSQL · Sequelize · TanStack Query · Zustand · AWS Lambda · SQS · Step Functions · ECS Fargate · AWS S3 · Docker · Terraform · GitLab CI

- Built a multi-tenant accounts receivable platform for dental offices to manage statements, invoices, payments, and claim status across admin, partner, and dental portals.
- Designed and implemented a React.js / TypeScript frontend with Vite, TanStack Query, and Zustand, backed by Node.js / Express.js APIs on PostgreSQL (Sequelize).
- Built async dental-record and payment sync using AWS Lambda, Step Functions, SQS, and an ECS Fargate worker with Docker, Terraform, and GitLab CI.
- Delivered role-based payment and statement flows (card, Apple Pay, bank) to improve collection visibility and reduce support handoffs between teams.
- Automated recurring AR follow-ups by integrating statement generation and payment workflows, reducing manual billing effort.

RESO — Governance Meeting and Proxy Voting Platform

React.js · TypeScript · Vite · Redux · Node.js · Express.js · TypeORM · PostgreSQL · Snowflake · AWS S3 · Secrets Manager · CASL · Azure AD · SAML · Docker · Vitest

- Governance and proxy-voting platform for investment teams to manage meetings, voting instructions, and shareholding checks before ISS cutoffs.
- Designed dual manual and electronic submission paths for voting instructions, custodians, and shareholding verification under eligibility and cutoff rules.
- Automated ISS electronic vote-back — ballot ingest, vote submission, and status reconciliation — with due-date sync and support tickets for uninstructed ballots.
- Integrated Snowflake holdings sync and AWS S3 storage for ISS notices and proxy PDFs, plus an ML service to extract notice data and match proxies to resolutions.

Tronic — Loyalty, Rewards and Wallet Platform

Next.js · TypeScript · Nest.js · Prisma · PostgreSQL · Kafka · Temporal · Docker · Kubernetes · Auth0 · TanStack Query · Chakra UI · Turbo Monorepo

- Loyalty and rewards platform with an admin console and end-user wallet for campaigns, offers, billing, and blockchain-backed reward programs.
- Created Next.js / TypeScript apps in a Turbo monorepo (TanStack Query, Chakra UI) with Nest.js APIs using Prisma / PostgreSQL, Auth0, Docker, and Kubernetes.
- Implemented event-driven campaign and reward automation with Kafka and Temporal so marketing and ops could launch programs with less engineering support.
- Integrated Auth0 authentication with Kafka and Temporal workflows to support secure, event-driven campaign and reward automation.

EDUCATION

Bachelor of Engineering — Computer Engineering

Apollo Institute of Engineering and Technology · Gujarat Technological University