

WORK EXPERIENCE

- Self-Employed

Freelance Full-Stack Developer

(06/2023 - 03/2026)

 - Developed and maintained scalable full-stack applications using React.js,Node.js, TypeScript, and Tailwind CSS, enhancing user engagement through
 - intuitive UI/UX improvements Implemented RESTful APIs with Express.js, integrating both MySQL (relational) and MongoDB (NoSQL) databases; optimized queries through indexing and schema normalization.
 - Built CI/CD pipelines using Jenkins and AWS, deploying to EC2 and S3 with automated testing and monitoring, ensuring 99.9% uptime
 - Secured 1,000+ user accounts by integrating JWT and OAuth2 authentication, implementing RBAC and token expiration strategies that reduced unauthorized access attempts.
 - Designed system integrations using RESTful APIs and WebSockets, developing reusable TypeScript components and modular Node.js services to reduce technical debt
 - Leveraged Git/Git Flow for version control, leading refactoring initiatives that improved application latency and reduced bug density.

kanaka durga vasthralayam

Operations Manager

(03/2018 - 04/2023)

PROJECTS

Hospital Resource Management

A scalable microservices-based Hospital Resource Management Backend system.

Backend

- Architected a scalable **microservices-based backend** using Node.js, Express.js, and TypeScript within an Nx Monorepo
- Designed and implemented an **API Gateway** for centralized routing, proxying, rate limiting (Redis), CORS, and Helmet security
- Developed an **Authentication Service** using PostgreSQL with Drizzle ORM for user management and role-based access control (Admin, Doctor, Patient)
- Implemented **JWT-based stateless authentication** with secure password reset via Nodemailer
- Integrated **Google OAuth 2.0** for third-party authentication
- Built an **Appointment Service** using MongoDB with Mongoose for doctor availability, slot management, and booking workflows
- Implemented **atomic booking operations using Mongoose transactions** to prevent double-booking
- Applied **Redis caching (cache-aside pattern)** for improved read performance and rate limiting
- Used **Zod** for runtime schema validation across services
- Integrated **Pino** for structured JSON logging
- Documented APIs using **Swagger/OpenAPI**

Key Functionalities

- User Authentication & Authorization:** Secure login using JWT, Google OAuth 2.0 integration, and RBAC implementation
- Role-Based Access Control:**Separate permissions and access flows for Admins, Doctors, and Patients
- Appointment Management:**Doctor availability configuration, time-slot management, and secure booking system
- Concurrency Handling:** Transaction-based booking logic to ensure data consistency and prevent race conditions
- Performance Optimization:** Redis-based caching and rate limiting for high-traffic scenarios
- Email Services:** Secure password reset flow using SMTP integration
- API Gateway Architecture:** Centralized request handling with proxy services and security middleware
- Containerization:** Dockerized services with Docker Compose for consistent multi-container development and production environments
- Monorepo Architecture:** Managed scalable codebase using Nx with shared libraries for common schemas, error handling, and types

<https://github.com/saicharanars/hospital-resource-management-system>

nodejs, expressjs, cors, ratelimitting, Authentication, Authorization, Oauth, JWT, redis, RBAC, Mongodb, Postresql, Drizzle-Orm, Mongoosejs, zod, data validation, Logging, pino, monorepo, Docker, Docker Compose, git, Typescript, Javascript, nx

SKILLS

Backend Development & Devops



Expressjs, Node JS, SocketIO, MongooseJS,Sequelize, JWT, git version control, jenkins, CI/CD, pm2, nestjs, typeorm, swagger, data validation and json serialization

DataBases



MongoDB, MySql, PostgreSQL

EDUCATION

ACE Engineering College

B.Tech, Mechanical

MyStore

A full-stack e-commerce application with a clean, responsive UI for an intuitive shopping experience.

Frontend

- Created a responsive UI with Next.js, React.js, and TypeScript
- Applied Tailwind CSS and Shadcn UI components for modern design
- Designed modular architecture with reusable React components
- Implemented Redux Toolkit and RTK Query for state and API management
- Established dynamic routes for product listings and user profiles Incorporated Tanstack Table for product,order and shipment management
- Developed role-based features for buyers and sellers

Backend

- Constructed a scalable backend using Express.js with Node.js and TypeScript
- Created a RESTful APIs for product processing,order processing, shipment tracking, and user management
- Implemented JWT authentication and password hashing
- Utilized PostgreSQL with Sequelize for buyer-api
- Integrated MongoDB with Mongoose for seller-api
- Used Swagger UI for API documentation
- Set up AWS S3 for image uploads with Sharp for optimization

Key Functionalities

- User Authentication: Secure login with JWT and Bcrypt
- Buyer Features: Product search, purchase process,and order tracking
- Seller Features: Product listing and order management tools
- Profile Management: Comprehensive profiles for buyers and sellers
- Dashboard: Sales statistics and order status visualizations
- Advanced Features Pagination: Efficient handling of product listings and orders
- Containerization: Containerized the application using Docker to ensure consistent and reproducible environments with multi-stage docker builds

<https://github.com/saicharanars/mystore>

postgresql, mongodb, mongoosejs, sequelize, Javascript, Typescript, Nodejs, Rest API, React js, Nextjs, Reduxjs, Redux Toolkit, RTK Query, Tanstack Table, Tan Stack Router, React Context, HTML, CSS, Tailwindcss, Shadcn UI, pnpm, nx, monorepo, bcrypt, jwt, aws-s3, nginx, git, ExpressJs, Docker, containerization

Front End Development



HTML, CSS, JavaScript, React js, redux js, Tailwindcss, material ui, Bootstrap, nextjs, shadcn, RTK Query, Tanstack Table

Prgramming Languages



JavaScript, TypeScript, Python

Chaitanya Junior College

Intermediate, MPC