

Harish Rana

✉ harishrana1144@gmail.com

☎ 8872512811

📍 Mohali, India

🌐 harishrana5492

🔄 Harish5492

SUMMARY

Skilled MERN Stack & Blockchain Developer with expertise in Node.js, Express.js, React.js, MongoDB, SQL, REST APIs, JWT, Redis caching, and responsive UI with Redux, focused on clean code and performance.

WORK EXPERIENCE

Antier Solutions Pvt. Limited

MERN Stack & Blockchain Developer

Jan 2024 - Present

- Designed scalable cross-chain transaction systems using **Node.js**, **BullMQ**, and **TokenBucket algorithm**, boosting performance and efficiency.
- Built real-time dashboards using **React.js** and **Redis caching**, integrating **CMC API** for blockchain performance insights and data-driven decision-making.
- Ensured **JWT** user ID matches DB and transaction IDs, added **Ethers.js** signature verification during wallet creation, and addressed **CertiK** audit points. Also ran **RabbitMQ** through **Docker** and configured the environment in **Zookeeper**.
- Conducted **code reviews** for **Miracle Wallet**, optimizing time and space complexity, and significantly reducing the **API execution time by 20%**.
- Managed **RabbitMQ** and **Zookeeper** servers through **Docker** for streamlined project execution and scalability.
- Deployed the project on **AWS Lambda**, ensuring scalability and seamless performance under varying workloads.

EDUCATION

Chandigarh Group of Colleges, Jhanjeri (Mohali, India)

Bachelor Of Technology (Computer Science) - 6.96

June 2016 - June 2020

CIVIL SERVICES PREPARATION

Full-time preparation | 2020–2023

Developed strong analytical, time management, and self-discipline skills.

PROJECTS

Note Wallet

2024 - 2024

- Developed a highly **secure authentication system**, implementing **JWT** and **refresh tokens** to ensure session persistence.
- Built a **Contract Factory**, empowering admins with **advanced role-based access control** for approving sub-admins and managing token creation.
- Engineered a **profit-driven Buy/Sell Mechanism**, integrating **automated profit-sharing models** to optimize **crypto trading strategies**.
- Implemented **Role-Based Access Control (RBAC)** to enforce **secure and efficient permission management** across different user roles.
- Built using **TypeScript**, **NestJS**, **PostgreSQL**, **Redis**, **RabbitMQ**, **Twilio**, **SendGrid**, **Nodemailer**, **JWT**, and **Refresh Token** to deliver scalable, secure, and real-time backend services

- Designed and implemented a **feature-rich admin panel**, streamlining financial transactions, NFT marketplace, blockchain data visualization, and referral programs.
- Developed real-time dashboard analytics and dynamic modules like **Transaction Data** and **NFT Transactions** in **React.js**, with **time-based filters**, pagination, and seamless API integration.
- Created a **multi-chain portfolio tool** by scraping data from 13+ sources like **Etherscan** and **BscScan**, and optimized large-scale wallet fetching (50,000+) using **BullMQ** and **Token Bucket Algorithm** for high-performance data handling.
- Integrated **AppsFlyer** in backend to capture user interactions and app installs, enhancing data-driven decisions.
- Implemented **Role-Based Access Control (RBAC)** across backend and frontend — Admins had full module access, while Sub-admins had Read/ReadWrite access as per permissions.
- Developed an automated **blockchain health monitoring system** using **Cron Jobs** to fetch latest blocks from **testnet** and **mainnet RPCs**, compare block differences, and update real-time health status with **Redis caching** and **CMC API**.
- Built using **TypeScript**, **Node.js**, **NestJS**, **PostgreSQL**, **MySQL**, **Redis**, **BullMQ**, **RabbitMQ**, **AppsFlyer**, **JWT**, **Refresh Token**, **SendGrid**, **Twilio**, and **Cron Jobs** for robust, scalable, and secure architecture.

Miracle Wallet

2025 - 2025

- Strengthened app security by resolving **CertiK audit points**, including strict validation to ensure **JWT user ID** matches the database before allowing address book or notification updates.
- Implemented a secure **App Lock mechanism** with increasing wait intervals (30s, 1m, 1h) after 3 failed PIN attempts to prevent brute-force access.
- Implemented wallet creation validation by verifying **user signature** using **Ethers.js**, ensuring only authorized users can create or restore wallets if one already exists.
- Migrated blockchain interaction logic from **Web3.js** to **Ethers.js** for **Ethereum** and **BSC**, enhancing reliability, performance, and future maintainability.
- Built using **TypeScript**, **Node.js**, **MySQL**, **Redis**, **RabbitMQ**, **JWT**, **Refresh Token**, **Zookeeper**, **Docker**, **SendGrid**, **Twilio**, and **Cron Jobs** for robust, scalable, and secure architecture.

MPC Vendor Wallet

2025 - 2025

- Developed backend functionality to manage vendor onboarding with optional free trial configuration by Admin.
- Implemented logic to track vendor API hit usage, restricting app access once the predefined limit is exhausted.
- Designed and enforced a post-trial monetization model requiring payment for continued app usage or API access.
- Added grace period support configurable by Admin — vendors receive up to **two 5-day grace periods** before mandatory billing is enforced.
- Built using **TypeScript**, **Node.js**, **MySQL**, **Docker**, **Zookeeper**, and integrated scalable, modular service logic to maintain reliability under high demand.

SKILLS

- **Languages:** JavaScript, TypeScript, C++, Python, GoLang, Solidity.
- **Frameworks & Libraries:** React.js, Redux, Tailwind CSS, Material-UI, Node.js, Express.js, NestJS.
- **Databases:** PostgreSQL, MySQL, MongoDB (Mongoose), Redis.
- **DevOps & Tools:** Docker, Kubernetes, AWS, Git, CI/CD Pipelines.
- **Architecture & Messaging:** Microservices Architecture, RabbitMQ, BullMQ, WebSockets, Zookeeper.
- **APIs & Integrations:** GraphQL, REST APIs, AppsFlyer, CMC API.