

Omid Javaheri

Frontend Engineer

✉ omidjavaheri.eng@gmail.com

☎ +989138100702

📍 Tehran

🌐 github.com/OmidJavaherii

🌐 linkedin.com/in/omidjavaheri

🌐 omidjavaheri.vercel.app

📄 SUMMARY

Frontend Engineer with **3 years** of experience building production web apps with **React**, **Next.js**, and **TypeScript**, plus some **full-stack** work with Next.js. Experience across fintech/trading products, multi-brand web platforms, and Web3 features. Focused on **performance**, **reusable UI**, and shipping practical product improvements with small teams.

🧠 SKILLS

- Languages: TypeScript, JavaScript
- Frontend: React, Next.js (App Router, SSR, SSG, ISR), Tailwind CSS, Shadcn UI
- State & Data: Zustand, React Hook Form, Zod, REST APIs, WebSocket
- Architecture & UX: Design systems / shared UI, Storybook, i18n, PWA
- Tooling: Git, Docker, Lighthouse
- Web3: Wagmi, Viem, wallet connection

🎓 Education

Bachelor of Computer Software Engineering

Islamic Azad University Central Tehran Branch

2023 – 2025 | Tehran, Iran

Associate of Electrical and Electronics Engineering

Esfahan Technical and Vocational College (Mohajer University)

2021 – 2023 | Esfahan - Iran

📁 EXPERIENCE

Frontend Engineer - Tara Chain 🔗

09/2025 – 07/2026 | Tehran, Iran

- Implemented Web3 wallet flows with Wagmi and Viem: provider setup, wallet connection, API integration, and wallet switching across product features.
- Built gamification and product UI for Web3 surfaces including EpicOfCastles, and Gas Station (not lunch yet).
- Implemented i18n for 4 languages (English, Spanish, Russian, Dutch) to support multi-language product experiences.
- Developed reusable UI components and shared frontend patterns to reduce duplicated UI work across projects.

Selected Projects: [EpicOfCastles](#) , [Shexon](#) , [MoonPad](#)

Frontend Engineer - Desna 🔗

06/2024 – 09/2025 | Tehran, Iran

- Improved frontend performance across 4~5 web applications with rendering strategy updates, code splitting, and lazy loading, raising Lighthouse scores from ~30 to ~82.
- Standardized UI development with reusable component libraries, Zustand-based state patterns, and shared form architecture using React Hook Form and Zod.
- Collaborated with product, design, and backend to ship production features for brand and product sites including GhajariFood and RoyalJeans.

Selected Projects: [Polaris](#) , [RoyalJeans](#) , [GhajariFood](#)

Frontend Engineer - OptionBaaz 🔗

08/2023 – 06/2024 | Remote

- Contributed to the frontend of an Iranian stock-market platform, working within the existing architecture to deliver trader and admin-facing features.
- Built real-time market visualization and navigation using WebSocket data sync and optimized rendering for active trading views.
- Developed unified admin and end-user dashboards to simplify operational workflows and surface key platform metrics.

📁 Projects

LonoTech Community 🔗

Next.js, React, TypeScript, MongoDB, MinIO, Docker

- Took an existing static/base site and evolved it into a full-stack community platform with Next.js, Server Actions, and additional product features.
- Built reusable UI components and admin workflows to support faster feature development.
- Implemented MongoDB for data persistence and MinIO for media storage.
- Containerized the app with Docker and deployed it on a VPS.

Gas Station

- Built a Web3 gaming platform with a 3-person frontend team, integrating Solana Wallet and in-game asset transactions.
- Developed swap, P2P trading, marketplace, multi-account, and gamification features including levels, rewards, missions, and achievements.
- Implemented SSR/CSR, selective revalidation, i18n, Sass, and responsive experiences across desktop, tablet, and mobile.
- Brought the product to the final pre-launch stage, with the launch later postponed by the client.

Ghajari Food 🔗

- Developed a suite of Next.js production applications for restaurant management, accounting, online ordering, menus, and kitchen operations.
- Implemented NextAuth, role/permission-based access control, and real-time workflows using SignalR with a .NET backend.
- Improved performance through SSR, SSG, ISR, lazy loading, caching, and rendering optimization, supported by a shared design system.
- Scaled one application to approximately 10K daily users and 500 online orders/day.