

Mels Vagharshyan

Senior Front-end Developer

CONTACT INFORMATION

Email: mels.vagharshyandev@gmail.com
Address: Yerevan, Armenia
Phone: +37494541615

OBJECTIVE

Senior Frontend Developer with over 7 years of experience delivering production-grade, scalable web applications for diverse domains including health tech, gaming, and enterprise SaaS. Deeply specialized in React.js, Next.js, and TypeScript, with strong expertise in architecting maintainable frontend systems using modern state management libraries such as Redux Toolkit, MobX, Zustand, and RTK Query.

Adept in building responsive, accessible, and high-performance UIs using design systems and styling libraries including Ant Design, Material UI, Tailwind CSS, Sass/Less, Styled Components, and Framer Motion. Proven experience integrating RESTful and GraphQL APIs, handling real-time data via WebSockets, and optimizing data-fetching strategies with SWR and Axios.

Skilled in frontend testing and QA using Jest, Vitest, React Testing Library, Cypress, and component-driven development with Storybook. Embraces modern tooling including Vite, Webpack, Babel, ESLint, Prettier, Husky, and Lint-Staged to ensure code quality and performance.

Experienced with CI/CD pipelines using GitHub Actions, GitLab CI/CD, and deployments via Vercel, Netlify, and Docker. Collaborates effectively in Agile environments using tools like Jira, ClickUp, Notion, and wireframing platforms such as Figma, Zeplin, and Miro.

A strong communicator and mentor who actively contributes to improving team workflows, codebases, and DX in fast-paced, cross-functional teams.

EXPERIENCE

Feb 2023 - Apr 2025

Front-End Developer

Creatix Technologies

I've had the opportunity to lead and contribute to various software development projects across different industries, focusing on building robust, scalable platforms with seamless user experiences.

Key Projects:

Project Management Platform for Builders:

Developed a robust, feature-rich project management platform tailored for construction teams, focusing on seamless task tracking, timeline management, document handling, and real-time team collaboration.

Engineered the frontend using React.js with TypeScript for a type-safe, component-based architecture, and MUI (Material UI) for a modern, responsive, and accessible UI design system.

GraphQL-Driven Frontend Architecture

Leveraged GraphQL on the frontend to manage data fetching and caching efficiently, eliminating the need for external state management libraries such as Redux and resulting in a cleaner, more maintainable codebase.

Utilized Apollo Client for declarative data access, advanced cache control, and background synchronization to enhance data consistency and reduce network overhead.

Implemented pagination, query batching, and debounced fetch strategies to optimize dashboard performance, especially for large-scale projects with high data complexity.

Performance Optimization & UX Improvements

Identified and resolved rendering bottlenecks affecting user experience on high-load project dashboards. Used React.memo, dynamic imports (code splitting), and lazy loading to minimize initial load time and improve perceived performance.

Enhanced responsiveness and interactivity by optimizing component hierarchies and applying skeleton loaders and optimistic UI patterns.

Billing System (Frontend Implementation)

Designed and developed a full-featured billing module with support for subscription plans, invoice generation, and payment tracking.

Integrated secure, third-party payment services directly into the UI with proper input validation, error handling, and user feedback mechanisms.

Implemented dynamic forms, data-driven pricing models, and automated billing workflows within the frontend logic to ensure a smooth financial user journey.

Sales Engagement and Lead Management Platform:

Developed a sales engagement platform designed to optimize lead tracking, streamline follow-ups, and support automated communication workflows.

Built a modern, responsive, and enterprise-grade user interface using React.js, TypeScript, and Ant Design, ensuring visual consistency and accessibility across the platform.

REST-Driven Architecture & Frontend Design

Implemented a robust frontend architecture powered by RESTful APIs, enabling modular integration with backend services and improving data flow predictability.

Employed Axios with advanced features such as interceptors, retry logic, and global error handling for resilient API communication.

Designed dynamic data-driven components with strong typing using TypeScript, improving maintainability and reducing runtime errors.

UX & Performance Enhancements

Optimized component rendering with React.memo, virtualized lists, and code splitting, reducing load times and enhancing performance for large data tables and CRM dashboards.

Implemented form wizards, activity feeds, and notification systems to support lead nurturing and pipeline visibility.

Created flexible UI layouts with responsive grid systems and custom theming to support various screen sizes and enterprise branding requirements.

Communication & Workflow Automation

Integrated real-time updates and interaction tracking using WebSocket listeners on the frontend, enhancing responsiveness in lead communication and task follow-up modules.

Built reusable UI patterns for scheduling, reminders, and contact logging, reducing redundancy and accelerating feature delivery.

Mar 2021 - Feb 2023

Sr. Front-end Developer

Digitain Armenia

Key Projects:

Club Poker:

Led the frontend development of a real-time poker game platform, delivering a responsive and immersive user experience using React.js, TypeScript, and Tailwind CSS.

Focused on UI performance and usability to ensure smooth gameplay, optimized rendering, and intuitive interactions across all devices.

Designed and implemented modular, reusable components to streamline development and maintain consistency across the application.

Real-Time Synchronization & Gameplay Optimization

Solved a critical issue where gameplay actions (bets, folds, card reveals) were not updating in real time, leading to desynchronized player experiences.

Integrated Socket.IO to enable instant bi-directional communication, ensuring accurate and timely game state updates across all connected clients.

Leveraged Redux Toolkit for scalable and maintainable state management, enabling efficient handling of game logic and player interactions on the frontend.

Improved performance by minimizing unnecessary re-renders and applying advanced state selectors and memoization techniques.

Lottery Game Platform (Web3 Integration):

Developed a decentralized lottery game platform integrating blockchain-based smart contracts, offering users a transparent, provably fair, and trustless gaming experience.

Designed and implemented a responsive and interactive frontend using React.js, TypeScript, and Tailwind CSS, with a strong focus on usability, real-time feedback, and secure wallet interactions.

Web3 Integration & Smart Contract Interaction

Utilized modern Web3 libraries such as Ethers.js, Web3.js, and Wagmi to establish seamless communication between the frontend and Ethereum-based smart contracts.

Implemented wallet connection flows using RainbowKit and WalletConnect, supporting MetaMask, Coinbase Wallet, and other popular providers.

Handled contract reads/writes with Hooks-based abstraction (Wagmi + Ethers.js), including dynamic gas estimation, transaction state handling, and error recovery.

Real-Time Feedback & Transaction UX

Built intuitive UIs for ticket purchasing, live draw countdowns, and results visualization, backed by real-time event listeners using WebSocket connections and smart contract event subscriptions.

Integrated on-chain data display (e.g., jackpot totals, user participation history) with off-chain enhancements such as cache optimization and optimistic UI updates for a responsive experience.

Security & Best Practices

Applied frontend security best practices for Web3, including transaction confirmation prompts, network mismatch warnings, and user-friendly error messages for failed contract interactions.

Ensured compatibility across desktop and mobile browsers with graceful fallback behaviors for unsupported wallets or networks.

Jan 2018 - Mar 2021

React.js Developer

BeeWeb

Wellnest Guided Wellness & Mental Health App:

Developed a modern, scalable, and accessibility-first wellness platform designed to support users in building consistent mental health habits. The app enables users to log moods, write private journals, and receive personalized guidance using advanced tracking and recommendation logic. Key features include

real-time journal syncing, emotion tagging, analytics, and offline support. Emphasis was placed on scalability, form accessibility, performance, and clean architecture.

Key Responsibilities & Achievements:

Designed a modular React architecture using TypeScript and MobX, ensuring clear separation of concerns and minimal coupling.

Built a fully responsive UI with Sass, custom media query mixins, and semantic HTML for WCAG-compliant accessibility.

Created reusable, schema-driven form components using react-hook-form and Zod for robust client-side validation and type safety.

Implemented mood tracking analytics using custom SWR hooks with optimistic updates and caching strategies.

Integrated Socket.io for real-time journal syncing between sessions and devices.

Introduced localStorage-backed caching and offline form persistence for better mobile UX.

Followed atomic design principles, built a flexible design system, and documented components using Storybook.

Ensured high test coverage with unit and integration tests for forms, stores, and live updates logic.

PixelPilot Real-Time Game Analytics Platform:

PixelPilot is a cross-platform analytics solution tailored for game developers and studios. It visualizes real-time data streams from game servertracking player sessions, performance metrics, in-app purchases, and more. The platform supports multi-tenant access, modular widgets, and custom dashboards, enabling live insights across multiple titles in production.

Key Contributions & Highlights:

Developed a scalable front-end architecture using React, TypeScript, and Ant Design Pro.

Utilized Recoil for scoped, lightweight state management of live session data and chart filters.

Integrated real-time data pipelines with Socket.io, rendering live metrics and logs using Ant Design Tables, ECharts, and React Virtualized.

Designed a widget system with drag-and-drop layout and user-customizable dashboards.

Implemented role-based access control (RBAC) with Firebase Auth and conditional routing.

Created a global notification system and alert center using Ant Design Drawer + Badge components.

Focused on data-intensive UX performance using React.memo, useCallback, and batched updates.

Integrated i18n for multilingual support (English, Korean, Japanese).

Used Sentry and LogRocket for session replay and error tracking in production.

EDUCATION

Information Technologies

Bachelor's Degree

European University

LANGUAGES

Armenian, English, Russian, German

SKILLS

JavaScript (ES6+), TypeScript, HTML5, CSS3, Sass/Less
Tailwind CSS, Styled Components, Emotion, Ant Design, Material UI, Framer Motion
Jest, Vitest, React Testing Library, Cypress, Storybook
GitHub Actions, GitLab CI/CD, Docker, Vercel, Netlify
Git, GitHub, GitLab, Bitbucket
AWS, AZURE

React.js, Next.js, Redux Toolkit, RTK Query, Zustand, MobX, Vue 3, Pinia
REST APIs, GraphQL (Apollo), WebSockets, SWR, Axios
Vite, Webpack, Babel, ESLint, Prettier, Husky, Lint-Staged
Jira, ClickUp, Trello, Notion
Figma, Zeplin, Miro, Slack, Confluence

SOFT SKILLS

Problem-Solving & Critical Thinking Ability to analyze complex frontend challenges and implement efficient, scalable solutions.

Collaboration & Teamwork Experience working in cross-functional teams, coordinating with backend developers, designers, and stakeholders.

Communication Skills Clearly articulating technical concepts to both technical and non-technical audiences.

Attention to Detail Writing clean, maintainable, and well-structured code with a focus on UI/UX best practices.

Adaptability & Continuous Learning Staying updated with the latest React ecosystem trends and best practices.

Time Management & Prioritization Efficiently handling multiple tasks, meeting deadlines, and optimizing workflows.

Mentorship & Leadership Guiding junior developers, conducting code reviews, and sharing knowledge with the team.