Kevin Diesenberg

Port Huron, Michigan | Email | Portfolio | LinkedIn | GitHub

WEB DEVELOPER · FULL STACK SOFTWARE ENGINEER

Full stack engineer with 2+ years of production experience at a national media company. Skilled in React, TypeScript, Node.js, AWS, and TanStack, with experience building SSR content platforms and internal tooling. Proven ability to ship user-facing features and APIs supporting 1.05M average daily multi-platform users (Web + Apps + OTT) and 709K average daily local users (YTD), while improving performance, accessibility, and developer velocity through automation and observability.

TECHNICAL SKILLS

JavaScript/TypeScript, Ruby; React, TanStack (Query/Table/Router/Form), useSWR, Axios, Styled Components, Tailwind, Shadcn/UI, Vite, Webpack, Headless UI, Radix UI, HTML/CSS/SCSS; Node.js, Express, Ruby on Rails, REST APIs, SST; Arc XP (Arc Fusion SSR, PageBuilder, Content API); Arc Identity SDK, SSO/OAuth, Google SSO; AWS (Lambda, S3, EC2, RDS, DynamoDB), Docker, Git, Vercel; PostgreSQL, SQL, DynamoDB; Jest, React Testing Library, Vitest; Sentry, Vercel AI SDK, Cursor AI; Slack, Jira.

PROFESSIONAL EXPERIENCE

Graham Media Group – Hybrid (Detroit, Michigan)

Website Developer · Software Engineer · 03/2024 - Present

- Built and maintained React + TypeScript applications across a multi-station broadcast portfolio, supporting 1.05M average daily multi-platform users (Web + Apps + OTT) and 709K average daily local users (YTD).
- Developed and maintained the core broadcast platform as a server-rendered React + TypeScript app (Arc Fusion) on Node.js and Webpack, integrating Arc XP PageBuilder and the Content API to power story/article and page rendering across multiple station sites.
- Delivered accessible, branded UI using a shared Styled Components design system by composing themed components over Headless UI and Radix UI primitives to ensure consistent UX across mobile and desktop.
- Optimized client-side data fetching with useSWR + Axios (cache + revalidation strategy, request de-duplication, conditional fetching), reducing redundant API calls 20% and improving perceived load performance.
- Implemented and maintained authentication workflows using Arc Publishing Identity SDK, supporting email/password login and Google SSO; centralized user state with React Context and ensured reliable session handling in an SSR environment (client-side initialization + session verification).
- Integrated subscription-aware experiences using Arc identity/subscription data to support gated content and account flows (login, registration, password reset), aligning UI behavior with content restriction rules.
- Built standalone internal apps (e.g., Trending Dashboard) using Vite + TanStack Query + Tailwind + Shadcn/UI, with SST backends (AWS + DynamoDB), enabling rapid iteration outside the main brand system.
- Developed Node.js/Express services in TypeScript and integrated AWS (Lambda, S3, RDS, DynamoDB) to support content workflows and media delivery; used Docker for repeatable staging and production builds.
- Integrated Vercel AI SDK to enable story summarization and consolidation, improving editorial workflows by grouping related/trending items across properties.
- Led a navigation architecture refactor by centralizing header state and data access, eliminating prop drilling and standardizing query patterns across six station sites.
- Built an internal Trending Stories tool (TanStack Query/Table/Form, SST, DynamoDB, Zod) enabling editors to submit, pin, expire, and manage story placements with real-time slot validation.
- Implemented Sentry error and performance monitoring to improve observability and accelerate production triage; contributed to code reviews and sprint delivery in a distributed Agile team using Slack and Jira.

TECHNICAL PROJECT

Broadcast Stories Management Platform

React, TypeScript, TanStack (Router, React Query, TanStack Form), Vite, Tailwind, Shadcn/UI, SST, DynamoDB, Vercel AI SDK

- Built a real-time dashboard for editors showing submitted and trending stories across multiple stations, with normalization, sorting, and expiration logic.
- Implemented AI-driven summarization with Vercel AI SDK to consolidate related stories into unified summaries for faster editorial review.
- Designed accessible interactive UI (carousels with reduced-motion support, chart/list toggles, live capacity tracking) to support editorial workflows.
- Integrated TanStack Query for caching and consistency across stations, improving responsiveness under frequent updates.

EDUCATION

Flatiron School – Full Stack Software Engineering Certificate (React, Ruby on Rails) - 10/2022 - 03/2023