Seif Essam

Full-Stack Developer

✉ seifessam60@gmail.com ☎ 01144654953 / 01211356458

https://www.linkedin.com/in/seif-essam-906b30181 https://github.com/seifessam60

LinkedIn Headline

Full-Stack Engineer Driving Scalable Node.js & React Architectures

About

Full-Stack Developer **7/2023 – Present**

Freelance

Seasoned full-stack engineer with 3+ years building and scaling production-grade applications from API to UI. I specialize in clean, maintainable architectures—fueled by domain-driven design on the backend (Node.js, Express, MongoDB, PostgreSQL) and modern front ends (React, Next.js, Vue, TypeScript, Tailwind CSS). I've led microservice platforms, optimized CI/CD pipelines, and mentored teams to adopt best practices in testing, security, and performance. Seeking a role where I can architect end-to-end solutions that delight users and empower engineers.

Core Competencies

- **Backend**: Node.js, Express.js, NestJS, PostgreSQL, MongoDB, Redis, GraphQL
- **Frontend**: React, Next.js, Vue.js, Vite, TypeScript, HTML5, CSS3, Tailwind CSS
- **Architecture**: Domain-Driven Design, Microservices, REST & GraphQL API Design
- **DevOps & CI/CD**: Docker, GitHub Actions, Heroku, Netlify, Vercel, Kubernetes basics
- **Testing**: Jest, Mocha, Supertest, React Testing Library, Cypress
- **Security & Auth**: JWT, OAuth2, Role-Based Access Control, PCI-DSS Compliance
- **Performance**: Caching Strategies, Code Splitting, Lighthouse Audits, SSR & SSG
- **Tools**: ESLint, Prettier, Swagger/OpenAPI, Postman, Storybook
- **Mentoring & Leadership**: Workshops, Brown-Bags, Pair Programming, Technical Documentation

Full Stack PROJECTS

AI_interview_platform

- A full-stack project designed to streamline the interview process by providing a platform for scheduling, conducting, and managing interviews.
- Developed and optimized RESTful APIs using Node.js and Next.js to handle interview scheduling, user management, and data retrieval.
- Implemented server-side rendering (SSR) with Next.js to improve performance and SEO for the platform.
- Designed and managed database schemas using Firebase to store user data, interview details, and application logs.
- Created a responsive and user-friendly front-end interface using React.js, TailwindCSS, and shadcn for seamless user interaction.
- Integrated authentication and authorization mechanisms using JWT to secure user accounts and restrict access to sensitive features.
- Deployed the application on Vercel for efficient and scalable hosting.

Technologies used: JavaScript, TypeScript, Node.js, Next.js, React.js, TailwindCSS, Firebase, JWT, Git/GitHub, Vercel. [Deployment Link]

Natours App

- Social media prototype built with Next.js 13 and TypeScript, leveraging the App Router for SSR and dynamic routing
- Developed responsive UI: Navbar, Sidebar, Profile Page, and Suggested Users components, styled with CSS Modules and PostCSS
- Integrated UploadThing for secure, serverless image uploads and cloud storage handling
- Modeled relational data (users, posts, media) with Prisma ORM and PostgreSQL, including schema definitions and migrations
- Tuned Next.js config (`next.config.ts`) for image optimization, custom font loading via `next/font` (Geist), and environment variable management
- Deployed to Vercel with CI/CD workflows, ensuring zero-downtime releases and fast global delivery

Technologies used: Next.js 13, TypeScript, Prisma, PostgreSQL, UploadThing, CSS Modules, PostCSS, Vercel. [Link]

University Library

- · A large-scale microservice project designed to manage books for users and administrators.
- Developed the front-end using Next.js and TypeScript, ensuring seamless user experience with modern design practices using Tailwind CSS.
- Implemented backend services to handle book management, user authentication, and data retrieval, leveraging PostgreSQL as the database.
- Integrated OAuth for secure authentication, facilitating a user-friendly login and registration process.
- Conducted thorough testing of all flows to ensure reliability and performance.

Technologies used: Next.js, TypeScript, Tailwind CSS, PostgreSQL, OAuth, Node.js, and various other tools and libraries for enhanced functionality. [link]

Front End PROJECTS

Mojito Cocktails

- A single-page React application built with Vite, showcasing a curated collection of cocktail recipes through an engaging, responsive UI.
- Developed the front end with React and Vite, leveraging hot-module replacement for rapid iteration and configuring optimized production builds.
- Architected a modular component structure (Hero, Layout, ContactSection) and centralized static recipe/art data in a constants module for maintainability and performance.
- Styled layouts using CSS, ensuring full responsiveness across mobile and desktop, and applied BEM-inspired naming for clear, scalable styles.
- Enforced code quality and consistency with custom ESLint rules and Prettier formatting, streamlining team collaboration.
- Deployed to Vercel via continuous integration, enabling zero-downtime updates and fast global delivery.

Technologies used: React, Vite, JavaScript (ES6+), CSS, ESLint, Prettier, Vercel. [link]

React Movies

- Developed a minimal template for a React application using Vite, enabling fast refresh and code optimization.
- Implemented core features for browsing and displaying movies, enhancing user experience through a responsive design.
- Utilized modern JavaScript, HTML, and CSS technologies for efficient performance and maintainability.
- Structured the project with best practices in mind, including proper configuration and linting for code quality.

Technologies used: React, Vite, JavaScript, HTML, CSS, Tailwind. [Deployment Link]

Apple Store Clone

- Developed a web application using React and Vite, focusing on performance optimization with Hot Module Replacement (HMR).
- Implemented a structured project layout, including essential configuration files for ESLint, PostCSS, and Tailwind CSS to ensure coding standards and responsive design.
- Integrated 3D models and dynamic content management, enhancing user engagement and visual appeal.
- Integrated a page that enables the user to join an ongoing group chat or create a new group chat.

Technologies used: React, Vite, JavaScript, HTML, CSS, Tailwind CSS. [Deployment Link]

Back End PROJECTS

Subscription Tracker

- A backend-focused project designed to help users track and manage their subscriptions efficiently.
- Developed RESTful APIs to handle subscription creation, updates, and deletions, ensuring robust and scalable backend functionality.
- Implemented user authentication and authorization using JWT to secure user data and restrict access to sensitive endpoints.
- Designed and optimized database schemas using MongoDB to store subscription details and user information.
- Integrated third-party APIs for payment tracking and notifications, enhancing the application's usability.

Technologies used: Node.js, Express.js, MongoDB, JWT, Postman, and other backend tools and libraries. [link]

Wallet Backend

A Node.js/Express.js API powering a React Native multi-currency wallet, handling wallet generation, transaction signing, and balance retrieval.

- Structured a modular folder layout (controllers, routes, services, utils, config) for clear separation of concerns.
- Configured CORS, JSON body parsing, and environment variables via dotenv.
- Implemented wallet-creation endpoints using bip39 and ethereumjs-wallet to derive Ethereum and Binance Smart Chain key pairs.
- Built transaction-signing and broadcasting routes with Web3.js and JSON-RPC, ensuring private keys remain server-side.
- Developed balance-fetching endpoints that query on-chain data from Infura and BSC nodes.
- Enforced request validation with express-validator and centralized error-handling middleware.
- Automated the dev workflow with nodemon, standardized code style with ESLint/Prettier, and wrote unit tests using Jest.
- Deployed the backend on Heroku with CI/CD integration for zero-downtime updates.

Technologies used: Node.js, Express.js, Web3.js, ethereumjs-wallet, cors, dotenv, express-validator, Jest, ESLint, Prettier, nodemon, Heroku.. [Link]

Mern Notes

MERN Notes Backend A RESTful Express.js API for creating, reading, updating, and deleting notes with topic, status, and content fields.

- Architected a modular folder structure (controllers, routes, models, middleware, config) for maintainability.
- Configured environment-driven MongoDB connections using Mongoose and dotenv.
- Built CRUD endpoints with comprehensive HTTP status handling and centralized error middleware.
- Designed a Note schema with timestamps to track creation and updates.
- Enabled CORS, JSON body parsing, and request throttling via express-rate-limit.
- Automated development workflow with nodemon and standardized code with ESLint/Prettier.
- Deployed production backend on Render.com with seamless environment variable management.
- Delivered a robust API consumed by a React/Vite frontend for real-time note operations.
- Strengthened skills in Express middleware patterns, Mongoose data modeling, rate limiting, and cloud deployment.

Technologies used: Node.js, Express.js, MongoDB with Mongoose, dotenv, CORS, express-rate-limit, nodemon, ESLint, Prettier, Render.com. [link]

SKILLS

Programing Languages: JavaScript, TypeScript
Frontend Development: HTML, CSS, Bootstrap, React.js, Redux , Angular, Vue.js, Next.js, TailwindCSS, VueJs, shadcn
Backend Development: Node.js, Express.js, Next.js, RESTful API, SSR, JWT
Databases: MongoDB, Mongoose, PostgreSQL, Firebase
Others: Version Control **(**Git/GitHub), ESLint, npm, PostCSS, Deployment (Vercel/Netlify), Environment Variable Management (.env files)