Morteza Mazrae

Front-End Developer

Iran, Ahwaz Mor.mazrae@gmail.com

+989169318223 /Morteza-Mazrae

Education

Bachelor of Mathematics Science — 2020 - present

ABOUT ME

I am a mathematics student and front-end enthusiast developer who enjoy programming and interested in breaking challenges and reducing smooth, fast and amazing web apps.
The challenges around web performance and developer experience interest me and keep me excited! Challenges like React rendering issues, HTML & CSS performance.

Experience

Coursee — Front-End Developer — 2021 - present

In Coursee we learned about combining the front-end and the back- end in an innovative way, we simply tried getting inspired by RPCs in general and we came up with a library called Telefunc which uses the power of the modern bundlers to combine the client and server code that made it easy for the team to collaborate. As a database, we used PostgreSQL which made the work really easy to be scaled, for PostgreSQL, we used Prisma which is a fairly modern ORM for PostgreSQL that makes it easy to handle SQL functionalities! One of the interesting parts of the coursee project, is the way we handled bundling and transpiling of the project, which is a tool called vite. With Vite and vite-plugin-ssr we have been able to get the full control of our apps so we'd handle each part and piece of our app. That's the reason it let me to handle many of the front-end burdens. One of the other important pieces of our app, is that we made our app based on dynamic imports (Lazy import and Suspense) and React concurrency model, so users don't face tearing issues and some TTFB problems, well, it succeeded and that's what our users report to us.

Snapp Group — Front-End Developer — Aug 2022 - Dec 2022

Used Vite-plugin-ssr the NextJS alternative that use Vite as a bundler and for API used Telefunc which is modern and DX friendly RPC. For the UI we used a library that called Mantine which is a React components library focused on providing great user and developer experience and also covering customized hooks for State management, UI, Dom, utilities and lifecycle events. We also used Windi CSS very similar to tailwind, If you are already familiar with Tailwind CSS, think about Windi CSS as an on-demand alternative to Tailwind, which provides faster load times, full compatibility with Tailwind and a bunch of additional cool features. In different cases for breaking down React tree and sharing data that can be considered "global" for a tree of React components, such as the current authenticated user, theme, or preferred language we used a library that called Jotai, simpler than Context-React or Redux, that is able build state by combining atoms and renders are optimized based on atom dependencies. This solves the extra re-render issue of React context and eliminates the need for the memoization technique

Karoo — Front-End Developer — 2020 - 2021

Creating products around ReactJS and moving old Nextjs versions to Next 11 with using modern features like get Server SideProps.

Skills & Tools

- Advanced HTML | CSS & SASS
- Bootstrap | Tailwind |Windicss
- JavaScript | TypeScript (Concurrency, Multithreading, ...)
- ReactJS | NextJS | ViteJS | (Hooks, State management, Performance, Concurrent rendering)
- Source Control (Git)
- API (Rest API, GraphQL, RPC)

Languages

- Arabic
- Persian
- English