Steven Bober

TECHNICAL SKILLS

Javascript/ES6, TypeScript, React, React Native, React Router, Redux, Recoil, Mui library, Evergreen library, Yup, Apollo Client, Apollo Server, GraphQL, GraphQL API's, HTML, CSS, Node/Express, SQL, PostgresSQL, LoDash, Next.js, Framer-Motion, Prisma, Langchain, Tailwind CSS, Docker, Sass, Bootstrap, MongoDB, Cypress, Mongoose, Typeorm, shadcn/ui, PlanetScale

PROFESSIONAL EXPERIENCE

Overland Code | ConnectSTR | Full Stack Engineer | Typescript, React, Node, GraphQL, SQl, Docker | July, 2022 - Present

- Utilizing Typescript, Evergreen UI and Styled components, built out UI/UX based on Figma mocks, with localized state, styles and encapsulated components.
- Utilized Apollo Client's useQuery and useMutation hooks for Backend interaction among frontend components, storing globalized state in Recoil.js.
- Developed and styled an interactive Dashboard page using React Hooks, Apollo Client and Recoil.js, creating a responsive and intuitive interface for all users; each widget was encapsulated and responsible for its own data query allowing for efficient, decoupled data fetching and renders.
- Built out React-router pattern for the sake of application navigation, leveraging secure routes and Router hooks to constrict UI/UX based on user roles.
- Built multiple frontend forms both within and outside of modals for user interaction; utilizing Yup for form validation and Apollo client for backend mutations.
- Was a key contributor in multiple database migrations that allowed for an application pivot; the database migrations were completed utilizing Prisma and PostgresSQL and demanded my fixing of multiple GQL resolver and schema based bugs caused by changing the database architecture.
- On the BE, utilized Typescript and GraghQL Codegen to ensure property type safety among the returns from our SQL queries and our GQL schema/resolvers.
- Established backend development patterns across the GQL schema, resolvers, and service methods that allowed for coherent app organization across.
- Utilized RAW SQL for backend data insertion and querying that subsequently facilitated efficient manipulation and retrieval for frontend component use.
- Employed PgTyped for SQL queries file generation, generating TypeScript types for parameters and results, ensuring type safety for raw SQL code.
- Applied Agile methodology, promoting iterative development and team collaboration for efficient delivery of frontend and backend components.
- Utilized Codegen for backend development, generating TypeScript code to provide detailed type information. This enabled front-end type safety, ensuring accurate and consistent data types were maintained between front and backend components, enhancing development efficiency and reducing potential errors.
- Expanded raw SQL queries via a join against multiple tables enabling the GQL resolver to fetch across multiple tables and synthesize data prior to FE return.
- Built out multiple frontend widgets and coded numerous functions to filter through backend data, facilitating calculations and display of key metrics vital to the user experience. This integration seamlessly synthesized data processing from the backend to frontend, enabling effective data representation.
- Refactored multiple FE components to align with overall UI theme in order to accomplish total frontend UI/UX consistency without taking on tech debt.
- Utilized react router for nested routes with dynamic sub paths coming from successful GQL mutations that facilitated CRUD operations.

Overland Code | Pucks Gambit | Front End Engineer | Typescript, React, Node, GraphQL, SQl, Docker | November, 2021 - July, 2022

- Leveraged React-Router-Dom to streamline and rectify UI/UX navigation on the login page for improved user engagement and application access.
- Engineered interactive side-bar component, incorporating a navigation system using React-Router-Dom, Styled Components, and Evergreen-UI, to allow an interactive UI/UX, that enables the user to navigate seamlessly from point to point within the application interface according to Figma designs.
- Employed Apollo Client and GraphQL to retrieve and present accurately inputted user data throughout key FE components, enhancing data relevance.
- Utilized React and Evergreen UI to architect a real-time component for diverse sports teams, creating a dynamic and interactive user interface. Integrated the global state management library, Recoil, to handle data transformations upon user interactions, amplifying the system's responsiveness and user experience.
- Built and refined a multi-faceted FE feature, with interactive elements, built from a variety of sub-components, that managed intricate aspects of data handling across the application, by utilizing local state, YUP, and TypeScript to ensure specific criteria and key metrics were met prior to render.
- Utilized react-select and TypeScript, to develop a tailored drop down feature within the UI/UX that facilitates a dynamic filtering of data based on key criteria.

U.S. Army Reserve | Transportation Officer(Active Secret Clearance) | 2020–Present

• Have performed various duties and tasks on an active and reserve status regarding logistics for the U.S. Army.

EDUCATION