# **Ota Prokopec**

Junior full-stack developer with considerable proficiency seeking opportunities to further hone skills and expertise. Capable of handling a wide array of tasks competently, with potential for significant contributions in various aspects of development.

#### Frontend

Svelte, Sveltekit, React, TanStack router, Next.js, Remix, Typescript, Urql, Apollo client, Tanstack query, Appwrite, Supabase, reacthook-form, Tailwind, Styled components Tanstack Router, next-intl, react-intl, Shadcn, HeroUI

#### Backend

Rest, GraphQL, tRPC, Nexus, Pothos, Express, Apollo GraphQL, GraphQl yoga, NextAuth, BetterAuth, Drizzle ORM, Prisma, Typescript, Zod, TSUP, Translations

### School

I studied: Střední průmyslová škola a Vyšší odborná škola Písek I study: Západočeská univerzita v Plzni

#### **Tourmate**

Tourmate is a web application designed to help users discover nearby places and experiences. The project is organized as a monorepo using Turborepo, separating the frontend, backend, and shared utilities. The frontend is built with SvelteKit, while the backend uses Apollo GraphQL in combination with Nexus. For the GraphQL client, Svelte-Urql is utilized. All application data is managed through Appwrite, which also handles user authentication. Cloudinary is used to store images, and Mapbox, along with MapTiler and MapLibre, powers the map features. The entire codebase is written in TypeScript, and the UI leverages several libraries, including Shadon, Flowbite, Tailwind UI, and DaisyUI.

Front end

Sveltekit (SSR, CSR), Svelte Urql, Typescript, Tailwind, Mapbox (MapTiler, MapLibre) Back end:

GraphQL, Typescript, Appwrite

#### Appwrite-ssr

Appwrite-ssr helps developers to create SSR applications with Appwrite.

The Library is built for TypeScript/JavaScript projects. It can be used with any SSR web framework, but it is especially built for Sveltekit.

Appwrite doesn't natively support SSR, so this library assists developers in managing authentication on the backend with Appwrite. Additionally, Appwrite-SSR offers type-safe collections and buckets, a feature not provided by Appwrite itself.

#### **My Pets**

MyPets is a TypeScript-based web app crafted to help people with finding their lost pet. It follows a monorepo architecture powered by Turborepo, cleanly splitting the codebase into frontend, backend, and utility packages. On the frontend, the app runs on SvelteKit, with Svelte-Urql handling GraphQL queries and mutations. The backend stack features Apollo GraphQL paired with Nexus for schema building. Data storage and authentication are managed by Appwrite, while media assets (like images) are hosted on Cloudinary. Interactive maps are rendered using Mapbox. For styling and UI components, the project takes advantage of multiple libraries: Shadcn, Flowbite, Tailwind UI, and DaisyUI, to deliver a flexible and modern user interface.

Front end:

Sveltekit (SSR, CSR), Svelte Urql, Typescript, Tailwind, Mapbox (MapTiler, MapLibre)

Back end: GraphQL, Nexus, Typescript, Appwrite, Cloudinary sdk

### Musify

Musify, the application that helps people to discover music.

The application is structured as a monorepo using Turborepo, and is powered by Next.js on the frontend with a GraphQL Yoga backend (using Nexus). For database access, it uses Drizzle ORM, with all data stored in a PostgreSQL database. Authentication is managed via NextAuth, and Urql is used to handle GraphQL queries and mutations. The entire project is written in TypeScript, and the UI is styled using component libraries like Shadcn and HeroUI.

Front end

Next.js, Urql, Apollo client, Tanstack query, Typescript, Tailwind Back end:

GraphQL yoga, Typescript, Drizzle, NextAuth

## Date app

The connection pocket friend for introverts.

The application follows a monorepo setup using Turborepo. The frontend is powered by Next.js, while the backend utilizes GraphQL Yoga and Pothos for schema management. Database interactions are handled through Drizzle ORM, with PostgreSQL as the data storage solution. BetterAuth manages authentication, and Apollo Client is used to handle GraphQL queries and mutations. The entire codebase is written in TypeScript, ensuring robust type safety throughout. For UI styling, libraries such as Shadcn and HeroUI are incorporated. TSUP serves as the build tool for utilities within the application, and all forms rely on React Hook Form. Next Intl is responsible for managing translations, while Cloudinary is used for storing images, including profile pictures and other media content.

#### Front end

Next.js, Apollo client, Typescript, Tailwind, react-hook-form, Shadcn, HeroUI, next-intl Back end:

GraphQL Yoga, Pothos, Typescript, Drizzle ORM, BetterAuth, TSUP, PostgreSQL, Cloudinary, Zod