SSR & Leaflet

Leaflet Anomalies with SvelteKit SSR

Server Side Rendering (SSR)

- A feature of modern 'meta' frameworks.

- When a route is loaded, the framework may load the page on the server first - build the complete page, and then send the complete page to the browser

- Matches closely conventional 'Server Rendered Applications' - like Node/Hapi/Handlebars or PHP

- Seen as better for Search Engine Optimisation (SEO)

# Server Side Rendering (SSR) - Client Side Hydration

- **Client-side hydration** a unique feature of modern frameworks (SvelteKit, Next, Nuxt)

- While the initial HTML is generated on the server:

  - Automatically include client-side JavaScript code that "hydrates" the page once it's loaded in the browser.

  - After the initial page load, client-side JavaScript takes over to handle user interactions and dynamic updates

  - Potentially provides a smoother user experience

- Seen as 'Best of Worlds' combination of SPA and traditional Server Render Applications

Server Side Rendering (SSR)

- **Improved Initial Page Load Performanc**e: SSR routes provide fully rendered HTML content to the client, which can significantly reduce the time it takes for users to see content on the page. This is particularly beneficial for content-heavy websites or applications where fast initial page load times are crucial for user experience.

- **Search Engine Optimization (SEO):** Search engine crawlers can more easily index content from SSR routes because they receive fully rendered HTML from the server. This can lead to better search engine rankings and increased visibility for the application in search engine results pages (SERPs).

- **Accessibility and Progressive Enhancement**: SSR routes ensure that content is accessible to users who have JavaScript disabled or are using devices with limited capabilities. By providing server-rendered content as a baseline, applications can progressively enhance the user experience with client-side JavaScript interactions.

- **Improved Performance on Low-Powered Devices**: SSR routes can provide a better user experience on low-powered devices or slower network connections by reducing the amount of client-side processing required to render the initial page.

- **Consistent User Experience:** SSR routes help ensure a consistent user experience by providing a predictable rendering process across different devices and browsers. This can help minimize issues related to client-side rendering inconsistencies.
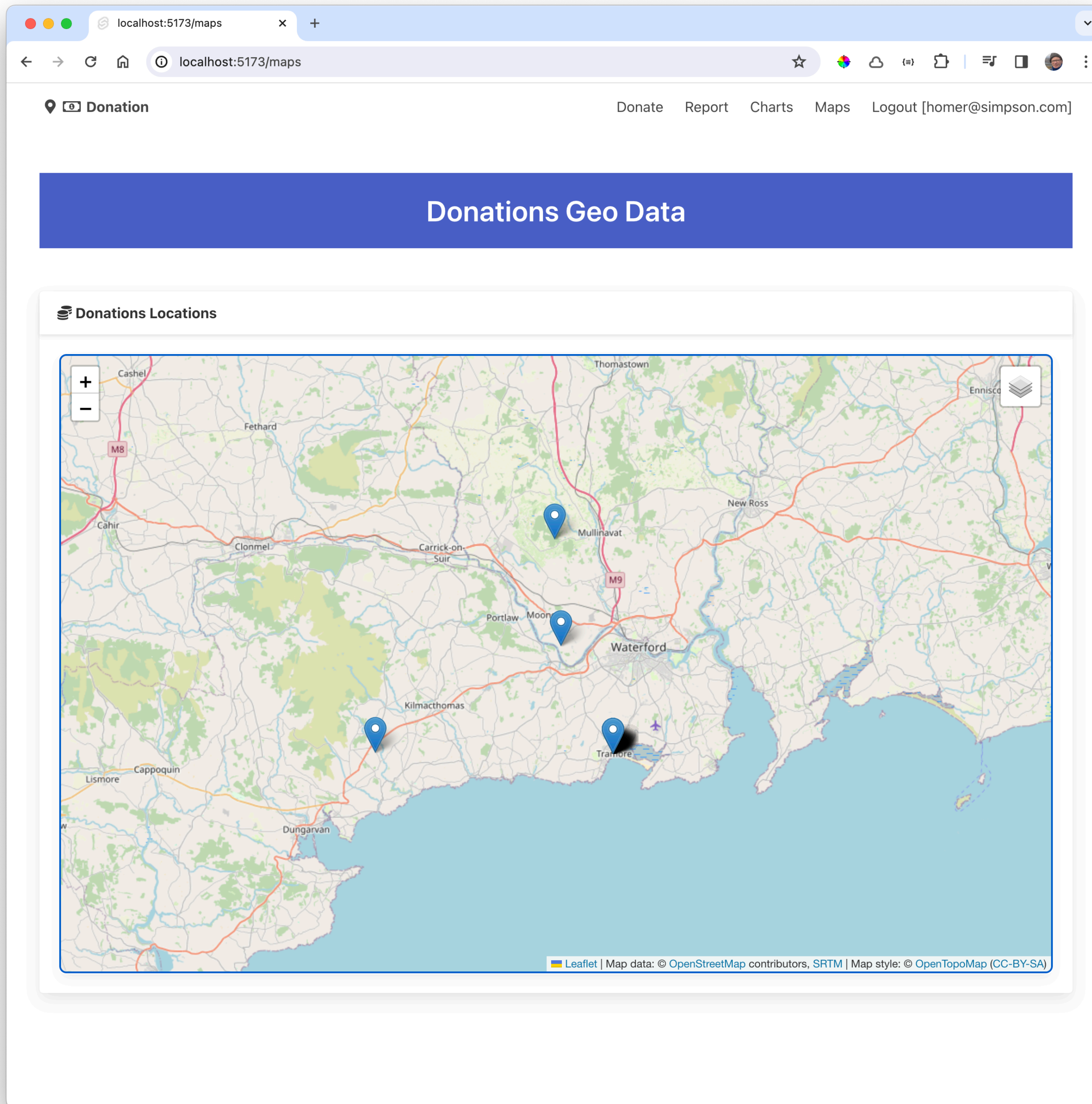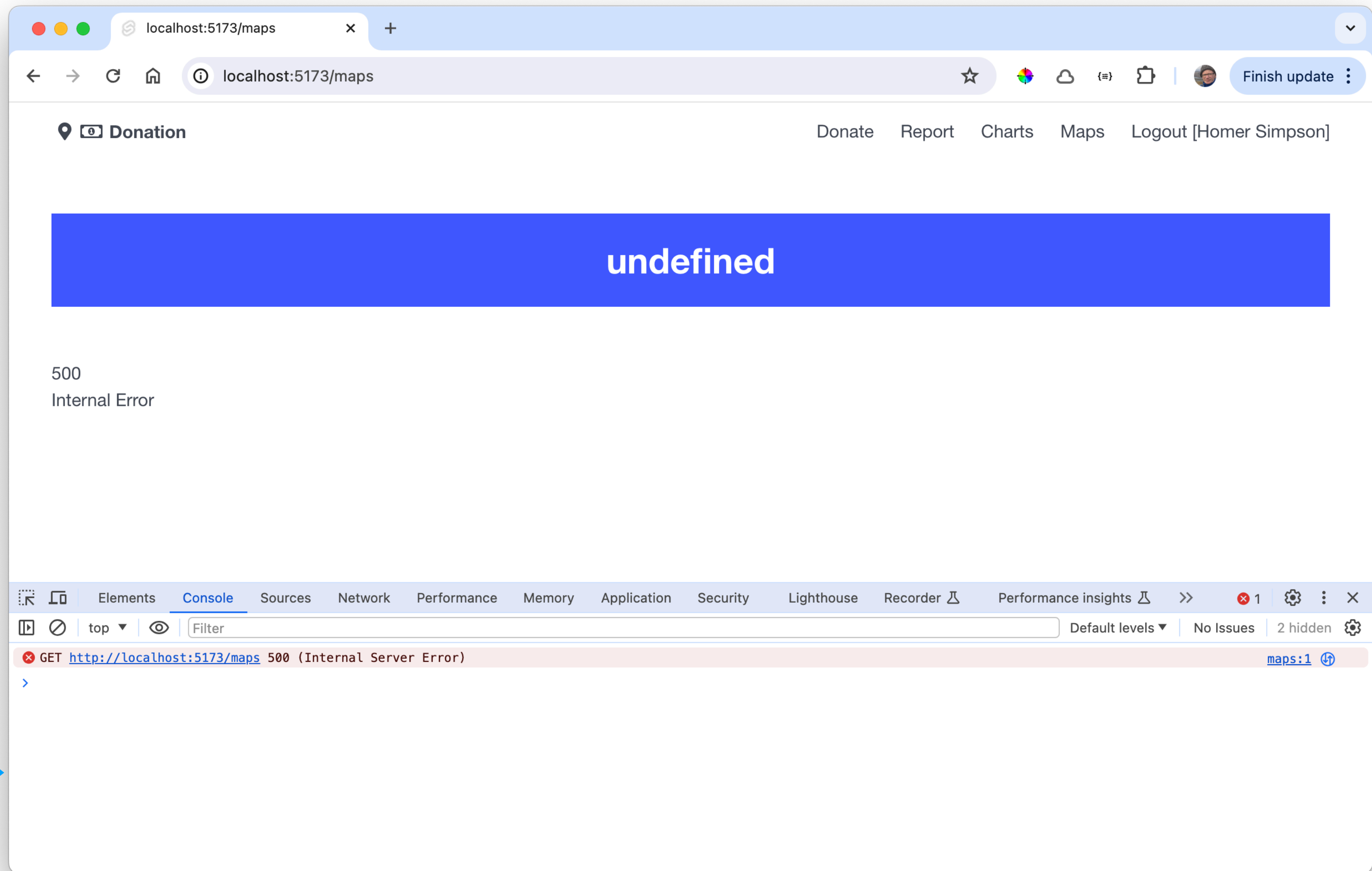
Maps Route

- Contains a LeafletMap component

- Leaflet is a 'vanilla' Javascript component - not a Svelte or SvelteKit aware

- Leaflet component requires a browser to initialise correctly.
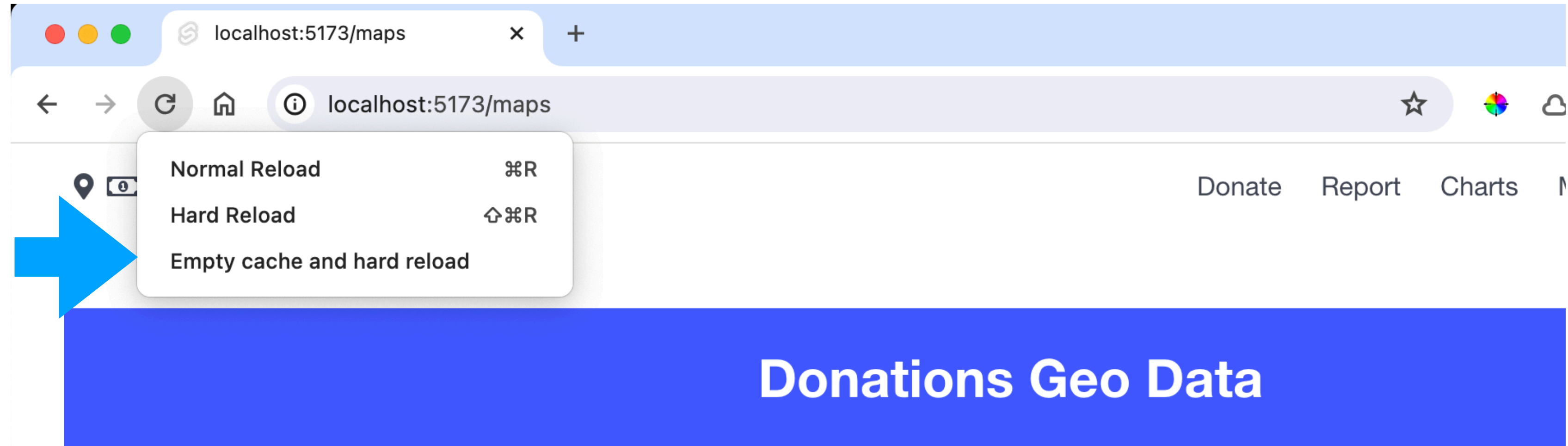
- Can be confused by SSR

Maps Route Error

- Maps will often fail to load on first visit to a page

- BUT: map load on subsequent visits

- Difficult to diagnose error, as browser onsole log not very informative

- Launch terminal error, however, is more useful:

- Force error by triggering 'Empty Cache and Hard Reload'

**- ReferenceError:
window is not defined**

- Indicates an attempt to load leaflet on the server (SSR).

- There are multiple ways of working around this error.

- Simplest solution is to disable SSR for a specific route

- Introduce a new file into the route called **+page.ts** with a single line disabling SSR:
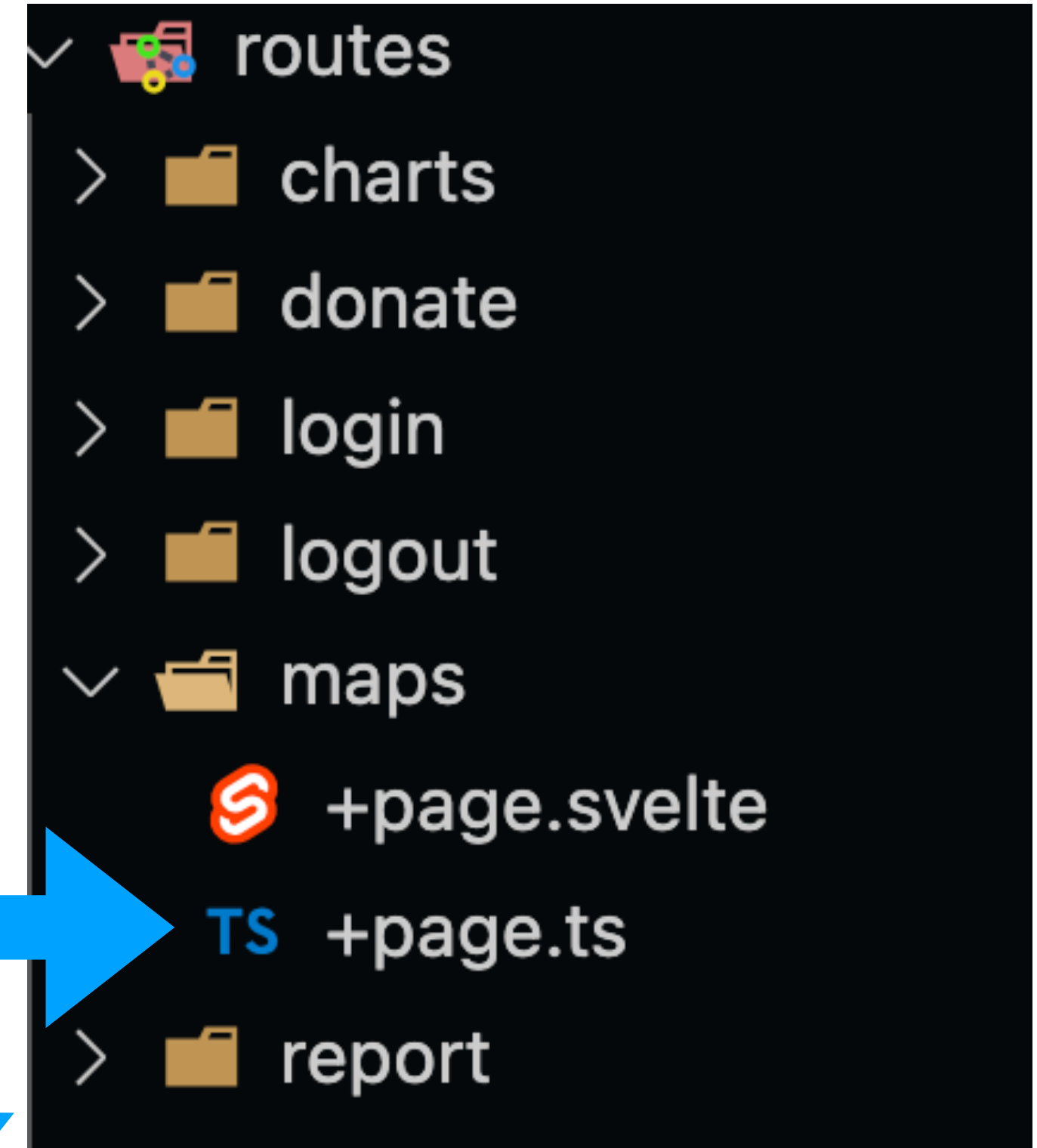
Disable SSR for a Route

```
export const ssr = false;
```

SSR & Leaflet

Leaflet Anomalies with SvelteKit SSR