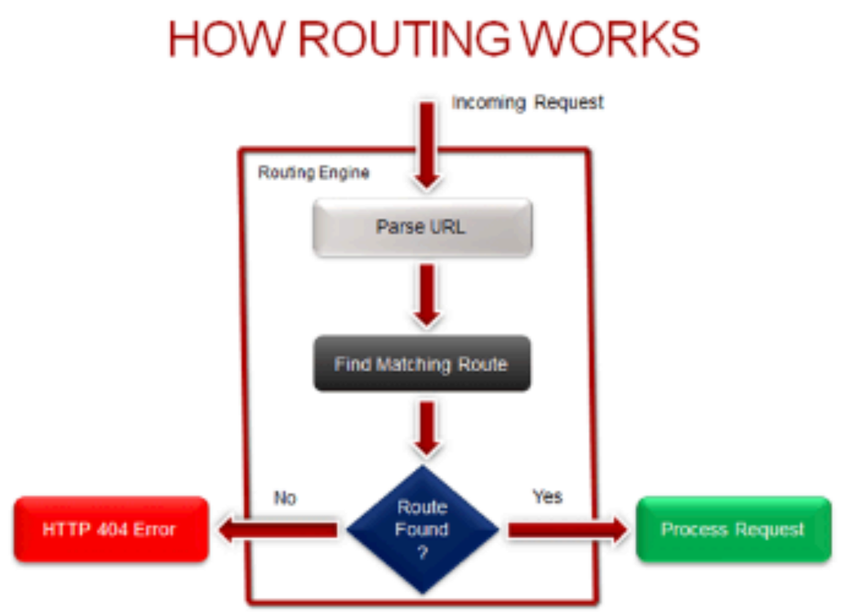
Routing

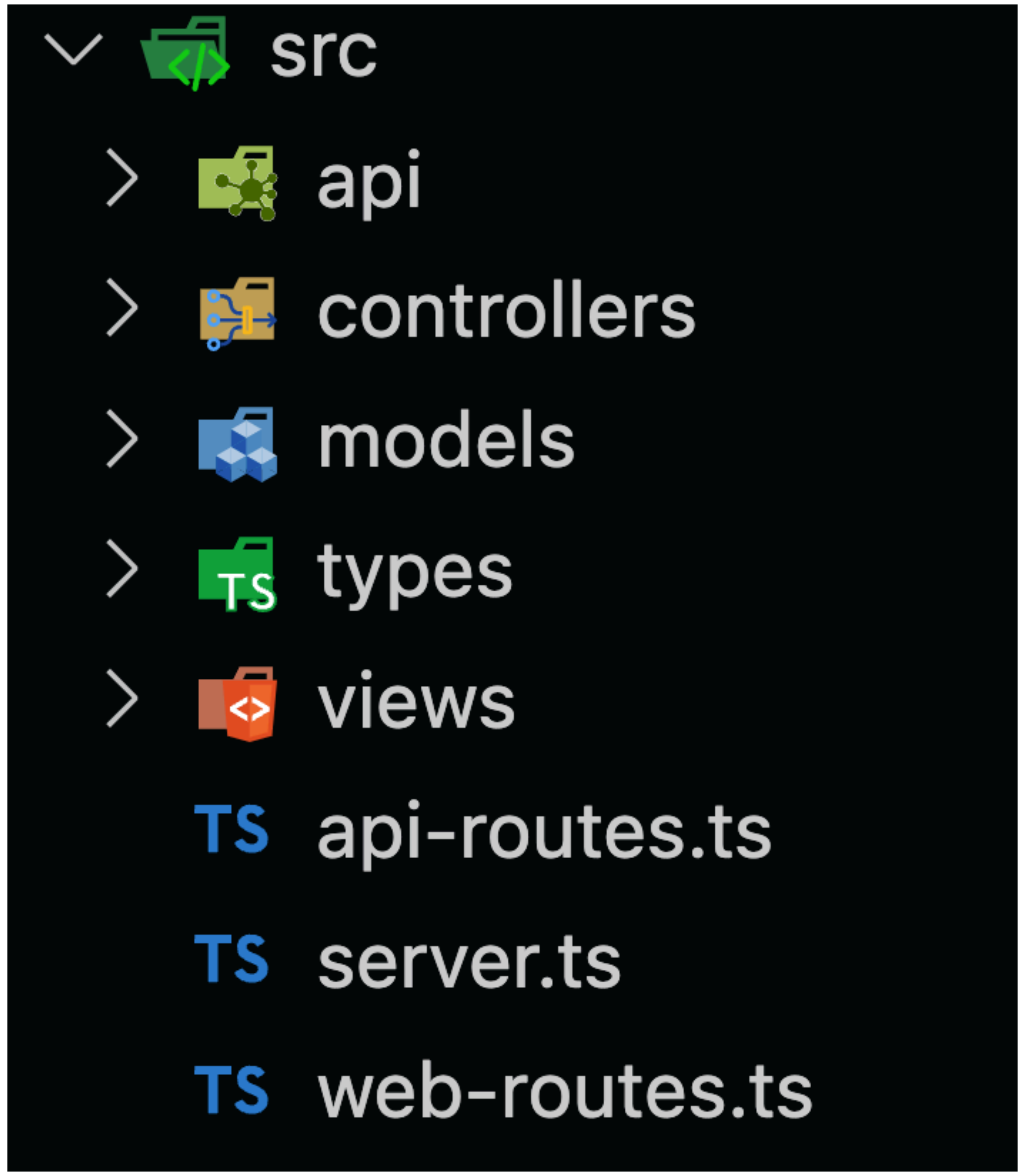


How Routing works in SvelteKit

Hapi Node App

SvelteKit Web Application

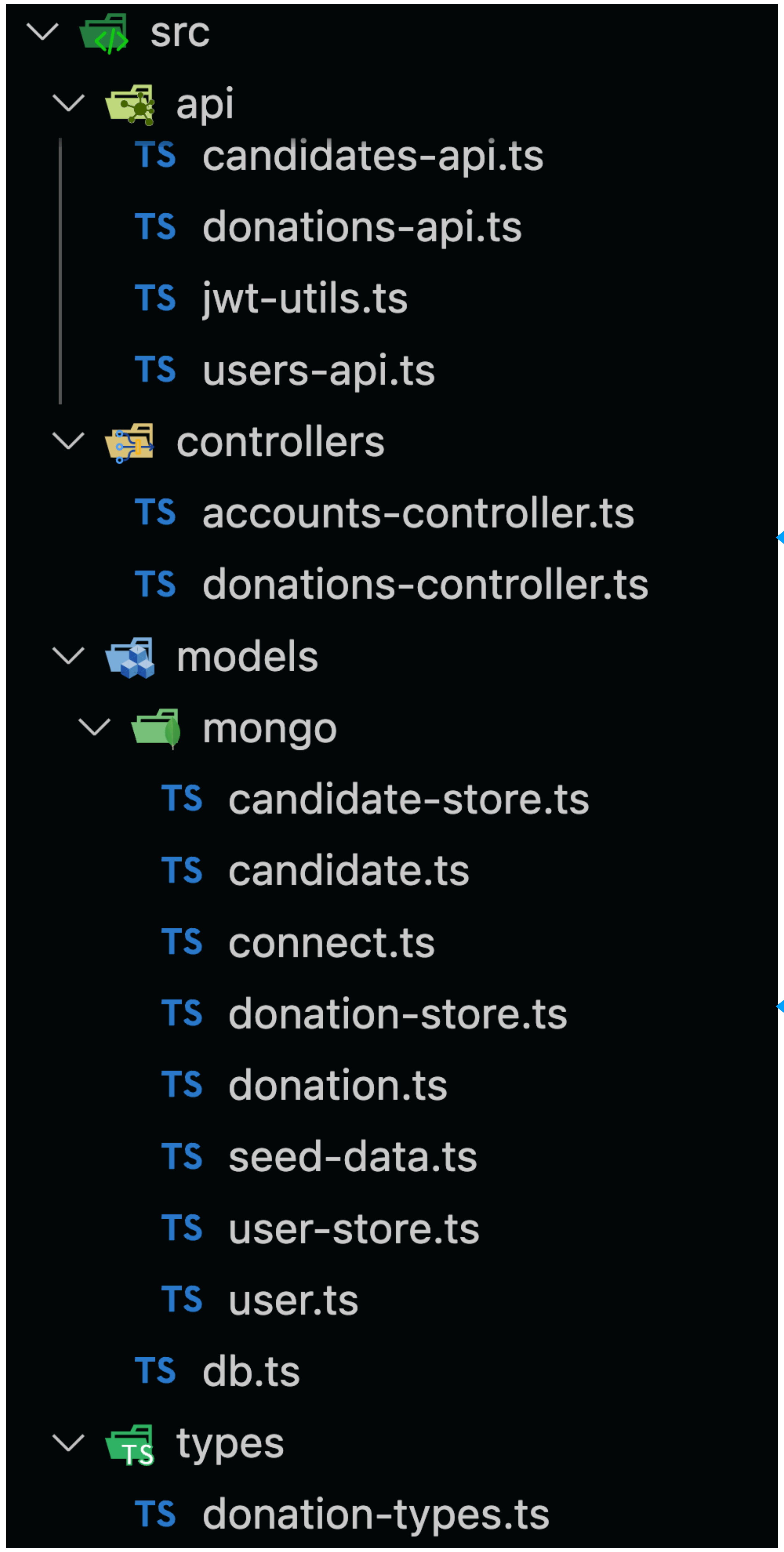
Hapi Node App



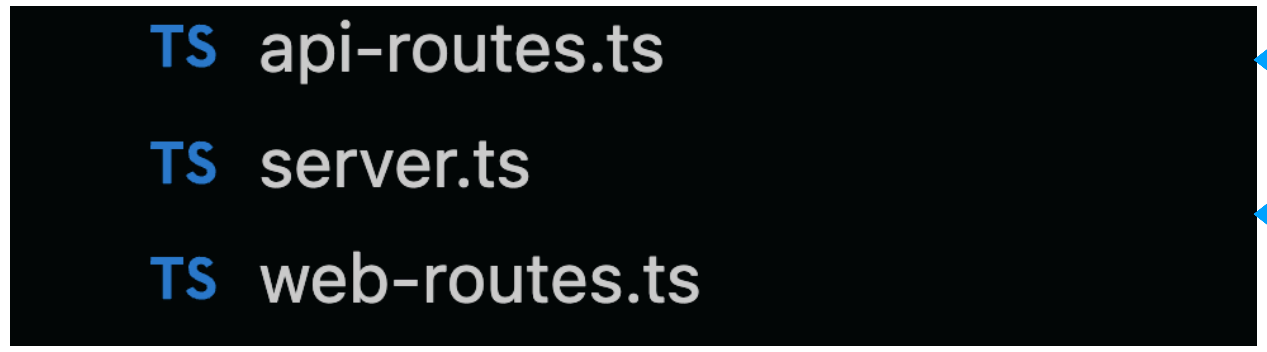
```
src
  api
  controllers
  models
  types
  views
  api-routes.ts
  server.ts
  web-routes.ts
```

- Server Rendered Application
- Node.js with Hapi Framework
- Hapi Plugins
- Npm Modules

Hapi Node App



```
src
  api
    candidates-api.ts
    donations-api.ts
    jwt-utils.ts
    users-api.ts
  controllers
    accounts-controller.ts
    donations-controller.ts
  models
    mongo
      candidate-store.ts
      candidate.ts
      connect.ts
      donation-store.ts
      donation.ts
      seed-data.ts
      user-store.ts
      user.ts
    db.ts
  types
    donation-types.ts
```



```
api-routes.ts
server.ts
web-routes.ts
```

Controllers

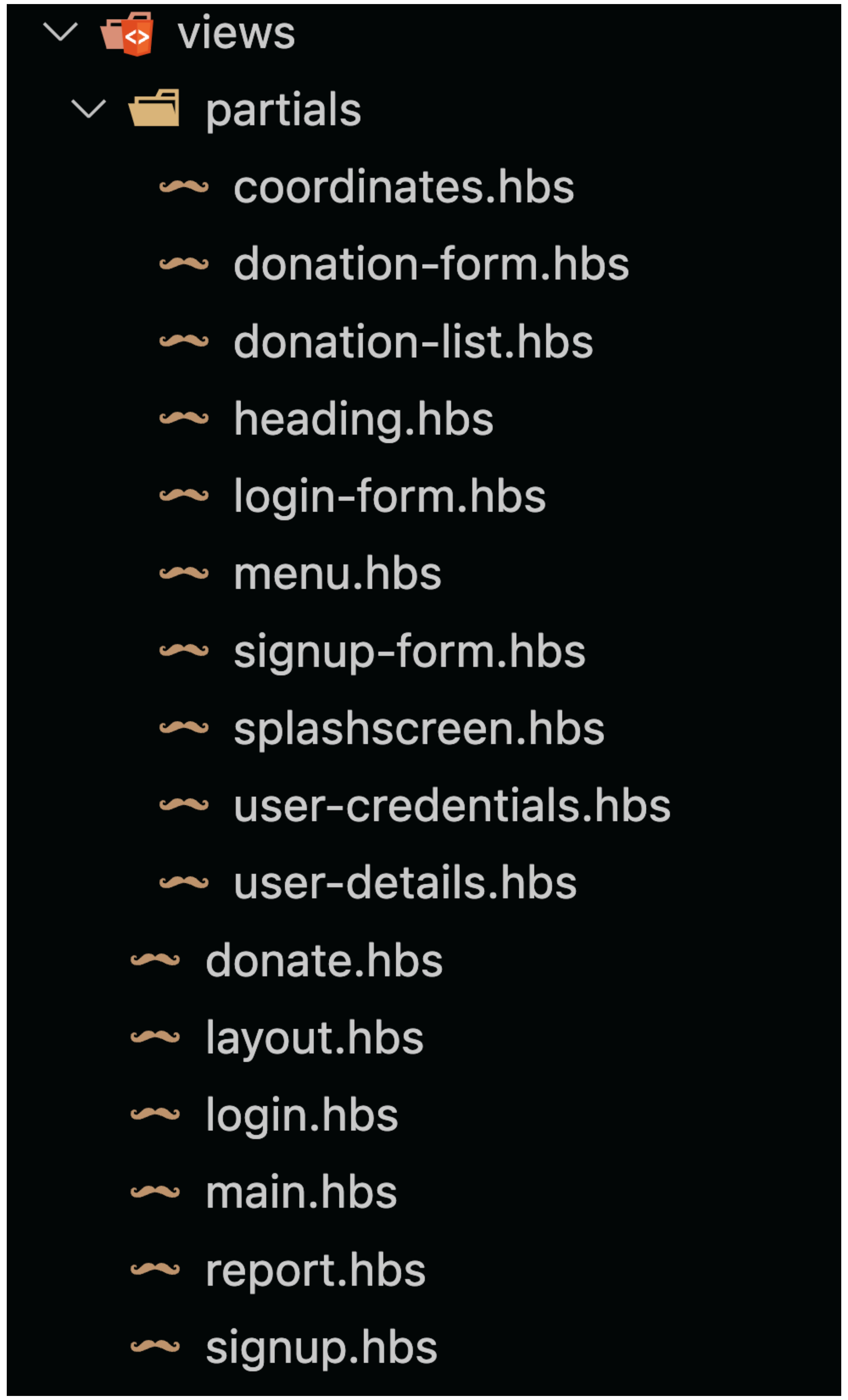
Models

Routes

Main

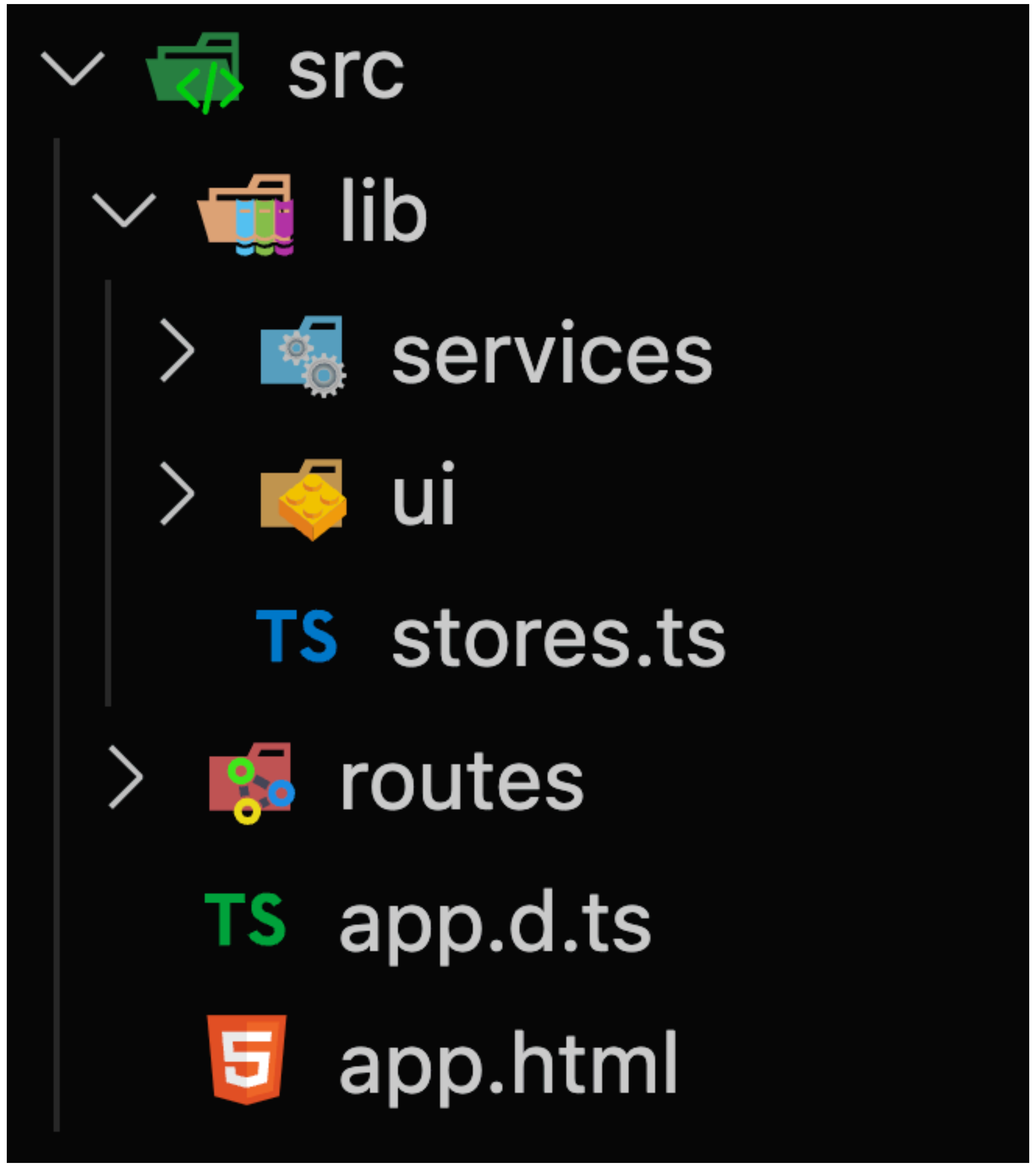
Components

Views



```
views
  partials
    coordinates.hbs
    donation-form.hbs
    donation-list.hbs
    heading.hbs
    login-form.hbs
    menu.hbs
    signup-form.hbs
    splashscreen.hbs
    user-credentials.hbs
    user-details.hbs
  donate.hbs
  layout.hbs
  login.hbs
  main.hbs
  report.hbs
  signup.hbs
```

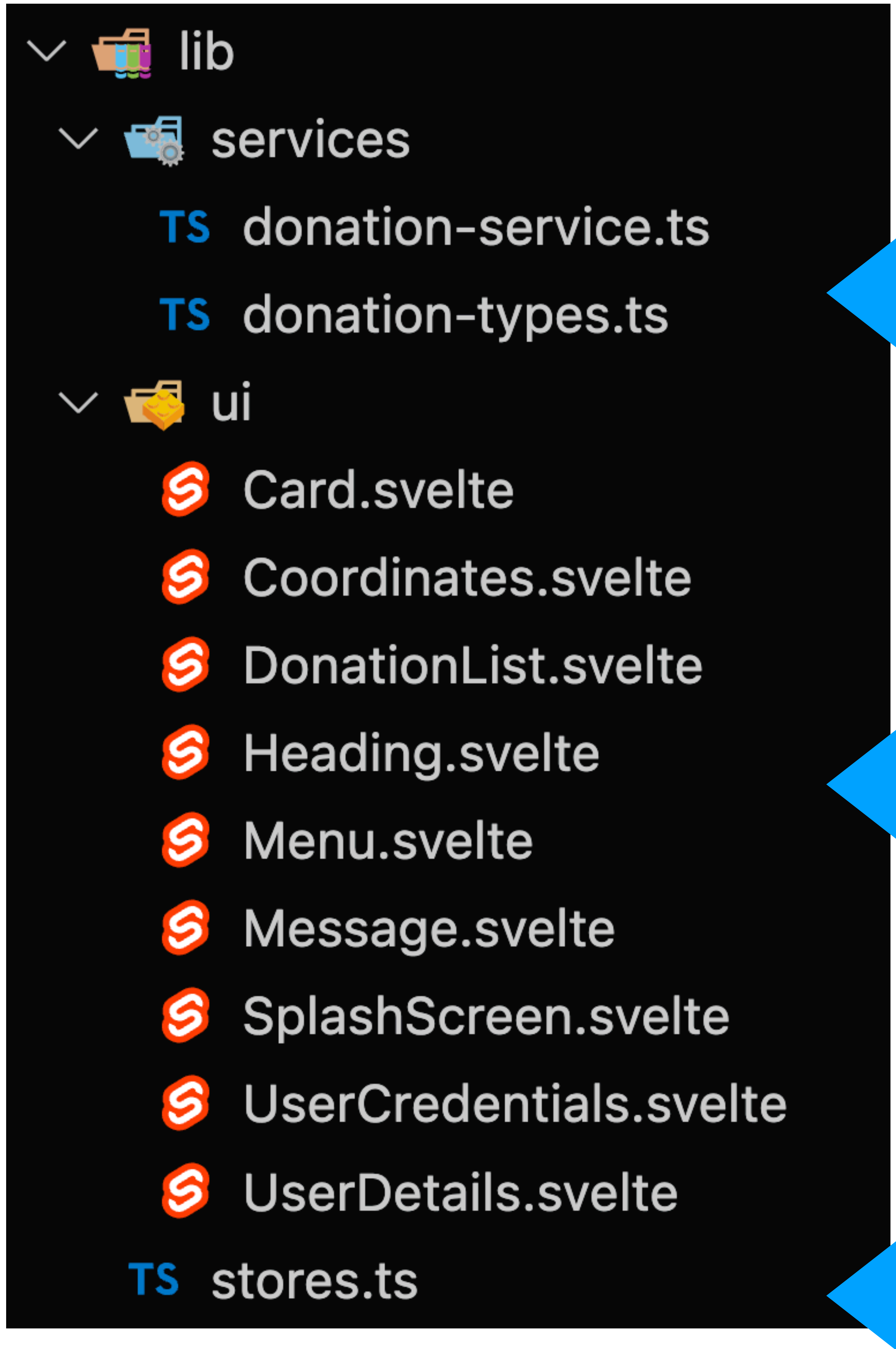
SvelteKit Web Application



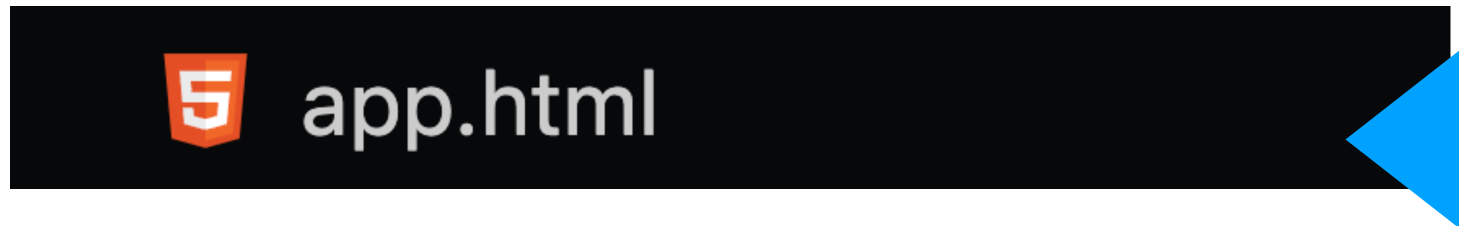
```
src
  lib
    services
    ui
    stores.ts
  routes
  app.d.ts
  app.html
```

- Front End Javascript Application
- SvelteKit Framework
- Interacts with remote services over HTTP (REST APIs)
- Npm Modules

SvelteKit Web Application



```
lib
  services
    donation-service.ts
    donation-types.ts
  ui
    Card.svelte
    Coordinates.svelte
    DonationList.svelte
    Heading.svelte
    Menu.svelte
    Message.svelte
    SplashScreen.svelte
    UserCredentials.svelte
    UserDetails.svelte
  stores.ts
```



```
app.html
```

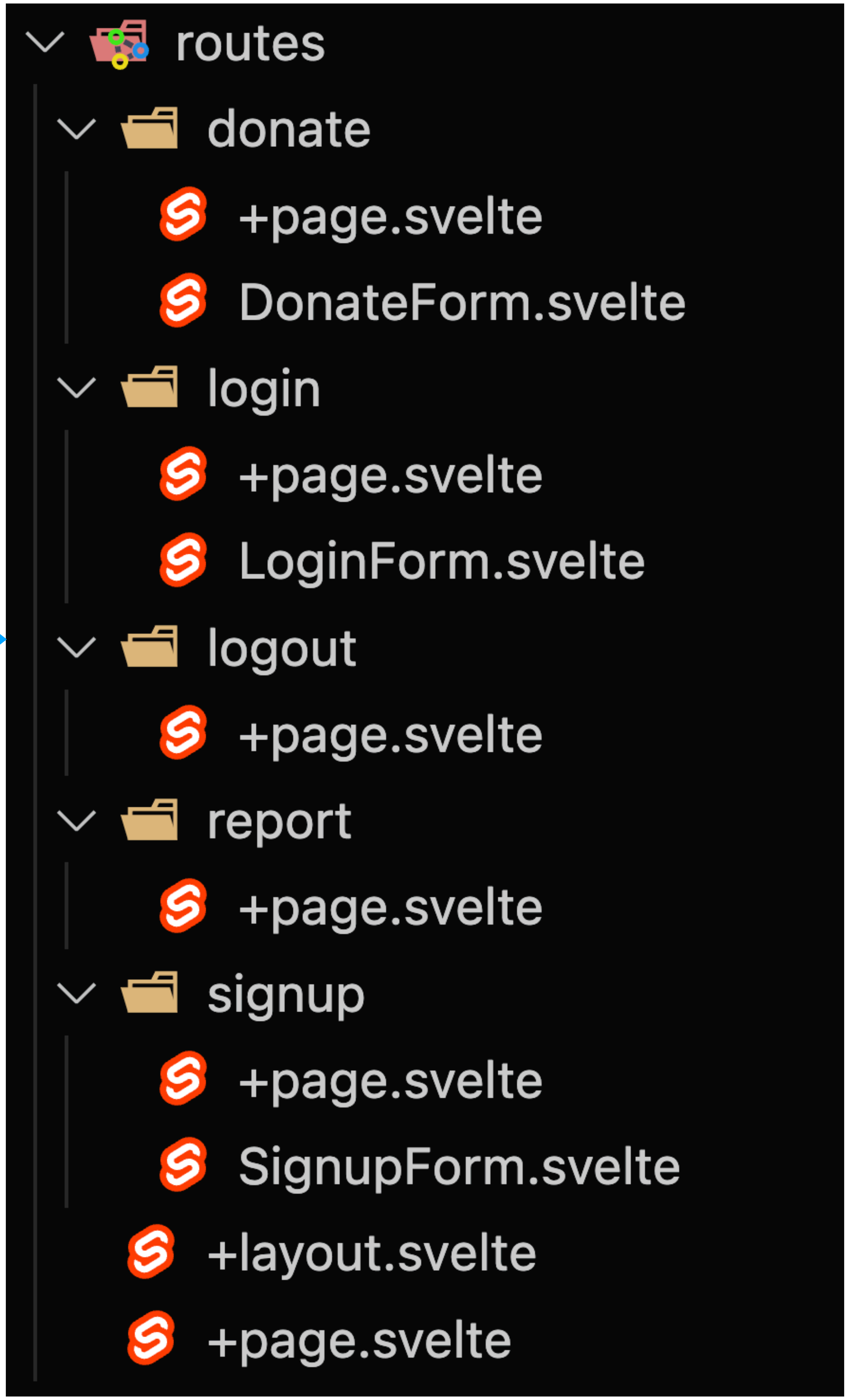
Interface to Remote API

Components

Shared Stores

Main

Routes

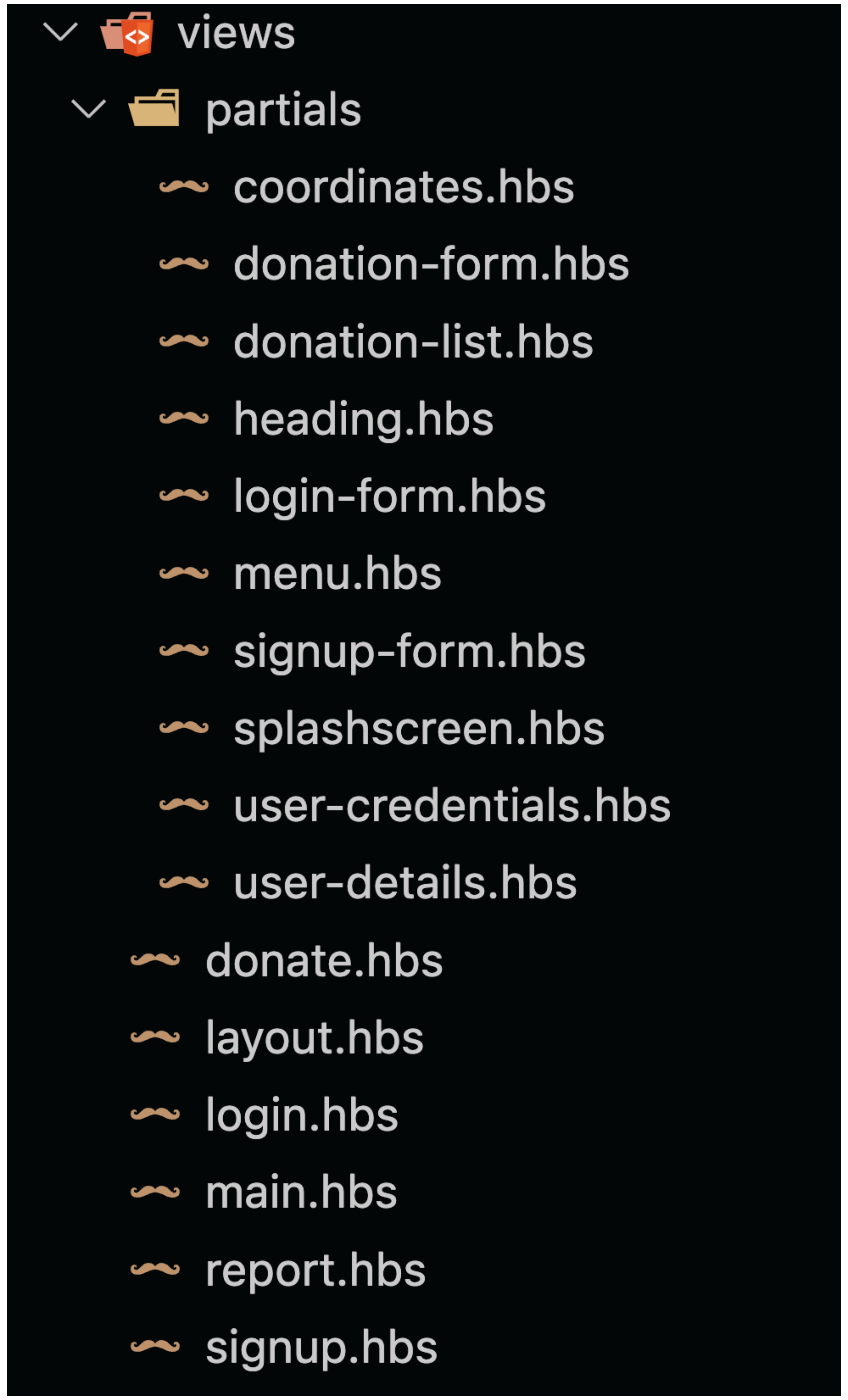


```
routes
  donate
    +page.svelte
    DonateForm.svelte
  login
    +page.svelte
    LoginForm.svelte
  logout
    +page.svelte
  report
    +page.svelte
  signup
    +page.svelte
    SignupForm.svelte
  +layout.svelte
  +page.svelte
```

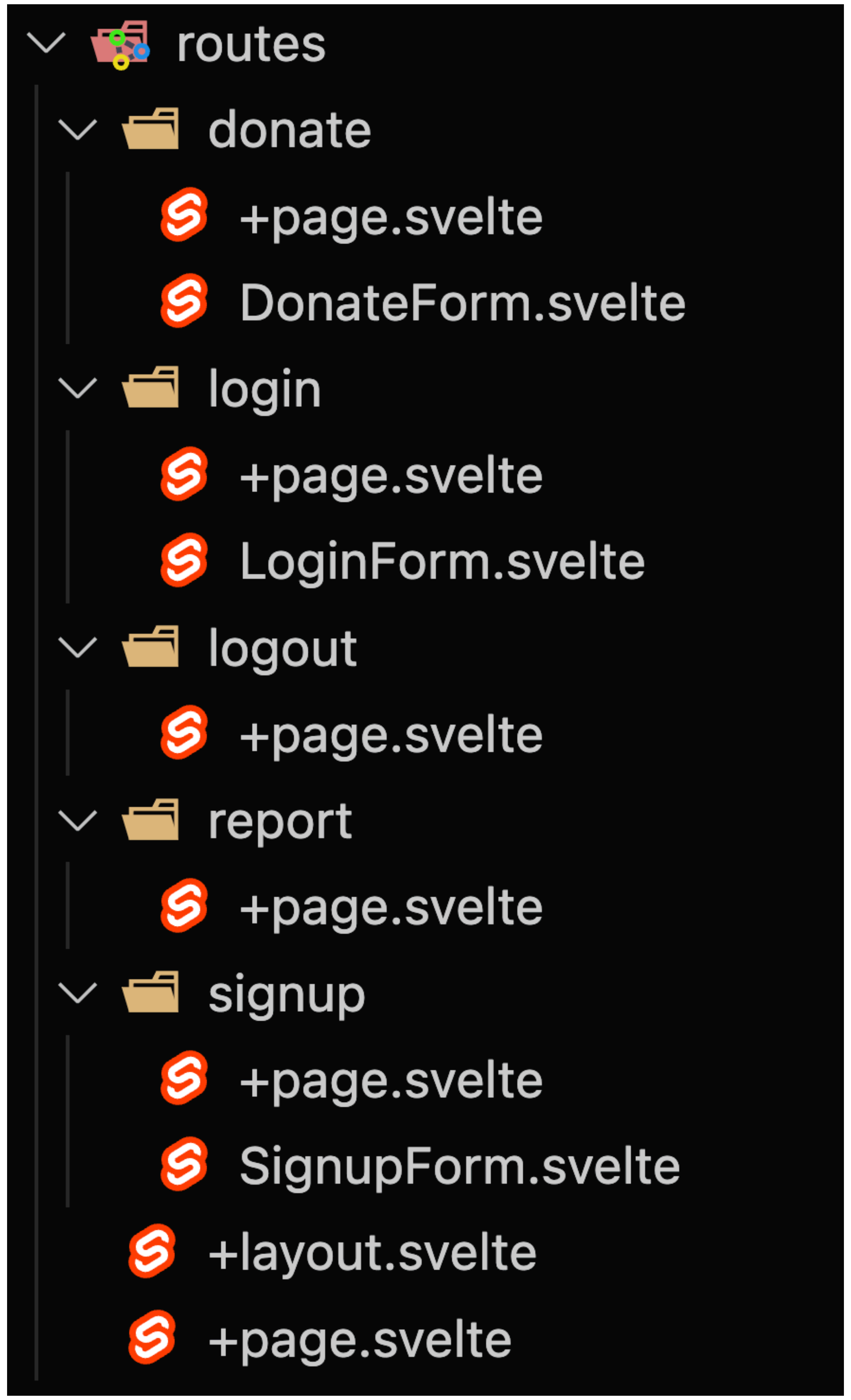
Hapi Node App

SvelteKit Web Application

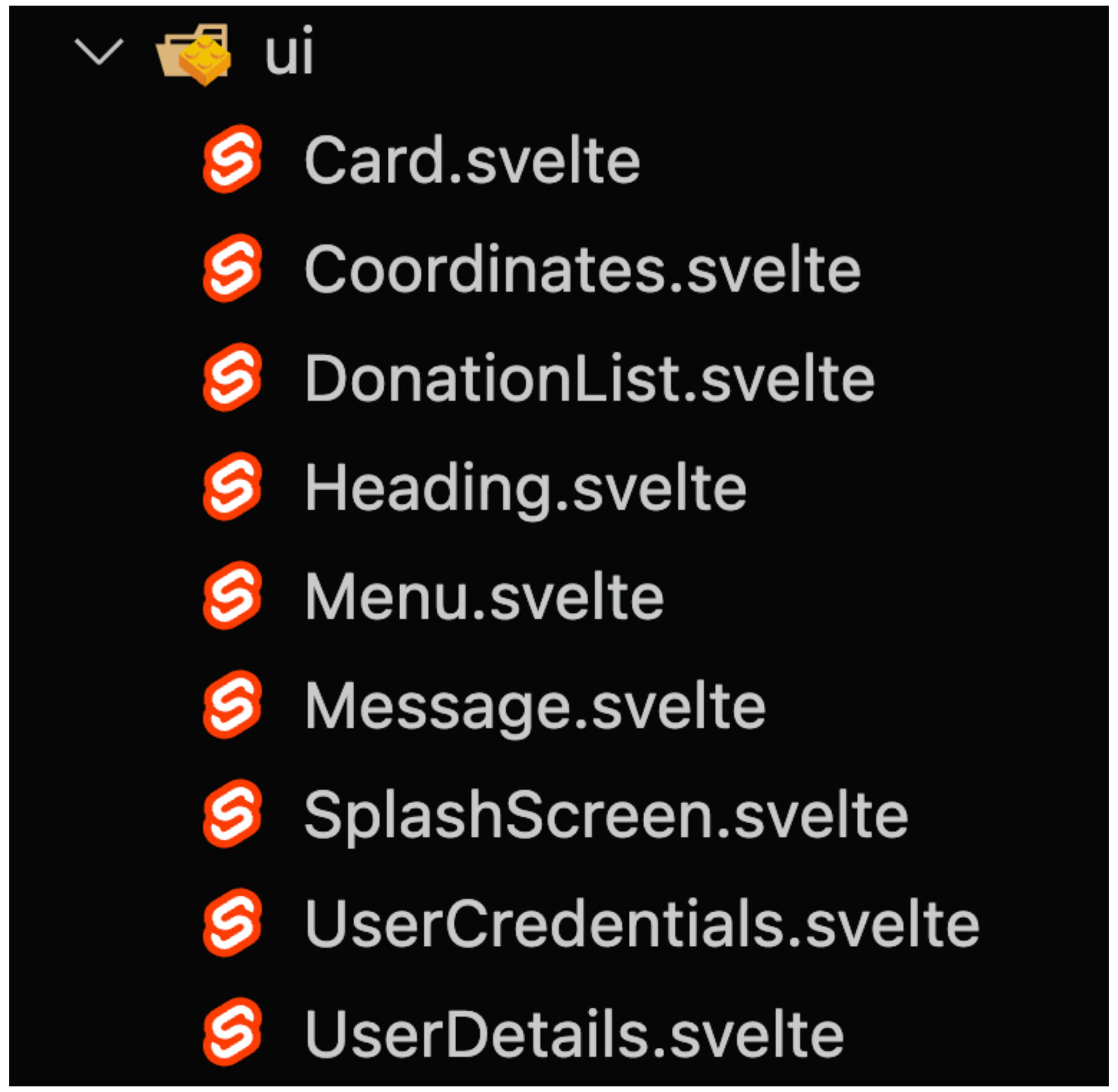
Handlebars Visual Components & Views



```
views
  partials
    coordinates.hbs
    donation-form.hbs
    donation-list.hbs
    heading.hbs
    login-form.hbs
    menu.hbs
    signup-form.hbs
    splashscreen.hbs
    user-credentials.hbs
    user-details.hbs
  donate.hbs
  layout.hbs
  login.hbs
  main.hbs
  report.hbs
  signup.hbs
```



```
routes
  donate
    +page.svelte
    DonateForm.svelte
  login
    +page.svelte
    LoginForm.svelte
  logout
    +page.svelte
  report
    +page.svelte
  signup
    +page.svelte
    SignupForm.svelte
  +layout.svelte
  +page.svelte
```

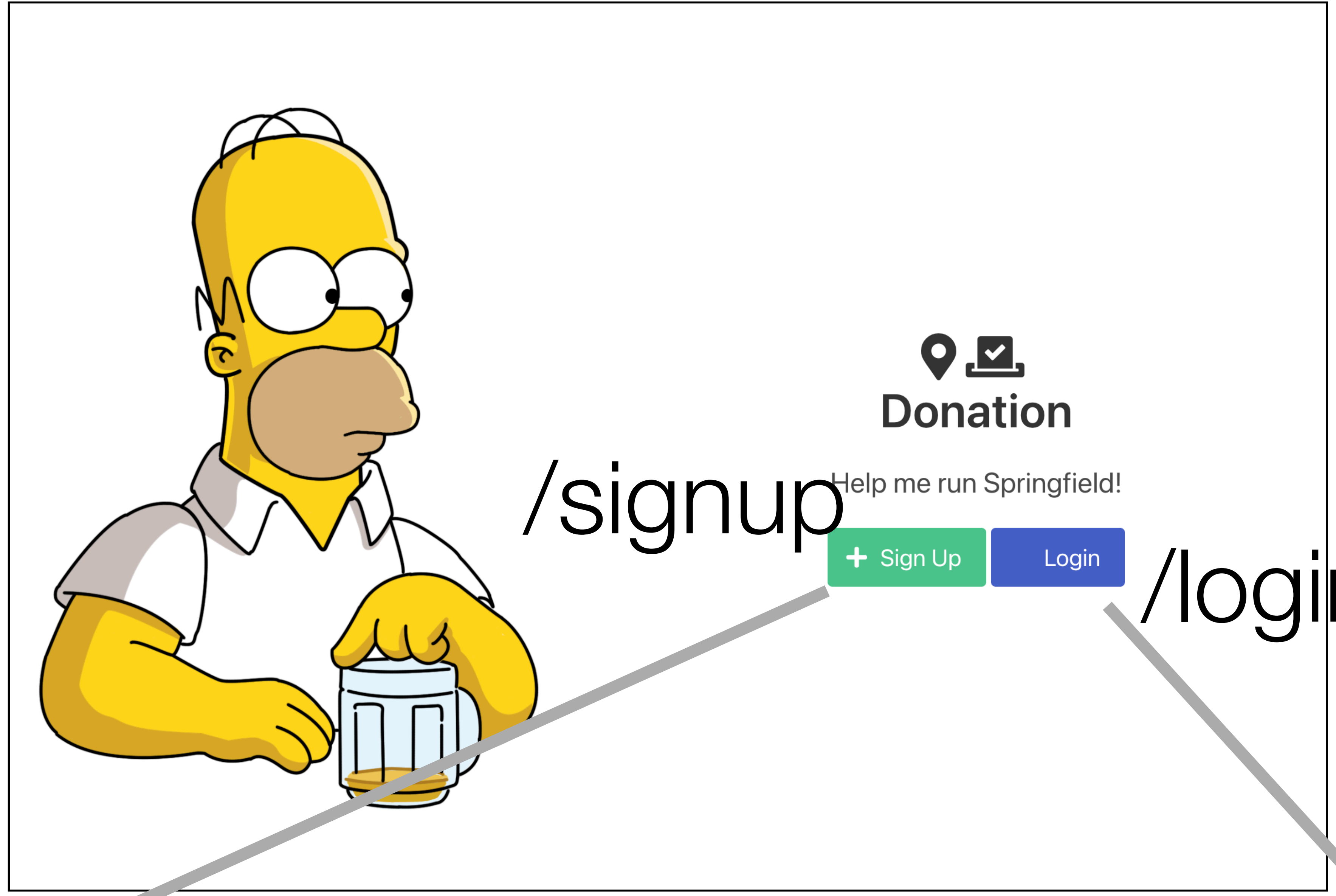


```
ui
  Card.svelte
  Coordinates.svelte
  DonationList.svelte
  Heading.svelte
  Menu.svelte
  Message.svelte
  SplashScreen.svelte
  UserCredentials.svelte
  UserDetails.svelte
```

Svelte Visual Components & Views

Hapi Node App

Donation HAPI Application

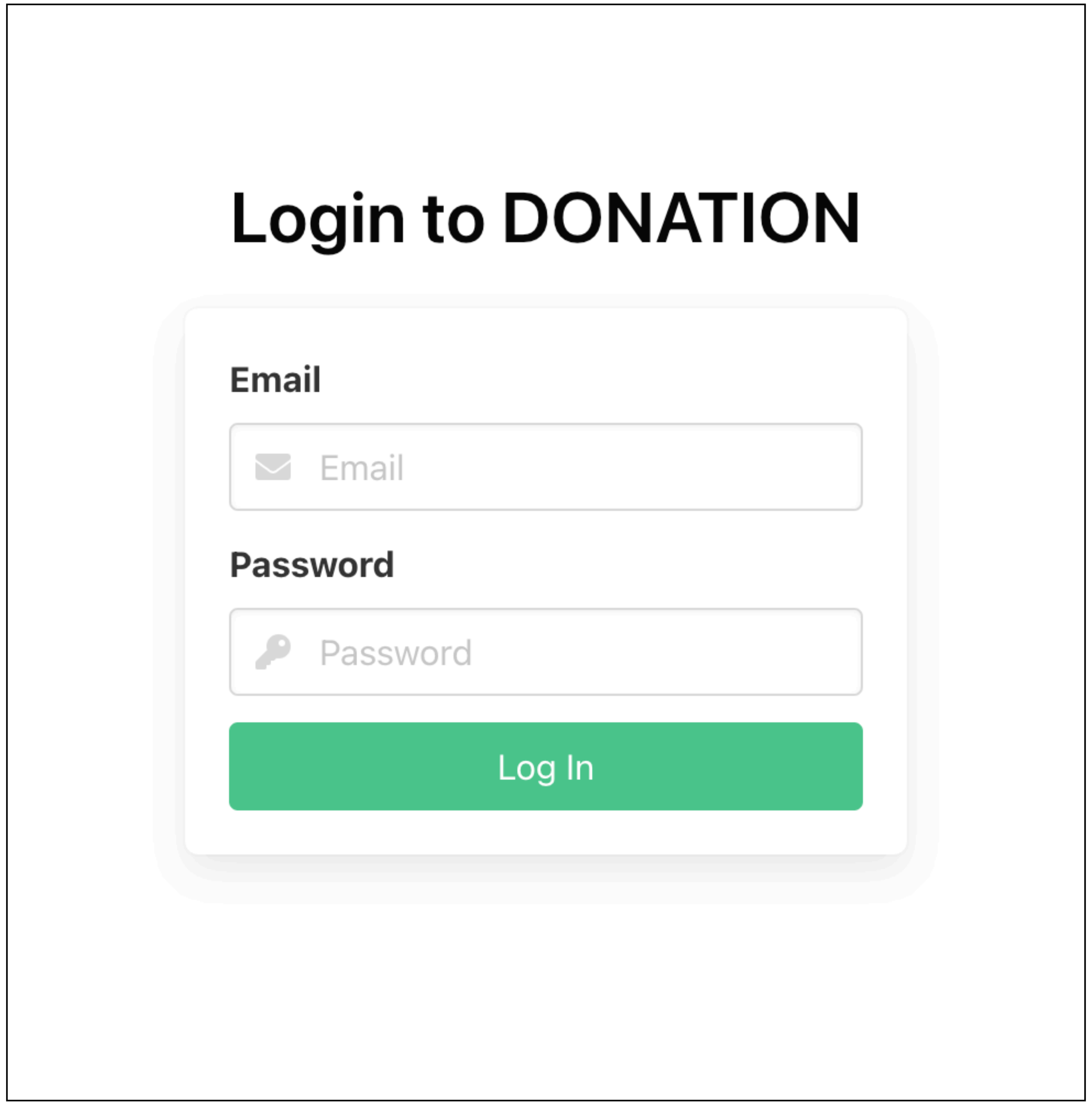


Donation

Help me run Springfield!

/signup

/login

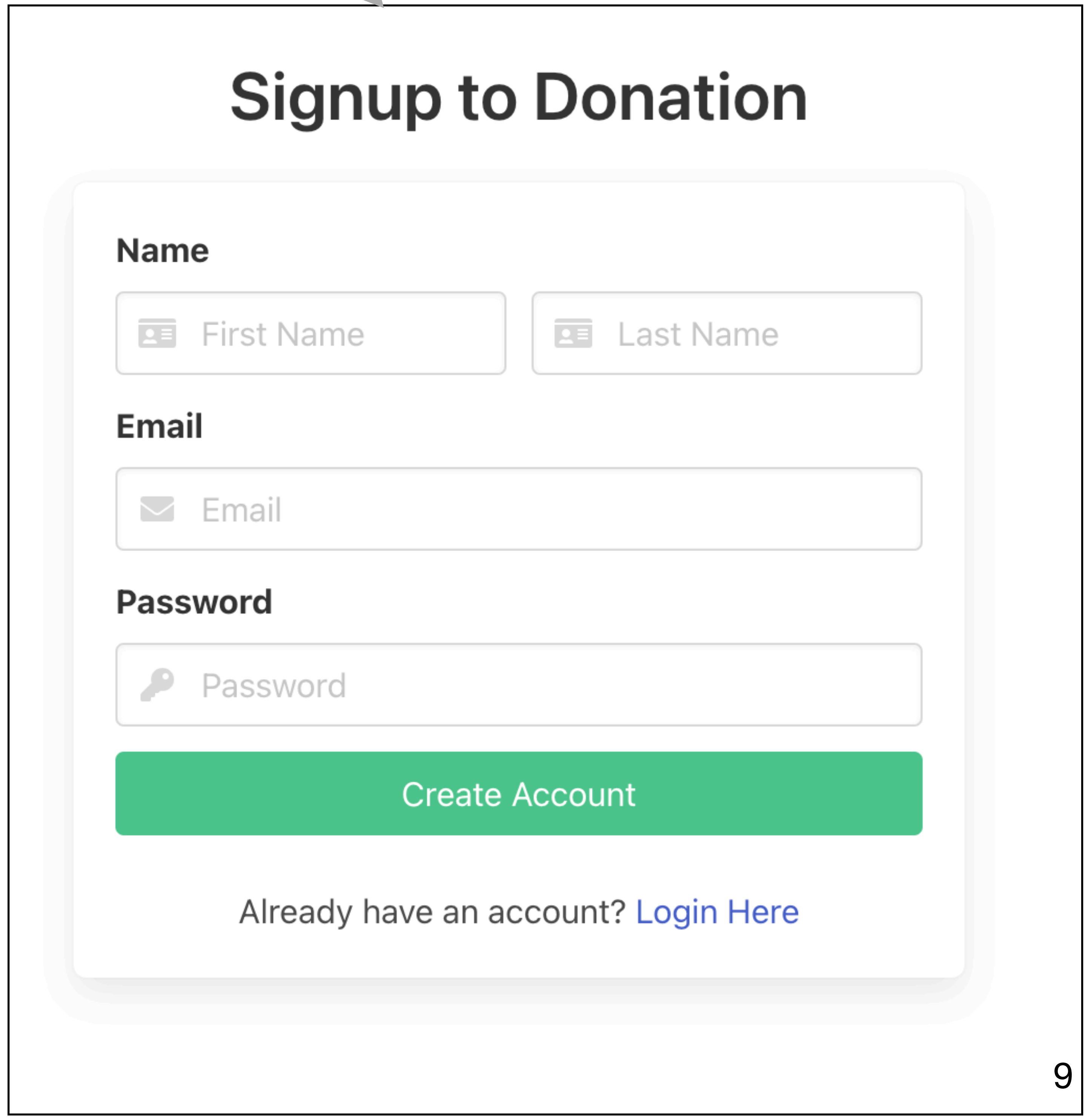


Login to DONATION

Email

Password

- Routing an intrinsic part of Server Rendered Applications (eg Hapi)



Signup to Donation

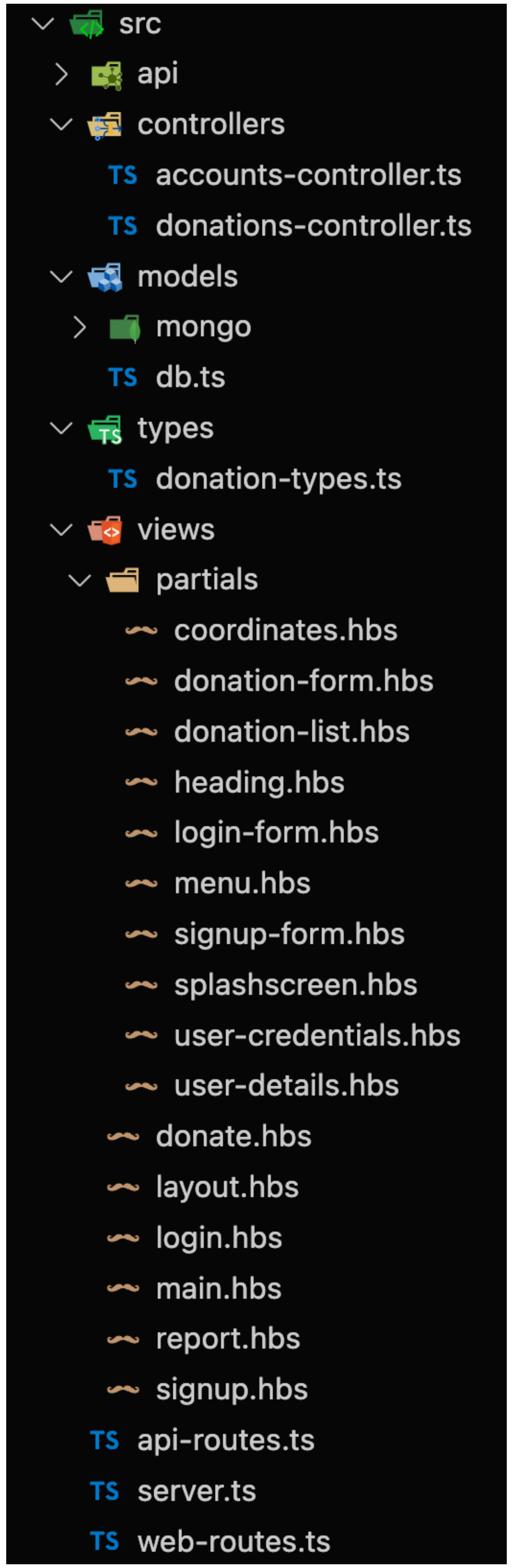
Name

Email

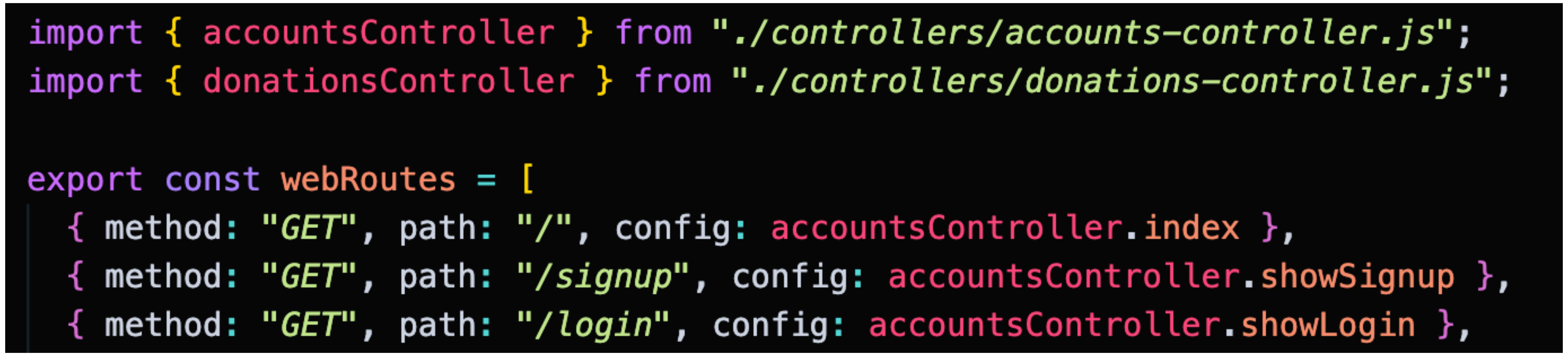
Password

Already have an account? Login Here

- Hapi Routes specified in a table

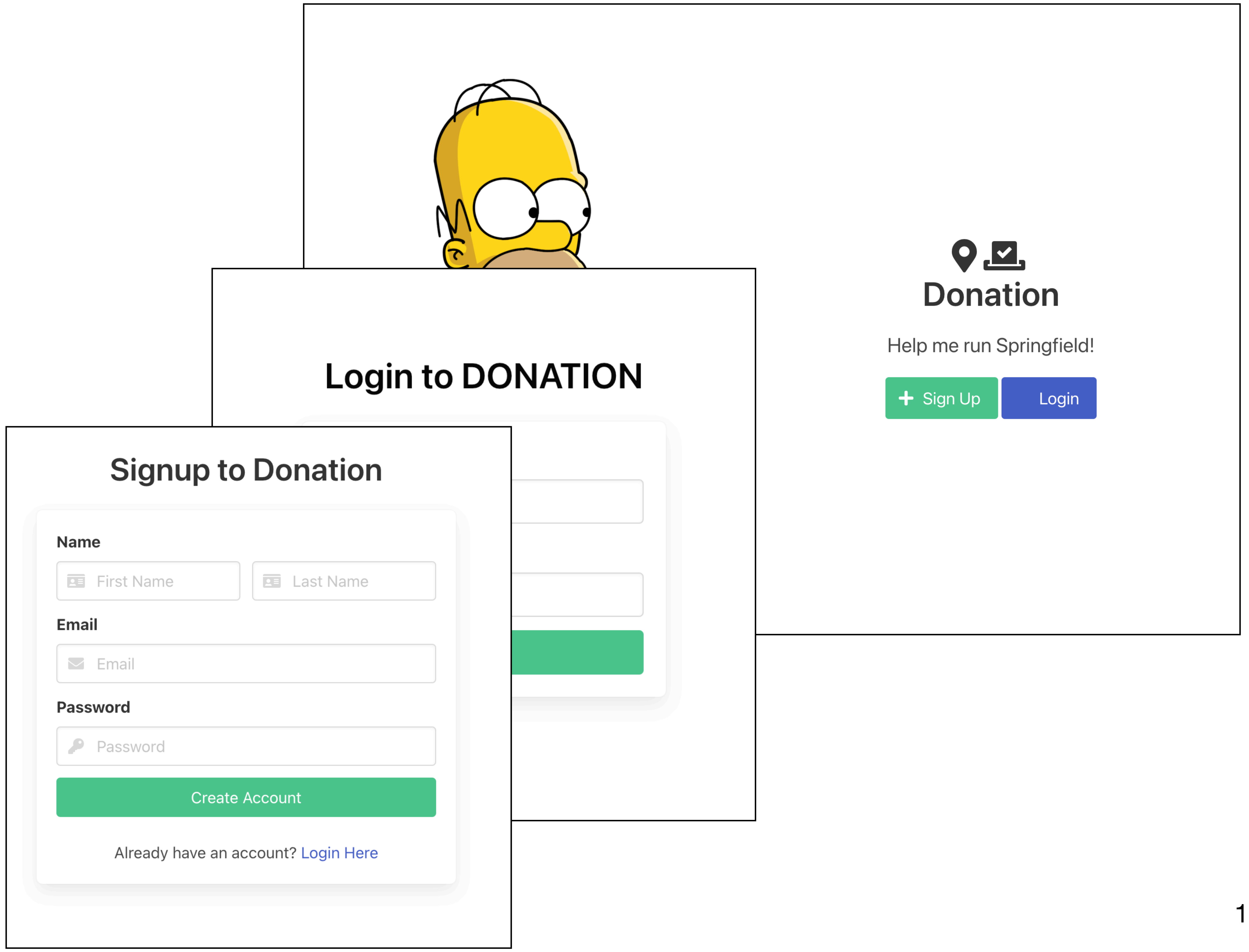


```
src
  api
  controllers
    TS accounts-controller.ts
    TS donations-controller.ts
  models
    mongo
    TS db.ts
  types
    TS donation-types.ts
  views
    partials
      coordinates.hbs
      donation-form.hbs
      donation-list.hbs
      heading.hbs
      login-form.hbs
      menu.hbs
      signup-form.hbs
      splashscreen.hbs
      user-credentials.hbs
      user-details.hbs
    donate.hbs
    layout.hbs
    login.hbs
    main.hbs
    report.hbs
    signup.hbs
  TS api-routes.ts
  TS server.ts
  TS web-routes.ts
```



```
import { accountsController } from "./controllers/accounts-controller.js";
import { donationsController } from "./controllers/donations-controller.js";

export const webRoutes = [
  { method: "GET", path: "/", config: accountsController.index },
  { method: "GET", path: "/signup", config: accountsController.showSignup },
  { method: "GET", path: "/login", config: accountsController.showLogin },
```



Donation

Help me run Springfield!

Login to DONATION

Signup to Donation

Name

First Name Last Name

Email

Email

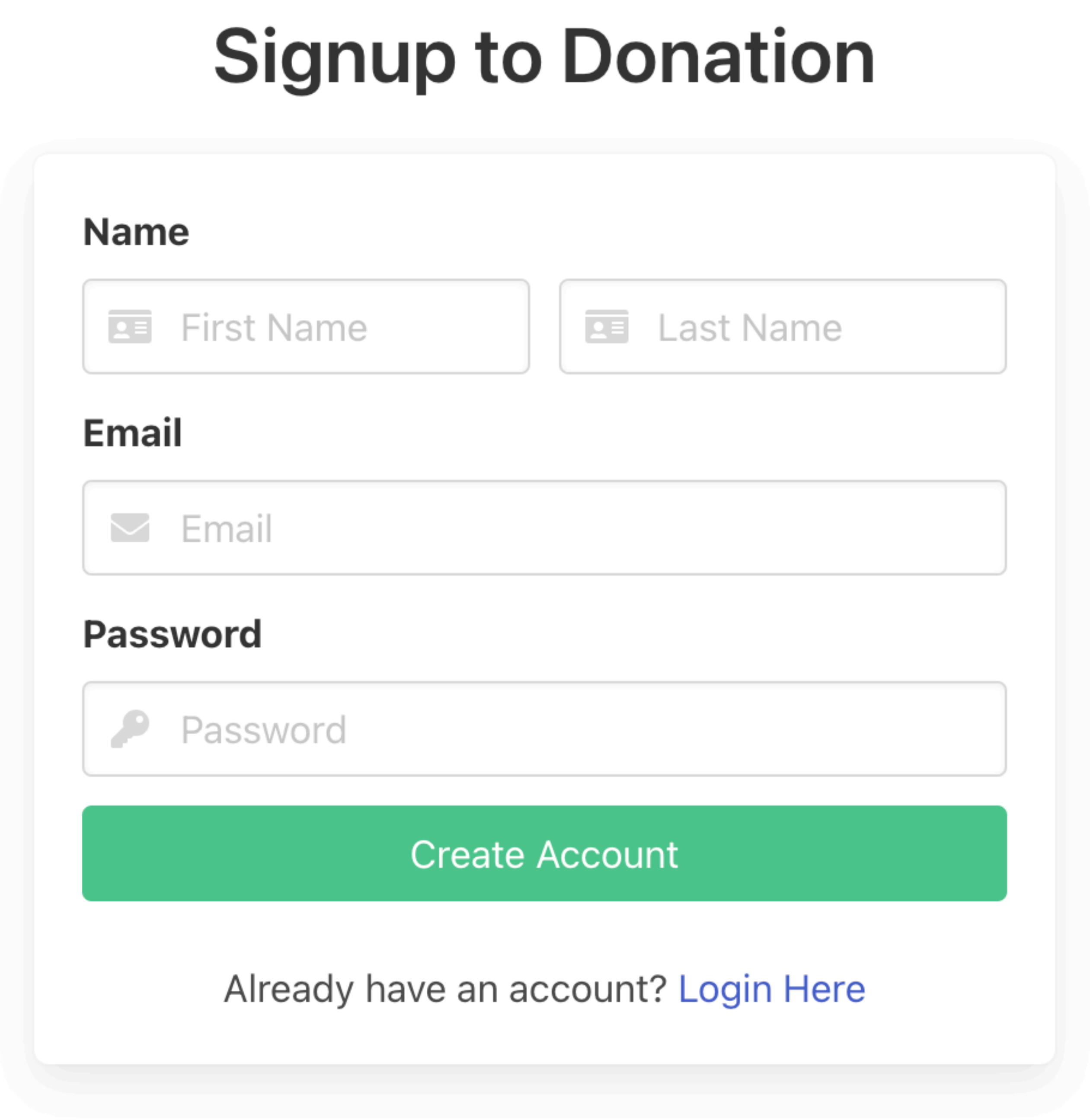
Password

Password

Already have an account? Login Here

- Router triggers Controller

- Controller loads View



Signup to Donation

Name

First Name | Last Name

Email

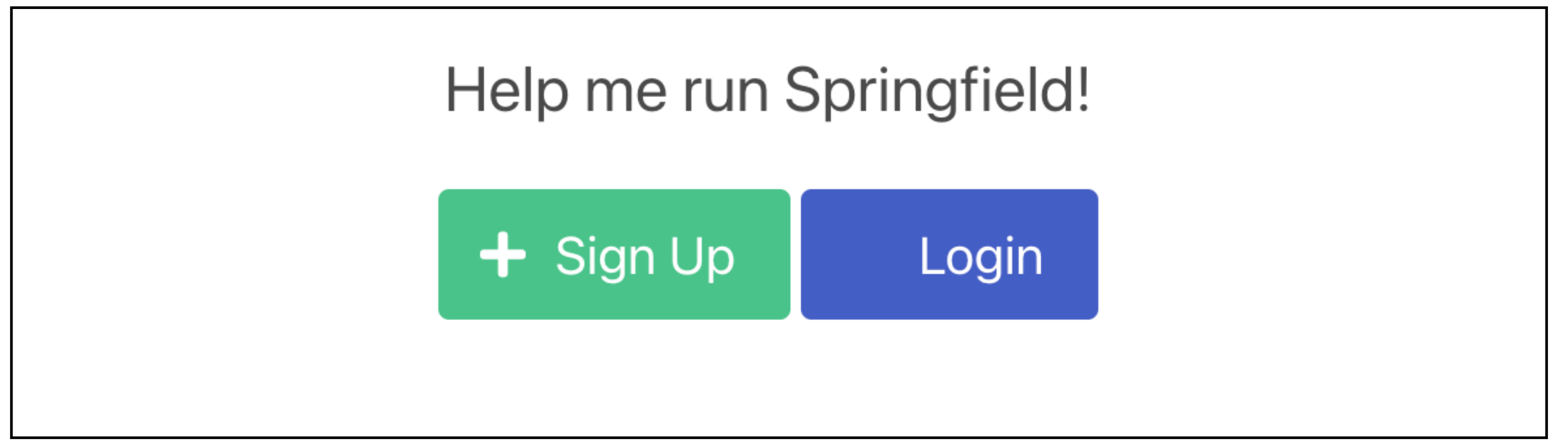
Email

Password

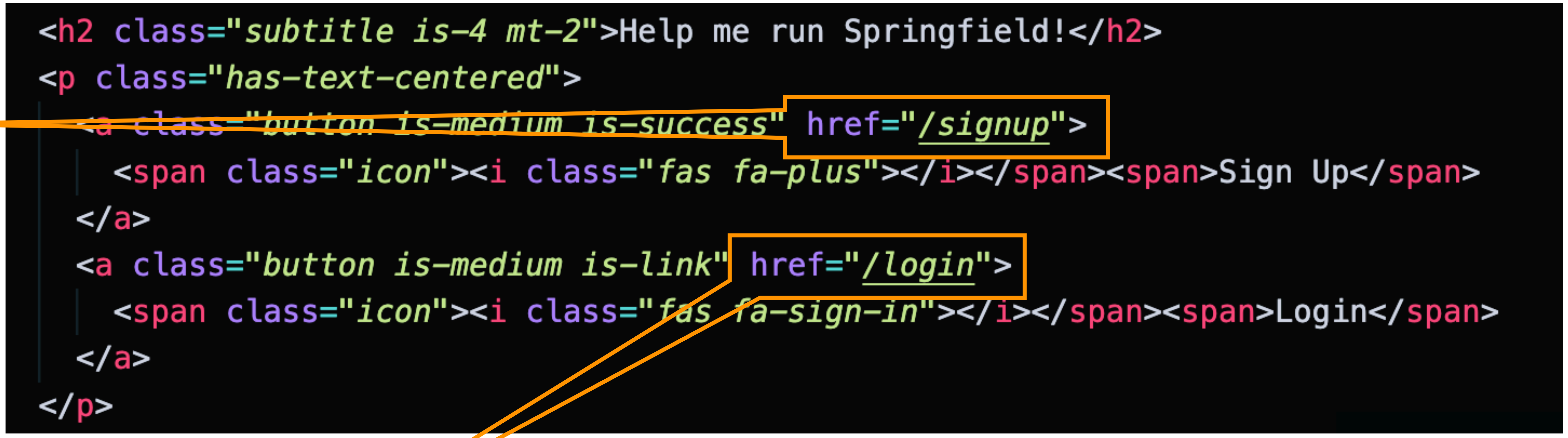
Password

Create Account

Already have an account? Login Here



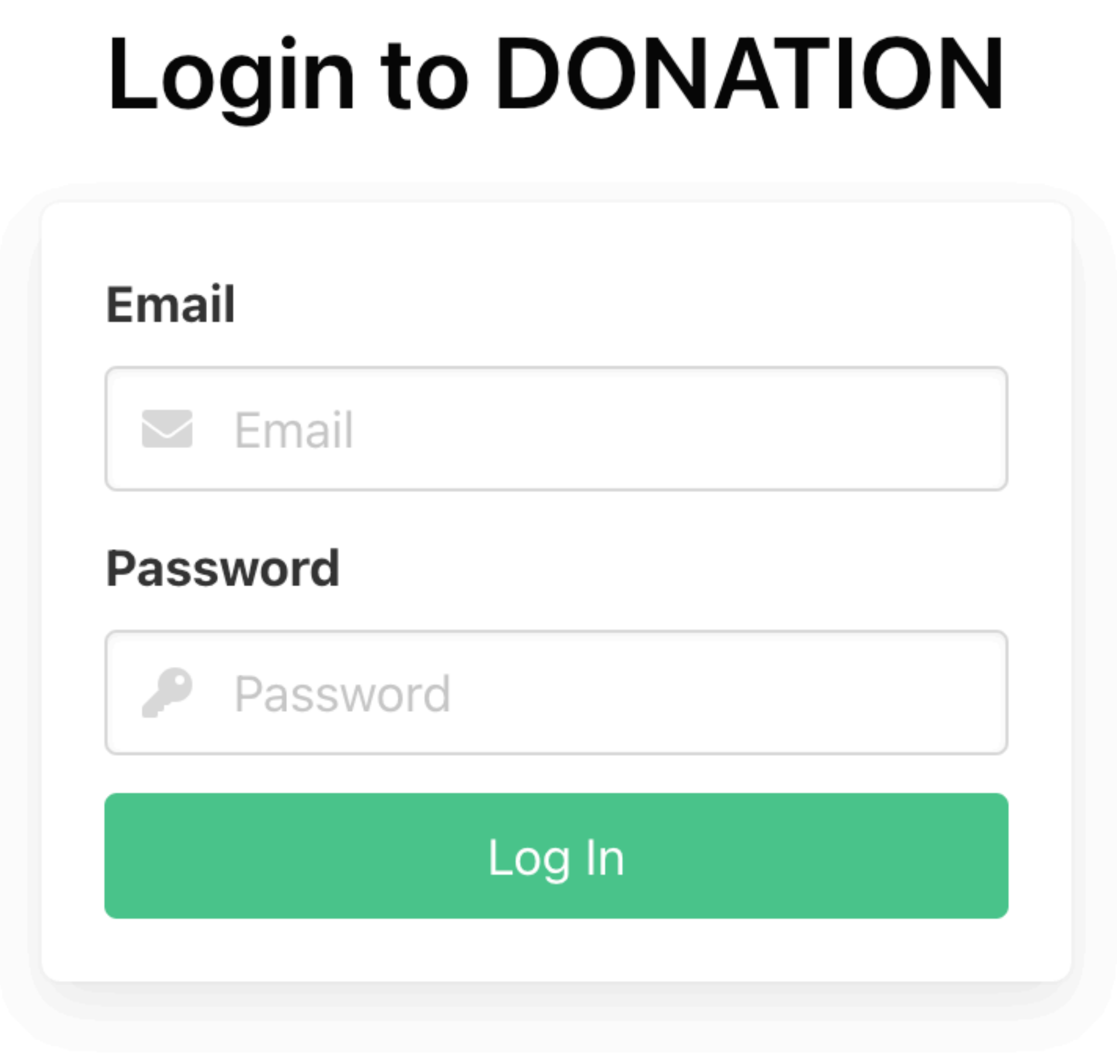
Help me run Springfield!



```
<h2 class="subtitle is-4 mt-2">Help me run Springfield!</h2>
<p class="has-text-centered">
  <a class="button is-medium is-success" href="/signup">
    <span class="icon"><i class="fas fa-plus"></i></span><span>Sign Up</span>
  </a>
  <a class="button is-medium is-link" href="/login">
    <span class="icon"><i class="fas fa-sign-in"></i></span><span>Login</span>
  </a>
</p>
```

- Routes specified in <a> tags

- Hapi Router located and executes the correct controller/ action on the server



Login to DONATION

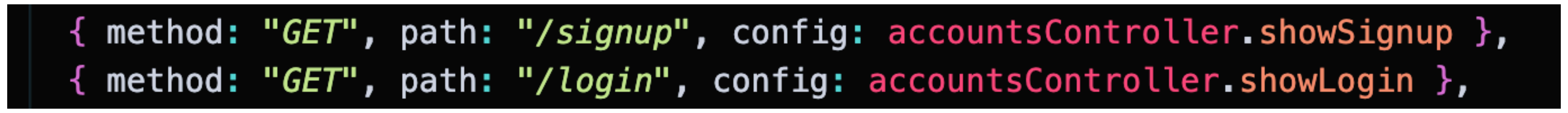
Email

Email

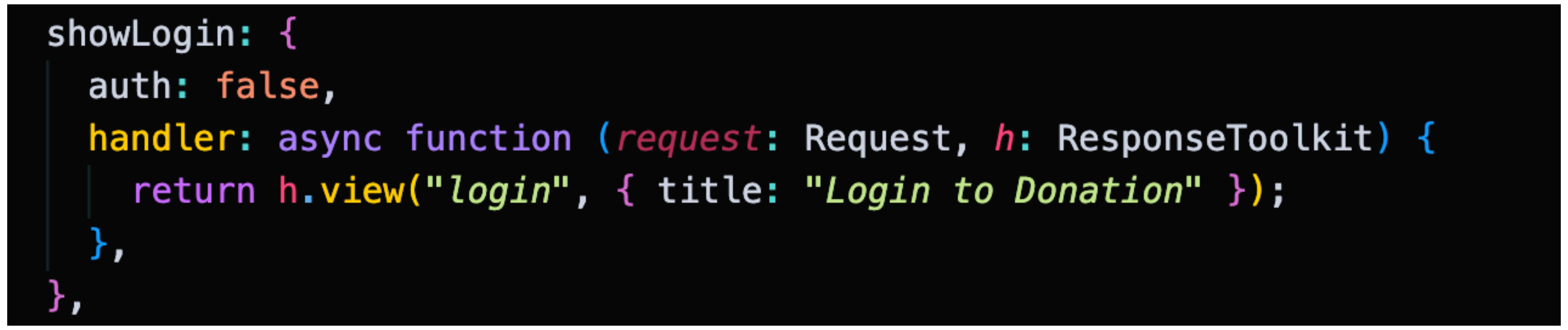
Password

Password

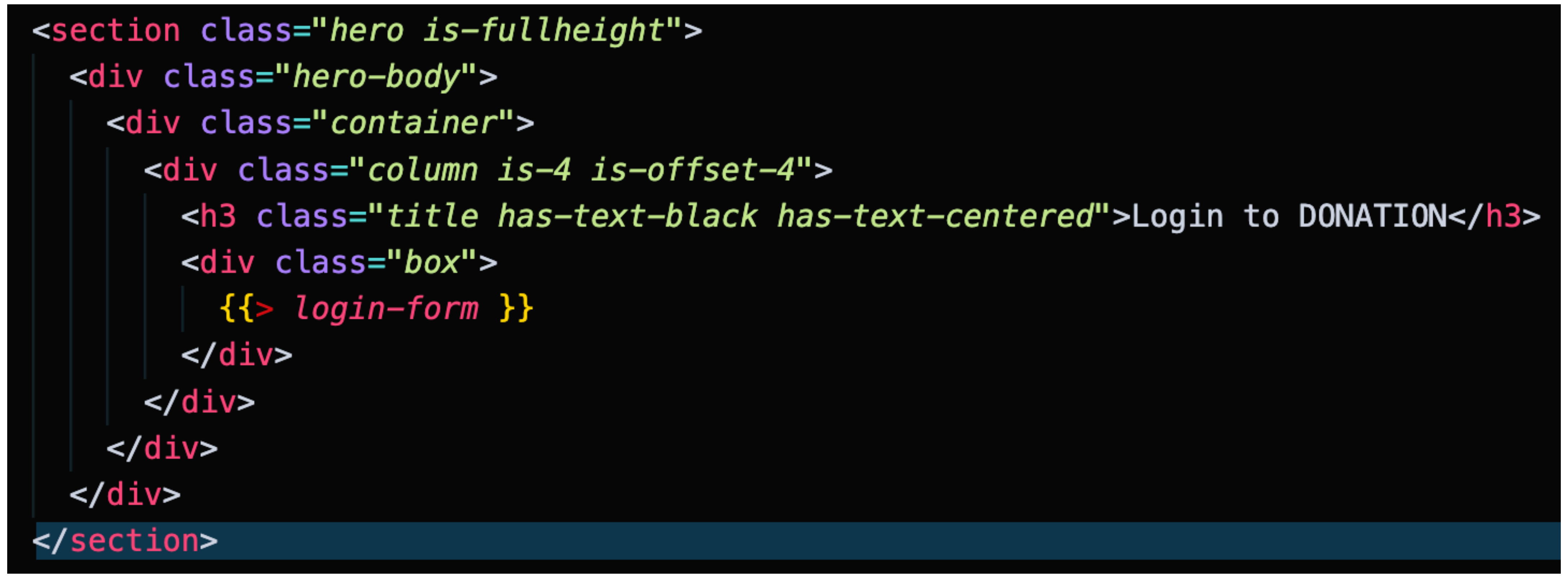
Log In



```
{ method: "GET", path: "/signup", config: accountsController.showSignup },
{ method: "GET", path: "/login", config: accountsController.showLogin },
```



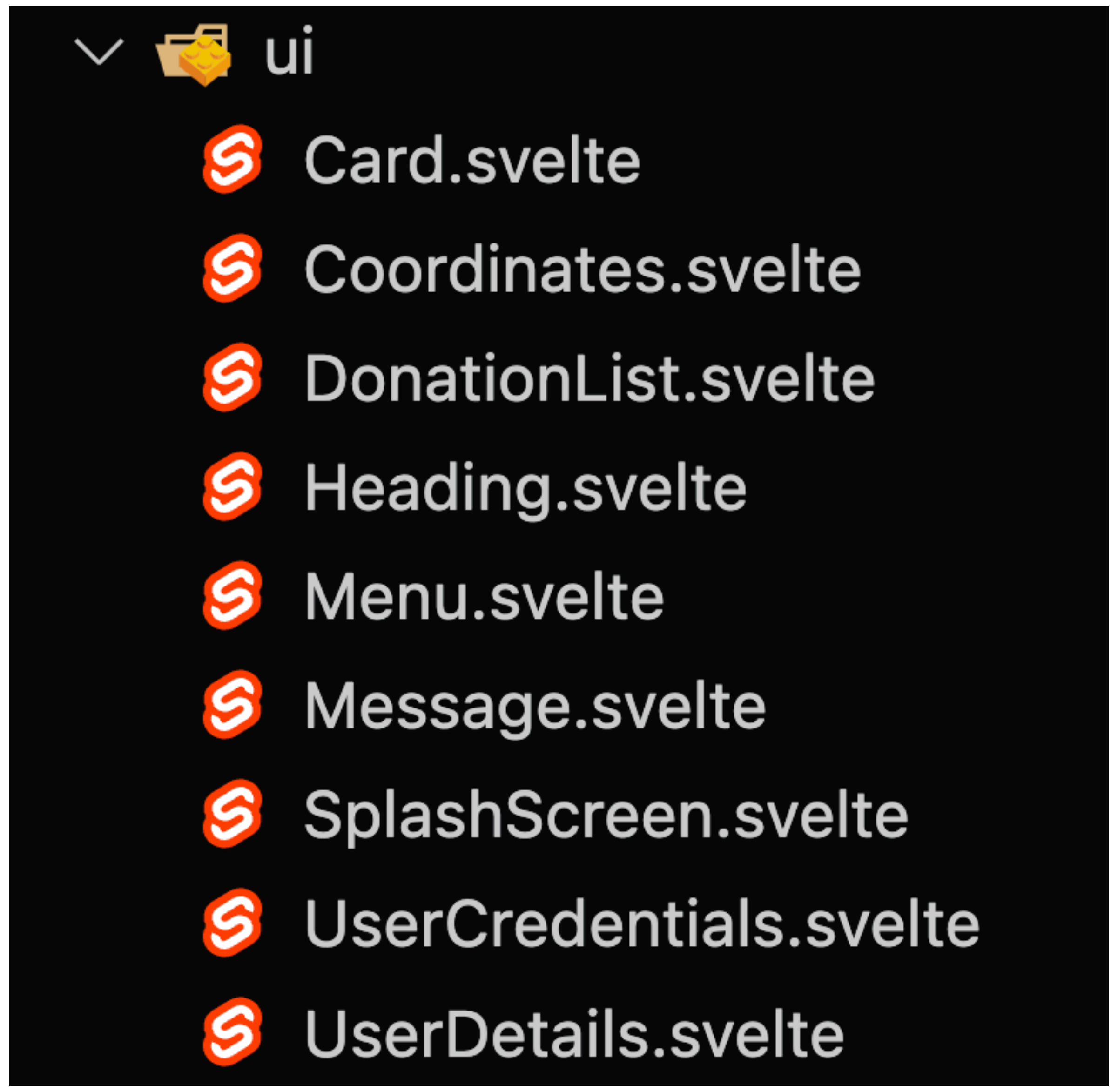
```
showLogin: {
  auth: false,
  handler: async function (request: Request, h: ResponseToolkit) {
    return h.view("login", { title: "Login to Donation" });
  },
},
```



```
<section class="hero is-fullheight">
  <div class="hero-body">
    <div class="container">
      <div class="column is-4 is-offset-4">
        <h3 class="title has-text-black has-text-centered">Login to DONATION</h3>
        <div class="box">
          {{> login-form }}
        </div>
      </div>
    </div>
  </div>
</section>
```

SvelteKit Web Application

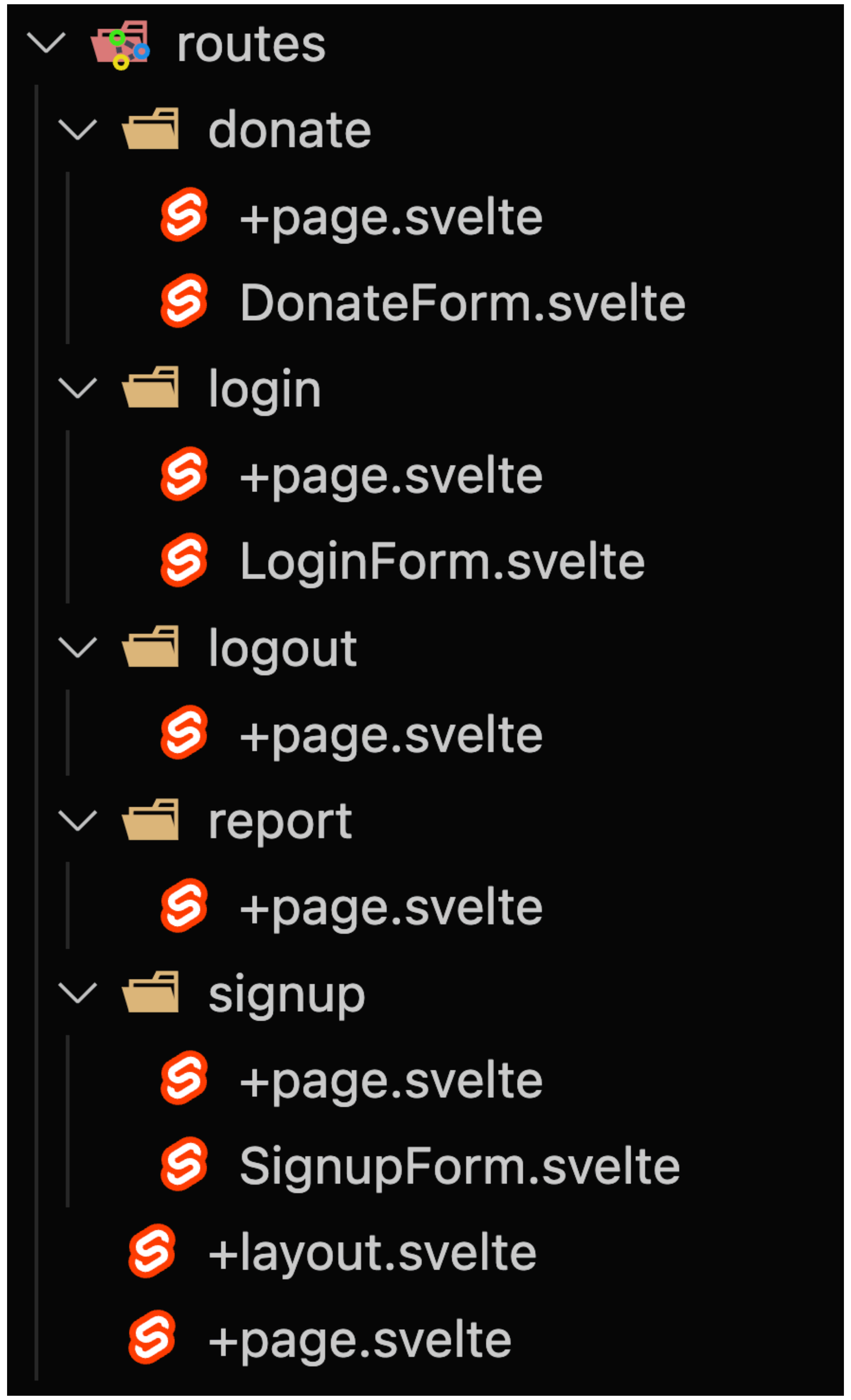
Application Structure



```
ui
    Card.svelte
    Coordinates.svelte
    DonationList.svelte
    Heading.svelte
    Menu.svelte
    Message.svelte
    SplashScreen.svelte
    UserCredentials.svelte
    UserDetails.svelte
```

Visual Components

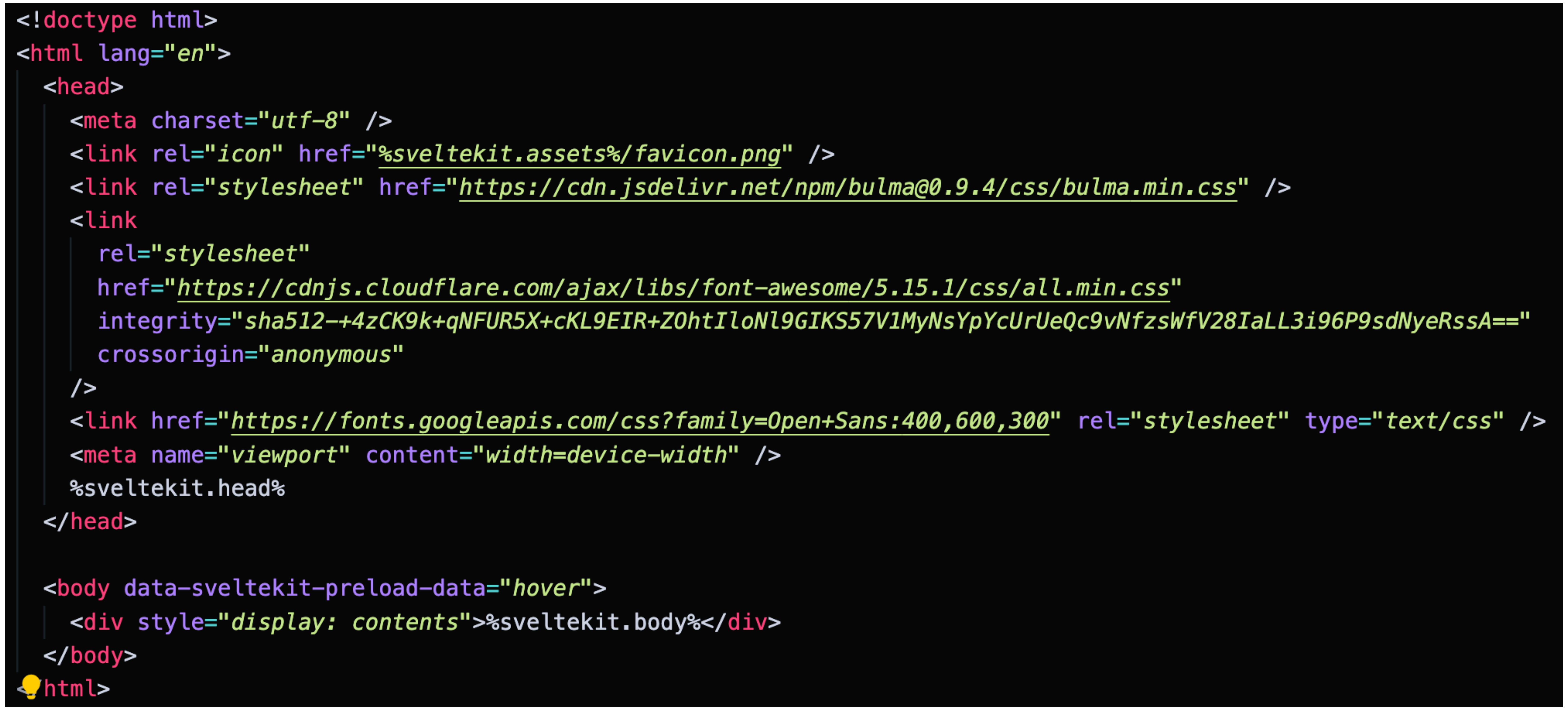
Web Pages/Views



```
routes
  donate
    +page.svelte
    DonateForm.svelte
  login
    +page.svelte
    LoginForm.svelte
  logout
    +page.svelte
  report
    +page.svelte
  signup
    +page.svelte
    SignupForm.svelte
  +layout.svelte
  +page.svelte
```

app.html

- All pages in our app are based on this template

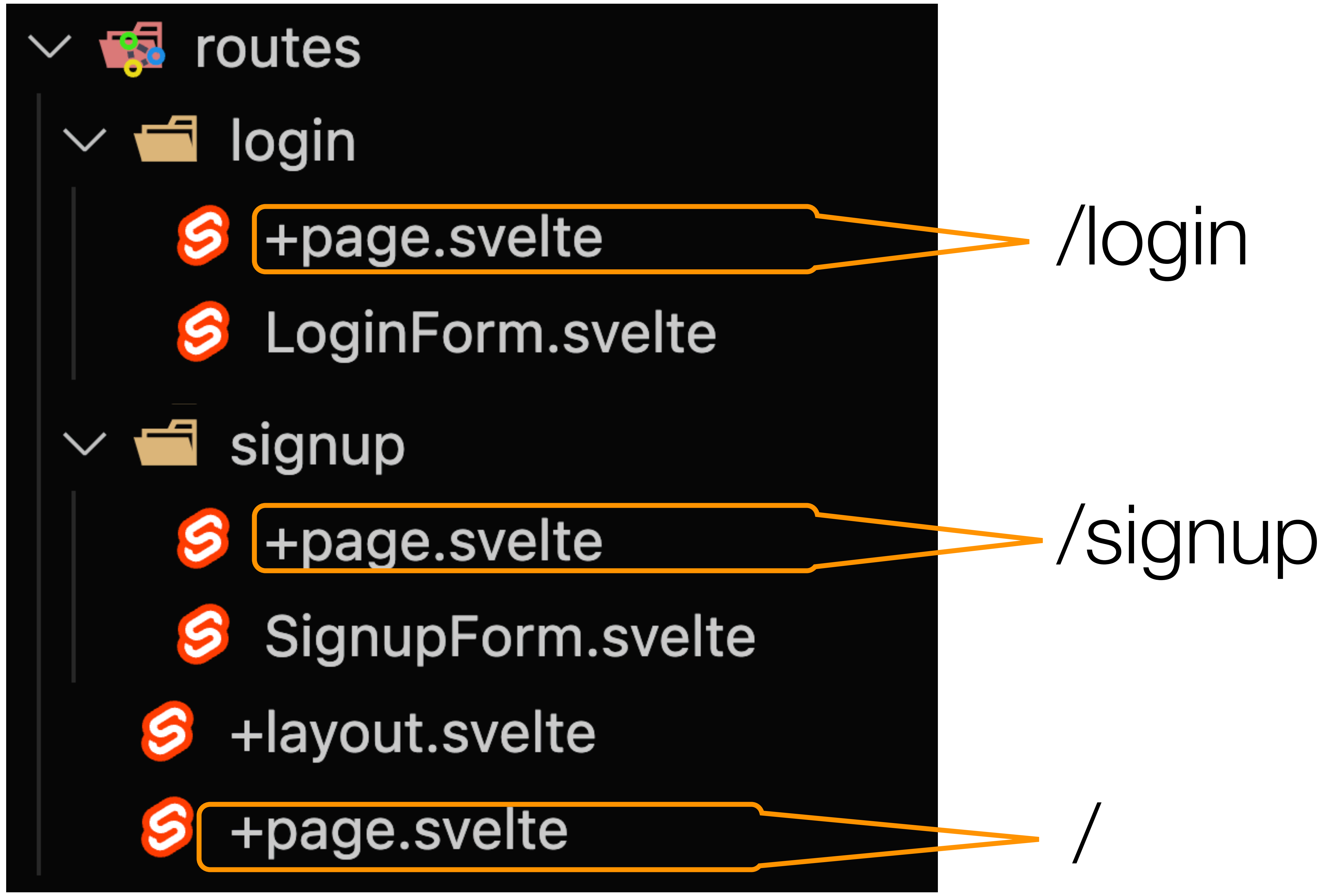


```html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%sveltekit.assets%/favicon.png" />
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.4/css/bulma.min.css" />
    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/css/all.min.css"
      integrity="sha512-+4zCK9k+qNFUR5X+cKL9EIR+ZOhtIloNl9GIKS57V1MyNsYpYcUrUeQc9vNfzsWfV28IaLL3i96P9sdNyeRssA=="
      crossorigin="anonymous"
    />
    <link href="https://fonts.googleapis.com/css?family=Open+Sans:400,600,300" rel="stylesheet" type="text/css" />
    <meta name="viewport" content="width=device-width" />
    %sveltekit.head%
  </head>

  <body data-sveltekit-preload-data="hover">
    <div style="display: contents">%sveltekit.body%</div>
  </body>
</html>
```

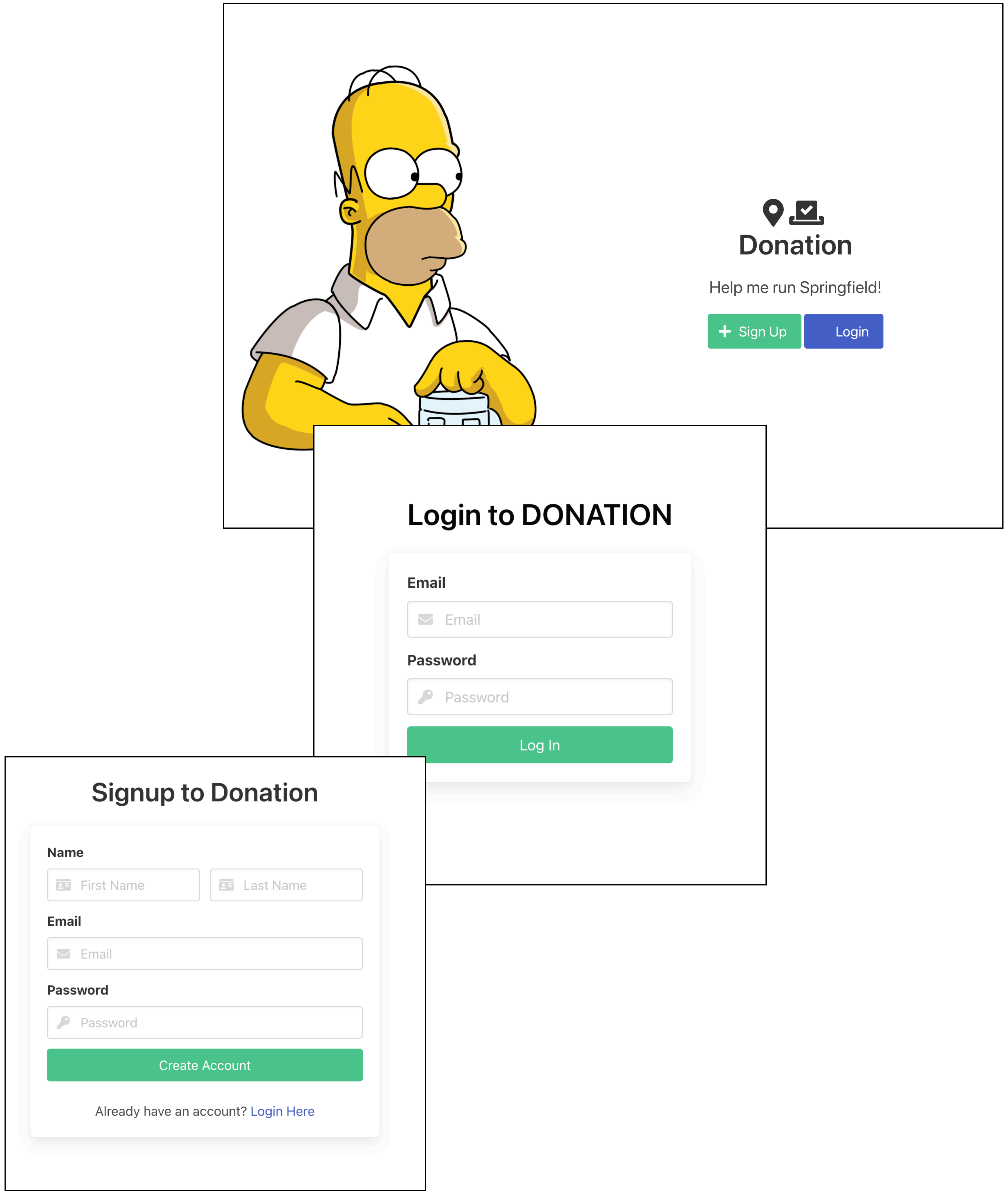
- This file is rarely updated

SvelteKit Router



```
routes
  login
    +page.svelte
    LoginForm.svelte
  signup
    +page.svelte
    SignupForm.svelte
  +layout.svelte
  +page.svelte
```

- The route are defined by the subfolder names in “routes” folder



Donation

Help me run Springfield!

Login to DONATION

Email: Email

Password: Password

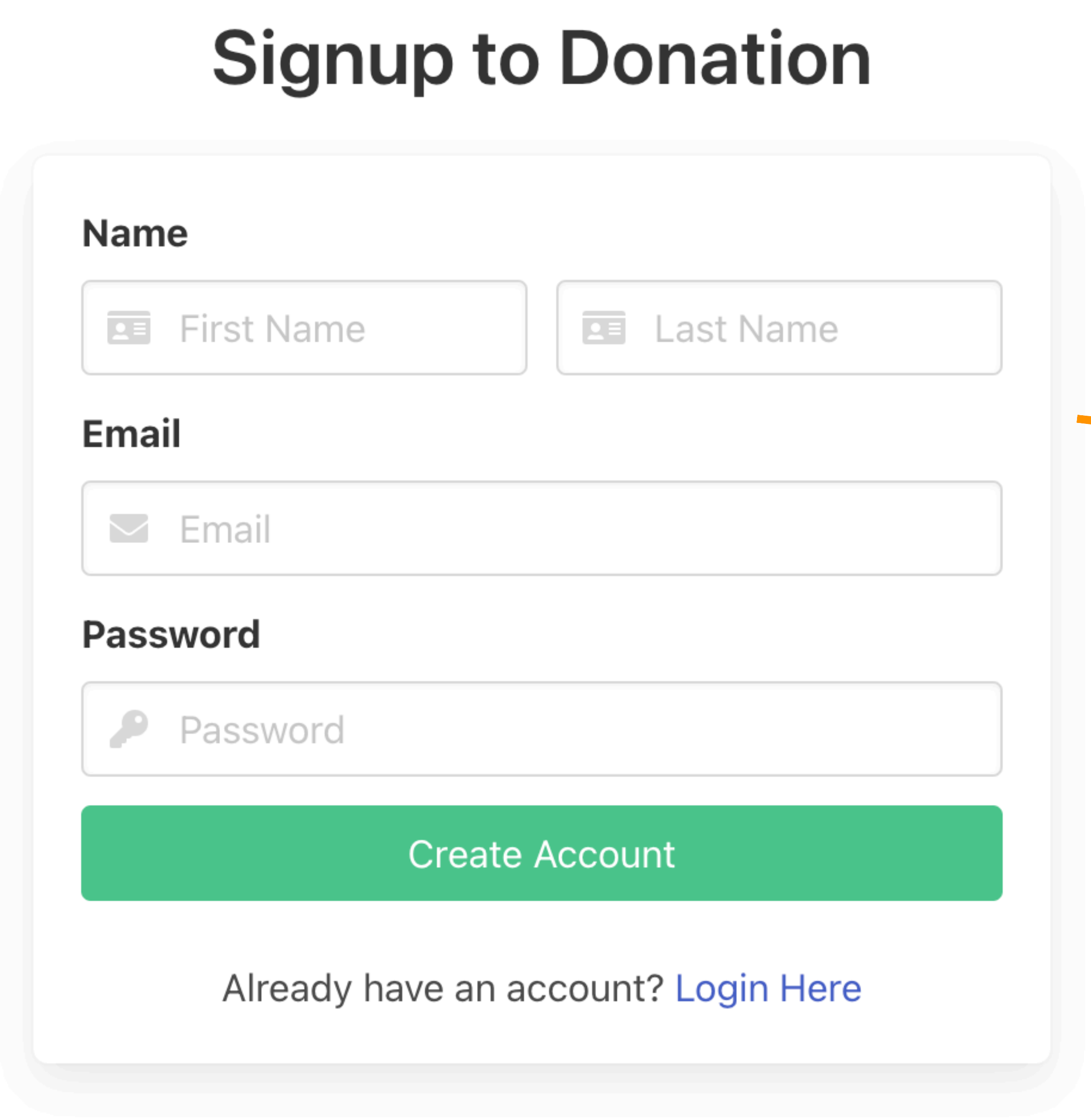
Signup to Donation

Name: First Name | Last Name

Email: Email

Password: Password

Already have an account? Login Here



Signup to Donation

Name

First Name | Last Name

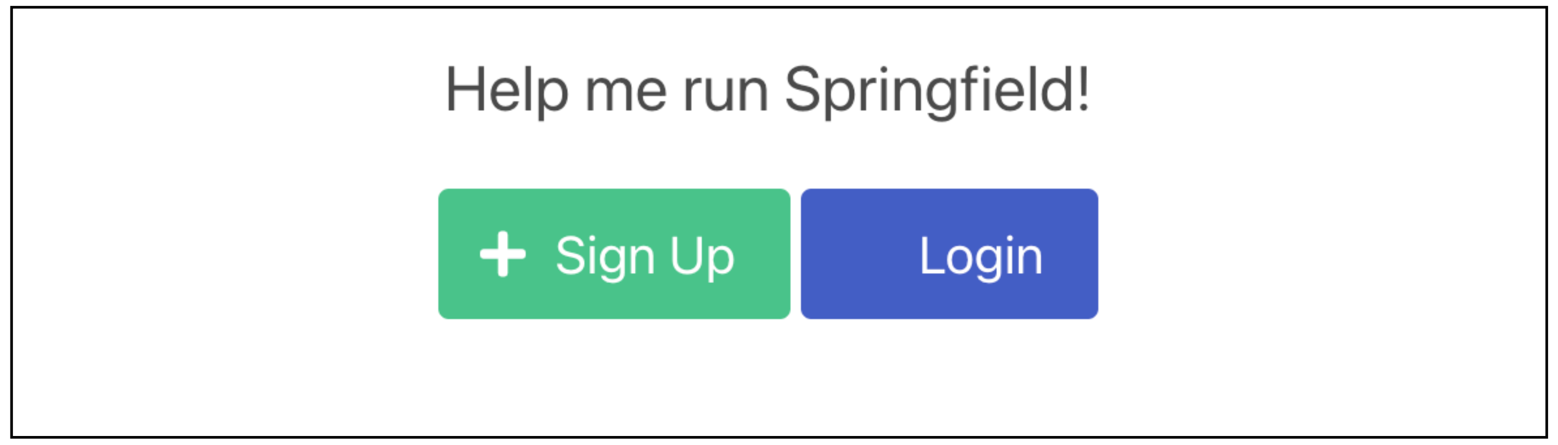
Email

Email

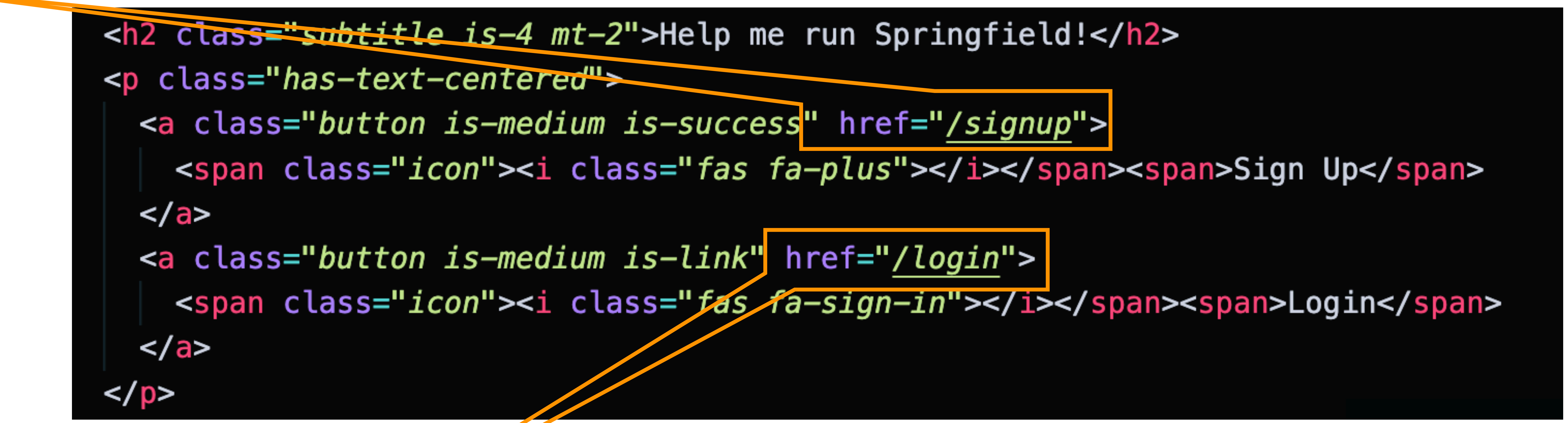
Password

Password

Already have an account? Login Here



Help me run Springfield!



```
<h2 class="subtitle is-4 mt-2">Help me run Springfield!</h2>
<p class="has-text-centered">
  <a class="button is-medium is-success" href="/signup">
    <span class="icon"><i class="fas fa-plus"></i></span><span>Sign Up</span>
  </a>
  <a class="button is-medium is-link" href="/login">
    <span class="icon"><i class="fas fa-sign-in"></i></span><span>Login</span>
  </a>
</p>
```

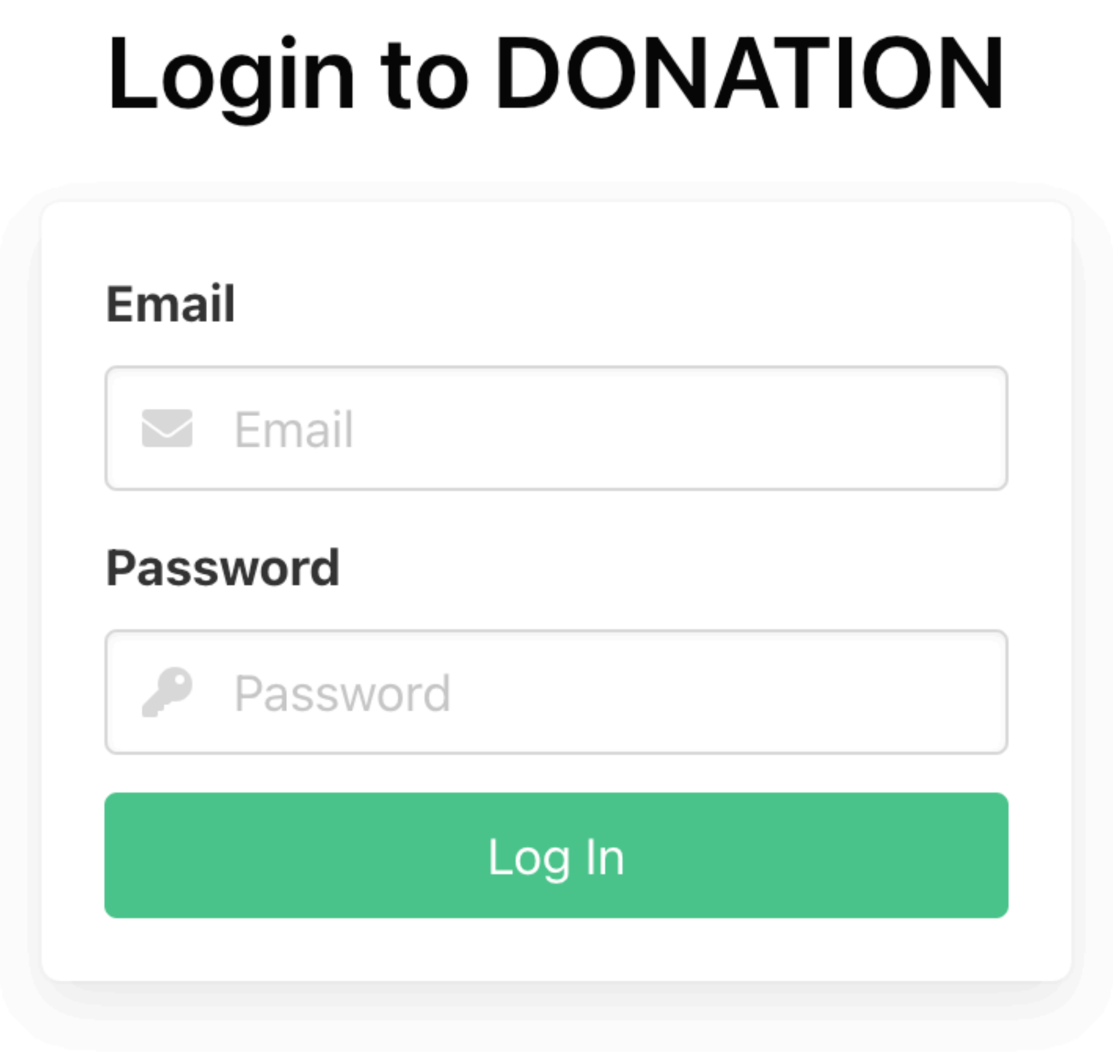
- Routes specified in <a> tags

- Sveltkit router locates correct page to load

- This page will be loaded from the server

OR

- It may already be in the client, and rendered from browser cache



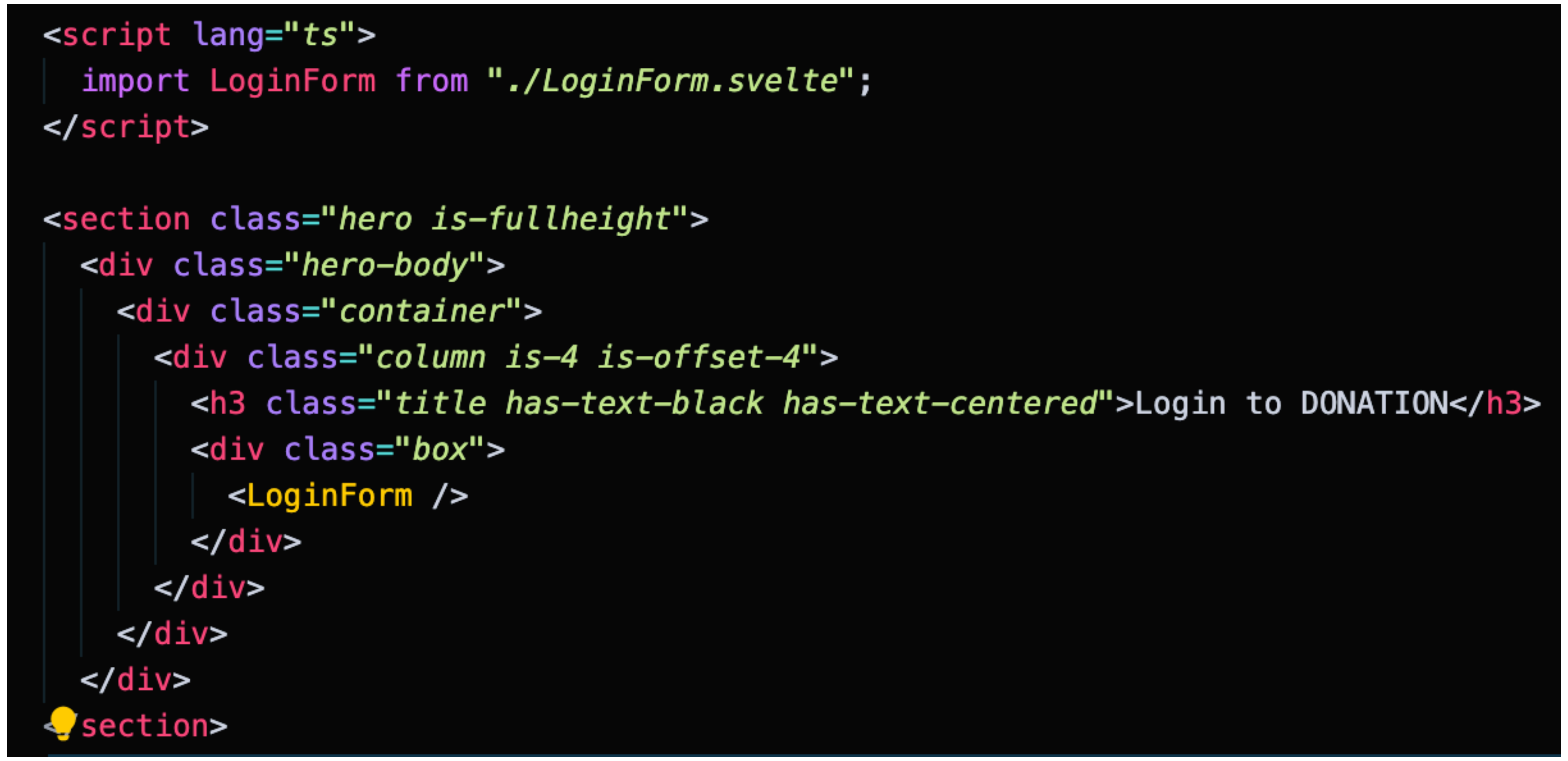
Login to DONATION

Email

Email

Password

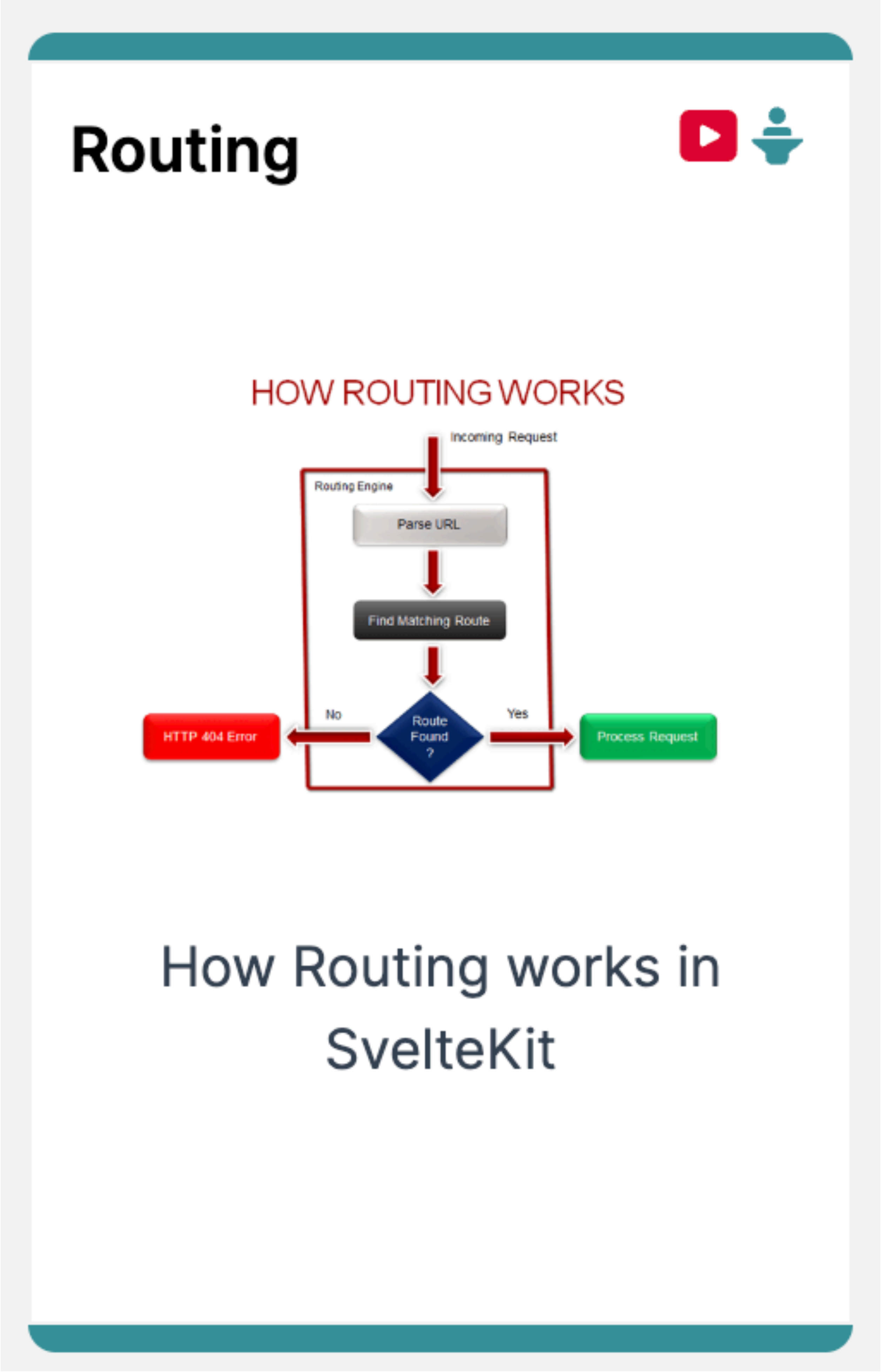
Password



```
<script lang="ts">
  import LoginForm from "./LoginForm.svelte";
</script>

<section class="hero is-fullheight">
  <div class="hero-body">
    <div class="container">
      <div class="column is-4 is-offset-4">
        <h3 class="title has-text-black has-text-centered">Login to DONATION</h3>
        <div class="box">
          <LoginForm />
        </div>
      </div>
    </div>
  </div>
</section>
```

routes/login/+page.svelte



Routing

HOW ROUTING WORKS

How Routing works in SvelteKit