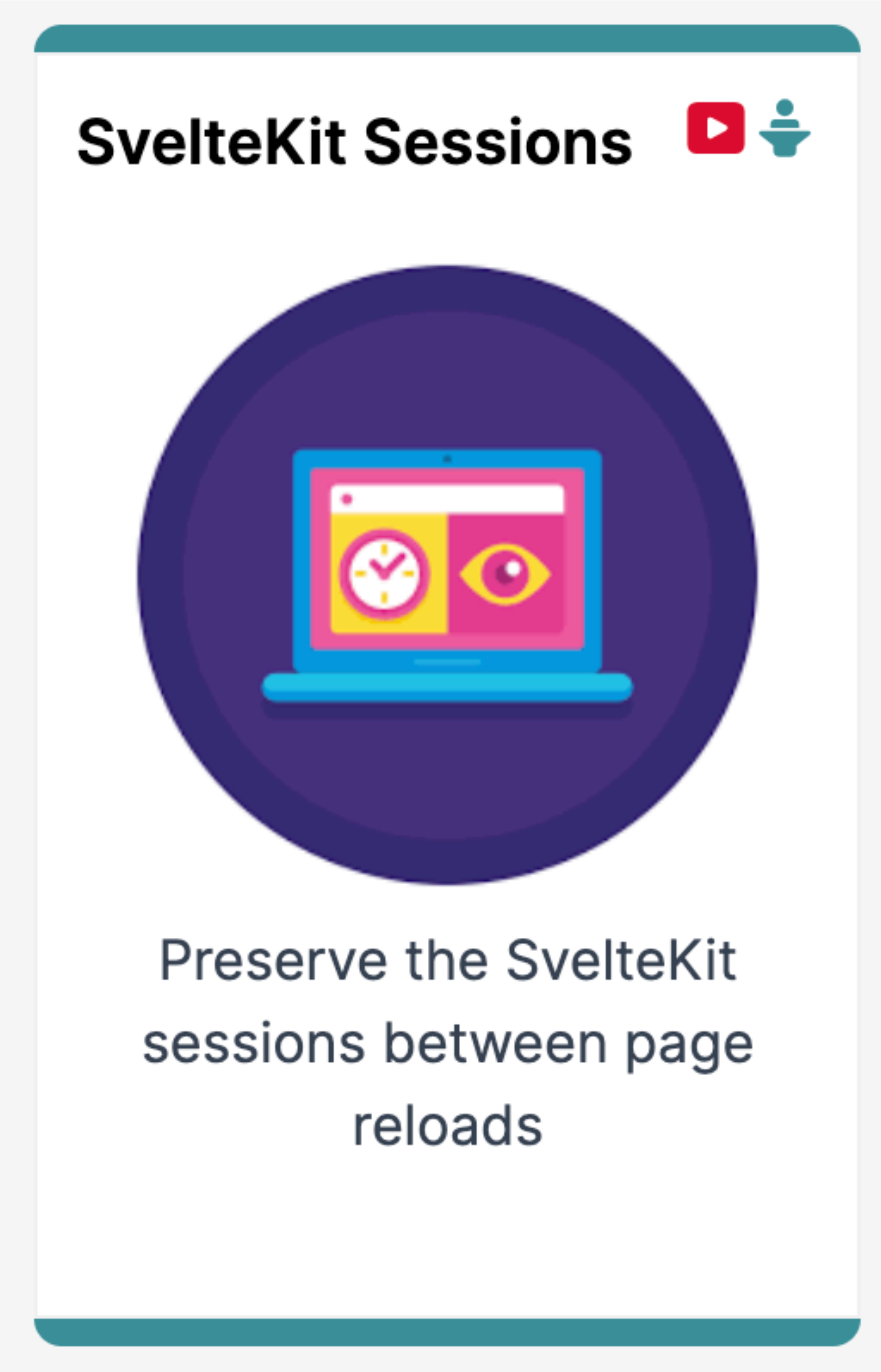
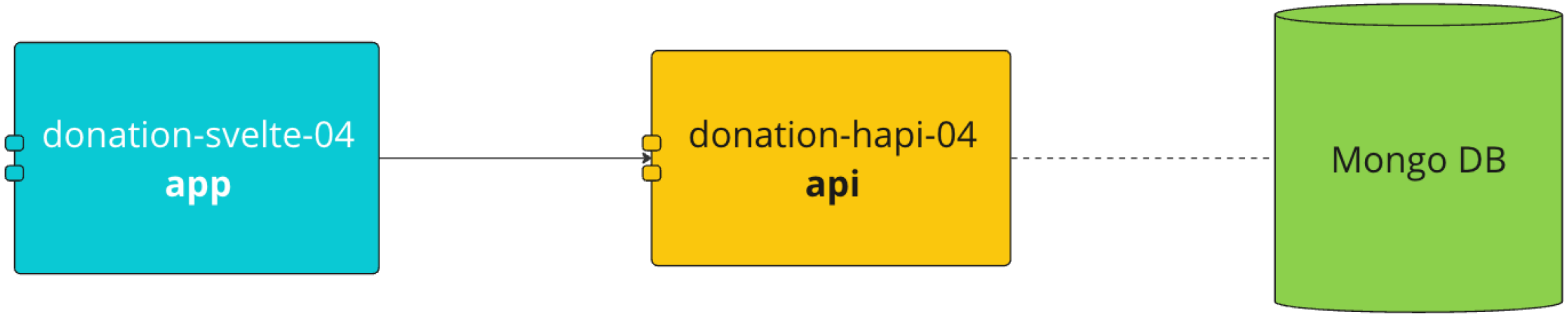


SvelteKit Sessions

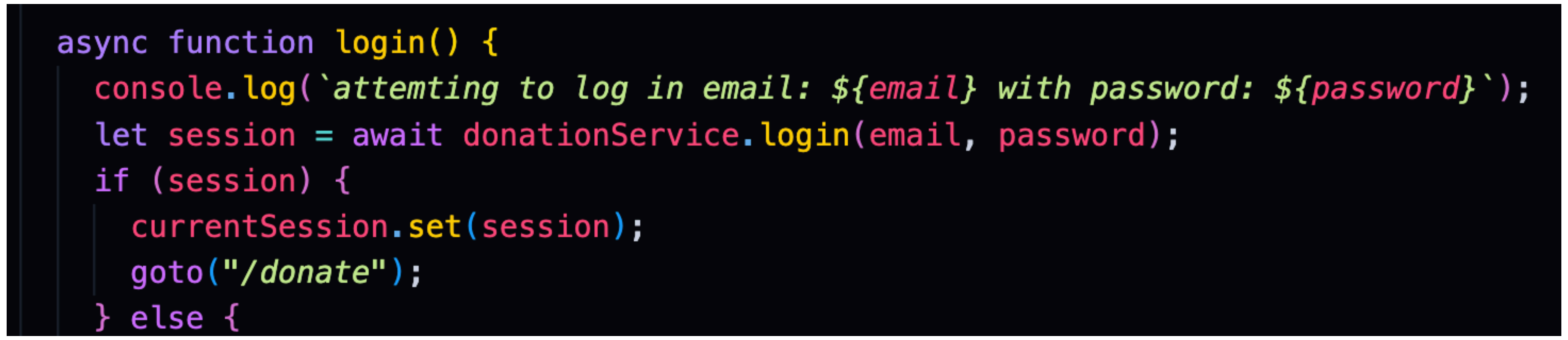
Preserve the SvelteKit sessions between page reloads

Svelte Authentication



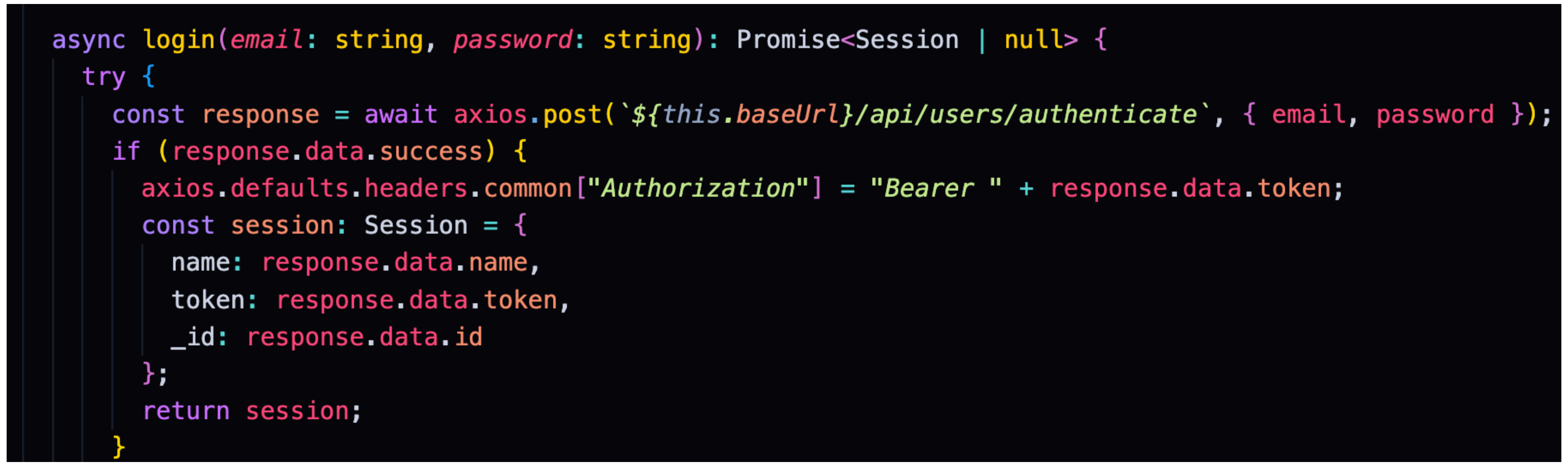
- Attach to the Donation-svelte API
- Signup / Login using Users API

LoginForm.svelte



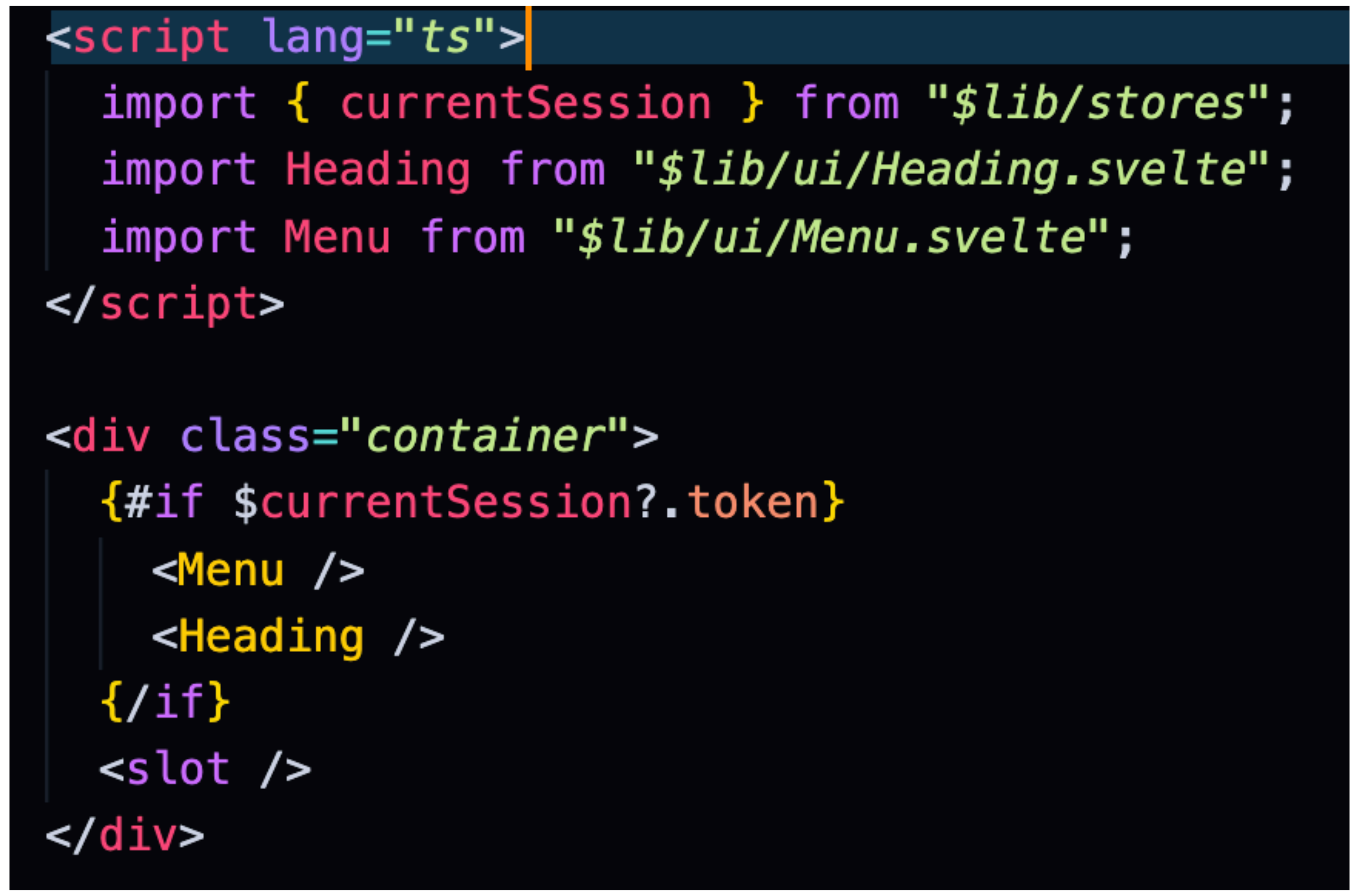
```
async function login() {
  console.log(`attemting to log in email: ${email} with password: ${password}`);
  let session = await donationService.login(email, password);
  if (session) {
    currentSession.set(session);
    goto("/donate");
  } else {
```

donation-service.ts



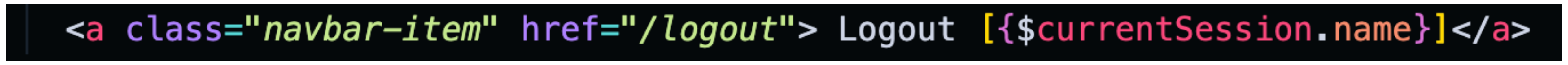
```
async login(email: string, password: string): Promise<Session | null> {
  try {
    const response = await axios.post(`${this.baseUrl}/api/users/authenticate`, { email, password });
    if (response.data.success) {
      axios.defaults.headers.common["Authorization"] = "Bearer " + response.data.token;
      const session: Session = {
        name: response.data.name,
        token: response.data.token,
        _id: response.data.id
      };
      return session;
    }
```

+layout.svelte

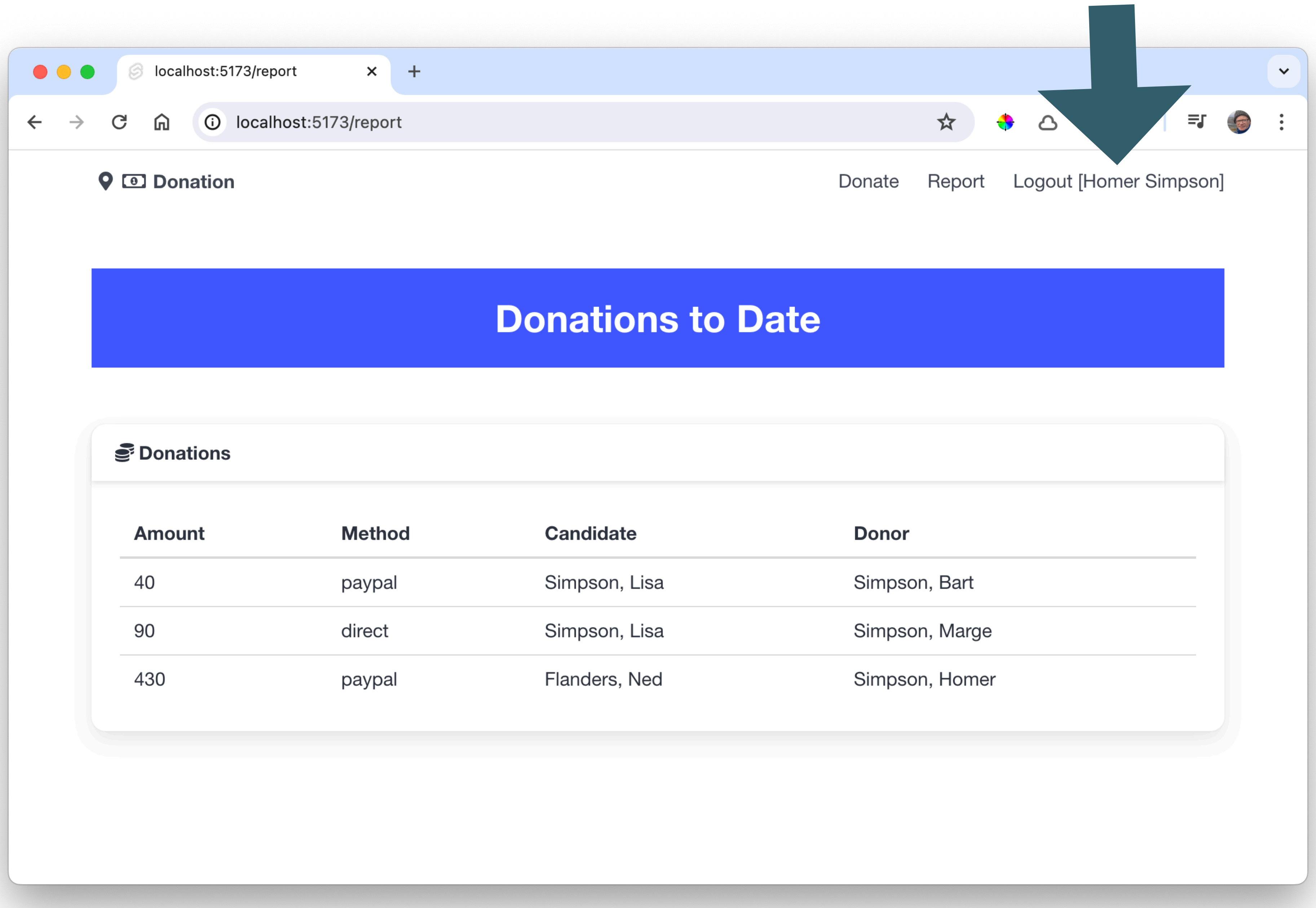


```
<script lang="ts">
  import { currentSession } from "$lib/stores";
  import Heading from "$lib/ui/Heading.svelte";
  import Menu from "$lib/ui/Menu.svelte";
</script>

<div class="container">
  {#if $currentSession?.token}
    <Menu />
    <Heading />
  {/if}
  <slot />
</div>
```

- Logged in user name displayed in menu



```
<a class="navbar-item" href="/logout"> Logout [{$currentSession.name}]</a>
```

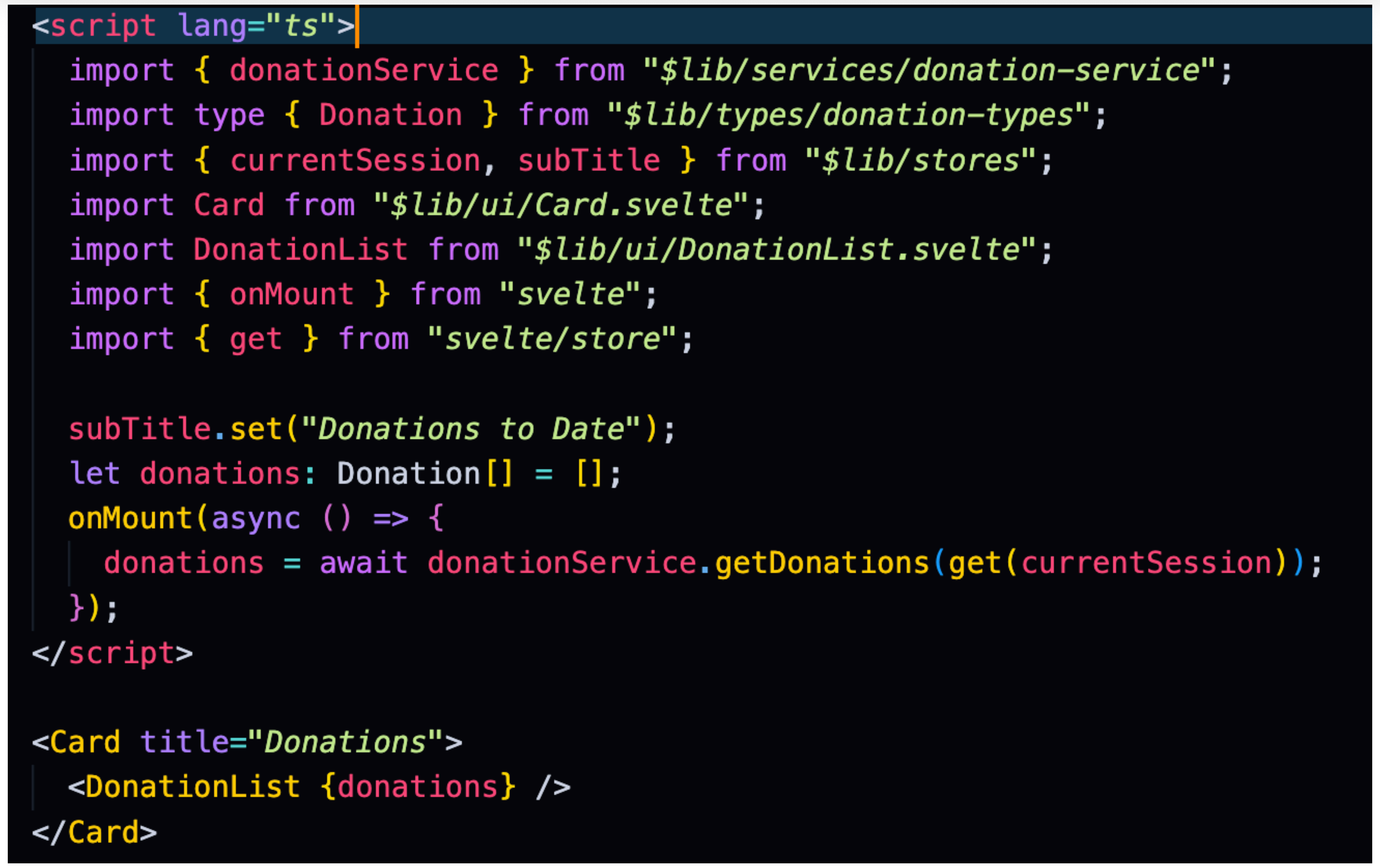


Donations to Date

Donations

Amount	Method	Candidate	Donor
40	paypal	Simpson, Lisa	Simpson, Bart
90	direct	Simpson, Lisa	Simpson, Marge
430	paypal	Flanders, Ned	Simpson, Homer

+report.svelte

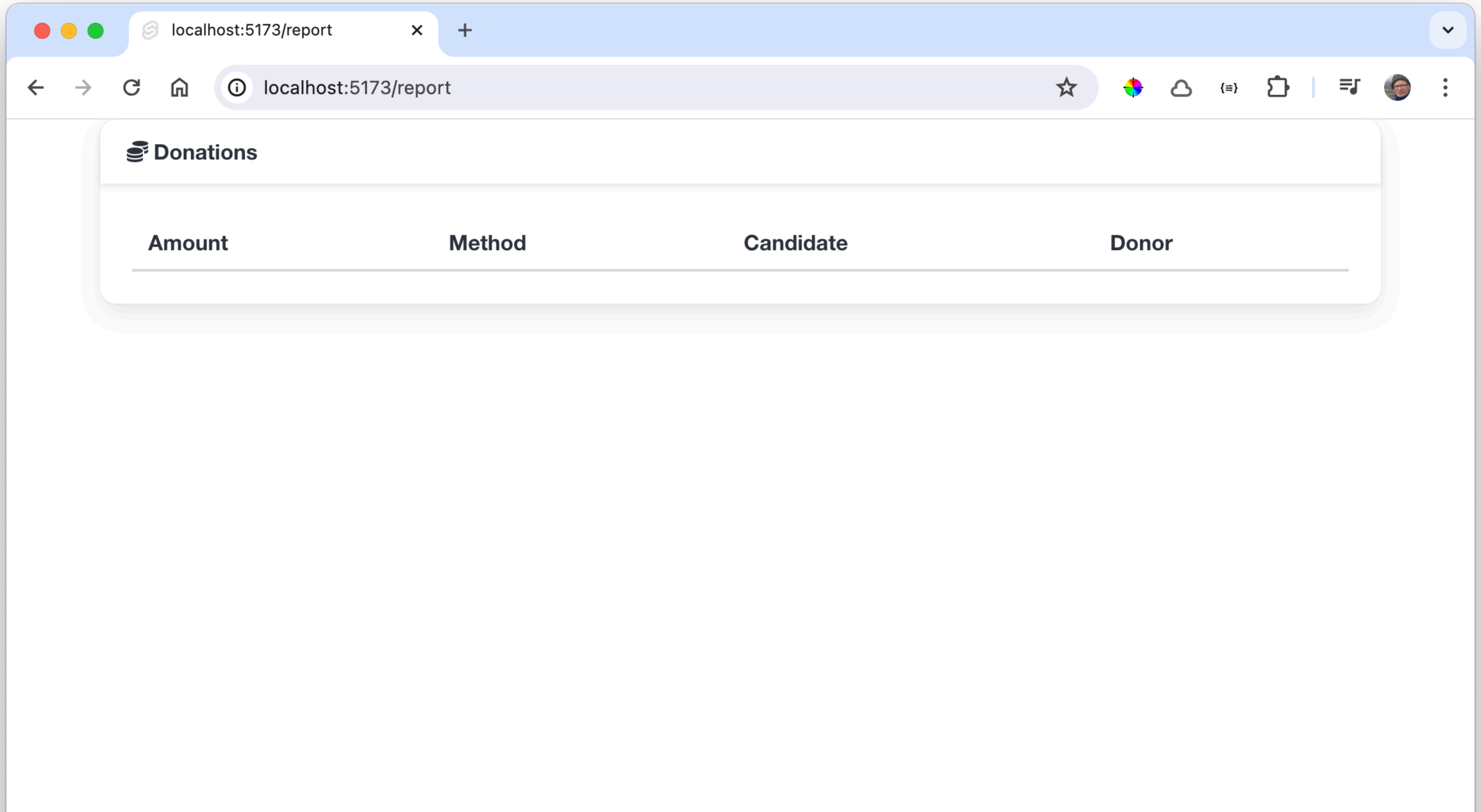


```
<script lang="ts">
  import { donationService } from "$lib/services/donation-service";
  import type { Donation } from "$lib/types/donation-types";
  import { currentSession, subTitle } from "$lib/stores";
  import Card from "$lib/ui/Card.svelte";
  import DonationList from "$lib/ui/DonationList.svelte";
  import { onMount } from "svelte";
  import { get } from "svelte/store";

  subTitle.set("Donations to Date");
  let donations: Donation[] = [];
  onMount(async () => {
    donations = await donationService.getDonations(get(currentSession));
  });
</script>

<Card title="Donations">
  <DonationList {donations} />
</Card>
```

Browser Reload Button

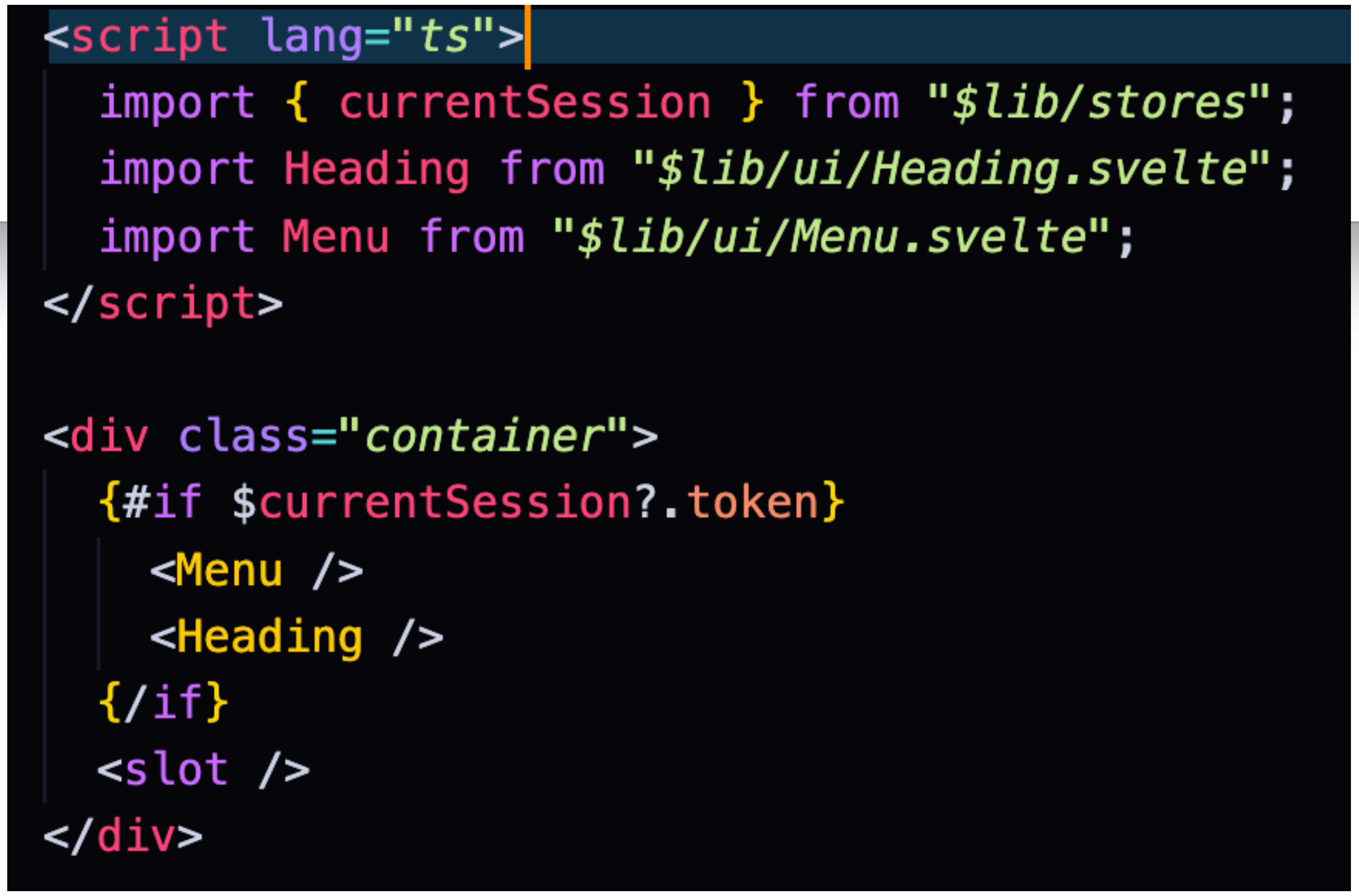
- If browser reload button pressed:
  - Menu + Header disappears
  - List of donations no longer displayed

- When browser reloads:
  - All stores reset
  - Session cleared - so layout will remove menu + Header

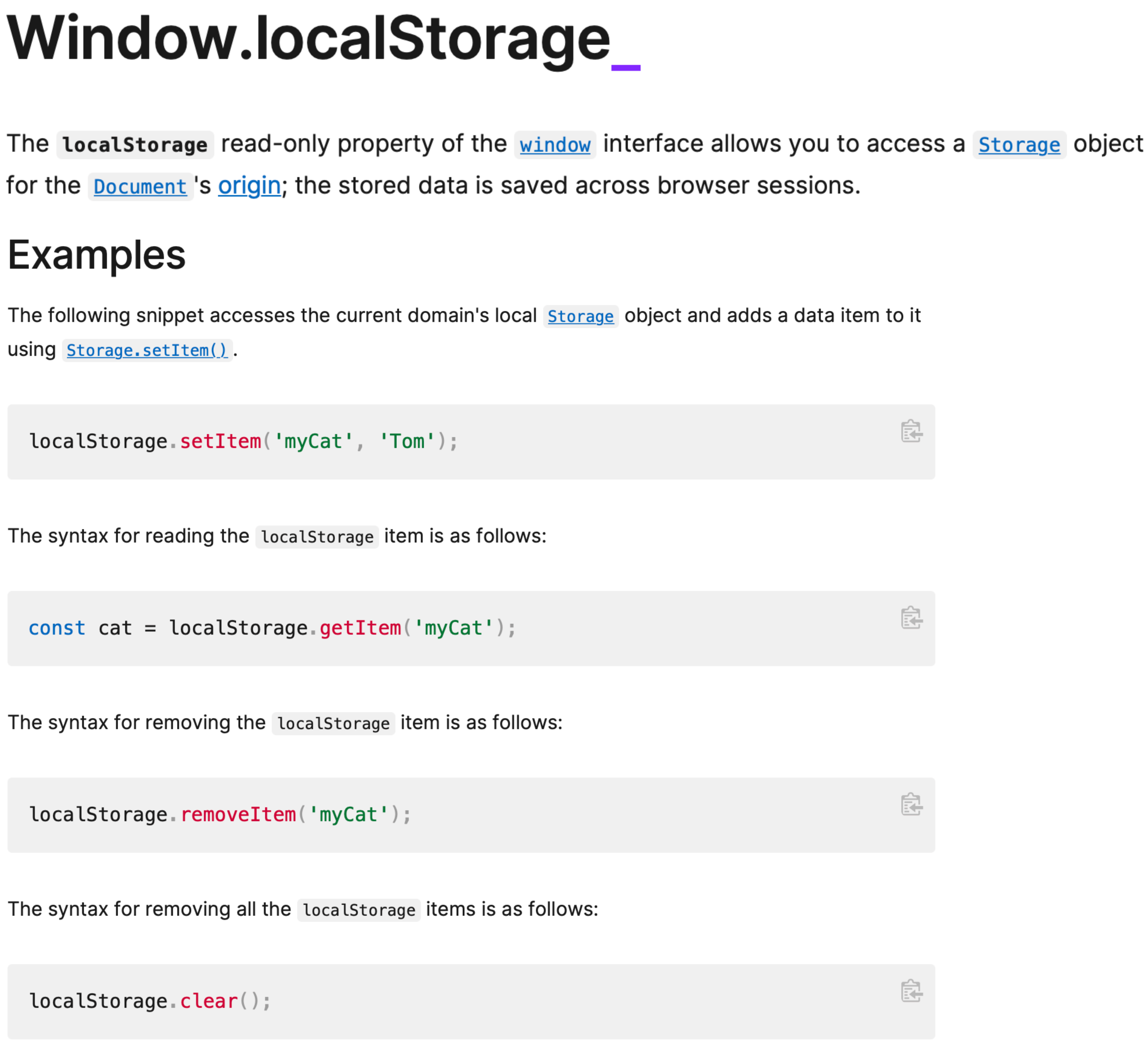


Donations

Amount	Method	Candidate	Donor



```
<script lang="ts">
  import { currentSession } from "$lib/stores";
  import Heading from "$lib/ui/Heading.svelte";
  import Menu from "$lib/ui/Menu.svelte";
</script>

<div class="container">
  {#if $currentSession?.token}
    <Menu />
    <Heading />
  {/if}
  <slot />
</div>
```



Window.localStorage

The `localStorage` read-only property of the `window` interface allows you to access a `Storage` object for the `Document`'s origin; the stored data is saved across browser sessions.

Examples

The following snippet accesses the current domain's local `Storage` object and adds a data item to it using `Storage.setItem()`.

```
localStorage.setItem('myCat', 'Tom');
```

The syntax for reading the `localStorage` item is as follows:

```
const cat = localStorage.getItem('myCat');
```

The syntax for removing the `localStorage` item is as follows:

```
localStorage.removeItem('myCat');
```

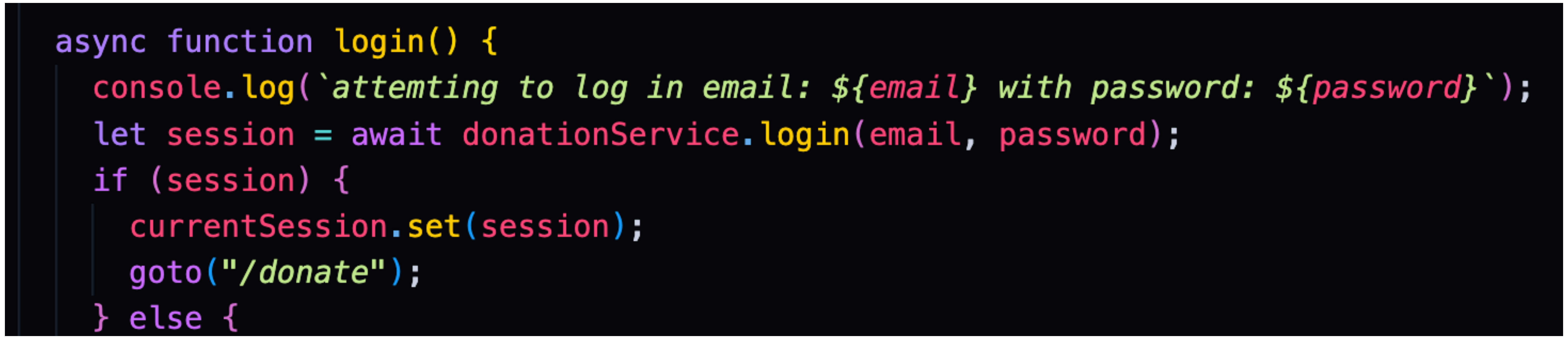
The syntax for removing all the `localStorage` items is as follows:

```
localStorage.clear();
```

https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage

Login

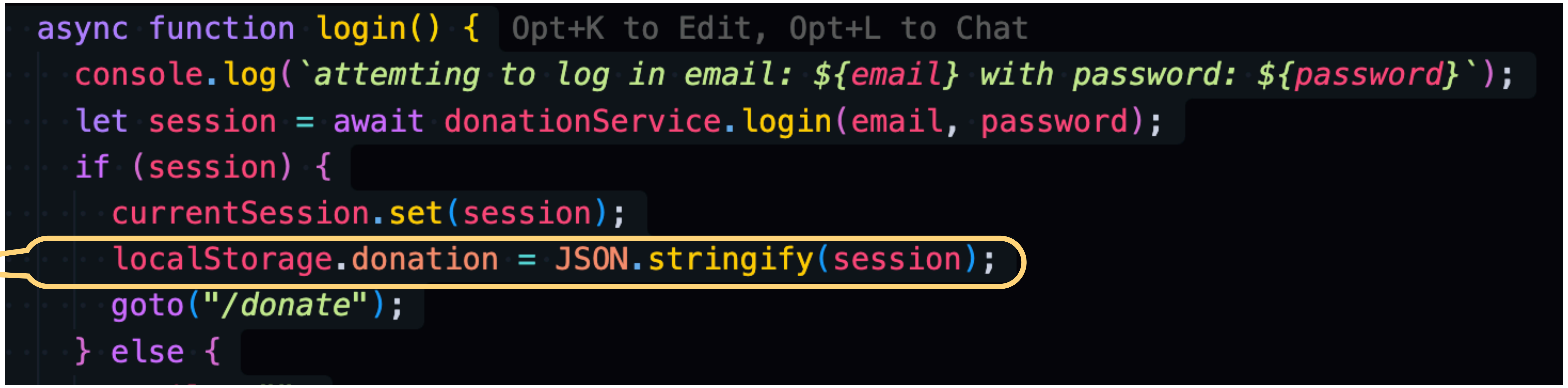
- On login, write user details to local storage



```
async function login() {
  console.log(`attemting to log in email: ${email} with password: ${password}`);
  let session = await donationService.login(email, password);
  if (session) {
    currentSession.set(session);
    goto("/donate");
  } else {
```

LoginForm.svelte

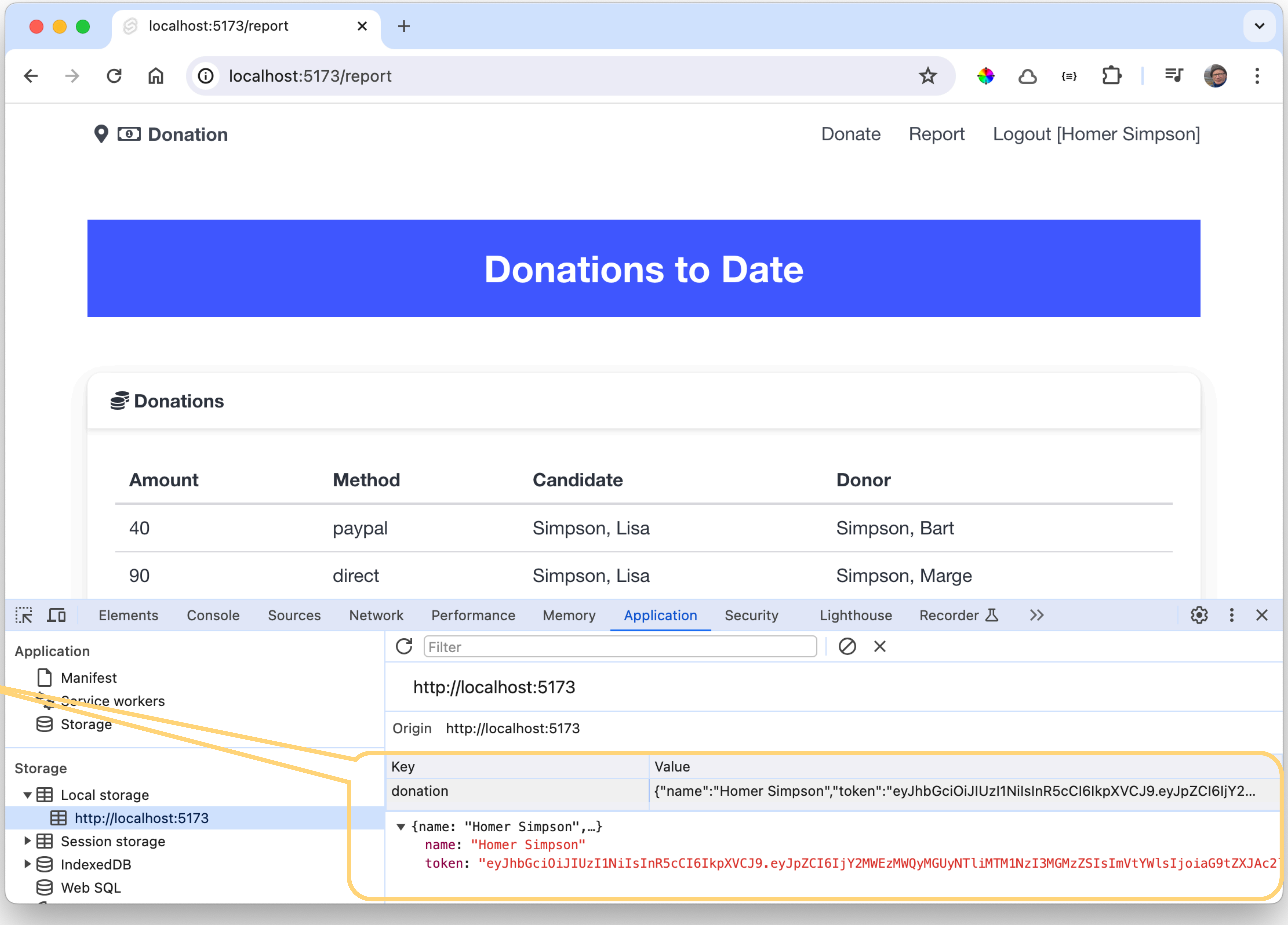
- This will persist in the browser until explicitly cleared



```
async function login() {  Opt+K to Edit, Opt+L to Chat
  console.log(`attemting to log in email: ${email} with password: ${password}`);
  let session = await donationService.login(email, password);
  if (session) {
    currentSession.set(session);
    localStorage.donation = JSON.stringify(session);
    goto("/donate");
  } else {
```

Login

- This will persist in the browser until explicitly cleared



Donations to Date

Donations

Amount	Method	Candidate	Donor
40	paypal	Simpson, Lisa	Simpson, Bart
90	direct	Simpson, Lisa	Simpson, Marge

http://localhost:5173

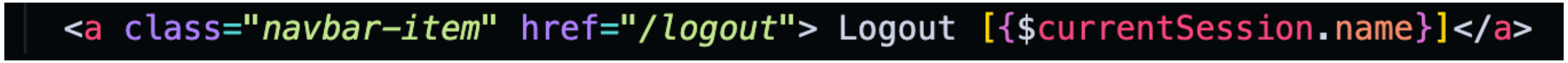
Origin http://localhost:5173

Key	Value
donation	{"name":"Homer Simpson","token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY2...

```
▼ {name: "Homer Simpson",…}
    name: "Homer Simpson"
    token: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY2MWEzMWQyMGUyNTliMTM1NzI3MGMzZSIsImVtYWlsIjoiaG9tZXJAc2
```

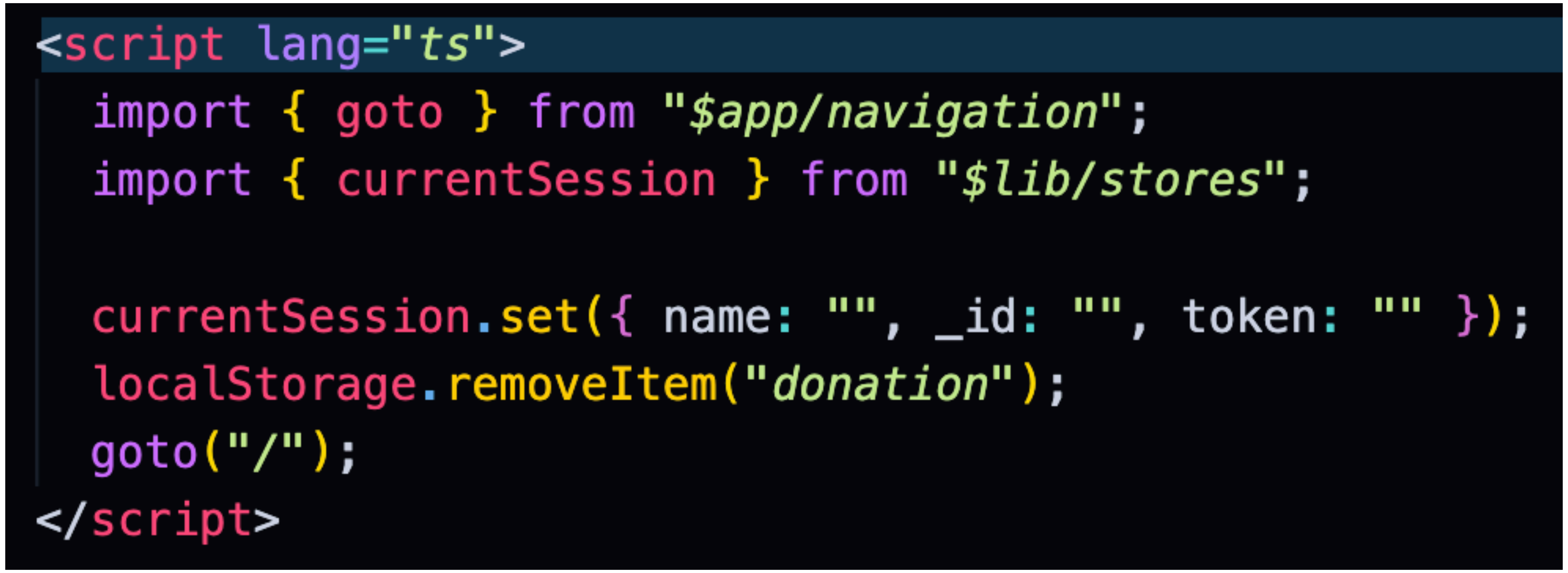
Logout

Donation
Donate
Report
Logout [Homer Simpson]



```
<a class="navbar-item" href="/logout"> Logout [{$currentSession.name}]</a>
```

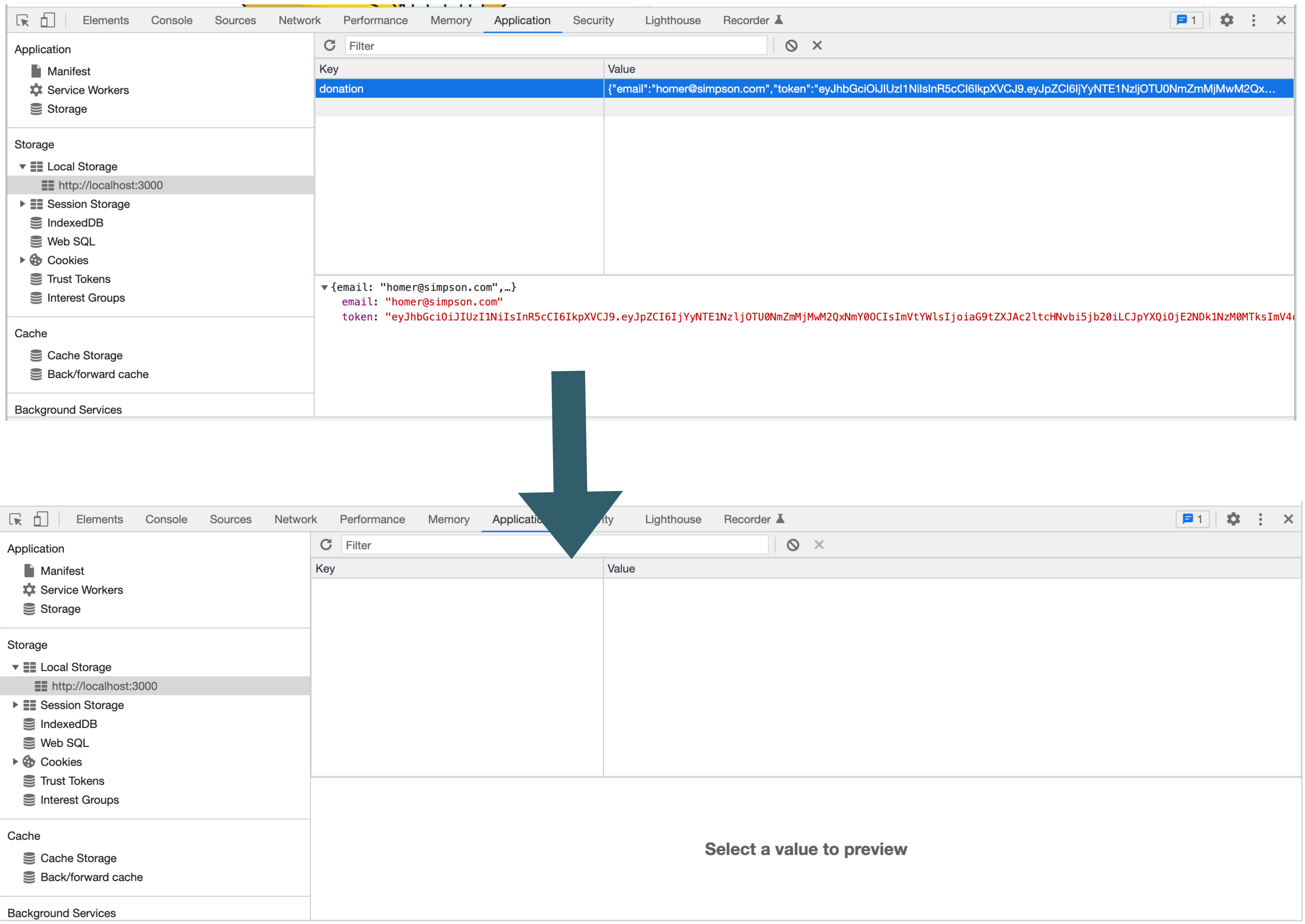
- Logout route:
  - Clears the curentSession store
  - removes the store contents from local storage
  - Navigate to the start page



```
<script lang="ts">
  import { goto } from "$app/navigation";
  import { currentSession } from "$lib/stores";

  currentSession.set({ name: "", _id: "", token: "" });
  localStorage.removeItem("donation");
  goto("/");
</script>
```

Logout

- When Logout button pressed, local storage will be cleared.



Key	Value
donation	{"email":"homer@simpson.com","token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyNTE1NzljOTU0NmZmMjMwM2Qx...

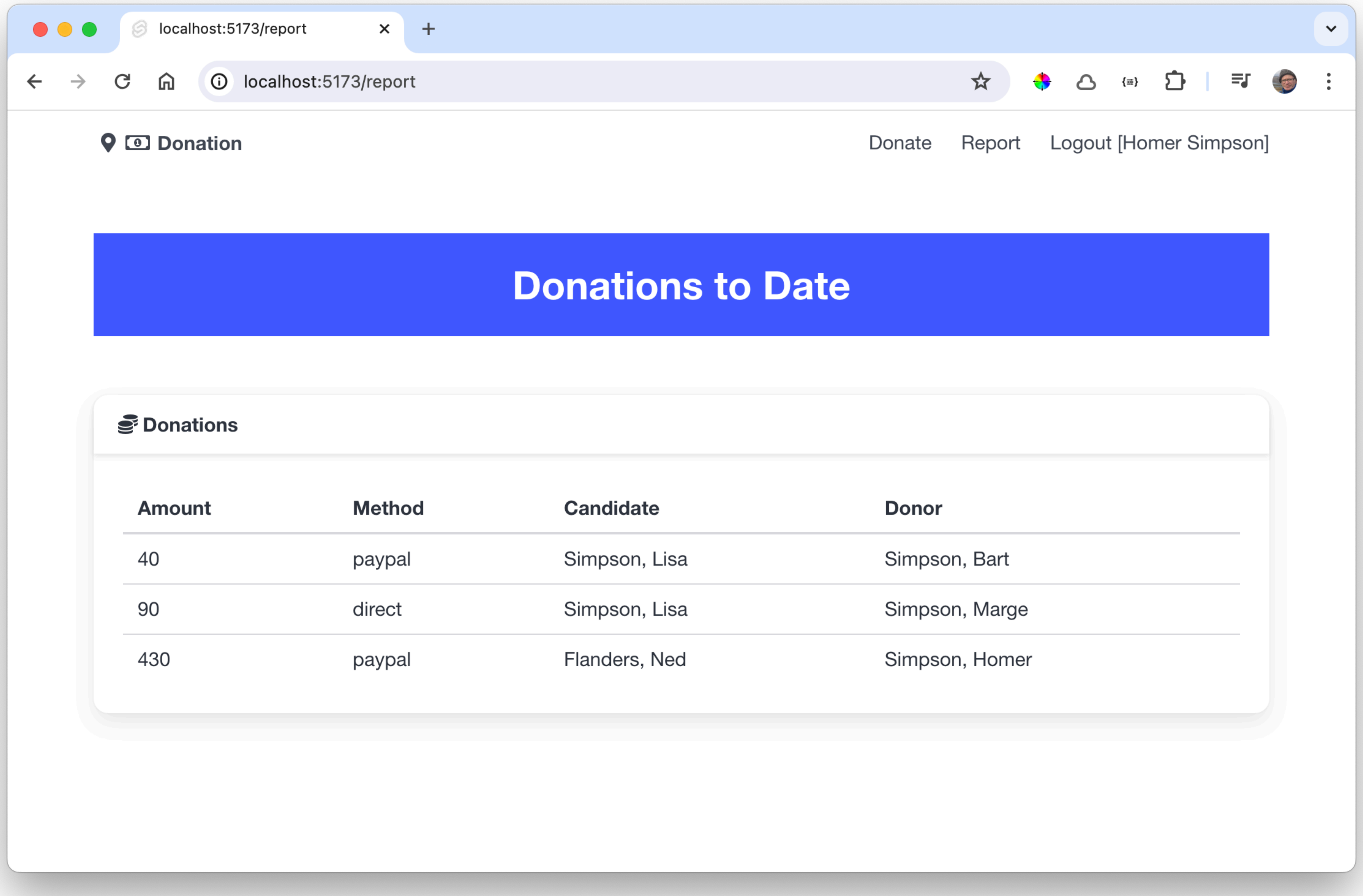
```
▼{email: "homer@simpson.com",…}
   email: "homer@simpson.com"
   token: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyNTE1NzljOTU0NmZmMjMwM2QxNmY0OCIsImVtYWlsIjoiaG9tZXJAc2ltcHNvbi5jb20iLCJpYXQiOjE2NDk1NzM0MTksImV4
```

Key	Value

Select a value to preview

Browser Reload Button

- There should be no change in application status

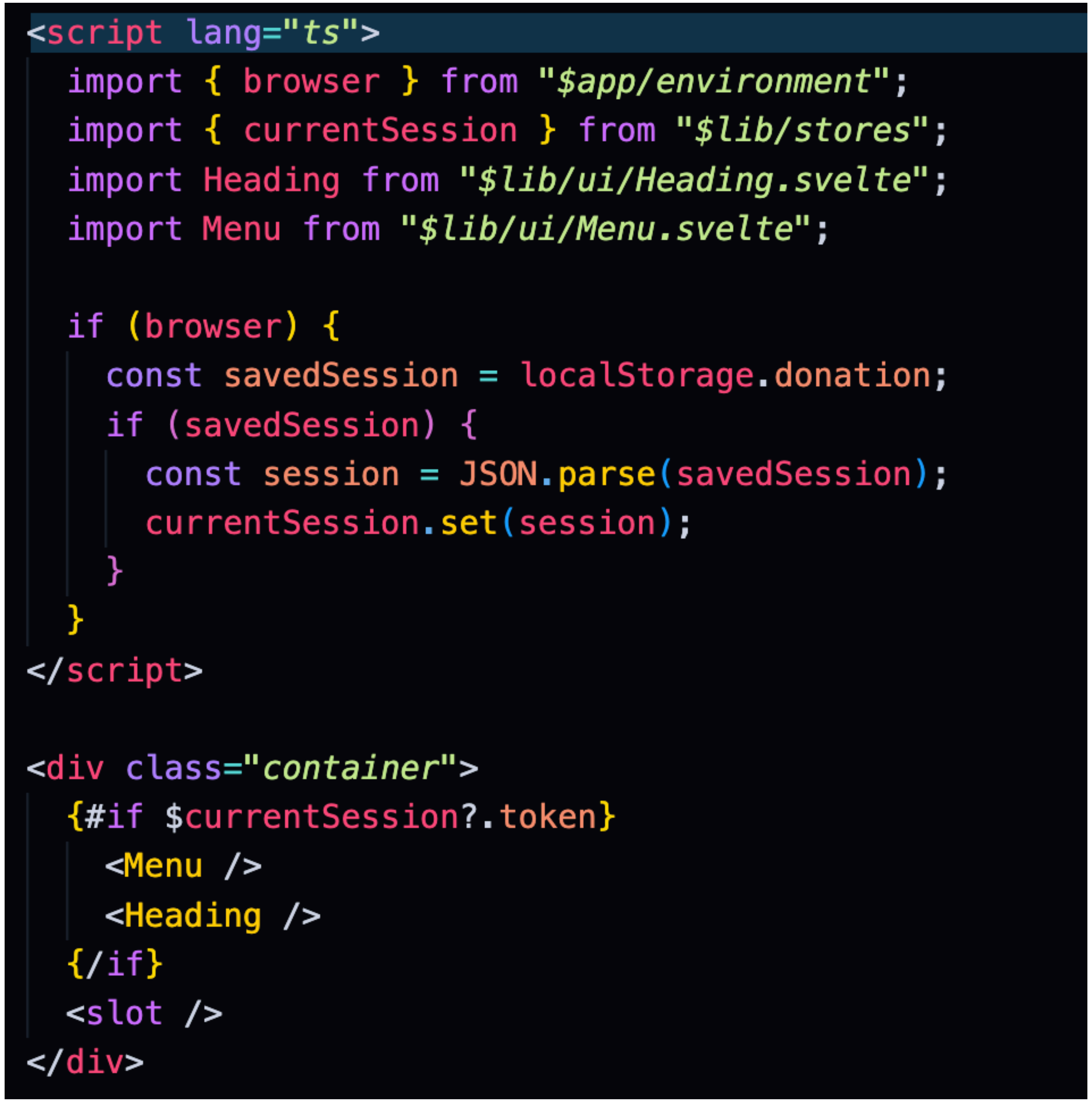


Donations to Date

Donations

Amount	Method	Candidate	Donor
40	paypal	Simpson, Lisa	Simpson, Bart
90	direct	Simpson, Lisa	Simpson, Marge
430	paypal	Flanders, Ned	Simpson, Homer

+layout.svelte

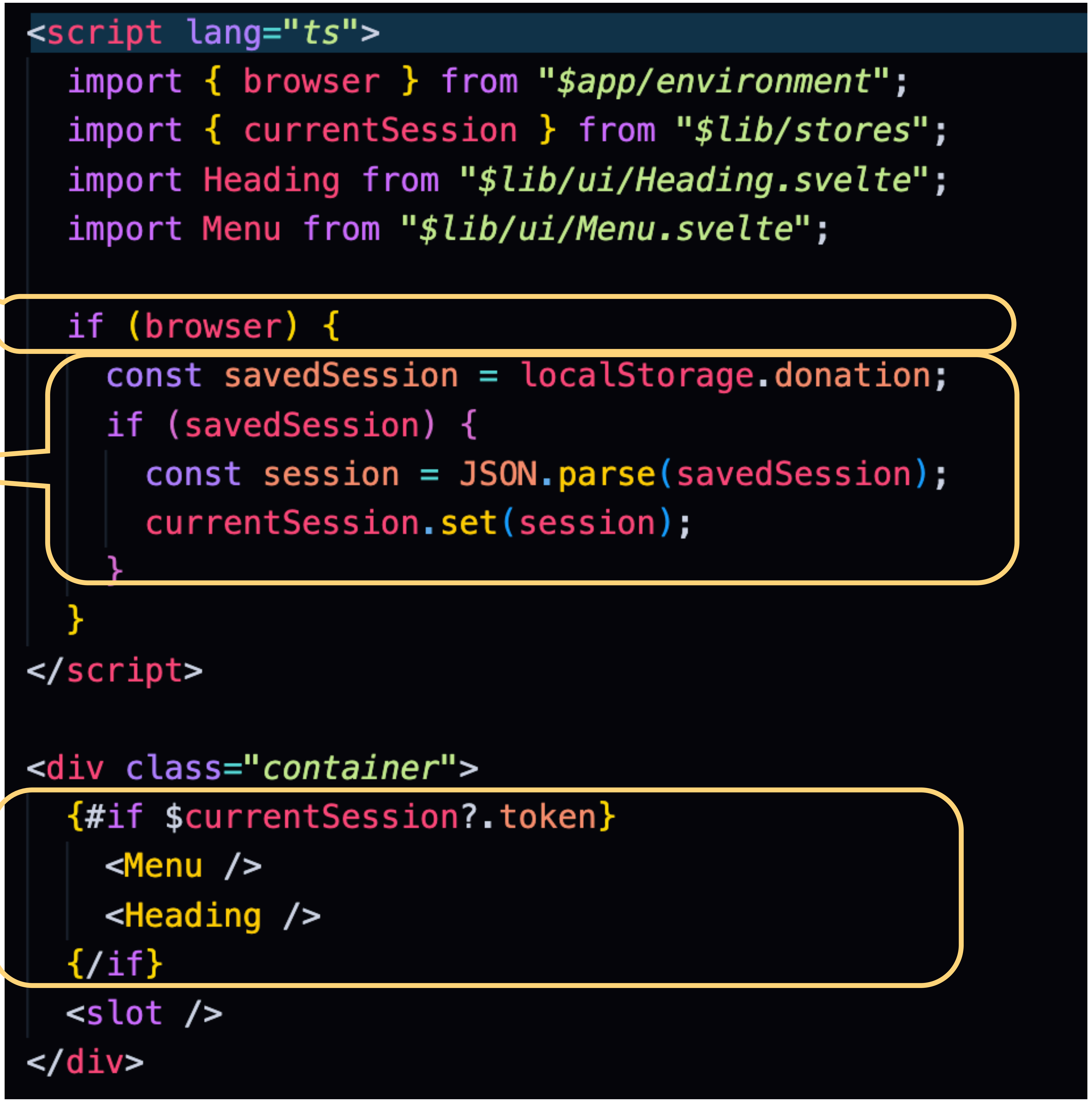


```
<script lang="ts">
  import { browser } from "$app/environment";
  import { currentSession } from "$lib/stores";
  import Heading from "$lib/ui/Heading.svelte";
  import Menu from "$lib/ui/Menu.svelte";

  if (browser) {
    const savedSession = localStorage.donation;
    if (savedSession) {
      const session = JSON.parse(savedSession);
      currentSession.set(session);
    }
  }
</script>

<div class="container">
  {#if $currentSession?.token}
    <Menu />
    <Heading />
  {/if}
  <slot />
</div>
```

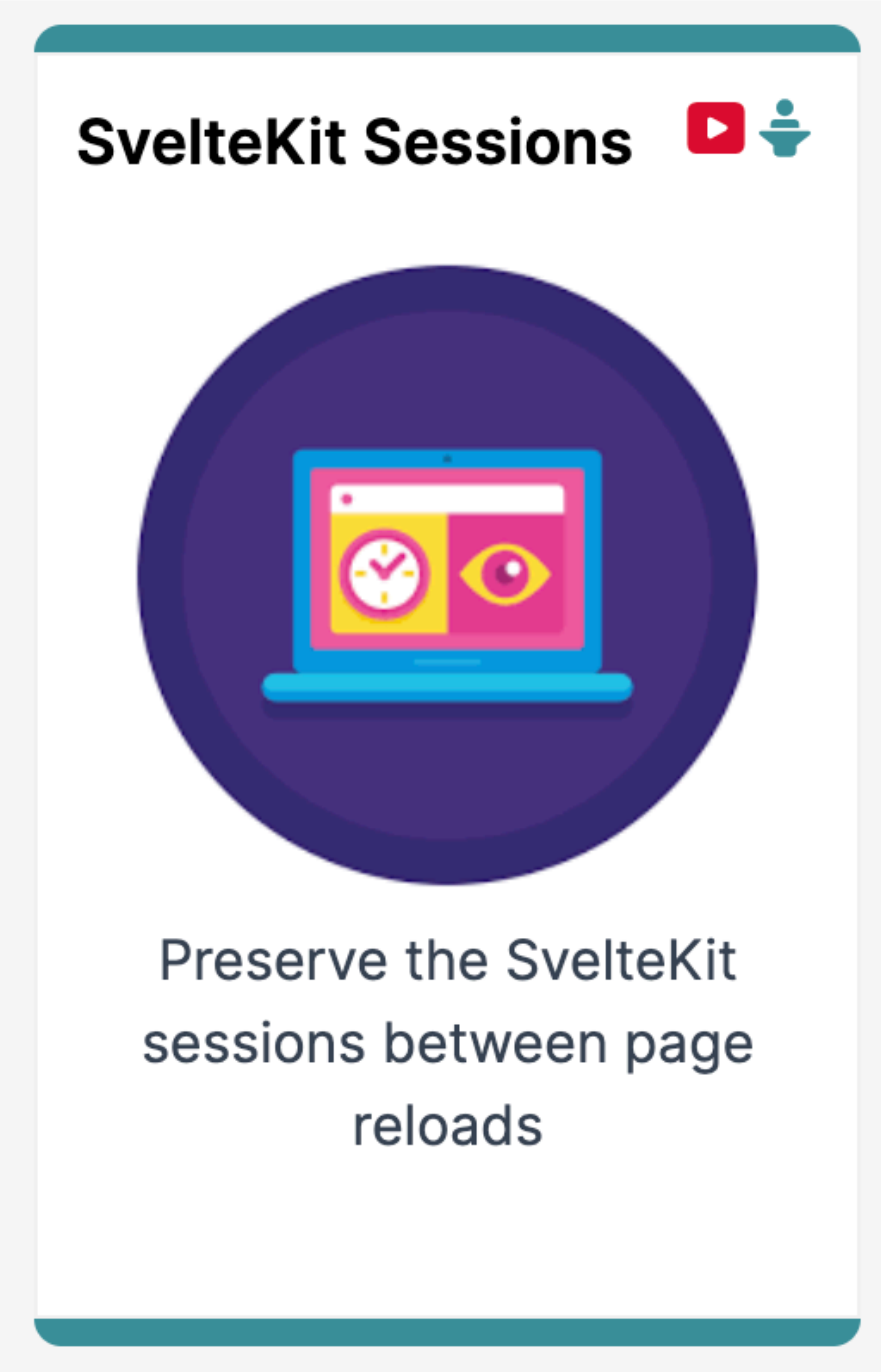
Browser Reload Button

- Are we running in the client?
- If there is a session in localStorage
  - Recover the session
  - Set the currentSession store
- currentSession values will ensure Menu + Heading will be displayed
-

+layout.svelte



```
<script lang="ts">
  import { browser } from "$app/environment";
  import { currentSession } from "$lib/stores";
  import Heading from "$lib/ui/Heading.svelte";
  import Menu from "$lib/ui/Menu.svelte";

  if (browser) {
    const savedSession = localStorage.donation;
    if (savedSession) {
      const session = JSON.parse(savedSession);
      currentSession.set(session);
    }
  }
</script>

<div class="container">
  {#if $currentSession?.token}
    <Menu />
    <Heading />
  {/if}
  <slot />
</div>
```



SvelteKit Sessions

Preserve the SvelteKit sessions between page reloads