Example app using ReasonML & ReasonReact components

This example builds upon the original `with-reasonml` example to show how a global state object can be used to track state across page within the application.

It is intended to show how to build a simple, stateful application using hooks without the added complexity of a redux type library.

This example features:

- An app that mixes together JavaScript and ReasonML components and functions
- An app with two pages which has a common Counter component
- That Counter component maintain the counter inside its module. This is used primarily to illustrate that modules get initialized once and their state variables persist in runtime

Deploy your own

Deploy the example using Vercel:

Deploy

How to use

Execute `create-next-app` with npm or Yarn to bootstrap the example:

```
npx create-next-app --example with-reasonml-todo with-reasonml-app
# or
yarn create next-app --example with-reasonml-todo with-reasonml-app
# or
pnpm create next-app -- --example with-reasonml-todo with-reasonml-app
```

Deploy it to the cloud with Vercel (Documentation).

Recommendation:

Run BuckleScript build system `bsb -w` and `next -w` separately. For the sake of simple convention, `npm run dev` run both `bsb` and `next` concurrently. However, this doesn't offer the full colorful and very, very, veeeery nice error output experience that ReasonML can offer, don't miss it!

There are 2 convenience scripts to facilitate running these separate processes:

1. `npm run dev:reason` - This script will start the ReasonML toolchain in watch mode to re-compile whenever you make changes.
2. `npm run dev:next` - This script will start the next.js development server so that you will be able to access your site at the location output by the script. This will also hot reload as you make changes.

You should start the scripts in the presented order.