NILS HARTMANN
https://nilshartmann.net

Nächste Generation Webframeworks

Alles Fullstack oder was?

Slides: https://react.schule/fullstack-oder-was

NILS HARTMANN

nils@nilshartmann.net

Freiberuflicher Entwickler, Architekt, Trainer aus Hamburg

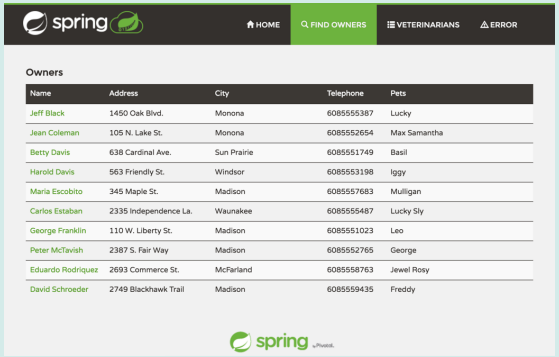
Java, Spring, GraphQL, React, TypeScript

https://graphql.schule/video-kurs

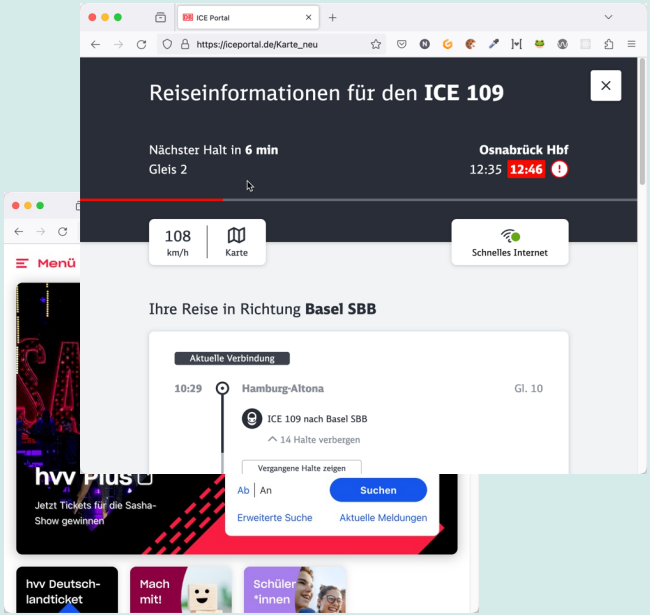
https://reactbuch.de

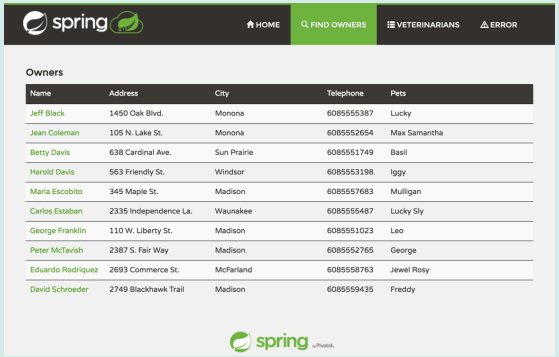
Webanwendungen...

WEBANWENDUNGEN

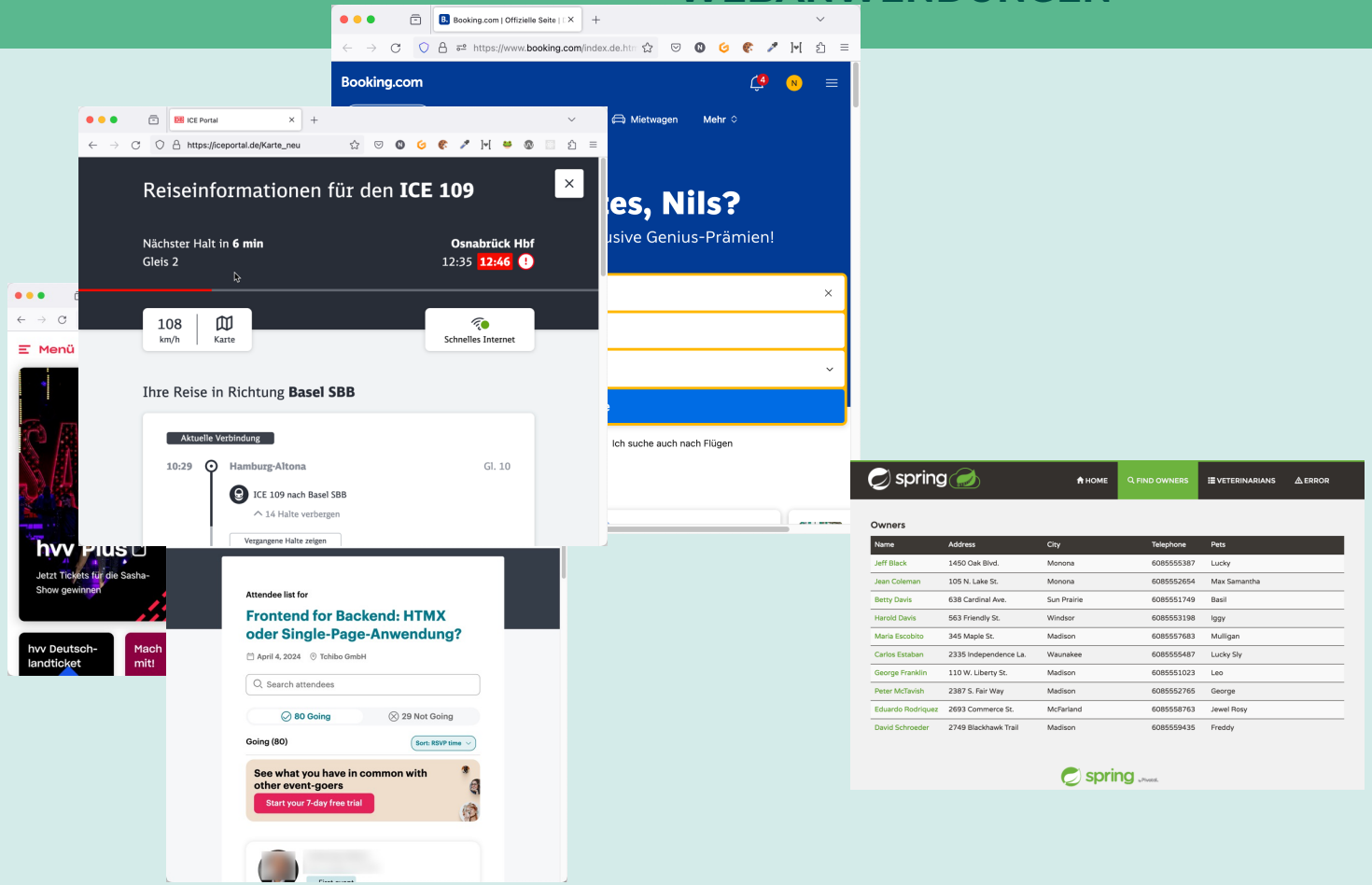


WEBANWENDUNGEN

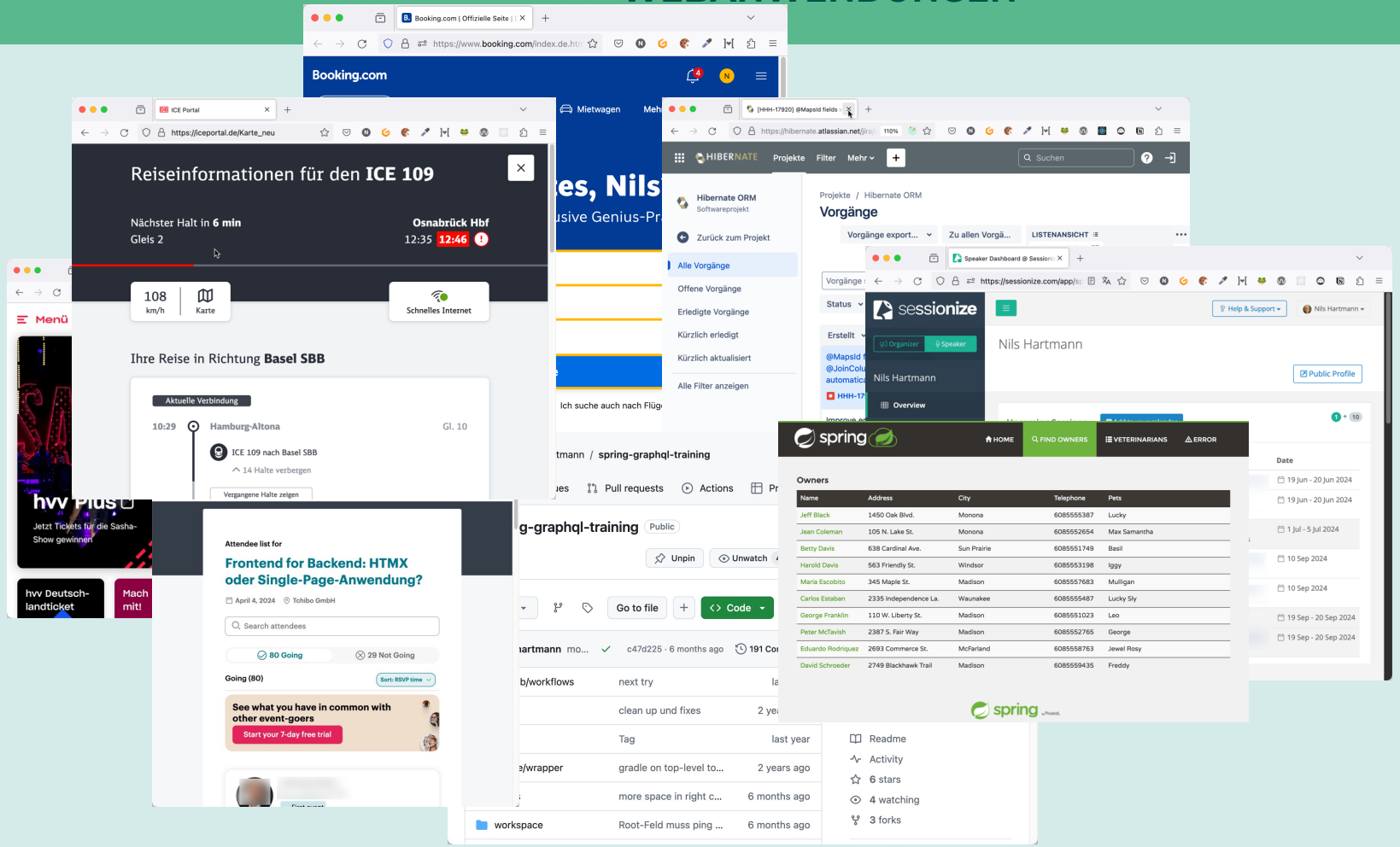




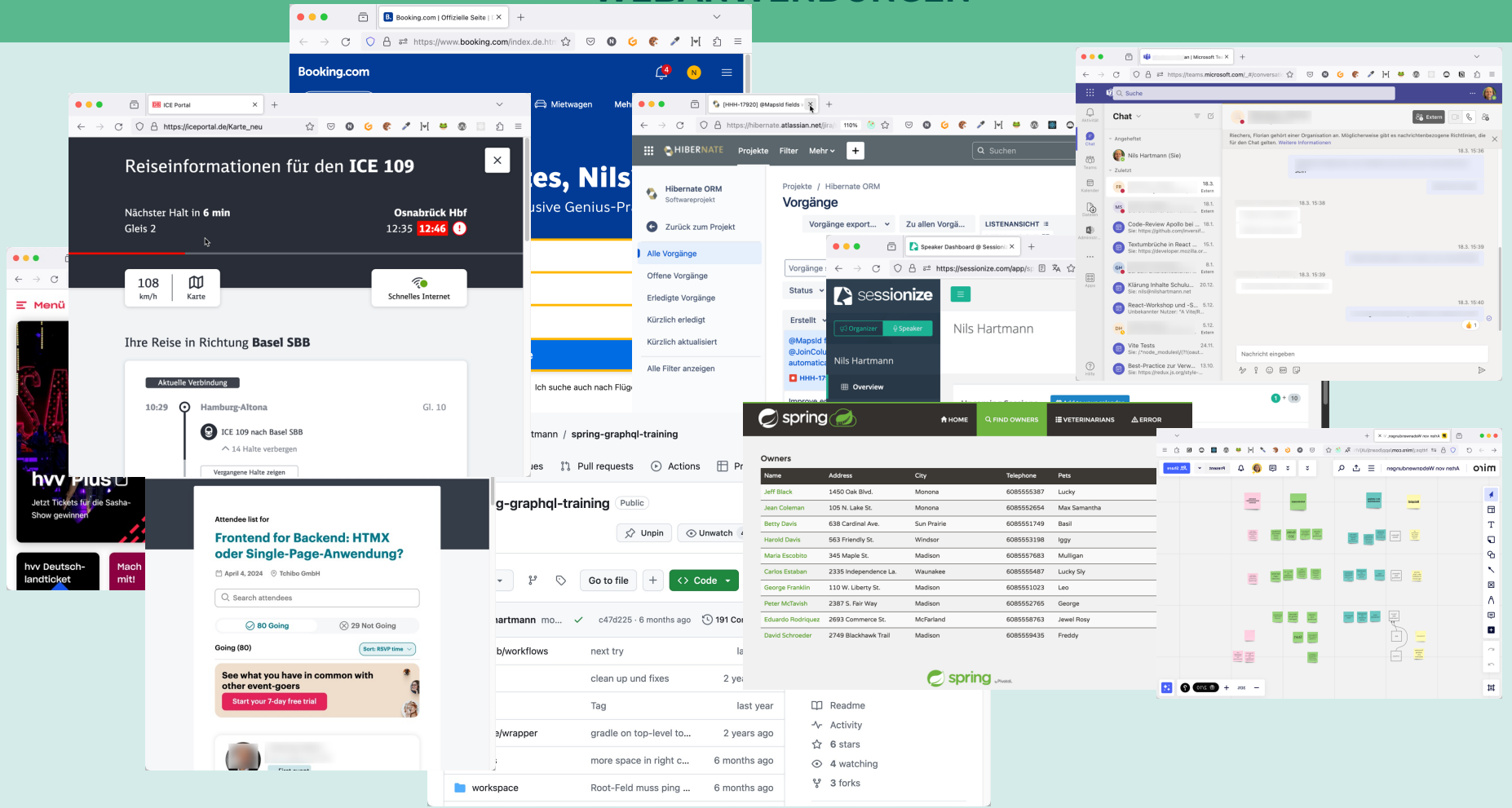
WEBANWENDUNGEN



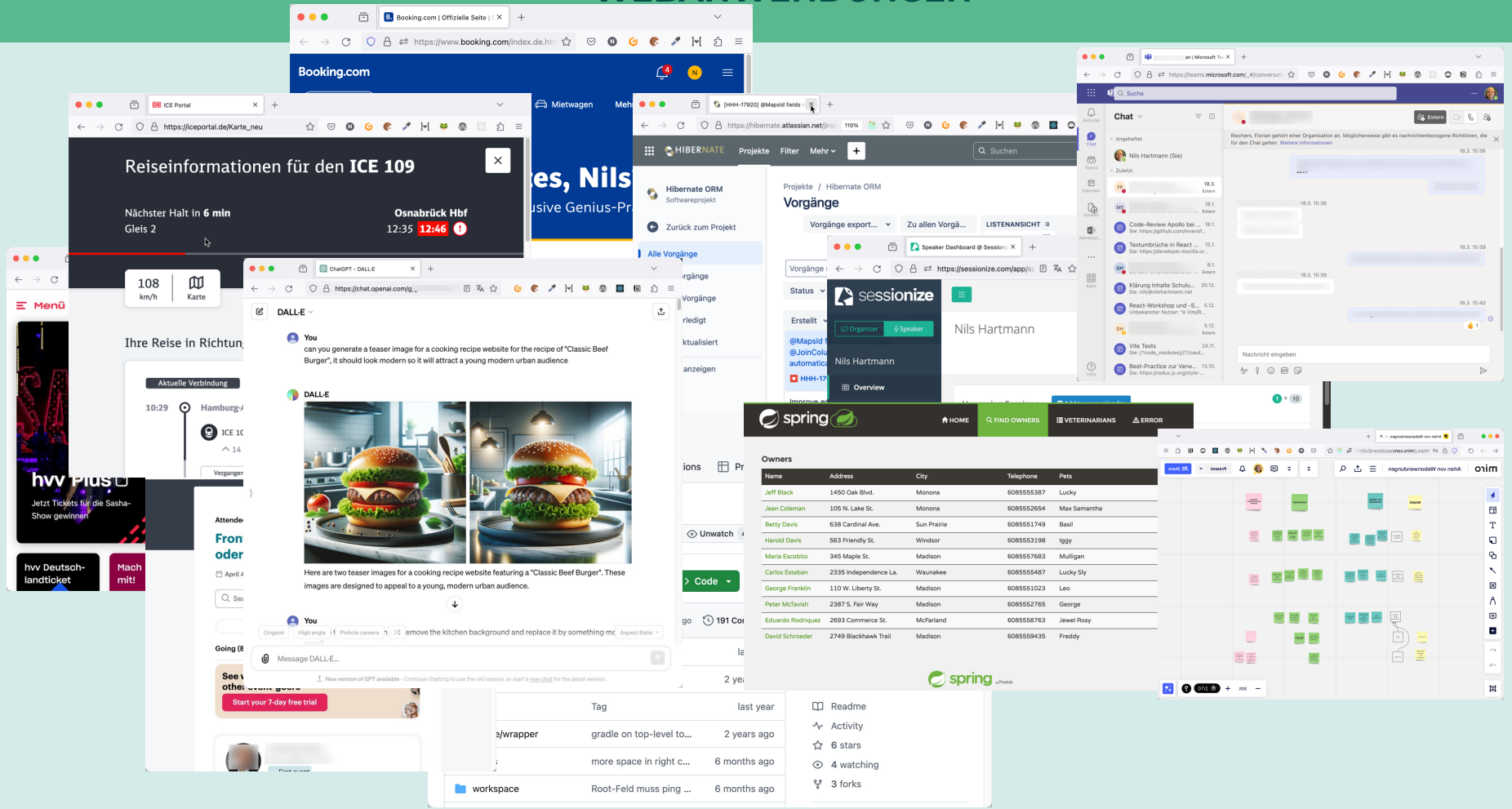
WEBANWENDUNGEN



WEBANWENDUNGEN

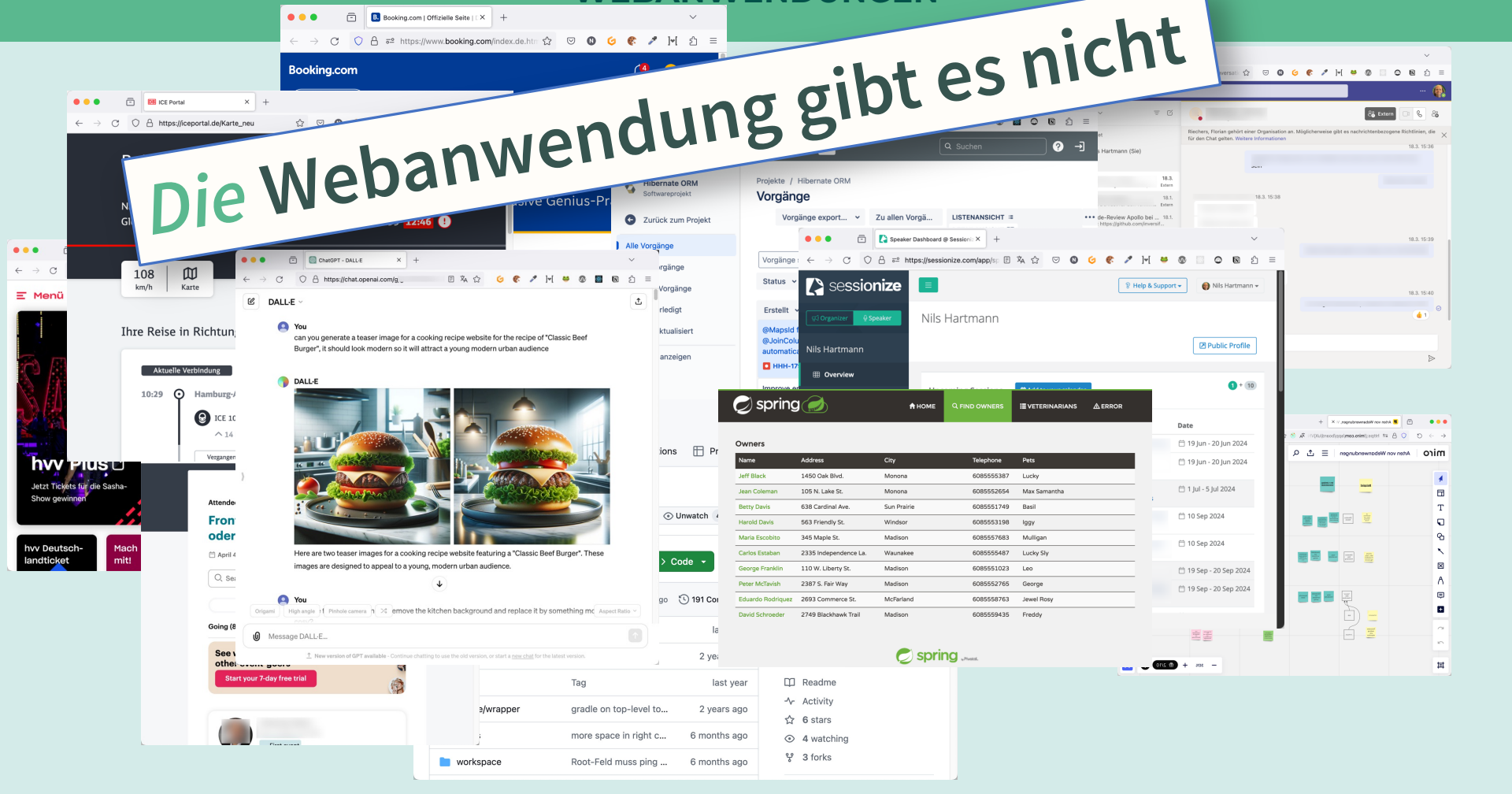


WEBANWENDUNGEN



WEBANWENDUNGEN

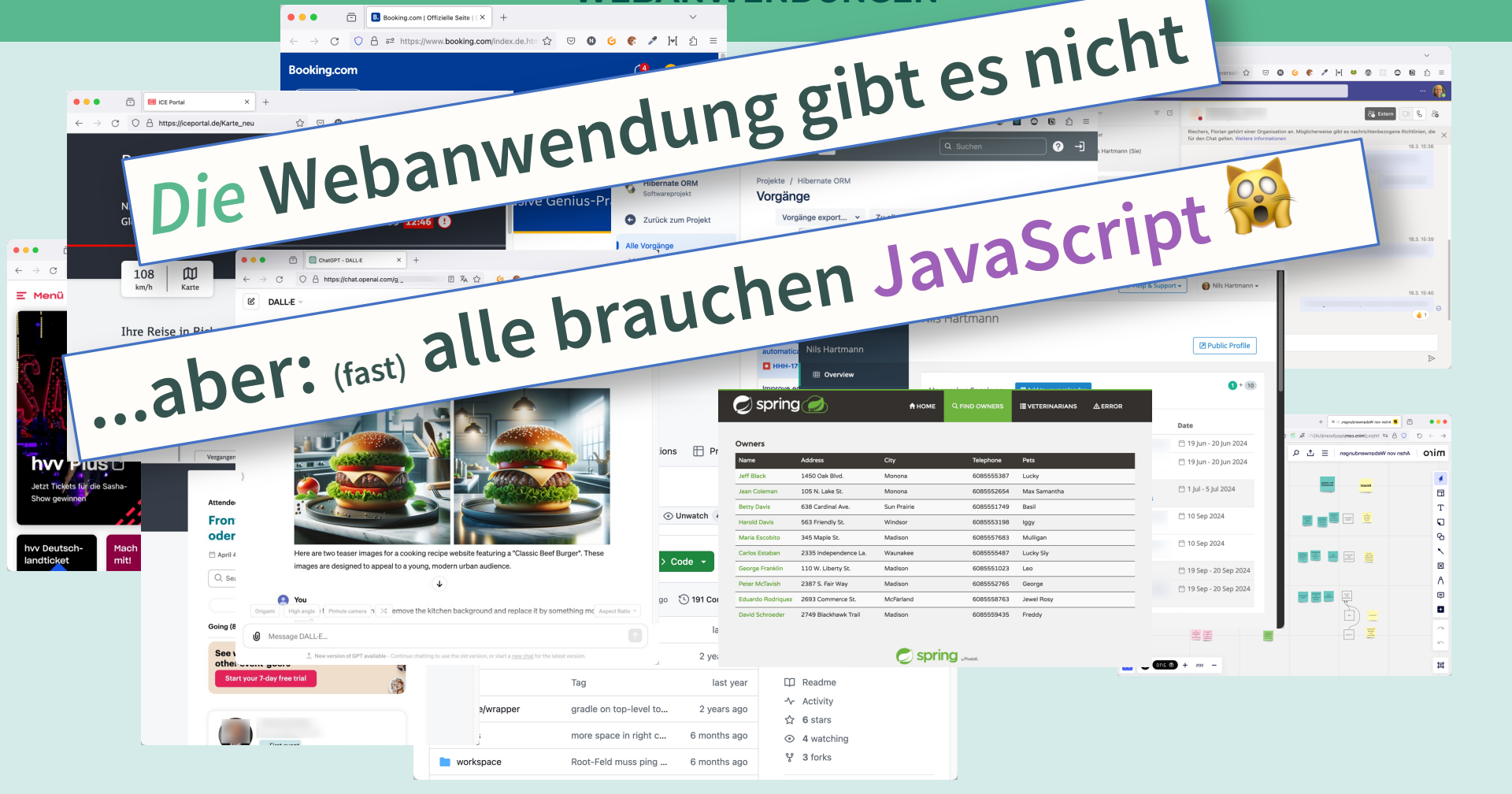
Die Webanwendung gibt es nicht



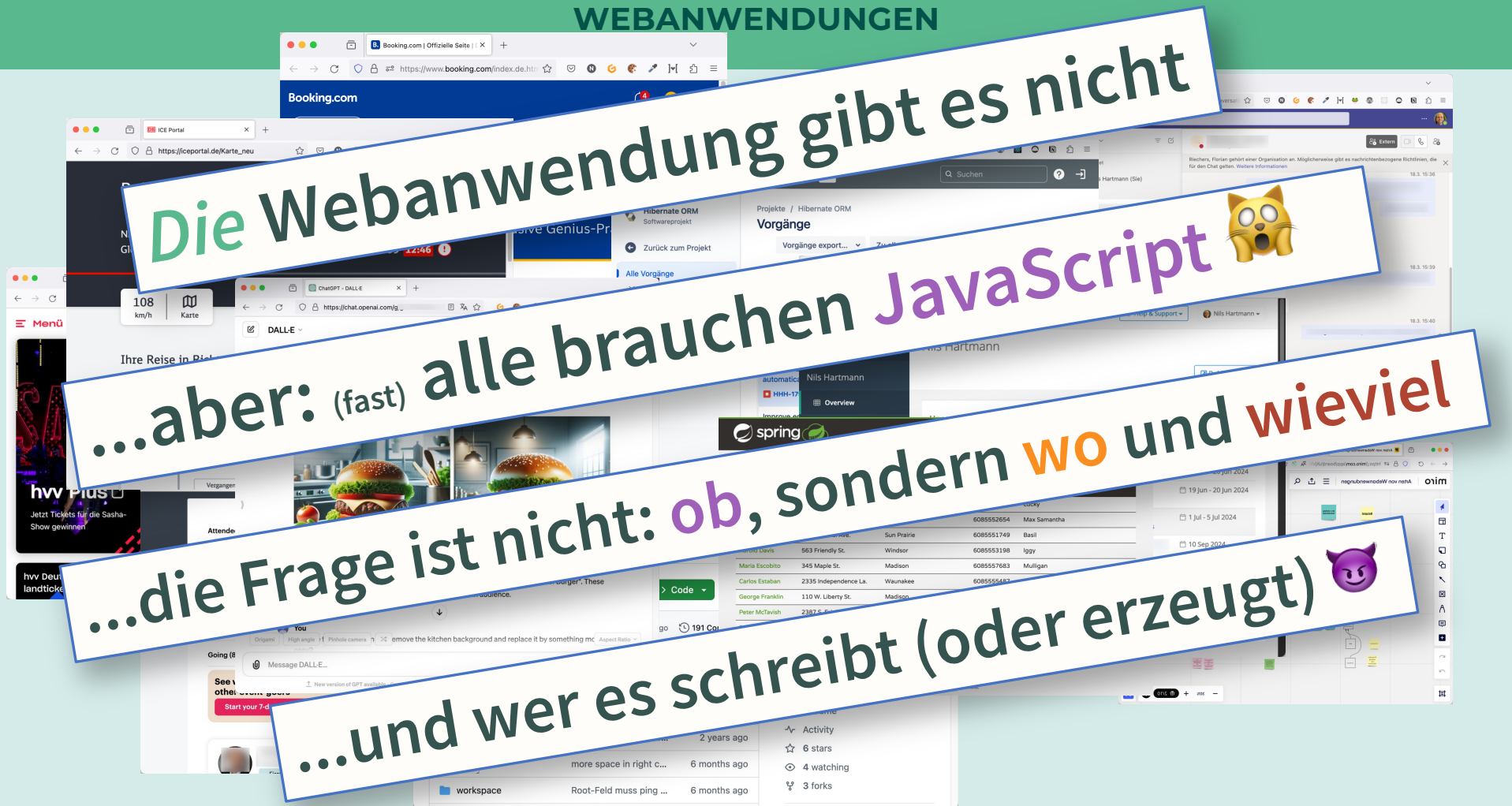
WEBANWENDUNGEN

Die Webanwendung gibt es nicht

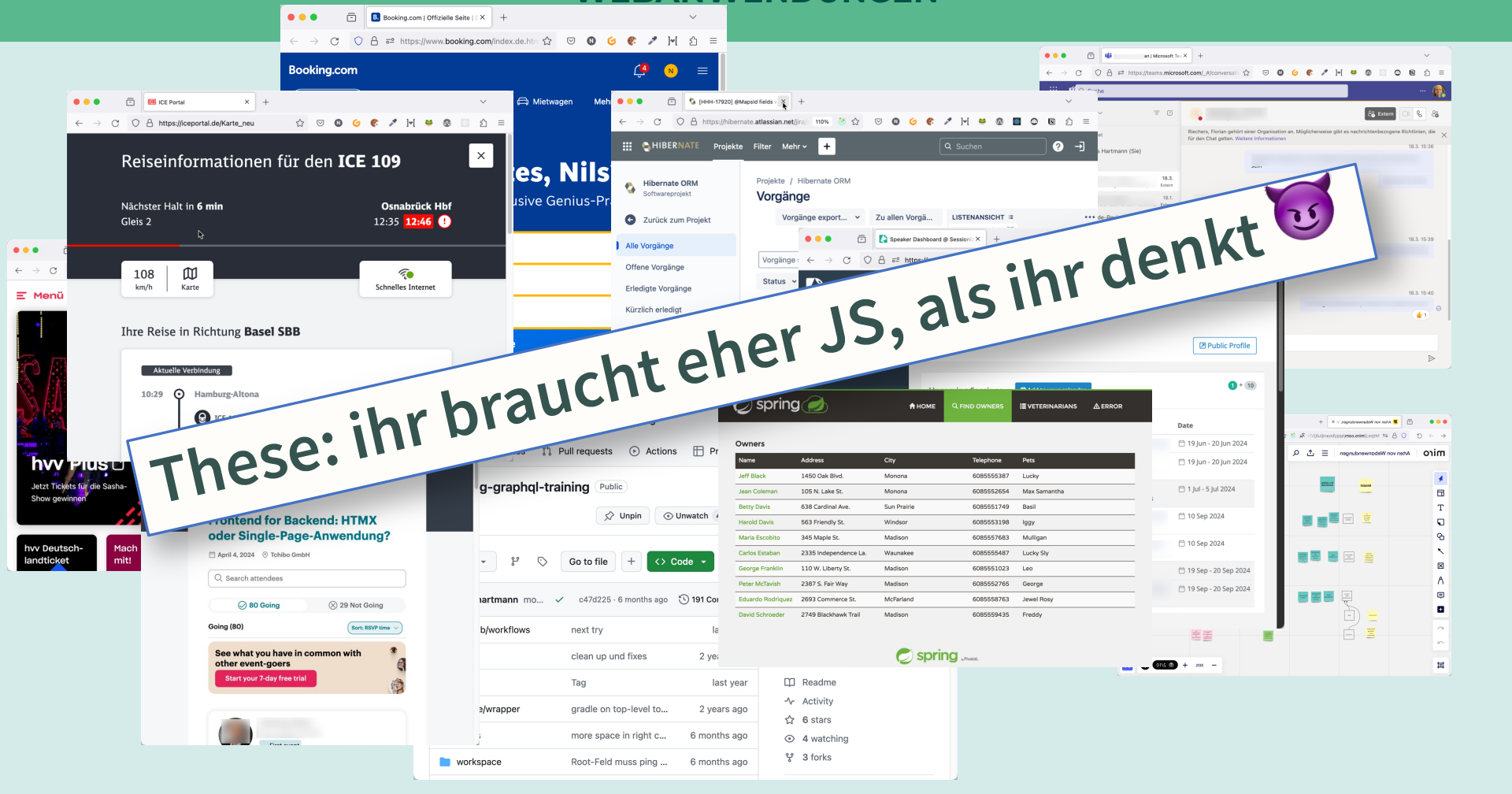
...aber: (fast) alle brauchen JavaScript



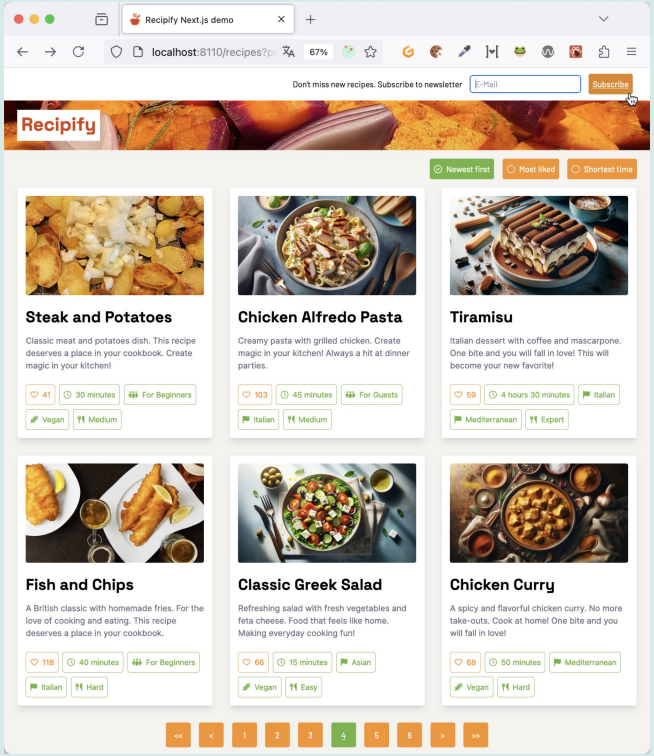

Die Webanwendung gibt es nicht
...aber: (fast) alle brauchen JavaScript
...die Frage ist nicht: ob, sondern wo und wieviel
Booking.com
Vorgänge
Projekte / Hibernate ORM
Zurück zum Projekt
Alle Vorgänge
Vorgänge export...
DALL·E
Message DALL·E...
Start your 7-day free trial
Nils Hartmann
Overview
spring
Code
Readme
Activity
6 stars
4 watching
3 forks
workspace
Root-Feld muss ping ...
6 months ago
2 years ago
hvv Plus
Jetzt Tickets für die Sasha-Show gewinnen


Die Webanwendung gibt es nicht
...aber: (fast) alle brauchen JavaScript
...die Frage ist nicht: ob, sondern wo und wieviel
...und wer es schreibt (oder erzeugt)

WEBANWENDUNGEN



BEISPIEL-ANWENDUNG



- http://localhost:8110

BEISPIEL-CODE

Ihr findet das "Recipify"-Beispiel hier:

Next.JS

https://github.com/nilshartmann/alles-fullstack

Es geht also "nur" darum, wie wir mit JavaScript umgehen:

Es geht also "nur" darum, wie wir mit JavaScript umgehen:

Welche Konsequenz hat JavaScript zur Laufzeit

Es geht also "nur" darum, wie wir mit JavaScript umgehen:

Welche Konsequenz hat JavaScript zur Laufzeit

...und bereits während der Entwicklung?

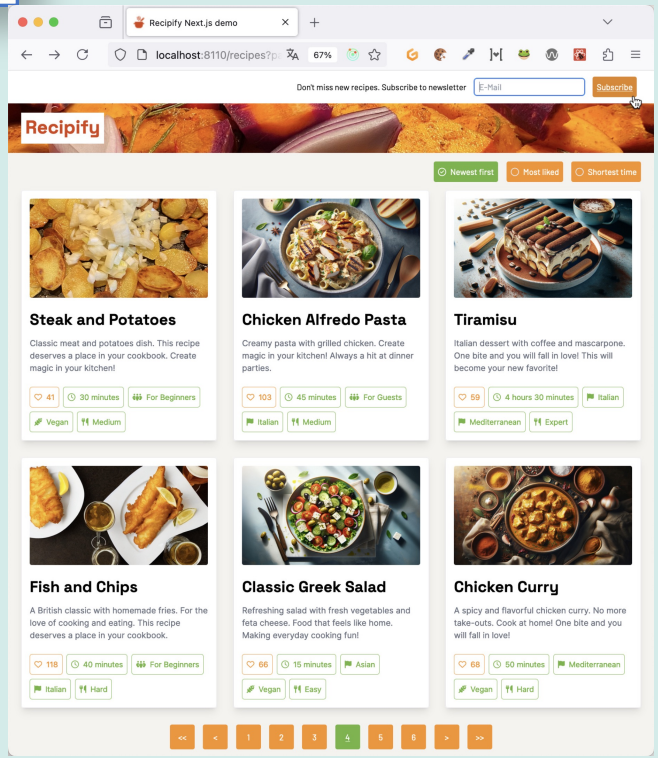
Klassische serverseitige Anwendung...

Klassische serverseitige Anwendung...

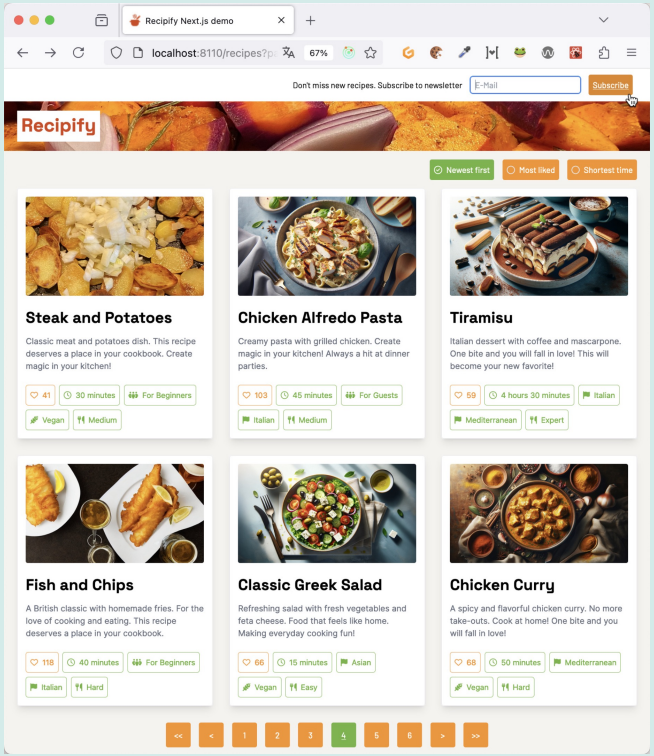
- Typische Vertreter zum Beispiel Spring MVC, PHP, .NET, Node.JS
- Oft Model-View-Controller (MVC) Pattern
- Templatesprache (z.B. Thymeleaf, Handlebars)

Unser Server erzeugt eine HTML-Seite...

```java
@Controller
public class RecipeController {

    private final RecipeRepository repository;

    RecipeController(RecipeRepository repository) {
        this.repository = repository;
    }

    @GetMapping("/")
    String getRecipesList(Pageable pageable, Model model) {

        var recipes = this.repository.findAll(pageable);

        // Model mit allen Informationen erzeugen, die
        // das Template zum rendern braucht
        model.addAttribute("recipes", recipes);
        model.addAttribute("pageable", pageable);

        // Name des Templates, das den HTML-Code für die
        // Seite erzeugt
        return "index";
    }
}
```

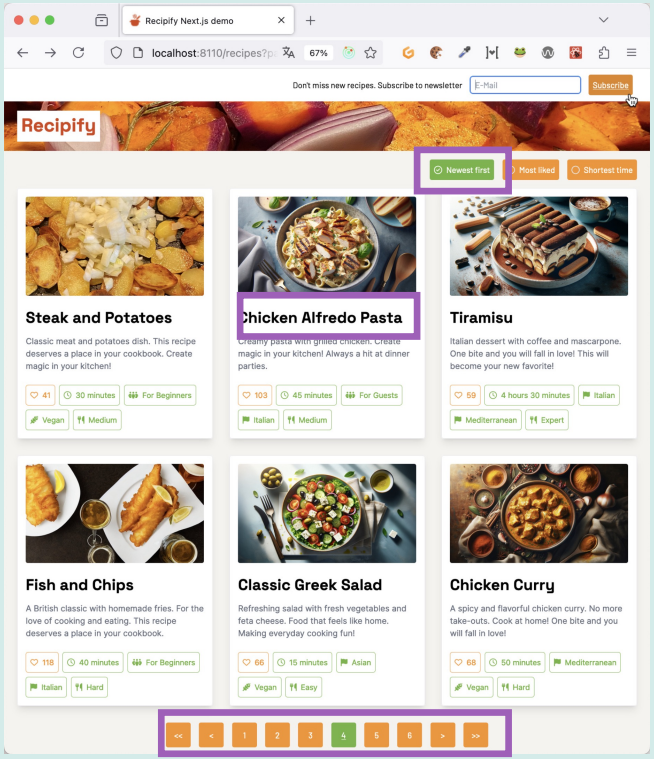


...die ist aber komplett statisch



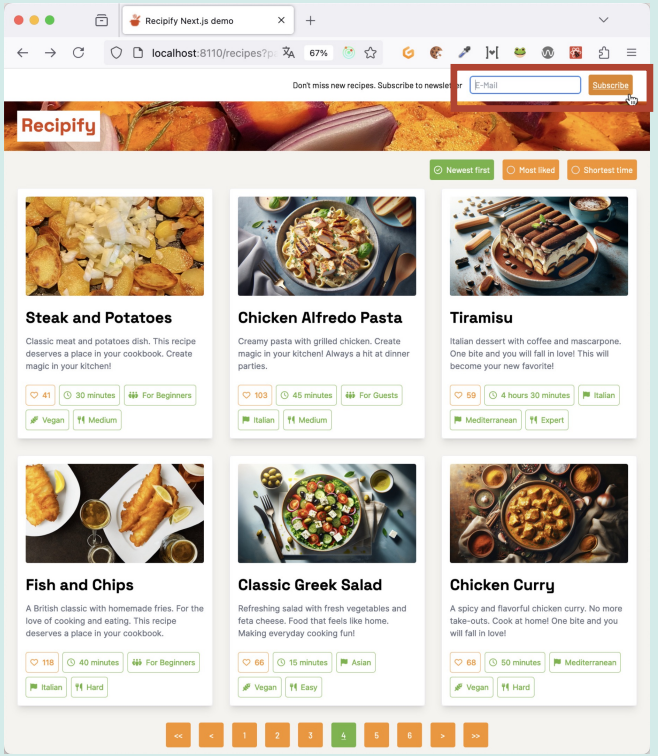
...die ist aber komplett statisch

- Interaktion per **Link**



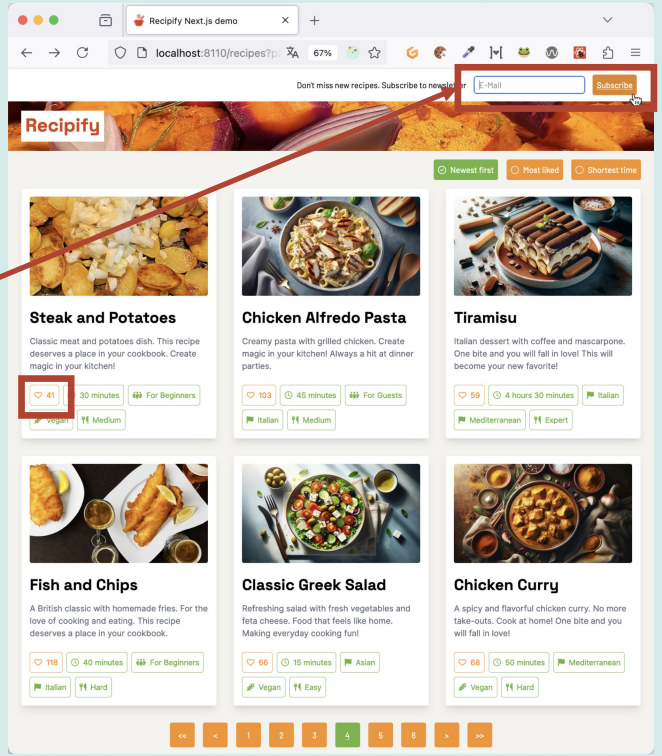
...die ist aber komplett statisch

- Interaktion per **Link**
- oder **Formular**

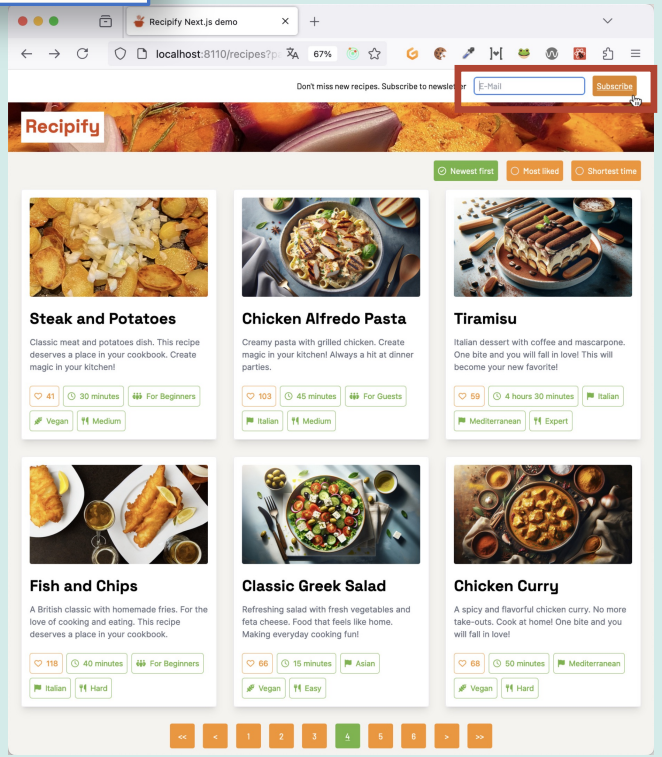


...die ist aber komplett statisch

- Interaktion per **Link**
- oder **Formular**
- in jedem Fall wird eine komplett neue Seite abgefragt



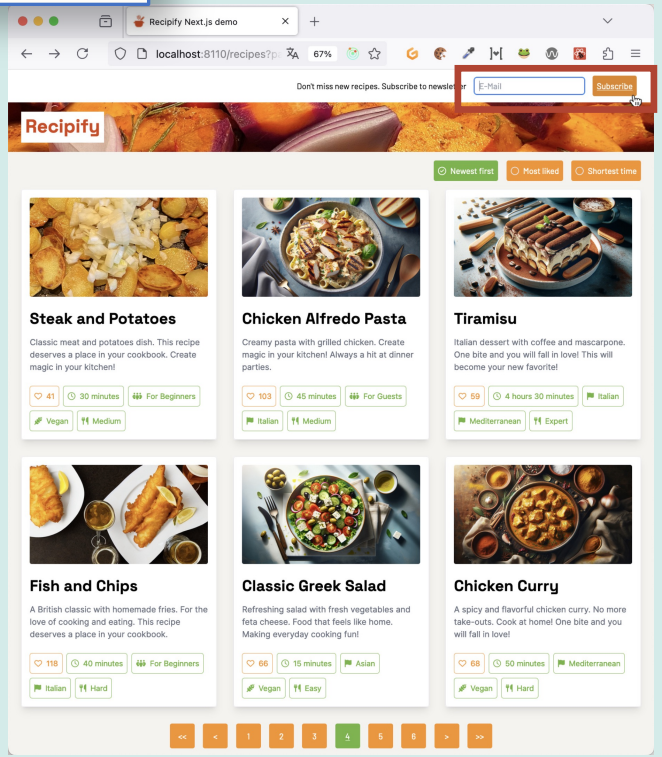
Bei jeder Interaktion: ein Server-Roundtrip…



- ist das ein Problem? 🤔
- was hat das für Konsequenzen? 🤔

Bei jeder Interaktion: ein Server-Roundtrip…

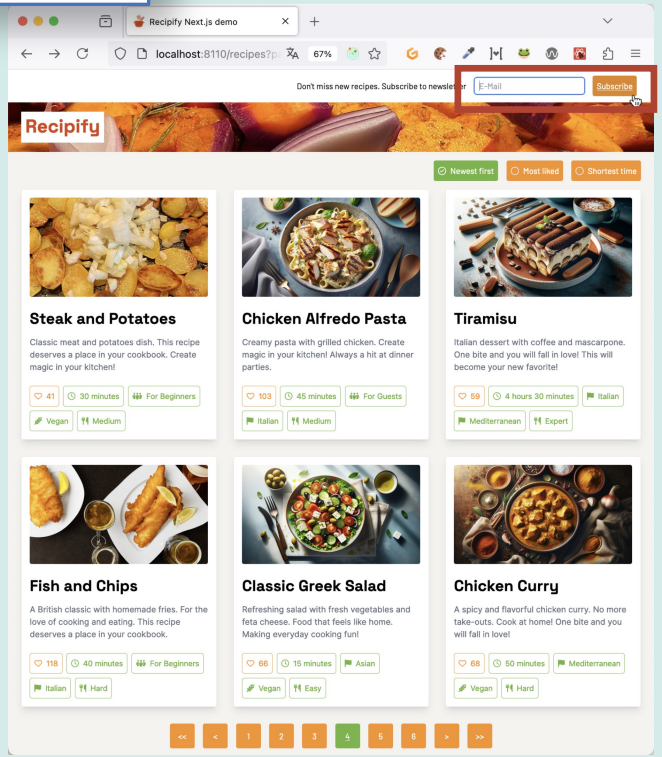
- Menge an Daten, Latenz
- Zustand in Eingabefeldern geht verloren
- Animationen etc. laufen nicht weiter



Bei jeder Interaktion: ein Server-Roundtrip…

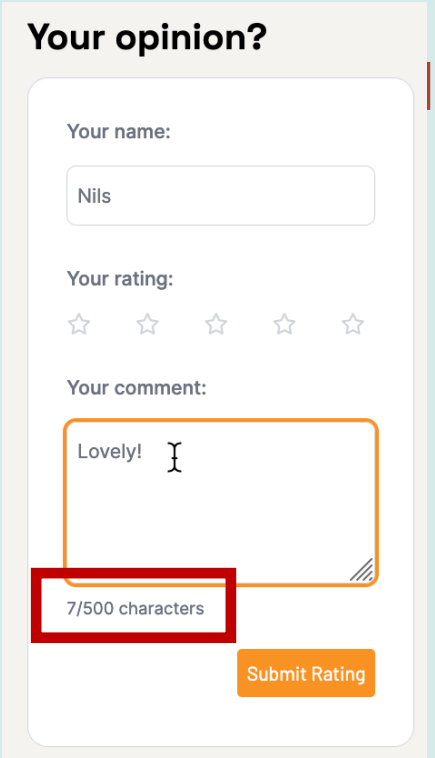
- Menge an Daten, Latenz
- Zustand in Eingabefeldern geht verloren
- Animationen etc. laufen nicht weiter

- 🕵️ Newsletter
- 🕵️ Like-Button
- 🕵️ Timer



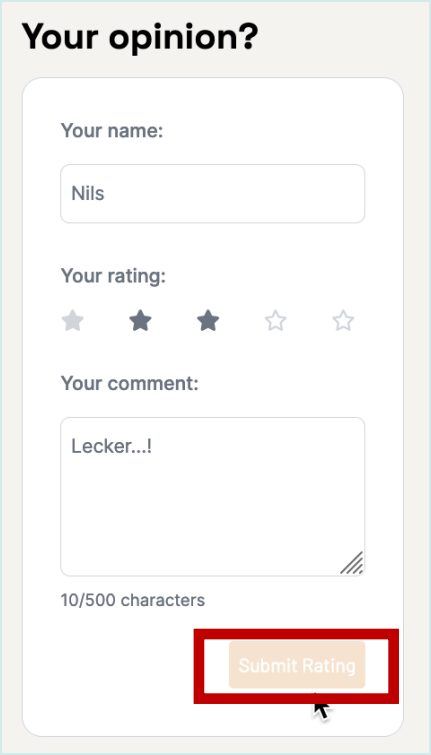
Weitere Interaktionen...

- Feedback direkt beim Tippen
- z.B. Zeichen-Zähler

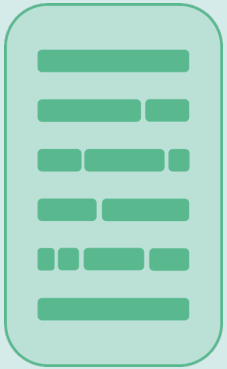


Weitere Interaktionen...

- Feedback direkt beim Tippen
- z.B. Zeichen-Zähler
- Doppeltes Submit verhinden



"Können wir nicht hier und da, ad-hoc JavaScript hinzufügen?"



HTML-Seite

Ja, das geht!

HTML-Seite

- JavaScript Schnipsel einstreun ("vanilla JS" oder zum Beispiel jQuery)

Ja, das geht!

HTML-Seite

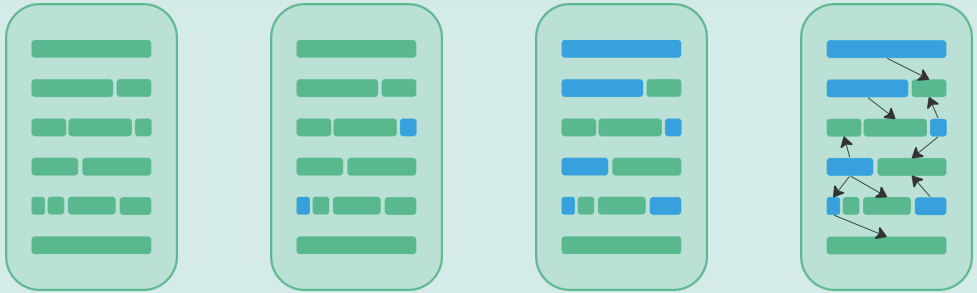
- Eigentlich optimal:
  - wir haben JavaScript nur da, wo wir es wirklich brauchen, für Interaktivität
  - alles andere kann statisches HTML und CSS sein ❤️

"...hier und da müsste auch noch schnell was interaktives hin..."

HTML-Seite

- Wir schreiben also noch etwas mehr JavaScript Schnipsel

"...und hier... und hier ... und hier... und ..." au weia!



HTML-Seite

Das Problem von Schnipsel-Architektur

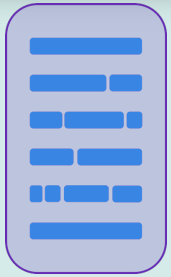
- Bunter Strauß an Server- und Client-Technologien (Backend-Sprache, Template-Sprache, JavaScript)
- Verantwortlichkeit willkürlich auf Frontend und Backend aufgeteilt

Das Problem von Schnipsel-Architektur

HTML-Seite

- Die Probleme "schleichen sich ein"
- Spätestens wenn es um Interaktionen geht, die nicht nur lokal sind
- Plötzlich hat unser Code nicht "ein paar Probleme" sondern ist kaputt

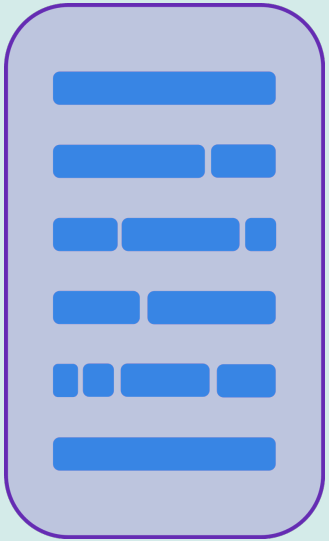
Dann besser alles in JavaScript? Single-Page-Anwendungen



JavaScript-Code

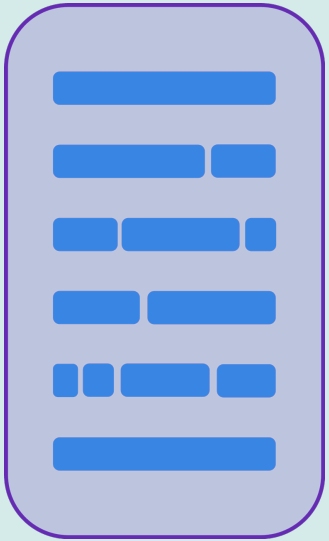
- ab ca. 2010
- Aus "Seiten" werden jetzt "Anwendungen"
- Klare Verantwortlichkeit: Server für Logik und Daten, Browser für UI
- Es gibt stabile und verbreitete Frameworks für jeden Geschmack

Single-Page-Anwendungen



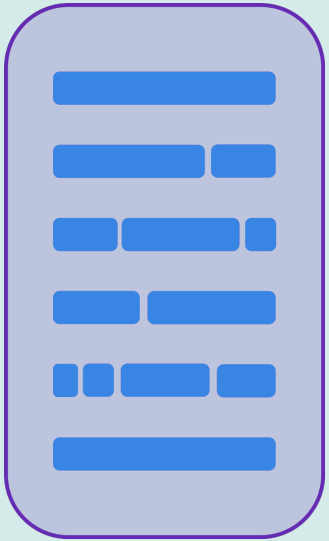
- Darstellung erfolgt vollständig mit JavaScript
- Statisches HTML spielt (fast) keine Rolle
- Die Anwendung kommuniziert mit dem Backend über API
- Ausgetauscht werden Daten, aber keine UI
- Vertreter: Angular, React, Svelte, Vue

Single-Page-Anwendungen



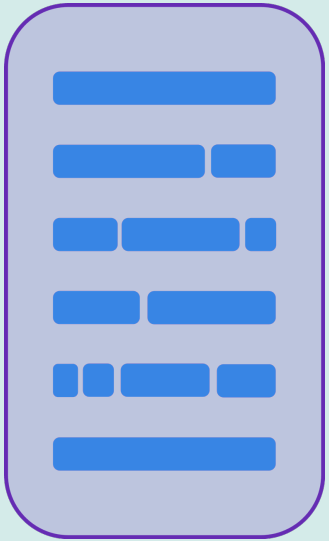
- Konsequenz #1: (viel) JavaScript zur Entwicklungszeit

Single-Page-Anwendungen



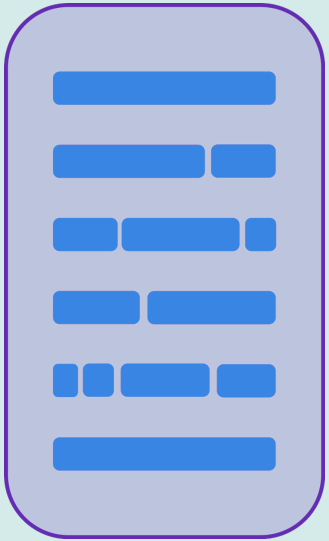
- Konsequenz #1: (viel) JavaScript zur Entwicklungszeit
- Konsequenz #2: (viel) JavaScript zur Laufzeit

Single-Page-Anwendungen



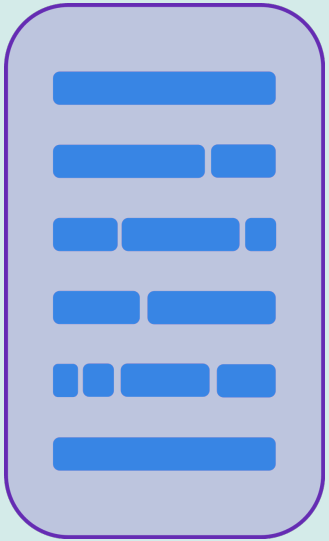
- **Konsequenz #2: viel JavaScript zur Laufzeit**
- Auch für statische Inhalte

Single-Page-Anwendungen



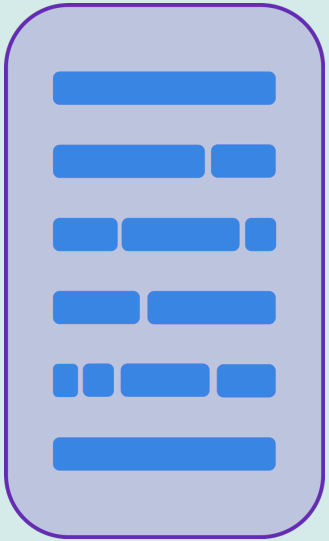
- JavaScript-Code:
    1. muss zum Browser gesendet werden
    2. muss vom Browser ausgeführt werden
    3. kann dann die darzustellenden Daten lesen
    4. kann dann erst die Daten anzeigen
    5. erst dann ist die Anwendung einsatzbereit
    6. Mit jedem Feature wird es mehr

Single-Page-Anwendungen



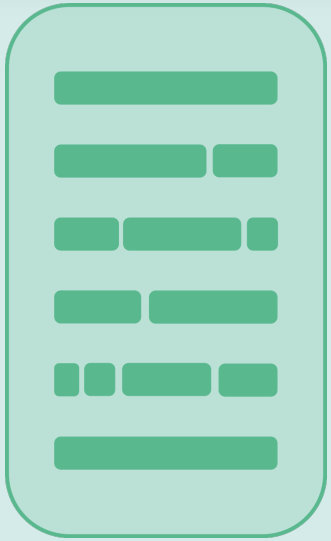
- Das hat Auswirkungen auf die **Laufzeit**-Performance
- Insbesondere beim Starten

Single-Page-Anwendungen

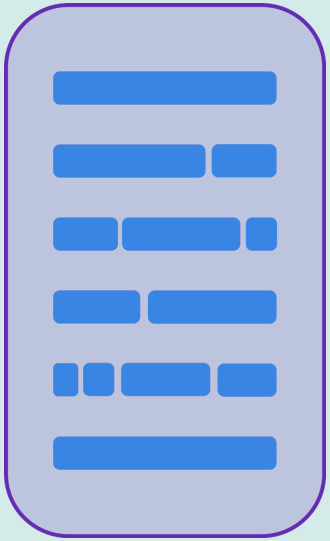


- Das hat Auswirkungen auf die **Laufzeit**-Performance
- Insbesondere beim Starten
- Ob das ein Problem für die eigene Anwendung ist, muss man von Fall zu Fall entscheiden
  - In-House- oder B2B-Anwendungen haben andere Anforderungen als ein Online-Shop

"Gibt es denn nichts dazwischen"?

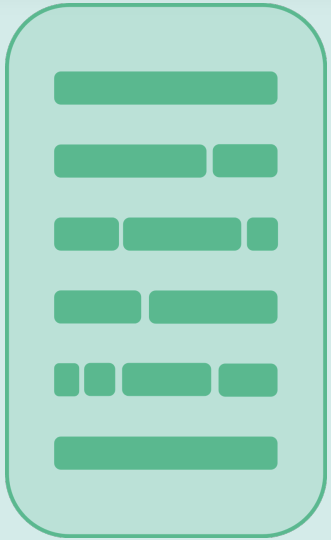


Nur Server (+JS)

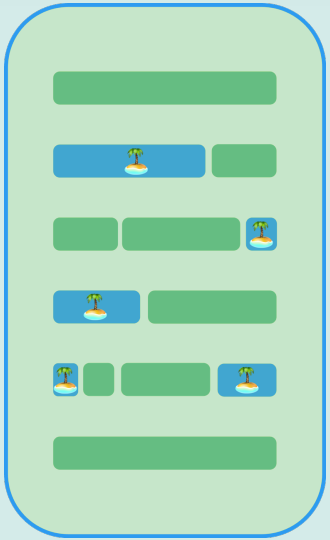


Nur Client (SPA)

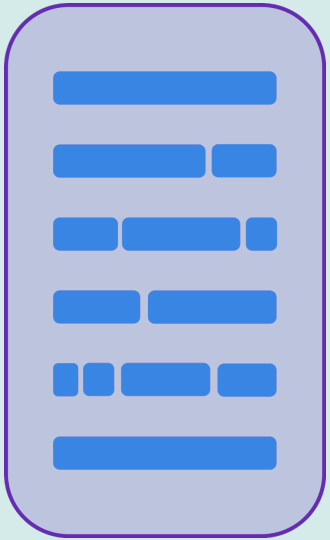
"Doch – Fullstack-Anwendungen 🥳"



Nur Server (+JS)

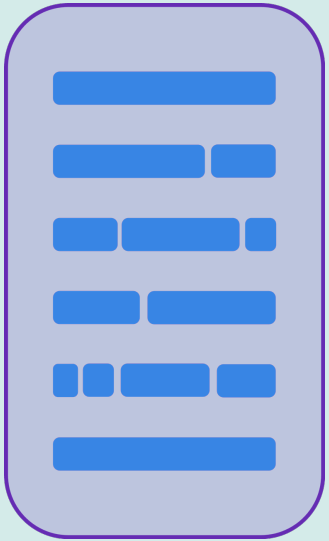


JS Fullstack Anwendung

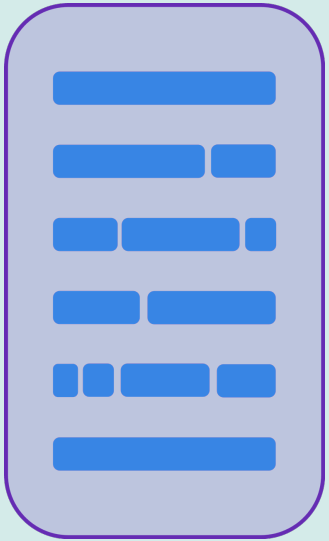


Nur Client (SPA)

Fullstack-Anwendungen (JavaScript)

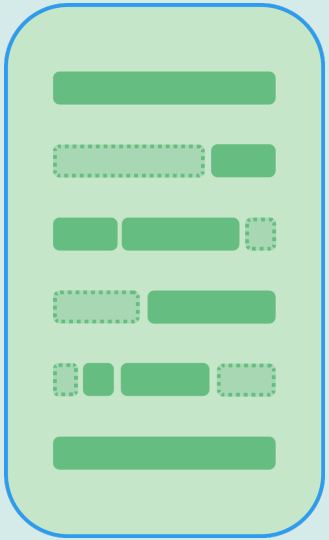


Fullstack-Anwendungen (JavaScript)



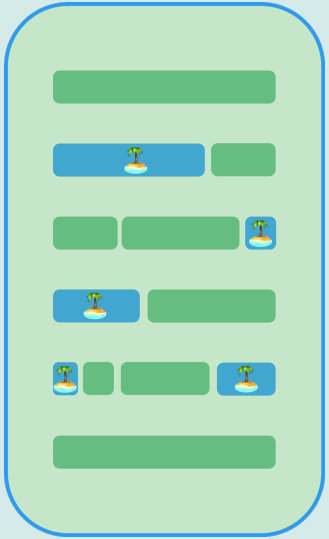
- Ebenfalls vollständig in JavaScript geschrieben

Fullstack-Anwendungen (JavaScript)



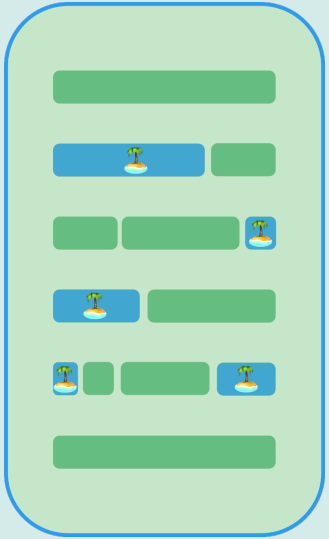
- Grundsätzliche Idee:
  1. **UI-Code** wird serverseitig vorgerendert
  2. **UI-Code** wird zum Browser gesendet und angezeigt

Fullstack-Anwendungen (JavaScript)



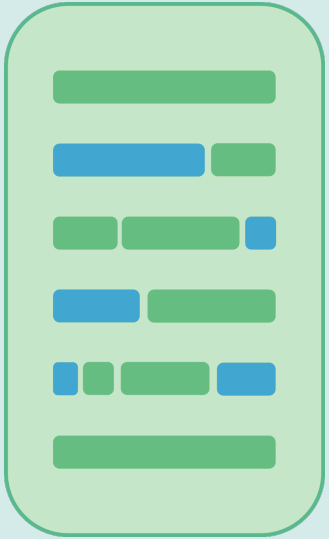
- Grundsätzliche Idee:
  1. **UI-Code** wird serverseitig vorgerendert
  2. **UI-Code** wird zum Browser gesendet und angezeigt
  3. Nur der JavaScript-Code ("Islands") für Interaktionen wird zum Browser geschickt

Fullstack-Anwendungen (JavaScript)



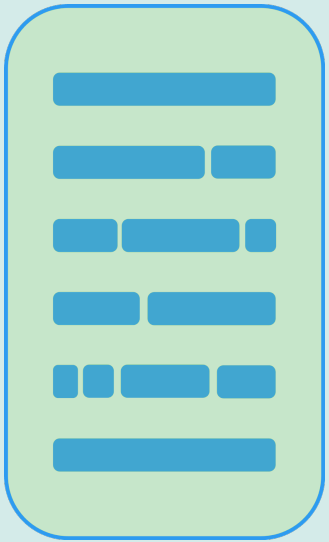
- Anwendung startet schneller:
  1. Browser bekommt UI-Code zur Darstellung
  2. Der notwendige JavaScript-Code wird nachgeladen
  3. Anwendung jetzt interaktiv

Fullstack-Anwendungen (JavaScript)



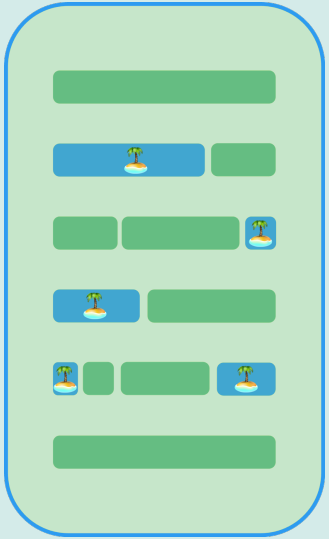
- Wir sind zurück zur JavaScript-Schnipsel-Architektur
  - aber: die Schnipsel werden automatisch vom Framework erzeugt
  - die Schnipsel existieren nur zur Laufzeit

Fullstack-Anwendungen (JavaScript)



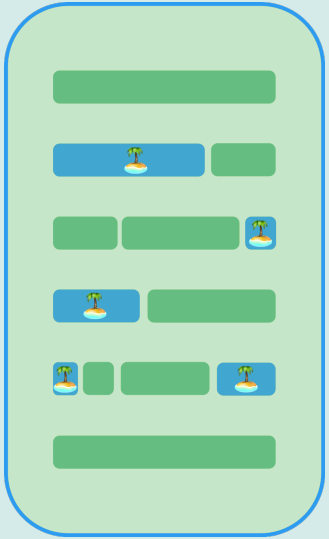
- Wir sind zurück zur JavaScript-Schnipsel-Architektur
  - aber: die Schnipsel werden automatisch vom Framework erzeugt
  - die Schnipsel existieren nur zur Laufzeit
  - In der Entwicklung ist "unser" Code aus "einem Guss"
  - aber weiterhin in JavaScript

Fullstack-Anwendungen (JavaScript)



- Bekannte Vertreter:
  1. Next.js (React)
  2. SvelteKit (Svelte)
  3. Nuxt.js (Vue)
  4. Astro (eigenes Framework + Support für alle SPAs)
  5. Qwik (eigenes Framework)

Fullstack-Anwendungen (JavaScript)



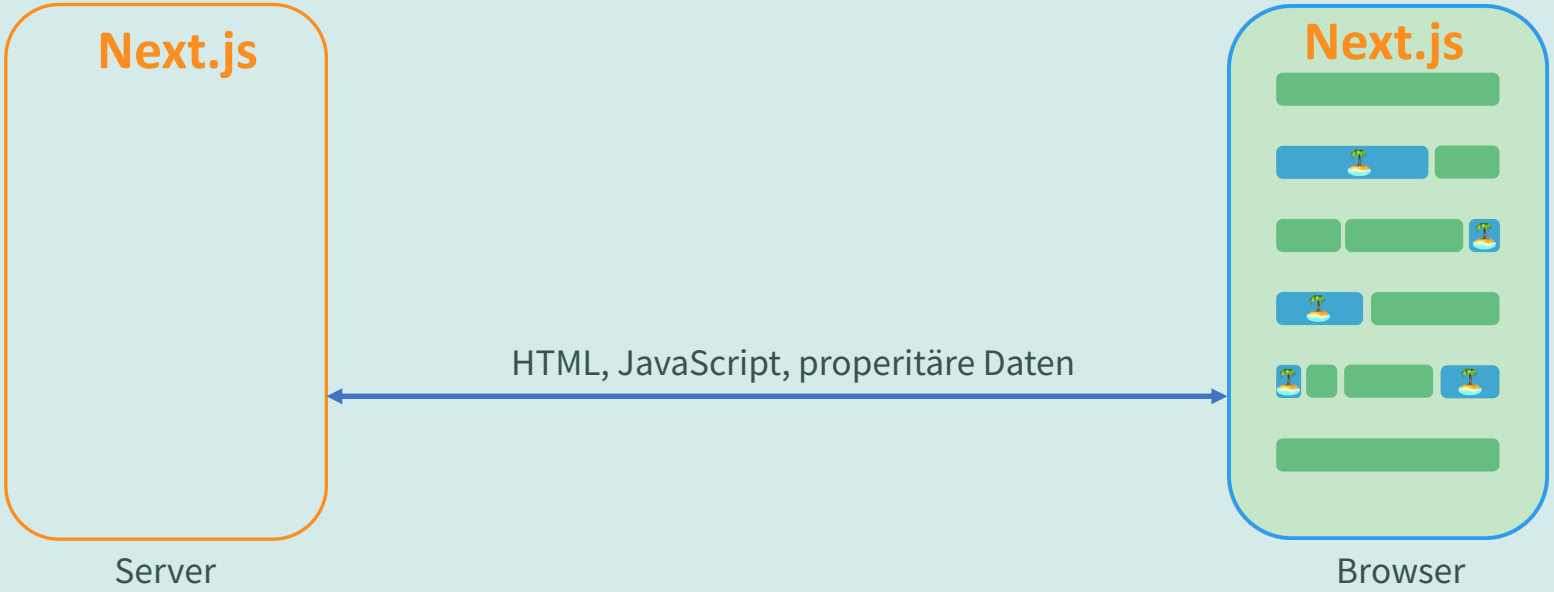
- Bekannte Vertreter:
  1. Next.js (React)
  2. SvelteKit (Svelte)
  3. Nuxt.js (Vue)
  4. Astro (eigenes Framework + Support für alle SPAs)
  5. Qwik (eigenes Framework)
- Funktionalität und Herangehensweise unterschiedlich

Beispiel:
Next.js

EIN PAAR BEISPIELE...

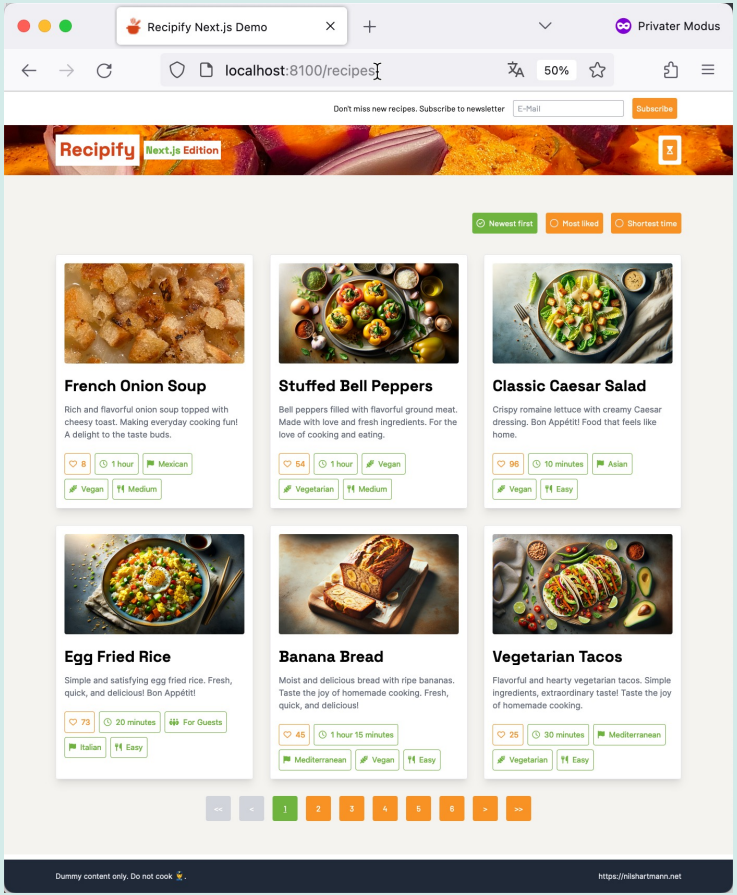
Next.js

- Basiert auf React
- Läuft im Client und im Server



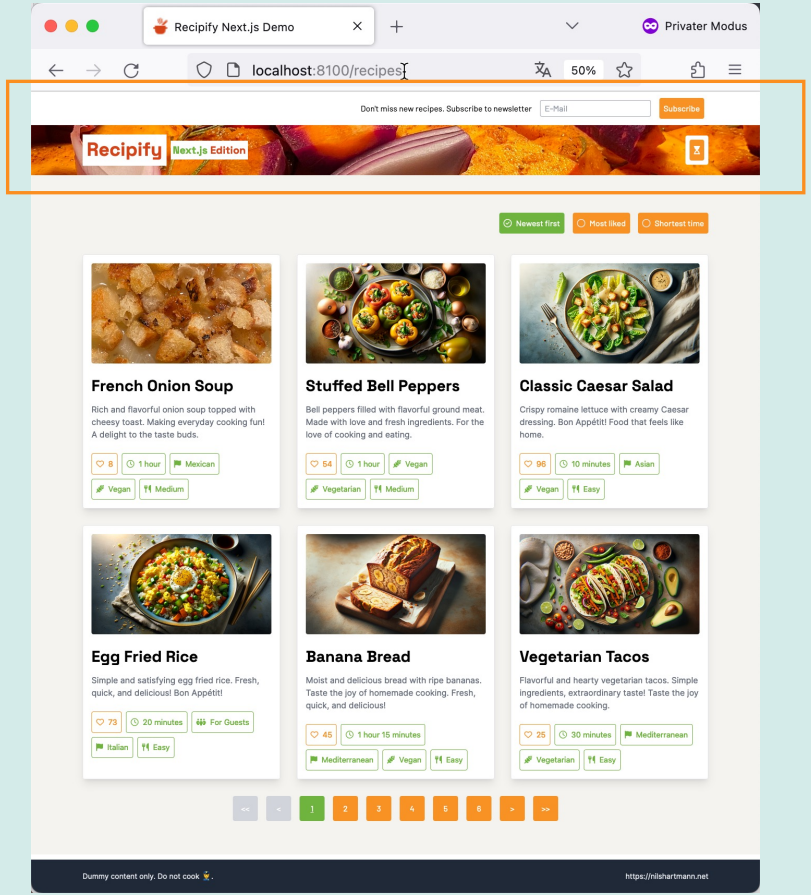
BEISPIEL: INITIALER SEITENAUFRUF

- **Beispiel: Seitenaufrufe**
- Demo: localhost:8100
- React-Komponente statt Controller
  - Aufgaben sind vergleichbar
- Seite kommt als HTML zurück
  - 👨‍🔍 Vergleichbar mit klassisch serverseitig
  - 👨‍🔍 Wo wird die Komponente ausgeführt?
  - 👨‍🔍 Seitenwechsel mit PaginationBar
  - 👨‍🔍 Open in new Tab
  - 👨‍🔍 Was passiert **ohne JavaScript**?



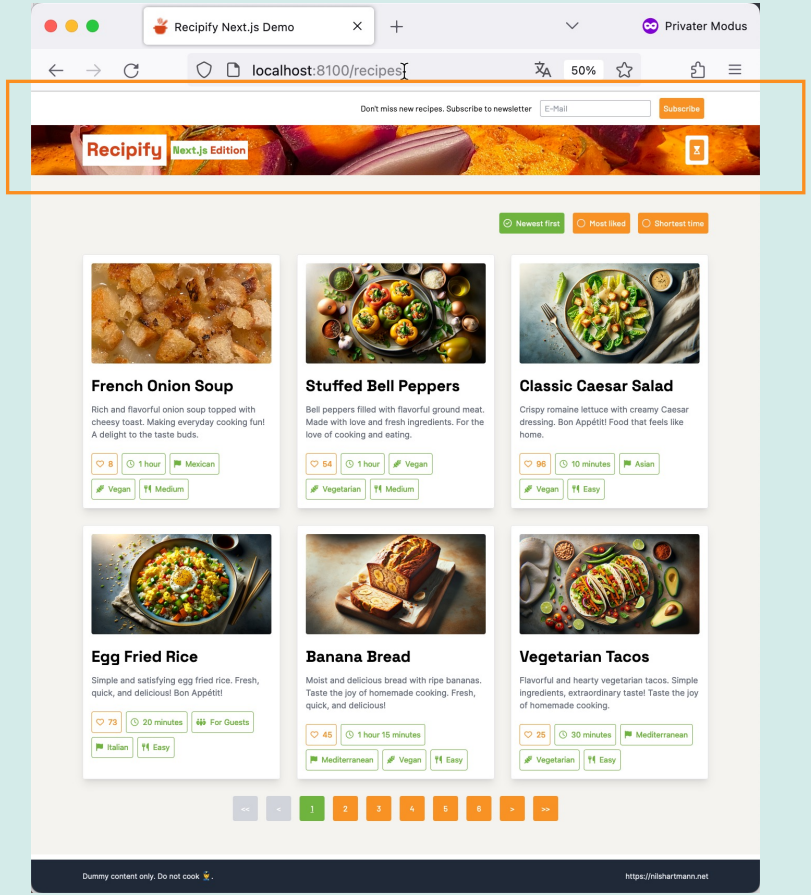
BEISPIEL: INITIALER SEITENAUFRUF

- **Beispiel: Layout**
- Wiederverwendbare Rahmen
  - 🕵️ Timer-Komponente im Header
  - 🕵️ Was passiert beim Seitenwechsel
  - 🕵️ Wie würden wir das "klassisch" machen?



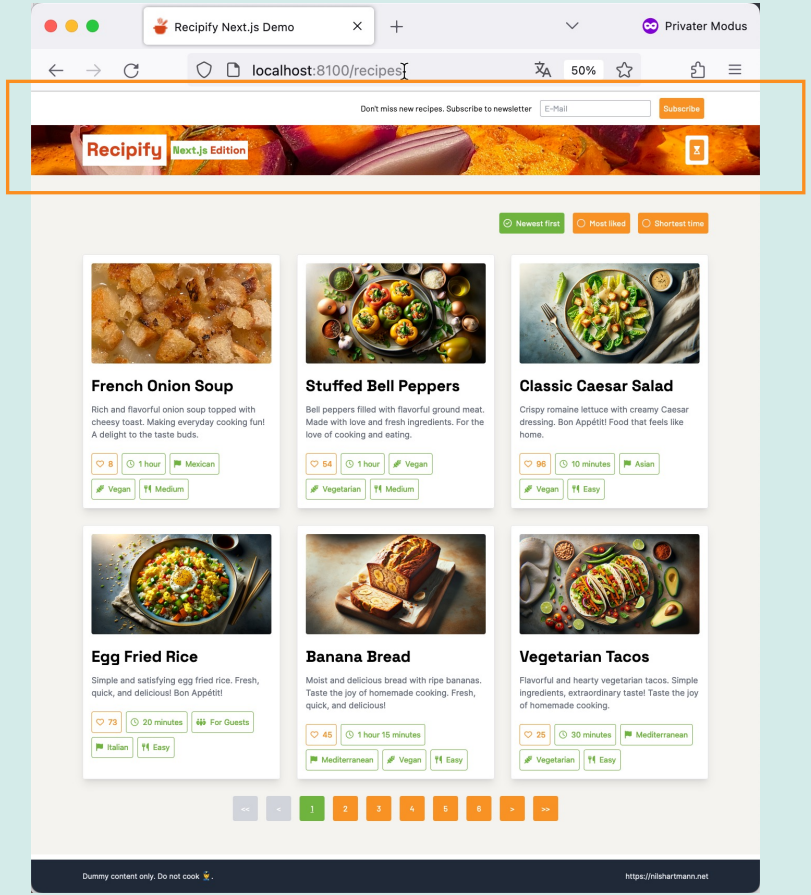
BEISPIEL: INITIALER SEITENAUFRUF

- **Beispiel: Interaktion**
- Zustand bleibt erhalten
  - 🕵️ LikeButton
  - 🕵️ Server-Action (impliziter Endpunkt)
  - 🕵️ Was passiert beim Submit mit dem Timer?
  - 🕵️ Was passiert ohne JS?



BEISPIEL: INITIALER SEITENAUFRUF

- **Beispiel: Interaktion #2**
- Portionszähler
  - 🕵️‍♂️ Reine Client-Information, Berechnung soll nur auf dem Client stattfinden
  - 🕵️‍♂️ Wieder ähnlich wie bei klassischem Ansatz:
    - kein fetch etc.
    - Templates aus einem Guß
    - aber: Logik wird jetzt auch im Client ausgeführt



FAZIT

- **Alles Fullstack oder was?**

FAZIT

- **Alles Fullstack oder was?**
- Nö.

FAZIT

- **Alles Fullstack oder was?**
- Nö.
- Aber interessante Alternative

FAZIT

- **Alles Fullstack oder was?**
- Nö.
- Aber interessante Alternative
- Wie immer: es kommt auf die Anforderung drauf an 🤓

NILS HARTMANN
https://nilshartmann.net

Vielen Dank!

Slides: https://react.schule/fullstack-oder-was

Fragen & Kontakt: nils@nilshartmann.net

Twitter: @nilshartmann