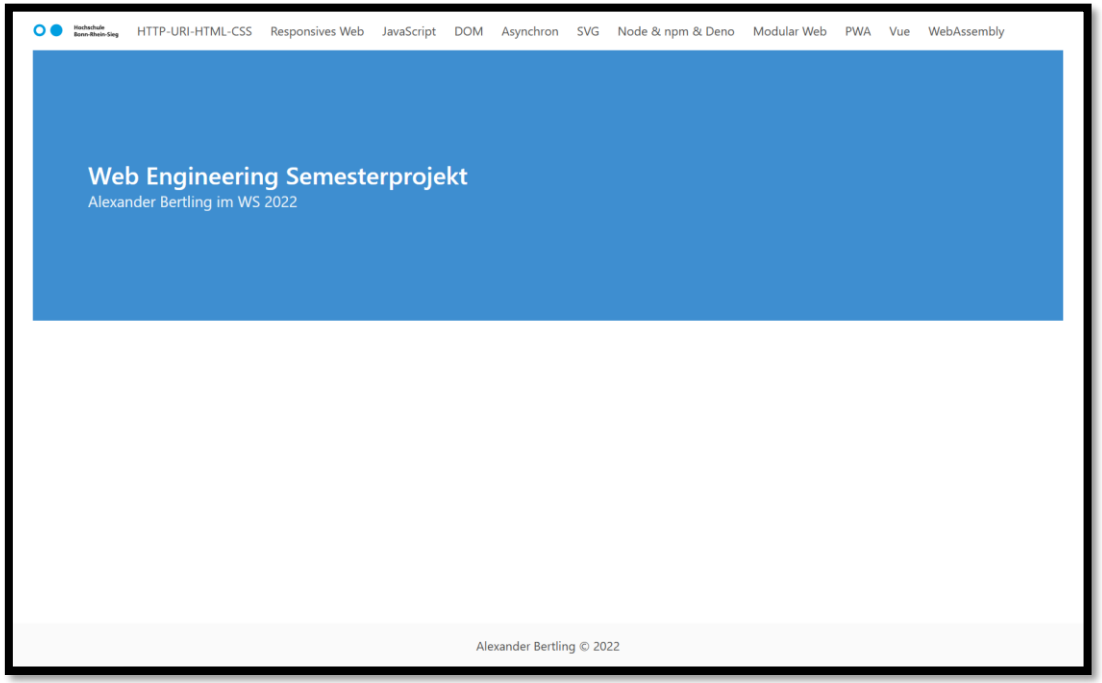
Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Web Engineering Semesterprojekt

Alexander Bertling | 15.03.2022



GitHub https://github.com/AlexBertling/web-engineering-semesterprojekt

netlify https://web-engineering-semesterprojekt.netlify.app/

chromatic https://master--61606f9d9b2bce003a59ac0b.chromatic.com

Ziel des Projekts

Fokus: Implementierung des Solution Navigators

1. **Frameworkless** mit Web Components
2. **Buildless** Development & **effizientes** Deployment
3. **Modernste** Tools (z.B. Import Maps)
4. **Configuration-Driven Application**

1) Frameworkless mit Web Components

- Keine Verwendung von Vue, React, Angular etc.
- **LitElement**: leichtgewichtige Superklasse von Webkomponenten
- **Vaadin Router**: leichtgewichtiger, client-side Router zur Navigation
- **Bulma**: leichtgewichtiges, modernes CSS-Framework

2) Buildless Development & effizientes Deployment

- Keine Buildstep notwendig während Entwicklung
- **web-dev-server**: File-Server mit Zusatzfunktionen
- **Rollup**: Build-Tool für produktive Bereitstellung der Anwendung
- **Storybook**: integrierter Test & Demonstration einzelner Komponenten

App generiert mit [Open Web Components Generator]

Modern Web rollup.js Storybook

3) Modernste Tools

- **ES-Modules**: ausschließlich ES6-import, kein CommonJS
- **Import Maps**: native Auflösung von Bare-Module-Imports im Browser
- **Dynamic Imports**: Laden von Modulen nur bei Bedarf
- **Generierter ServiceWorker**: automatisch durch Workbox-Konfiguration

Zusätzlich soll die Anwendung die **Lighthouse-Kriterien** erfüllen.

4) Configuration-Driven Application

- Inhalte kommen aus **zentraler Konfiguration**
- Trennung von Implementierung und Inhalten
- Wiederverwendung der Anwendung mit anderen Inhalten möglich

Herausforderungen

Organisatorisch / Mangelnde Erfahrung

- (Gleichzeitiges) **Deployment** auf Netlify, Chromatic, Azure
- **Wenig Führung** ohne Framework-Einsatz
  - Auswahl & Anwendung der Tools anspruchsvoll
  - Keine einheitliche Quelle für Informationen & Guides
- **Router-Umsetzung** mit Hervorheben des aktiven Links anspruchsvoll
- Umgang mit **Workbox / Service-Workern** ungewohnt
  - Precaching aller Ressourcen für komplette Offline-Verfügbarkeit

Herausforderungen

Technologisch

- **Import-Maps** noch nicht in allen Browsern verfügbar

  ➔ Auflösung auf Server-Seite mit Import-Maps

- **Dynamic Imports** mit Rollup nicht geschafft

  ➔ Import in appConfig.**js**

- **Shared-Stylesheets** (für Bulma) über Shadow-Doms mehrerer Module hinweg noch nicht möglich

  ➔ Link-Tag in Komponenten

- **HTML-Module** noch nicht verfügbar

  ➔ Einbettung HTML-Ressourcen mit IFrames