

webmap - Automatische Generierung von Websites zur Interaktiven Datenveranschaulichung

DIPLOMARBEIT

verfasst im Rahmen der

Reife- und Diplomprüfung

an der

**Höhere Lehranstalt für Informationstechnologie,
Ausbildungsschwerpunkt Medientechnik**

Eingereicht von:

Jonas Dorfinger
Sebastian scholl

Betreuer:

Dietmar Steiner

Projektpartner:

Christopher Stelzmüller, triply GmbH

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Leonding, April 2022

J. Dorfinger & S. Scholl

Zur Verbesserung der Lesbarkeit wurde in diesem Dokument auf eine geschlechtsneutrale Ausdrucksweise verzichtet. Alle verwendeten Formulierungen richten sich jedoch an alle Geschlechter.

Abstract

Brief summary of our amazing work. In English. This is the only time we have to include a picture within the text. The picture should somehow represent your thesis. This is untypical for scientific work but required by the powers that are. Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.



Zusammenfassung

Zusammenfassung unserer genialen Arbeit. Auf Deutsch. Das ist das einzige Mal, dass eine Grafik in den Textfluss eingebunden wird. Die gewählte Grafik soll irgendwie eure Arbeit repräsentieren. Das ist ungewöhnlich für eine wissenschaftliche Arbeit aber eine Anforderung der Obrigkeit. *Bitte auf keinen Fall mit der Zusammenfassung verwechseln, die den Abschluss der Arbeit bildet!* Suspendisse vel felis.

Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.



Inhaltsverzeichnis

1	Einleitung	1
1.1	Kurzbeschreibung	1
1.2	Aufgabenstellung	1
1.3	Zielsetzung	1
1.4	Geplantes Ergebnis	1
2	Technologien	2
2.1	Frontend Framework Angular	2
2.2	Map Frameworks	3
2.3	Static Site Generators	4
2.4	Deployment Pipeline	4
2.5	Backend	5
2.6	Webserver	5
2.7	Reverse Proxy	5
2.8	Containerization	5
3	Umsetzung	6
3.1	Frontend Implementierung	6
4	Evaluation des Projektverlaufs	7
	Literaturverzeichnis	V
	Abbildungsverzeichnis	VI
	Tabellenverzeichnis	VII
	Quellcodeverzeichnis	VIII
	Anhang	IX

1 Einleitung

1.1 Kurzbeschreibung

Webmap ist ein Triply entwickelt hochwertige Softwarelösungen, die dabei helfen, bestehende Mobilitätssituationen (Verkehrsanalysen, Besucherströme) zu verstehen. Einer der Kernfunktionen liegt darin, Ergebnisse der Analysen einfach und verständlich darzustellen. Dafür soll eine Software geschaffen werden mit welcher interaktive Kunden-Demos einfach und schnell erzeugt werden können.

1.2 Aufgabenstellung

Bei dem Diplomarbeitsprojekt Webmap handelt es sich um die Implementierung eines Generators, welcher interaktive Webseiten automatisch erstellen soll. Für die erwartete Benützung ist es erforderlich, dass die ganze Software vollständig im Web-Browser und somit plattformnabhängig funktioniert.

1.3 Zielsetzung

Mitarbeitern von Triply soll es mit der Datenvisualisierungs-Pipeline möglich sein, schnell und einfach interaktive Websites auf der Basis von komplexen Datensätzen zu erstellen, um diese potentiellen Kunden anschaulich zu präsentieren.

1.4 Geplantes Ergebnis

Entwicklung eines effizienten Generators für interaktive Datenveranschaulichung. Weiters ist eines unserer Ziele die Benützung der Software so leicht und intuitiv wie möglich zu gestalten. Zusätzlich dazu soll der gesamte Prozess vom Starten der Konfiguration bis zur laufenden Website möglichst wenig Zeit in Anspruch nehmen.

2 Technologien

2.1 Frontend Framework Angular

Angular ist eine von Google entwickelte Entwicklungsplattform um Frontends zu implementieren. Als Plattform bietet Angular etliche Services und Features an, dazu gehören nicht nur die der Cross-Platform Support, sondern auch eine built-in Geschwindigkeits- und Performanceoptimierung. Auch für Unternehmen, mit sehr hohen Nutzerzahlen ist Angular ein gern gewähltes Tool. Ein großer Vorteil von Angular ist zudem, dass Angular unter eine Open-Source Lizenz veröffentlicht ist und somit nicht von dem Angular-Google Team sondern auf von der Community betreut und erweitert wird. Angular verfolgt bei der Implementierung das Konzept vom Model-View-Controller Design Pattern.

Angular zu verwenden ist eine Anforderung von der Betreuer Firma triply GmbH, es wird bereits für alle Frontend Projekte verwendet, deshalb auch für dieses.

Model-View-Controller Design Pattern

Dieses Pattern wird oft für die Entwicklung von User Interfaces verwendet, dabei wird die Logik in drei verschiedenen Elementen (Files) aufgeteilt. Das wird gemacht um interne Abbildungen und Referenzen von Daten beziehungsweise Informationen in einer Art und Weise aufzuteilen, wie der diese dem Nutzer angezeigt oder vom Nutzer akzeptiert werden. In Angular ergeben alle drei Files eine gemeinsame Component

Model Das Model File ist das Herzstück der Component, es beinhaltet alle Daten Strukturen und den Aufbau, dabei werden direkt Daten, Logik und Regeln der Component festgelegt und verwaltet.

View Die View beinhaltet alle Details zur Darstellungen der Daten und Informationen. Es kann mehrere Views für die gleiche Information geben, zum Beispiel ein Balkendiagramm für den Manager oder eine Tabelle für die Buchhaltung.

Controller Der Controller Accepts input and converts it to commands for the model or view.

Model

2.1.1 Alternativen für Angular

React

ReactJS ist die größte Konkurrenz von Angular, mit einem vielfachen an Marktanteilen. ReactJS ist eine JavaScript Bibliothek von Facebook. Im Gegensatz zu Angular, wird der DOM weg abstrahiert was ein einfacheres Programmiermodel und bessere Performance ermöglicht. Ein Alleinstellungsmerkmal ist dazu auch noch, dass React Server-Side Rendering unterstützt.

Vue

VueJS ist kaum verbreitet, und hat als größten Nachteil, dass es nur von einer Person regelmäßig erweitert und betreut wird, das kann man anhand der Contributor Activity von dem

Alle 3 Frameworks werden von den den größten IDEs unterstützt.

2.2 Map Frameworks

Um die GeoDaten im Browser darstellen zu können, wird eine Map benötigt. Es gibt dafür drei weit verbreitete Frameworks, MapBox, Leaflet und Google Maps.

2.2.1 Mapbox

2.2.2 Leaflet

2.2.3 Google Maps

2.3 Static Site Generators

2.3.1 Next.js

2.3.2 Jekyll

2.3.3 Scully

2.4 Deployment Pipeline

Nach dem Download des generierten Projekts sollte die Möglichkeit bestehen, kleine Änderungen daran vorzunehmen und es dann schnell und einfach auf einen Server zu deployen. Der Server ist dabei entweder eine Linux Virtual Machine oder eine *Firebase Hosting* Instanz. Dazu gab es mehrere Ansätze: Ist das Projekt ein Git-Repository, kann eine Automatisierungssoftware, wie Jenkins oder GitHub Actions verwendet werden, die bei jedem *Push*-Event das Projekt buildet und auf den Server deployed. Die zweite Möglichkeit ist ein Node.js-Skript, das im Projekt enthalten ist und von der Kommandozeile aus ausgeführt wird.

2.4.1 Jenkins

Jenkins ist ein Open-Source Automatisierungsserver. Das Projekt wird in Java entwickelt und kann mit Hilfe von Plugins an spezifische Anforderungen angepasst werden. Der Server wird vor Allem zur Automatisierung von Aufgaben, wie das Builden, Testen und Deployen von Softwareprojekten genutzt. Jenkins muss selbst gehostet werden. Dazu wird ein offizielles Docker Image im Docker Hub bereitgestellt. Die Konfiguration einer Pipeline wird in einem *Jenkinsfile* vorgenommen, das sich im Git-Repository befindet. Um die Dateien zur Virtual Machine zu senden, wurde das Plugin *Publish Over SSH* verwendet. Für das Deployen zu *Firebase Hosting* steht kein Plugin zur Verfügung. Stattdessen wurde auf die *Firebase CLI* zurückgegriffen.

2.4.2 GitHub Actions

2.5 Backend

2.5.1 JavaScript

2.5.2 Typescript

2.6 Webserver

2.7 Reverse Proxy

2.8 Containerization

2.8.1 Docker

2.8.2 Docker Compose

3 Umsetzung

3.1 Frontend Implementierung

3.1.1 Aufbau und Struktur

3.1.2 Sidebar

3.1.3 Mapview

3.1.4 Global Datasources

3.1.5 Custom Color Schemes

3.1.6 Navigation

4 Evaluation des Projektverlaufs

Aufzählungen:

- Itemize Level 1
 - Itemize Level 2
 - Itemize Level 3 (vermeiden)
- 1. Enumerate Level 1
 - a. Enumerate Level 2
 - i. Enumerate Level 3 (vermeiden)

Desc Level 1

Desc Level 2 (vermeiden)

Desc Level 3 (vermeiden)

Literaturverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

Anhang