



IT-HTL YBBS AN DER DONAU

HÖHERE TECHNISCHE LEHRANSTALT  
FÜR INFORMATIONSTECHNOLOGIE

AUSBILDUNGSSCHWERPUNKT NETZWERKTECHNIK



# DIPLOMARBEIT

## Echtzeit Visualisierung von Energiesystemen

**Ausgeführt im Schuljahr 2020/21 von:**

David Pöchacker 5AHITN

Marcel Entner 5AHITN

Tobias Kronsteiner 5AHITN

Ybbs an der Donau, am TT.MM.JJJJ

**Betreuer/Betreuerin:**

Dipl.-Ing. Johann Burgstaller

**Projektpartner:** Best GmbH

---

Abgabevermerk:

Datum:

Betreuer:

# Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

## Unterschriften der Projektmitglieder

Ybbs an der Donau, am TT.MM.JJJJ

---

Name 1

---

Name 2

---

Name 3

---

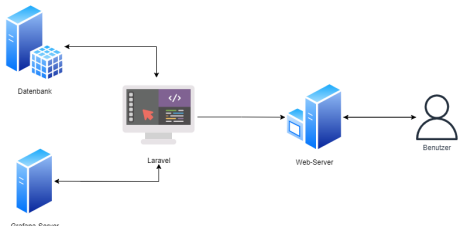
Name 4


# Kurzfassung der Diplomarbeit/Abstract

Hier bitte die ausgefüllten Formulare der Antragstellung in deutscher und englischer Sprache einfügen.  
Seitennummerierung mit B,C,...

	<b>HÖHERE TECHNISCHE LEHRANSTALT YBBS AN DER DONAU</b>	
	Fachrichtung:	<b>Informationstechnologie</b>
	Ausbildungsschwerpunkte:	<b>Netzwerk- und Medientechnik</b>

## DIPLOMARBEIT DOKUMENTATION

Namen der Verfasser/innen	David Pöchacker, Marcel Entner, Tobias Kronsteiner	
Jahrgang Schuljahr	5AHITN 2021/22	
Thema der Diplomarbeit	Echtzeit Visualisierung von Energiesystemen	
Kooperationspartner	Bioenergy and Sustainable Technologies GmbH	
Aufgabenstellung	Das Ziel der Diplomarbeit „Echtzeit Visualisierung von Energiesystemen“ ist, dem Unternehmen Best GmbH eine zentrale Verwaltung von Energiesystemen bereitzustellen. Zusätzlich zur Verwaltung soll es möglich sein, Echtzeitdaten von einer ausgewählten Energietechnologie in Form von Statistiken zu visualisieren.	
Realisierung	Die Weboberfläche wurde mit Laravel umgesetzt. Eingegebene Daten werden in einer Datenbank erfasst und mittels Grafana auf der Weboberfläche visualisiert. Abrufbar ist das Produkt über eine vom Auftraggeber bereitgestellte Domain mit dazugehörigen Webserver.	
Ergebnisse	Mithilfe des Produktes können Energiesysteme sowie Energietechnologien erstellt, bearbeitet und gelöscht werden. Ein rollenbasiertes Benutzersystem regelt den Zugriff auf die Verwaltung der einzelnen Energiesystemen. Der Administrator Benutzer hat als einziger die Möglichkeit neue Benutzer hinzuzufügen oder bestehende zu löschen. Jedem Benutzer ist es möglich, die Grafana Statistiken seiner selbst erstellen Energietechnologien anzeigen zu lassen.	
Architektur		
Teilnahme an Wettbewerben, Auszeichnungen	Mostviertler Schulinovationspreis noch keine	
Möglichkeiten der Einsichtnahme in die Arbeit	Bibliothek SZ-Ybbs	
Approbation Prüfer (Datum / Unterschrift)	Prüfer/Prüferin	Direktor bzw. Abteilungsvorstand

	<b>HÖHERE TECHNISCHE LEHRANSTALT YBBS AN DER DONAU</b> <b>COLLEGE of ENGINEERING</b>	
	Department: <b>Information Technology</b> Educational focus: <b>Network and Media Technology</b>	

## DIPLOMA THESIS

### Documentation

Author(s)		
Form		
Academic year		
Topic		
Co-operation Partners		
Assignment of Tasks		
Realisation		
Results		
Illustrative Graph, Photo (incl. explanation)		
Participation in Competitions Awards		
Accessibility of Diploma Thesis		
Approval (Date / Sign)	Examiner	Head of College / Department

# Danksagung

Wir möchten uns bei allen bedanken, die uns bei der Erstellung dieser Arbeit geholfen haben.

Insbesondere möchten wir uns bei unserem Diplomarbeitsbetreuer Herrn DIp. Johann Burgstaller bedanken. Er unterstützte uns vor allem bei Fachlichen und Inhaltlichen Fragen jeglicher Art und stand uns bei jeder Besprechung mit dem Auftraggeber bei. Zusätzlich zu den Besprechungen war er auch in der Freizeit jederzeit für uns zu Verfügung, um diverse Fragen zu beantworten. Bei unserem Kooperationspartner Best GmbH möchten wir uns ebenso für die Zusammenarbeit bedanken. Mit Herrn Stefan Aigenbauer, Armin Cosic, Michael Zellinger und Jürgen Mitterlehner hatten wir während der gesamten Projektphase ständigen Kontakt um Änderungen sowie Vorschläge des Projektes gemeinsam abzustimmen. Außerdem stellten sie uns diverse Hardware zur Verfügung, wodurch die Umsetzung der Diplomarbeit drastisch erleichtert wurde.

Abschließend möchten wir uns bei unseren Familien, Freunden sowie Klassenkollegen bedanken, die uns während dieser Phase immer zur Seite standen und uns immer wieder aufs neue motivierten.

Vielen Dank!

# Inhaltsverzeichnis

Eidesstattliche Erklärung	A
Kurzfassung der Diplomarbeit/Abstract	B
Danksagung	E
Inhaltsverzeichnis	i
<b>1 Einleitung</b>	<b>2</b>
1.1 Zielsetzung und Aufgabenstellung . . . . .	2
1.1.1 Beschreibung der Diplomarbeit . . . . .	2
1.1.2 Motivation zur Diplomarbeit . . . . .	3
1.1.3 Ziel der Arbeit . . . . .	3
1.1.4 Ergebnis . . . . .	3
1.2 Rollen und individuelle Zielsetzung der Teammitglieder . . . . .	4
1.2.1 David Pöchacker . . . . .	4
1.2.2 Marcel Entner . . . . .	4
1.2.3 Tobias Kronsteiner . . . . .	5
<b>2 Grundlagen und Methoden</b>	<b>6</b>
2.1 Analyse des vorhandenen Systems . . . . .	6
2.1.1 Begriffe . . . . .	7
2.2 Anforderungen an das Produkt . . . . .	8
2.2.1 Schutz von vertraulichen Informationen . . . . .	9
2.2.2 Statistische Auswertung . . . . .	9
2.3 Architektur des Zielsystems . . . . .	9
2.3.1 Endgeräte . . . . .	10

2.3.2	Serverseitig . . . . .	10
2.3.3	Clientseitige Interaktion des Benutzers . . . . .	10
2.3.4	Framework . . . . .	11
2.3.5	Front-End Templates . . . . .	16
2.3.6	Verbindung der Datenbank mit Laravel . . . . .	18
2.4	Visuelle Darstellung der Energiesysteme und Energietechnologien . . . . .	20
2.4.1	Kartendienste . . . . .	20
2.4.2	Geoinformationssystem . . . . .	22
2.4.3	CSS-System . . . . .	22
2.4.4	Auswahl des Anbieters . . . . .	22
2.5	Berechtigungssystem Benutzer . . . . .	23
2.5.1	Benutzerrollen . . . . .	23
2.5.2	Berechtigungen in Laravel . . . . .	24
2.6	Ui/Ux Design . . . . .	24
2.6.1	Wireframe . . . . .	25
2.6.2	Persona . . . . .	25
2.7	Template Layout . . . . .	25
2.7.1	Platzhalter Yield . . . . .	26
2.7.2	Sections . . . . .	26
2.7.3	Einbindung der definierten Sections . . . . .	26
2.8	Laravel Befehle . . . . .	26
2.8.1	Migration Befehle . . . . .	26
2.8.2	Seeder und Factory Befehle . . . . .	27
2.8.3	Model und Controller Befehle . . . . .	27
2.8.4	Starten des Laravel Develop Servers . . . . .	27
2.8.5	Befehle nach dem Git Pull . . . . .	27
<b>3</b>	<b>Ergebnisdokumentation</b>	<b>28</b>
3.1	Laravel . . . . .	28
3.1.1	Installation . . . . .	28
3.1.2	Bootstrap Einbindung . . . . .	28
3.1.3	Grafana Einbindung . . . . .	28
3.1.4	MVC . . . . .	28



3.2	Datenbankanbindung in Laravel . . . . .	29
3.2.1	Datenbank Anmeldeinformationen . . . . .	29
3.2.2	Mail Server Konfigurationen . . . . .	29
3.2.3	Migrations . . . . .	30
3.3	Routen in Laravel . . . . .	31
3.3.1	Resource Routen . . . . .	31
3.3.2	GET Routen . . . . .	31
3.3.3	Auth Routen . . . . .	31
3.4	Datenbankdesign . . . . .	31
3.4.1	Erstellen eines neuen Schemas . . . . .	31
3.5	Corporate Design . . . . .	31
3.5.1	Vorschläge . . . . .	31
3.5.2	Änderungsvorschläge . . . . .	32
3.5.3	Finales Design . . . . .	32
3.5.4	Definierte Farben . . . . .	32
3.5.5	Überschriften . . . . .	32
3.5.6	Interaktionsfarben . . . . .	32
3.5.7	Schriftarten . . . . .	32
3.5.8	Schriftgrade . . . . .	32
3.5.9	Logo . . . . .	32
3.5.10	Verwendete Icons und deren Bedeutungen . . . . .	32
3.5.11	Map Icons . . . . .	32
3.5.12	Icons in Formularen . . . . .	33
3.5.13	Icons im DataTable . . . . .	33
3.5.14	Buttons . . . . .	33
3.5.15	Tabelle mit generellen Informationen über einzelne HTML Elemente . . . . .	33
3.5.16	Datenformate . . . . .	33
3.6	Weboberfläche . . . . .	33
3.6.1	Backend . . . . .	33
3.6.2	Front-End . . . . .	37
3.6.3	Login . . . . .	38
3.6.4	Registrierung . . . . .	38
3.6.5	Kartendienst Funktionalitäten . . . . .	38

3.6.6	Anzeige von Energiesystemen und Energietechnologien auf der Karte . . . . .	39
3.6.7	Layoutvorlage der Website . . . . .	39
3.7	DataTable . . . . .	40
3.7.1	Individueller DataTable . . . . .	40
3.7.2	Sortierfunktion . . . . .	40
3.7.3	Suchfunktion . . . . .	41
3.7.4	Seitenanzahl . . . . .	41
3.7.5	Icons . . . . .	41
3.7.6	MoveToMarker . . . . .	41
3.8	Galerie Funktionen . . . . .	42
3.8.1	Auswahl eines Energiesystems . . . . .	42
3.8.2	Energietechnologien des Energiesystems anzeigen . . . . .	42
3.9	Grafana . . . . .	42
3.9.1	Automatisches Erstellen der Dashboards . . . . .	42
3.9.2	Automatisches Erstellen der Panels . . . . .	42
3.9.3	Energietechnologien Statistiken anzeigen . . . . .	42
3.10	Einbindung von Google Maps . . . . .	43
3.10.1	Google Cloud . . . . .	43
3.10.2	Google Cloud Platform Account erstellen . . . . .	43
3.10.3	Apis aktivieren und einbinden . . . . .	43
3.10.4	Individuelle Map erstellen und einbinden . . . . .	43
<b>4</b>	<b>Resümee und Ausblick</b>	<b>44</b>
<b>5</b>	<b>Quellen und Literatur</b>	<b>45</b>
	<b>Abbildungsverzeichnis</b>	<b>46</b>
<b>6</b>	<b>Tabellenverzeichnis</b>	<b>47</b>
<b>7</b>	<b>Codeverzeichnis</b>	<b>48</b>
<b>8</b>	<b>Begleitprotokoll gem. § 9 Abs. 2 PrO-BHS</b>	<b>49</b>
8.1	Begleitprotokoll David Pöchacker . . . . .	49
8.2	Begleitprotokoll Marcel Entner . . . . .	49
8.3	Begleitprotokoll Tobias Kronsteiner . . . . .	49

<b>9</b>	<b>Anhang</b>	<b>50</b>
9.1	Verfasser der Kapitel . . . . .	50
9.1.1	David Pöchacker . . . . .	50
9.1.2	Marcel Entner . . . . .	50
9.1.3	Tobias Kronsteiner . . . . .	50
9.2	Verwendete Software . . . . .	50
9.2.1	Visual Studio Code . . . . .	50
9.2.2	Apache WebServer . . . . .	51
9.2.3	Composer . . . . .	51
9.2.4	Windows Eingabeaufforderung (CMD) . . . . .	51
9.2.5	Github VCS und Github Desktop GUI . . . . .	51
9.2.6	phpMyAdmin . . . . .	51
9.2.7	Adobe XD . . . . .	51
9.2.8	Adobe Photoshop . . . . .	51
9.3	Projektplanung . . . . .	51
9.3.1	Projektkommunikation . . . . .	52
9.3.2	Projektstrukturplan . . . . .	52
9.3.3	Verantwortungsmatrix und Aufwandsschätzung . . . . .	52
9.3.4	Meilensteinplan . . . . .	52
9.3.5	Terminplan . . . . .	52
9.4	Inhalt von GitHub . . . . .	52

page1

# Kapitel 1

## Einleitung

In diesem ersten Kapitel wird auf die Aufgabenstellung und die Ziele dieser Diplomarbeit eingegangen.

### 1.1 Zielsetzung und Aufgabenstellung

Eine allgemeine Beschreibung, die Veranlassung, das Ziel und das Ergebnis dieser Diplomarbeit ist der Inhalt der folgenden Abschnitten.

#### 1.1.1 Beschreibung der Diplomarbeit

Im Rahmen der Diplomarbeit „Echtzeit-Visualisierung von Energiesystemen“ soll ein Produkt zur Verwaltung von Energiesystemen erstellt werden. Dabei soll auf die bereits bestehenden Komponenten wie Webserver, Datenbank sowie Grafana Server aufgebaut werden. Der Webserver präsentiert die Verwaltungsanwendung, welche öffentlich im Internet für jeden zugänglich ist. Auf dieser Anwendung werden die vorhandenen Energiesysteme sowie Energietechnologien dargestellt. Durch den Kooperationspartner „Best GmbH“ ist es möglich, gemeinsam wichtige Entscheidungen wie die Auswahl der anzuzeigenden Daten zu treffen, da gewisse Informationen über ein Energiesystem <sup>1</sup> sowie eine Energietechnologie <sup>2</sup> aus Datenschutzgründen nicht angezeigt werden sollen. Der Anwender soll die Möglichkeit haben, sein eigenes Energiesystem mit den dazugehörigen Energietechnologien zu erstellen und anschließend deren Echtzeitdaten <sup>3</sup> zu veranschaulichen.

---

<sup>1</sup>Eine genaue Begriffserklärung befindet sich in 2.1.1.3

<sup>2</sup>Eine genaue Begriffserklärung befindet sich in 2.1.1.2

<sup>3</sup>Eine genaue Begriffserklärung befindet sich in 2.1.1.1

### 1.1.2 Motivation zur Diplomarbeit

Die Diplomarbeit ist dadurch entstanden, dass Mitarbeiter der Best GmbH angefragt haben, ob das SZ-Ybbs mit ihnen eine Diplomarbeit durchführen möchte. Das Ziel des Projektes ist es, eine zentrale Verwaltung von Energiesystemen zu ermöglichen, da eine solche zentrale Verwaltung bei dem Auftraggeber noch nicht vorhanden ist. Da es sich dabei um ein Thema handelt, das im Lehrplan der IT-HTL Ybbs/Donau breiten Raum einnimmt, war die Kooperation mit der Schule für dieses Projekt eine gute Wahl. Daraufhin wurde das Projektteam auf diesen Vorschlag für eine Diplomarbeit aufmerksam, und entschloss sich kurze Zeit danach, dieses Produkt umzusetzen. Es sind zwar bereits eine Vielzahl solcher Produkte vorhanden, jedoch noch keine ideale Lösung, die den Anforderungen des Auftraggebers entspricht. Für das Projektteam war das ein Ansporn, genau dieses Produkt zu entwickeln und es soll mit wenig Aufwand und Vorwissen für einen Benutzer möglich sein, das Produkt zu bedienen.

### 1.1.3 Ziel der Arbeit

Das Ziel dieses Projektes ist eine Weboberfläche, welche es ermöglicht, Energiesysteme einfach und intuitiv zentral zu verwalten. Die Energiesysteme sollen auf einer Karte angezeigt werden, um einen Überblick aller vorhandenen Energiesysteme zu ermöglichen. Man soll Energiesysteme erstellen können, und es soll möglich sein, einem Energiesystem mehrere Energietechnologien hinzuzufügen. Bei den Energietechnologien soll es möglich sein, deren Echtzeitdaten in Form von Statistiken anzeigen zu lassen. Ebenso soll zu einer Energietechnologie ein Foto hinzugefügt werden können, welches in einer Bildergalerie dargestellt werden soll. Über die gesamte Anwendung ist zudem ein Design Handbuch sowie ein Benutzerhandbuch zu verfassen. Zusätzlich ist ein neues Datenbankschema zu entwickeln, da das bereits vorhandene Datenbankschema nicht dem heutigen Stand der Technik entspricht. Eine rollenbasierte Benutzerverwaltung soll zur Steuerung des Zugriffs auf die Website implementiert werden. Somit ist jeder Benutzer auf der Website nur dazu berechtigt, seine eigenen Energiesysteme zu verwalten, mit Ausnahme des Administrator-Benutzers.

### 1.1.4 Ergebnis

Das Ergebnis des Produktes ist eine lauffähige Weboberfläche, die das Verwalten von Energiesystemen mit dazugehörigen Energietechnologien ermöglicht. Die Verwaltung ist mit einem rollenbasierten Benutzerauthentifizierungssystem begrenzt, um unbefugten Besuchern die Verwaltung zu verwehren. Dafür hat der Administrator eine eigene Registrierungsseite, um neue Benutzer anzulegen und bestehende zu löschen. Die Funktionen „Energiesystem“ sowie „Energietechnologie Erstellen, Bearbeiten, Löschen“ und die dazugehörigen Statistiken anzeigen zu lassen erfüllen alle ihre Funktionalitäten. Die vorgesehene Bildergalerie wurde erfolgreich umgesetzt, und zeigt zu jeder Energietechnologie das dazugehörige Bild an. Unter folgendem Link ist das Ergebnis der Diplomarbeit ersichtlich. <https://visu.microgrid-lab.eu/>

## 1.2 Rollen und individuelle Zielsetzung der Teammitglieder

In Abbildung 1.1 sind alle Mitglieder, die bei dieser Diplomarbeit eine wichtige Rolle spielen, ersichtlich.

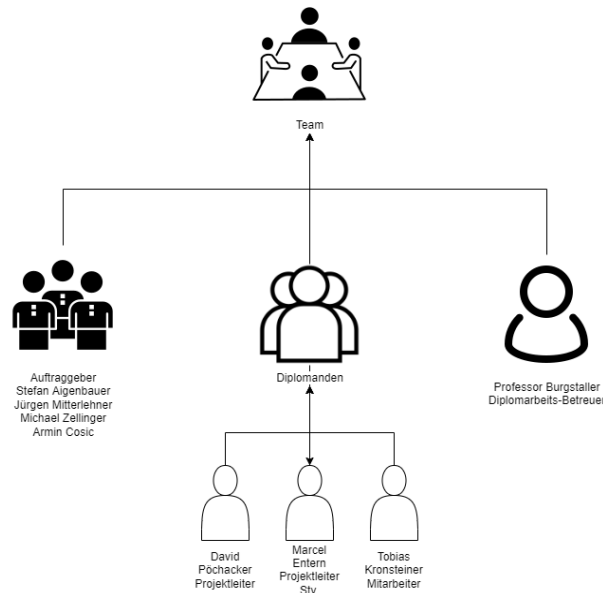


Abbildung 1.1: Team

### 1.2.1 David Pöchacker

Die implementierte Karte muss so konfiguriert werden, dass das Hinzufügen von Energiesystemen sowie Energietechnologien für den Benutzer möglich ist. Zusätzlich zum Erstellen eines Energiesystems oder einer Energietechnologie sollen die Funktionen, diese zu bearbeiten und zu löschen, ebenfalls gegeben sein. Um die Verwaltung der Energiesysteme unter Kontrolle zu haben und gegen unerwünschte Zugriffe zu schützen, ist ein rollenbasiertes Benutzersystem notwendig. Um diese Anforderungen zu erfüllen, müssen folgende Aufgaben umgesetzt werden:

- Kartendienst Funktionen implementieren
- Verwaltungsfunktionen der Energiesysteme sowie Energietechnologien erstellen
- Benutzerverwaltungssystem einrichten

### 1.2.2 Marcel Entner

Hauptaufgabe ist die generelle Konzeption und Umsetzung des Front-End Designs. Er ist dafür verantwortlich, Front-End Vorlagen zu entwerfen und diese mit dem Auftraggeber abzustimmen. Das Teammitglied ist auch für die Umsetzung des Vorschlages auf der Website verantwortlich und verfasst dazu ein Designhandbuch. Eine weitere Aufgabe ist es, eine Bildergalerie, die zu jeder Energietechnologie ein dazugehöriges Bild anzeigt, zu entwerfen. Das Teammitglied war auch dafür verantwortlich, dass eine Impressums- und eine Datenschutzseite vorhanden ist.

### 1.2.3 Tobias Kronsteiner

Das Teammitglied setzt sich mit der Analyse des vorhandenen Systems sowie der Datenbank auseinander. Weiters befasst es sich mit der Analyse, Konzeption und Implementierung von Visualisierungsmöglichkeiten der anwendungsspezifischen Statistiken. Der Zugriff sowie der Datenverkehr zwischen dem Produkt und der neu konzipierten Datenbank gehört ebenfalls zum Aufgabenbereich. Zum größten Teil befasst sich das Teammitglied mit dem Visualisierungs Tool Grafana und dessen HTTP Api<sup>4</sup>. Eine besondere Herausforderung stellt hier die Implementierung eines Algorithmus zum erweitern des Datenmodells eines Dashboards <sup>5</sup>, sowie das dynamische Teilen von Panels <sup>6</sup>, auf der Benutzeroberfläche des Produktes dar.

---

<sup>4</sup>Anwendungsspezifische Programmierbar Schnittstelle eines Programmes

<sup>5</sup>Genauere Erklärung erfolgt im Abschnitt xx

<sup>6</sup>Genauere Erklärung erfolgt im Abschnitt xx



## Kapitel 2

# Grundlagen und Methoden

In diesem Abschnitt werden die verwendeten Grundlagen und Methoden, die für die Umsetzung dieses Diplomarbeitsprojekts notwendig sind, dargestellt. Falls bei der Umsetzung mehrere Möglichkeiten zur Wahl standen, werden die einzelnen Möglichkeiten miteinander verglichen und nach einem Vergleich wird die besser geeignete Variante ausgewählt.

### 2.1 Analyse des vorhandenen Systems

Das bestehende System des Auftraggebers umfasst die Komponenten Grafana Server, Webserver und eine Datenbank mitsamt den notwendigen Algorithmen, um diese mit Echtzeitdaten zu befüllen. Bei der vorgegebenen Datenbank handelt es sich um einen MariaDB SQL-Server in der Version 10.1.48. Bei MySQL handelt es sich um eine quelloffene Implementierung des SQL Standards in der Version 5.0.12. In der nachfolgenden Abbildung ist das vorhandene System des Auftraggebers ersichtlich.



Abbildung 2.1: vorhandeneSystemAuftraggeber

Der Grafana Server wird übernommen, um Dashboards für die Energiesysteme und Panels für die Energietechnologien zu erstellen. Dabei können mehrere Statistiken, auch Panels genannt, zu einem Dashboard zusammengefasst werden, um somit eine Ordnerstruktur am Grafana Server zu erlangen. Der Webserver wird als Produktivserver benützt, um die fertige Website zu präsentieren. Die vorhandene Datenbank wird aufgrund deren Schemas nicht übernommen, stattdessen wird ein neues besseres Datenbankschema entwickelt. Die dafür notwendigen Algorithmen, um die neue Datenbank mit Echtzeitdaten zu befüllen, werden vom Auftraggeber entwickelt.

### 2.1.1 Begriffe

In diesem Abschnitt werden notwendige Begriffe, die in dieser Diplomarbeit eine wichtige Rolle spielen und zu Missverständnissen führen könnten, erklärt.

#### 2.1.1.1 Echtzeitdaten

Der Begriff Echtzeitdaten ist definiert durch die regelmäßige und in gleichen Zeitabständen erfolgende Erfassung sowie Verarbeitung von Daten einer Energietechnologie. Der Zeitabstand der Datenerfassung bei den Energietechnologien beträgt 30 Sekunden. Aufgrund des vorgegebenen Zeitabstandes werden die Echtzeitdaten als weiche Echtzeitanforderung definiert, da das Produkt alle einkommenden Daten schnellstmöglich mit einem konstanten Zeitabstand von 30 Sekunden bearbeitet. Ein Überschreiten dieser Zeitgrenze wird nicht als Versagen definiert, solange sich die Zeit noch in einem akzeptablen Toleranzbereich von wenigen Sekunden befindet. Eine Unterschreitung der Zeitangabe ist sehr selten möglich.

#### 2.1.1.2 Energietechnologie

Als Energietechnologie wird ein Stromerzeuger, Stromverbraucher oder ein Energiespeicher bezeichnet. Stromerzeuger sind PV-Anlagen oder Windkraftanlagen. Stromverbraucher sind E-Ladestationen oder Hausanschlusszähler. Batteriespeicher oder Wärmespeicher sind Speicher-Energietechnologien. Die von diesen Technologien erfassten Echtzeitdaten werden in einer Datenbank gespeichert und anschließend grafisch in Form von Statistiken visualisiert.

Dabei werden folgende Echtzeitdaten für die Statistiken verwendet:

- Erzeuger/Verbraucher - Leistung [kW]
- Erzeuger/Verbraucher - Energie [kW/h]
- Speicher – Kapazität/Temperatur [kW/h]/[°]

Die folgende Tabelle bietet eine Übersicht aller vorhandenen Energietechnologien. In der ersten Spalte der Tabelle befinden sich reine Erzeuger-Energietechnologien[PM1]. Verbraucher-Energietechnologien sind in der zweiten Spalte ersichtlich. Speicher-Energietechnologien in der dritten Spalte und in der vierten und zugleich auch letzten Spalte befinden sich Energietechnologien, die sowohl als Verbraucher-Energietechnologie und als Erzeuger-Energietechnologie definiert sind.

Erzeuger	Verbraucher	Speicher	Verbraucher & Erzeuger
PV-Anlage	Wasserstoff- Elektrolyse	Batteriespeicher	Biomasseheizkraftwerk
Stromnetzbezug	E-Ladestation	Wasserstoff-speicher	Biomasseheizwerk
Wasserstoff Brennstoffzelle	Hausanschlusszähler	Wärmespeicher	Biomassekessel
Windkraftanlage	Gebäude- Wärmebedarfszähler	Kältespeicher	Kompressionskältemaschine
Wärmenetzbezug	Gebäude- Kältebedarfszähler		Ab- oder Adsorptionskältemaschine
Solarthermieranlage			
Wärmepumpe			

Tabelle 2.1: Energietechnologien

### 2.1.1.3 Energiesystem

Der Begriff Energiesystem fasst mehrere Energietechnologien in einem bestimmten Gebiet, die genau einem Energiesystem zugeordnet sind, logisch zusammen. Ein Energiesystem kann somit eine Gemeinde, ein Gebäude oder ein einziger Haushalt mit mehreren Energietechnologien sein.

### 2.1.1.4 Front-End

Der Begriff Front-End beschreibt die Weboberfläche auf welcher der Benutzer verschiedene Interaktionen durchführen kann. Dabei umfasst dieser Begriff alle Unterseiten der Weboberfläche welche von einem Benutzer verwendet werden können.

### 2.1.1.5 Back-End

Mit dem Begriff Back-End ist das Laravel-Projekt, welches für die Ereignisse der Benutzer-Interaktionen zuständig ist, gemeint. Zusätzlich sind in diesem Begriff die Zugriffe auf die Datenbank sowie auf den Grafana Server mit eingebunden, welche durch das Laravel-Projekt durchgeführt werden.

## 2.2 Anforderungen an das Produkt

Die Datenbank des Auftraggebers ist die größte Schwachstelle des vorhandenen Systems. Aufgrund dessen wird ein neues Datenbankschema entwickelt. Neben der Entwicklung eines neuen Datenbankschemas sind folgende Anforderungen an das Produkt gegeben:

- Übersichtliche Darstellung von Energiesystemen und Energietechnologien auf einem Kartendienst
- Grafana-Statistiken mit Echtzeitdaten der Energietechnologien anzeigen
- Bildergalerie um Energietechnologien eines ausgewählten Energiesystems zu präsentieren
- Managementfunktion mit Hilfe von verschiedenen Benutzern für unterschiedliche Berechtigungen

### 2.2.1 Schutz von vertraulichen Informationen

Sämtliche Daten wie Adressen, Standorte (in Form von Koordinaten) oder die Namen der Ersteller von Energiesystemen oder Energietechnologien sind vertrauliche Informationen und sollten somit nicht für jeden einsehbar sein. Darum sind diese Daten nur in der Datenbank gespeichert, welche nur für Benutzer mit entsprechender Berechtigung zugänglich ist. Die Daten, die in den Grafana-Statistiken dargestellt werden, sind ebenso vertrauliche Daten. Die Informationen, die man einer solchen Statistik entnehmen kann, könnten in die falschen Hände gelangen und zu unerwünschten Tätigkeiten führen. Um dieses zu verhindern, sieht jeder angemeldete Benutzer nur seine eigenen Statistiken und keine anderen. Ein nicht angemeldeter Besucher sieht keine Statistiken.

### 2.2.2 Statistische Auswertung

Text

## 2.3 Architektur des Zielsystems

Die Abbildung 2.2 repräsentiert die Architektur des Zielsystems. Der Benutzer greift über die Weboberfläche auf den Webserver zu, wo sich das Laravel-Projekt befindet. Auf dieser Website kann der Benutzer verschiedene Interaktionen durchführen. Dabei ist die Weboberfläche ständig in Verbindung mit der Datenbank sowie dem Grafana Server. Sobald der Benutzer auf die Energiesysteme-Seite der Weboberfläche wechselt, wird eine Datenbankabfrage aller vorhandenen Energiesysteme durchgeführt. Zusätzlich dazu wird mittels API-Abfrage auf den Grafana Server zugegriffen, um Statistiken einer Energietechnologie anzuzeigen, falls dies vom Benutzer angefordert wird.

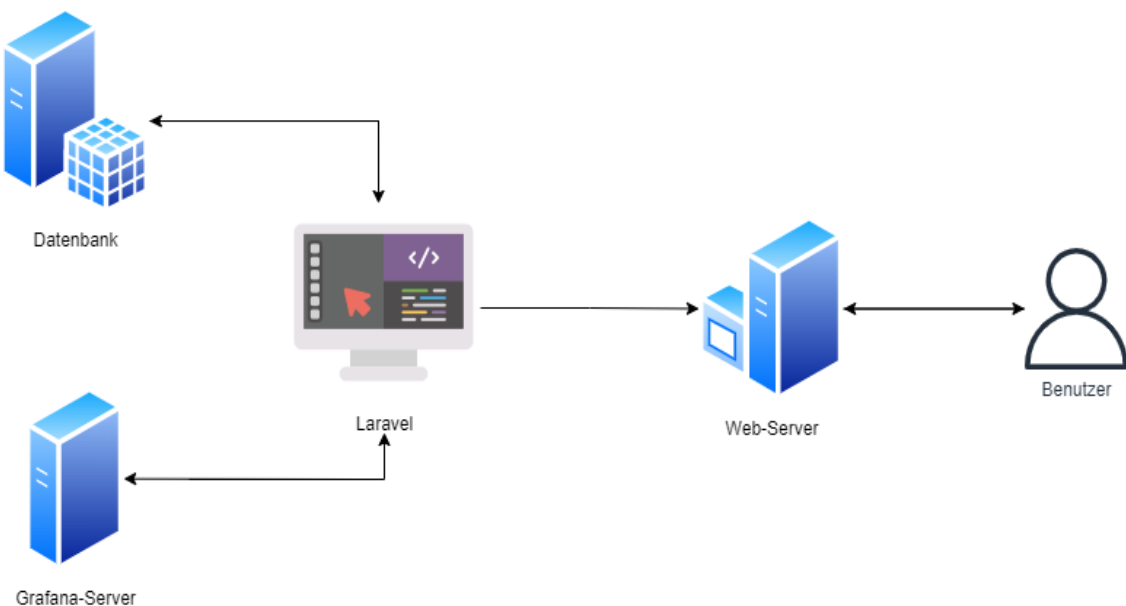


Abbildung 2.2: Architektur

### 2.3.1 Endgeräte

Das Produkt wurde als Browser Anwendung entwickelt, um überall lauffähig zu sein. Es ist in Chrome, Firefox und Edge Browser getestet worden und ist dort voll funktionsfähig. Die Oberfläche wurde für einen 13 bis 27 Zoll Monitor entwickelt und funktioniert dort in jeder Zoomstufe einwandfrei. Klares Ziel war es, das Produkt nicht für den Mobiltelefongebrauch zu programmieren. Bewusst wurde sich gegen eine native Anwendungsentwicklung entschieden, um so die Verwendbarkeit auf allen Plattformen zu garantieren. Darüber hinaus würde eine parallele Anwendungsentwicklung auf den drei gängigen Betriebssystemen (Windows, MacOS, Linux) den zeitlichen Rahmen der Diplomarbeit sprengen. Getestet wurde das Produkt in den Versionen:

- Chrom 99.0.4844.51
- Firefox 98.0
- Edge 99.0

### 2.3.2 Serverseitig

Text

### 2.3.3 Clientseitige Interaktion des Benutzers

Bei dem Punkt Clientseitig befindet man sich in der Architektur des Zielsystems bei dem Element JavaScript und mit den damit verbundenen Interaktionen des Benutzers. Map-Interaktionen sind Interaktionen mit einer Karte, dazu gehört das Erstellen, Bearbeiten und Löschen von Energiesystemen sowie Energietechnologien. Eine weitere Interaktion wäre die Verwendung des Adresssuchfeldes, welche mit einem Klick auf den Button „Suche“ oder durch Drücken der Enter-Taste durchgeführt wird. Tabellen-Interaktionen sind eine zusätzliche clientseitige Aktion, da die vom DataTable<sup>1</sup> bereitgestellten Funktionen wie die Suchfunktion, Sortierfunktion oder Seitennummerierung allesamt clientseitig stattfinden und somit keinen Einfluss auf das Back-End haben. Folgende Abbildung zeigt den Aufbau der einzelnen Elemente, die zusammenarbeiten, um die Website zu erstellen.

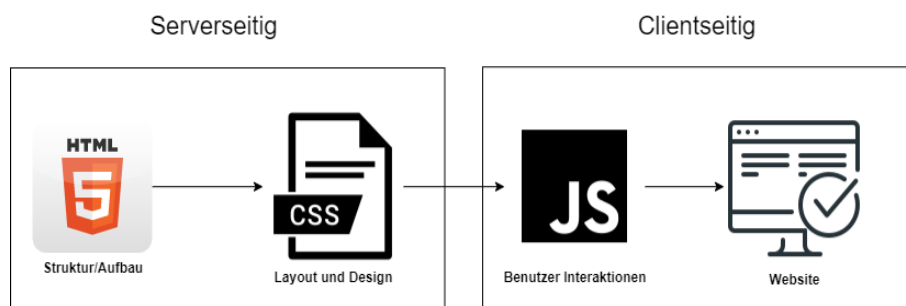


Abbildung 2.3: clientseitig

<sup>1</sup>Eine genaue Begriffserklärung befindet sich in 3.7

## 2.3.4 Framework

Die Wahl des Frameworks ist für die Umsetzung des Projekts entscheidend. Unter dem Begriff Framework versteht man eine Art Programmiergerüst, die es dem Programmierer deutlich erleichtert, ein Produkt zu erstellen. Jedes Framework bietet spezielle Lösungen und Lösungsansätze an und damit hat jedes einzelne sein eigenes Einsatzgebiet. In den nachfolgenden Unterkapiteln werden mögliche Frameworks für die Umsetzung des Projekts kurz erklärt. In 2.3.4.5 wird genauer darauf eingegangen, welches Framework gewählt wurde und aus welchem Grund.

### 2.3.4.1 Laravel

Das PHP-Framework Laravel, welches 2011 entwickelt wurde, basiert auf dem MVC Muster<sup>2</sup> und bietet damit eine sehr gute Strukturierung und Übersicht beim Arbeiten. Laravel wird in den meisten Fällen als Back-End Framework verwendet. Die meisten Projekte werden mit Laravel im Back-End in Kombination mit Vue.js im Front-End umgesetzt. Genauere Informationen zu Vue.js sind im Kapitel 2.6.3 nachzulesen. Laravel bietet jedoch die Möglichkeit, im Back- als auch im Front-End verwendet zu werden. Es lässt sich in folgende Einzelteile strukturieren:

- Migrations
- Views
- Controller
- Models
- Routen

Migrations sind die Abbildung der Tabellenstruktur und ermöglichen es, bei richtiger .env Datei Konfiguration, mit artisan Befehlen, ganz einfach eine Datenbank und die dazugehörigen Tabellen zu erstellen und diese auch genauso einfach wieder zu löschen oder zu leeren. Views fungieren als visueller Gliederungspart zwischen den Controllern und den Benutzern. In ihnen wird alles, was auf der Oberfläche ersichtlich ist, ausprogrammiert und gestaltet. Controller bilden die Verbindung zwischen den Views und den Models und ermöglichen es dem Benutzer, Daten mithilfe eines Zugriffs auf das Models zu verändern. Models bilden die Datenstruktur ab und ermöglichen den Zugriff auf die Daten und die Änderung dieser. Routen vermitteln die Benutzerabfrage von der View mit dem dazugehörigen Controller und werden in der Datei “web.php” definiert. Der ganze Prozess ist in Abbildung 2.4 visuell dargestellt.

---

<sup>2</sup>wird in Unterabschnitt 3.1.4 genauer erläutert

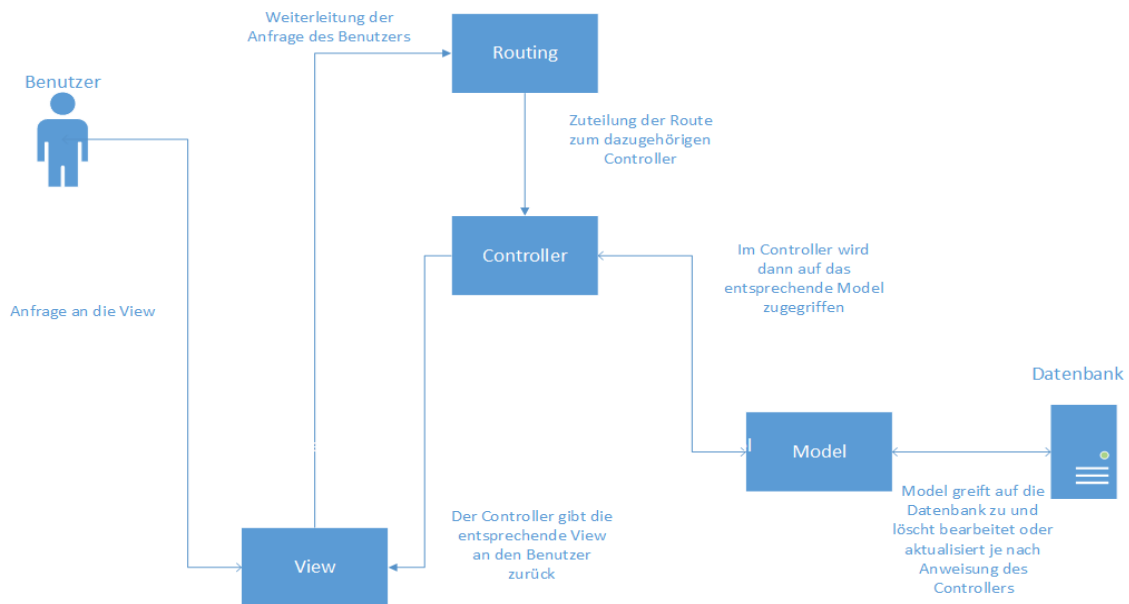


Abbildung 2.4: Laravel MVC

### 2.3.4.2 Angular

Angular ist ein clientseitiges JavaScript-Web-Framework und wird meistens zum Erstellen von sogenannten Single Page Applications, wie beispielsweise ein einfaches Bedienelement zur Steuerung einer Maschine, verwendet. Angular ist speziell für Web-,Desktop- und Mobile-Anwendungen entworfen. Ein weiterer Aspekt des Frameworks ist es, dass es von Google entwickelt wurde, und so ein langfristiger Support gewährleistet ist. Angular macht es mit der Kombination aus HTML und TypeScript möglich, so ziemlich jede mögliche Aufgabenstellung zu bewältigen. Es bietet auch besondere Features wie eine bidirektionale Datenbindung an und ist somit gut für Echtzeitanwendungen geeignet. Weitere Informationen zu Angular können unter der Quelle xy (<https://angular.de/artikel/was-ist-angular/>) nachgesehen werden.



Abbildung 2.5: Angular Logo

### 2.3.4.3 ASP.NET

ASP.NET (Active Server Pages) ist ein Web Applications Framework, welches 2002 von der Firma Microsoft veröffentlicht wurde. Es ist der Nachfolger des ASP Frameworks und bietet eine perfekte Grundlage, um dynamische Websites, Webanwendungen und Webservices zu entwickeln. Projekte werden hier in der Regel in der Sprache C# programmiert. Jedoch besteht auch die Möglichkeit, andere Sprachen wie beispielsweise Perl, Python oder Cobol zu verwenden. ASP.NET bietet nicht nur die Möglichkeit dynamische Websites zu erstellen, sondern auch Desktop Anwendungen. Durch die vordefinierte Datenstruktur, die ASP.NET bereitstellt, hilft es dem Programmierer Programmiersprachen nicht zu vermischen und einen übersichtlichen Programmierstil beizubehalten. Des Weiteren können viele Teile automatisch generiert werden und ersparen dem Programmierer damit einen immensen Aufwand.



Abbildung 2.6: ASP.net Logo

Genaure Informationen zu ASP.Net können unter der Quelle xy (<https://asp.mvc-tutorial.com/de/421/einfuehrung/was-ist-asp-net-mvc/>) nachgelesen werden.

### 2.3.4.4 React

React ist eine JavaScript-Softwarebibliothek, die es erleichtert, Benutzeroberflächen in Form von Web Applikationen zu erstellen. Es ist komponentenbasiert, was bedeutet, dass jedes Element in Blöcken aufgebaut ist und durch Zusammenfügen dieser einzelnen Codebits kann dann eine sogenannte View erstellt werden. Dies bietet die Möglichkeit, bereits erstellte Codebits auch in anderen Views zu verwenden. React wird oft zusammen mit ASP.NET verwendet.

Genauerer zu React kann unter der Quelle xy (<https://t3n.de/news/react-facebook-623999/>) nachgelesen werden.



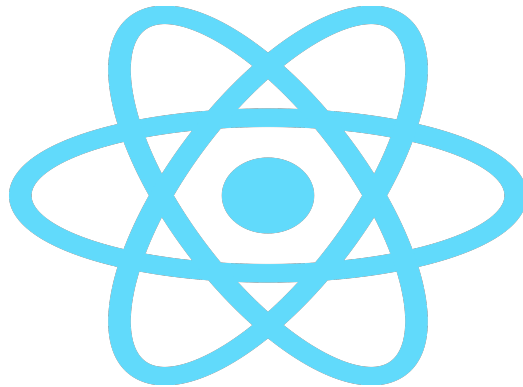


Abbildung 2.7: React Logo

### 2.3.4.5 Entscheidung des Frameworks

Positive und negative Aspekte jedes Frameworks wurden vom Projektteam, mithilfe der Tabelle 2.3.4.5, abgewogen.

Framework	Vorteile	Nachteile
React	Software Skalierbarkeit; hohe Flexibilität; Codebits ermöglichen es, die Redundanz zu erhöhen; schnelles Rendering; leichter Einstieg	Es nicht genug Möglichkeiten, eine Webapp zu erstellen und benötigt Zusatzbibliotheken; schweres debuggen; komplexe Benutzeroberfläche
ASP.NET	hohe Flexibilität; MVC Architektur; gute Skalierbarkeit; einfache Benutzung	Ressourcenaufwendig; Teuer aufgrund des Lizenzkaufes; keine aktuelle Dokumentation; alte Versionen sind nicht mehr kompatibel
Angular	frühzeitige Fehlererkennung; Konsistent	relativ starr und unflexibel; schwer zu erlernen;
Laravel	Unit Tests werden angeboten; einfache Erweiterbarkeit; Routing in der App; nutzt die neuesten Funktion von PHP; sehr gute Dokumentation; integrierte Benutzerverwaltung; integrierte Mailverwaltung; performant; einfache Handhabung; Bootstrap Einbindung ist einfach möglich	unterstützt keine Zahlungsfunktion; keine Mobil App Entwicklung; oft Probleme bei Updates

Tabelle 2.2: Entscheidung des Frameworks

Nach Begutachtung der Tabelle 2.3.4.5 und aufgrund der überwiegenden positiven Aspekte , entschied sich das Projektteam für das Framework Laravel. Entscheidend war beispielsweise die einfache Handhabung und die Möglichkeit Bootstrap in Kombination zu verwenden. Da dieses Framework auch im Schulfach Softwareentwicklung genauer bearbeitet wurde, besitzt das Projektteam bereits ein gutes Grundwissen. Es wurde darüber hinaus überlegt, ob auch das Front-End Framework Vue.js zusätzlich verwendet werden soll. Da Vue.js nur zusätzliche Komplexität ohne einen wirklichen Nutzen mit sich bringen würde, wurde diese Option vernachlässigt. Das Framework Laravel bietet alle nötigen Front- und Back-End Möglichkeiten, die es ermöglichen, Laravel im Front wie auch im Back-End zu verwenden. Entscheidender Faktor war auch, dass das Projektteam noch nie mit einem anderen Framework gearbeitet hat, und so eigentlich nur Laravel in Frage gekommen ist.

### 2.3.5 Front-End Templates

Die Auswahl des Front-End Templates ist essenziell für die Umsetzung der Weboberfläche des Produktes. Jedes Front-End Template bringt seine Vor- und Nachteile mit sich, wobei das Projektteam die Anforderungen an das Front-End Template mit den Vor- und Nachteilen jeden einzelnen Front-End Templates verglichen hat, um sich schlussendlich für eines zu entscheiden. Dabei sind die Anforderungen des Projektteams an das Front-End Template, dass es unkompliziert in das Projekt einzubinden ist und das Projektteam bereits Erfahrung damit hat, um das Arbeiten damit zu erleichtern. Weitere Anforderungen sind, dass es Responsive-Layouts ermöglicht, sowie Vorlagen von Komponenten, die eingebunden werden können, zur Verfügung stellt. Die drei Front-End Templates, die das Projektteam zur Auswahl stellt, sind Bootstrap, Tailwind und Vue.js.

#### 2.3.5.1 Bootstrap

Bootstrap ist ein Open Source Front-End-CSS-Framework. Es basiert auf den Programmiersprachen HTML, CSS sowie JavaScript und stellt Gestaltungsvorlagen wie Formulare, Tabellen, Buttons und weitere Oberflächengestaltungen bereit. Bootstrap ist weltweit eines der bekanntesten und beliebtesten Front-End-Frameworks. Ebenso bietet Bootstrap eine Open Source-SVG-Icon-Bibliothek an, womit die Kompatibilität zwischen Komponenten und Icons bestenfalls gegeben ist. Vorteil dieses Templates ist, dass es leichtgewichtig ist, weil nur Front-End-Daten geladen werden müssen und keine Backend-Funktionalitäten. Zusätzlich ist es problemlos in das Projekt einzubinden und stellt mehrere Vorlagen zur Verfügung, die ebenso eingebunden werden können. Bootstrap ist kostenfrei zu verwenden, und das Projektteam bringt bereits Erfahrung mit diesem Template mit. Die Nachteile sind, dass Bootstrap wenige Back-End-Funktionen bereitstellt, und es ist weniger gut geeignet für sehr große Applikationen.

#### 2.3.5.2 Tailwind.css

Tailwind ist ein Utility-First CSS-Framework, welches derzeit weltweit sehr beliebt ist. Diese Art von Framework soll mehr Flexibilität bieten als die traditionellen Vorgänger. Tailwind.css stellt Utility-Klassen zur Verfügung, mit denen man selbst Klassen zum Stylen der Komponenten definieren kann. Ein Inlinestyl ist bei diesem Framework ebenso möglich, womit externe CSS-Dateien überflüssig sind. Tailwind stellt Hilfsklassen zur Verfügung, welche das Arbeiten um einiges erleichtern. Zusätzlich bietet dieses Template eine hohe Flexibilität sowie mehrere Funktionen, um ein Responsive Design zu erreichen. Jedoch hat dieses Template keine vorgefertigten Komponenten, auf welche zurückgegriffen werden kann, und der Benutzer dieses Templates braucht sehr viel Zeit zum Erlernen der richtigen Verwendung. Die Größe der zu installierenden CSS-Datei ist sehr groß, da dieses Template eine breite Palette von Klassen bereitstellt, wobei die meisten oft in einem Projekt nicht verwendet werden. Das Projektteam hat mit diesem Template wenig Erfahrung, und die Dokumentation im Internet ist sehr mangelhaft.

### 2.3.5.3 Vue.js

Das JavaScript Framework Vue.js, welches für die Front-End-Entwicklung eingesetzt werden kann, wird immer populärer. Die Einfachheit und Benutzerfreundlichkeit ist ein Grund für die immer größer werdende Beliebtheit. Vue.js basiert auf dem virtuellen DOM<sup>3</sup>, was bedeutet, dass das Abarbeiten von Befehlen schneller funktioniert als beim herkömmlichen bekannten DOM<sup>4</sup>. Vue.js basiert auf den Programmiersprachen HTML, CSS und JavaScript, womit die wichtigsten verwendeten Sprachen abgedeckt sind. Vue.js stellt Funktionen wie UniTests, End-to-End-Tests sowie Routing-Systeme bereit. Die Lesbarkeit bei diesem Template ist sehr gut, da alle Komponenten sich in einer Datei befinden. Das Projektteam hat nur wenig Erfahrung mit diesem Template und es herrschen generelle Sprachbarrieren, da vieles nur in der Sprache Chinesisch vorhanden ist.

### 2.3.5.4 Entscheidung des Front-End Templates

Das Projektteam hat sich für das Front-End Template Bootstrap CSS entschieden aufgrund der Vorteile, die es mit sich bringt. Diese Entscheidung wurde unter anderem deswegen so getroffen, da das Projektteam bereits Erfahrung mit diesem Framework sammeln konnte und dieses Framework alle Anforderungen, die an das Front-End Template gestellt wurden, erfüllt. In der nachfolgenden Tabelle ist eine Gegenüberstellung der Vor- und Nachteile der Front-End Templates ersichtlich:

Front-End Template	Vorteile	Nachteile
<b>Bootstrap</b>	basiert auf HTML, CSS und JavaScript stellt Gestaltungsvorlagen bereit weltweit sehr bekannt und beliebt leicht in das Projekt einzubinden Projektteam hat bereits Erfahrungen damit	wenige Back-End Funktionen nicht geeignet für große Applikationen
<b>Tailwind.css</b>	weltweit sehr beliebt bietet mehr Flexibilität stellt Utility-Klassen zur Verfügung	keine vorgefertigten Komponenten Projektteam hat sehr wenig Erfahrung Online-Dokumentation ist mangelhaft erfahrungsloser Benutzer benötigt viel Einarbeitungszeit Größe der CSS-Datei ist enorm
<b>Vue.js</b>	einfach und benutzerfreundlich basiert auf HTML, CSS und JavaScript stellt Funktionen wie UniTests, End-to-End Tests und Routing-Systeme bereit hohe Lesbarkeit des Codes	Projektteam hat wenig Erfahrung Sprachbarrieren, vieles nur auf Chinesisch

Tabelle 2.3: Vor- und Nachteile der Front-End Templates

<sup>3</sup>Virtuelles Document Object Model, beim Aktualisieren schneller als das herkömmliche

<sup>4</sup>Document Object Model, stellt XML oder HTML Dokumente als Baumstruktur dar

### 2.3.6 Verbindung der Datenbank mit Laravel

In Laravel ist die Verbindung der Datenbank mithilfe der `.env` Datei möglich. Tabellen werden in sogenannten Migrations definiert. Auf die einzelnen Komponenten und deren Individualisierung wird in den Kapiteln 2.3.6.1 und 2.3.6.2 noch genauer eingegangen.

#### 2.3.6.1 Laravel `.env` Datei

In Laravel bietet die `.env` Datei, welche sich direkt im Projektordner befindet, die Möglichkeit, alle nötigen Umgebungsvariablen zu initialisieren. Ein Beispiel für Umgebungsvariablen sind Datenbank Anmeldeinformationen, Cache-Treiber, Mail-Server-Domain oder auch die App URL. Wichtig ist es, dass die `.env` Datei immer Gerät spezifisch konfiguriert wird. Das heißt, es ist wichtig beim Zusammenarbeiten, speziell mit Git, sehr gut aufzupassen. Folgedessen ist es empfehlenswert, die `.env` Datei mit einem `Gitignore`<sup>5</sup> zu versehen, um Komplikationen zu vermeiden. Es besteht auch die Möglichkeit, Änderungen in der `.env` Datei direkt aus dem Programm vorzunehmen.

#### 2.3.6.2 Migrations

Migrations bilden in Laravel eine Möglichkeit, die Tabellenstruktur im Programm selbst fest zu legen. Sie sind sozusagen der Blueprint<sup>6</sup>, in welchem die gesamte Tabellenstruktur vordefiniert ist. Für jede Tabelle, die in der Datenbank, welche vorher in der `.env` Datei definiert wurde, erstellt werden soll, gibt es eine eigene Migration. Migrations sind unterteilt in eine `up()` und in eine `down()` Methode<sup>7</sup>. In der `up()` Methode wird der Name der Tabelle der Name der einzelnen Tupel und deren Datentyp definiert. Der Datentyp `id()` ist standardmäßig immer der primary Key. In der `down()` Methode wird die Tabelle von der Datenbank gelöscht. Die beiden Methoden werden durch `artisan` Befehle, welche im Kapitel 3.10 ersichtlich sind, angesteuert. Erstellt werden Migrations mithilfe eines weiteren Laravel Befehls, welcher ebenfalls im Kapitel 3.10.1 beschrieben wird. Wenn die Migration erfolgreich erstellt wurde, ist diese folgendermaßen benannt : `JJJJ.MM.DD_tempname.php` und befindet sich im Ordner `Projektname\database \migrations`.

Alle Informationen liegen der Quelle xy (<https://laravel.com/docs/9.x/migrations>) zugrunde.

---

<sup>5</sup>Gitignore gibt an, welche Datei bei einem Git Pull nicht heruntergeladen werden soll

<sup>6</sup>ein Blueprint ist sozusagen der Bauplan oder eine Vorlage wie etwas auszusehen hat

<sup>7</sup>Methoden sind ein Art Container für den in ihnen auszuführenden Code

### 2.3.6.3 Seeder und Factories

Factories ermöglichen es, Fake Models<sup>8</sup> zu erstellen und mithilfe dieser auf die Datensätze in der Datenbank zuzugreifen und diese zu verändern. Daten werden mithilfe der Open Source-Bibliothek Faker generiert. Die Faker Bibliothek kann folgende Daten generieren:

- E-Mail
- einen Namen
- eine Telefonnummer
- Wörter
- Sätze
- Absätze
- Zufallszahlen

Datenbank-Factories erlauben es, Tabellen programmgesteuert mit Daten anzufüllen. Sie ermöglichen es, innerhalb von einigen Minuten tausende Daten in die Datenbank zu schreiben. Die Hauptaufgabe eines Seeder ist es jedoch nur die Factories aufzurufen.

### 2.3.6.4 Datenübergabe in Laravel

Ein Controller hat die Aufgabe mithilfe eines Models die Daten zu verändern und danach eine View zurückzugeben. Möchte man jedoch Daten wie Attribute, welche im Formular mitgeschickt werden, auf der zurückgegebenen View verwenden, müssen diese mitgegeben werden. Es gibt die Möglichkeit, Daten mithilfe von compact in das Return Statement<sup>9</sup> anzuhängen. Dies würde im Code folgendermaßen aussehen:

Quellcode 2.1: Return Statement mit Compact

```
1 return view(Viewname, compact('data'));
```

In der View kann dann mit folgendem Code auf die Daten zugegriffen werden:

Quellcode 2.2: Zugriff mittels Foreach Schleife

```
1 @foreach ($data as $d)
2     $d->Beispiel_Attribut
3 @endforeach
```

Die zweite Variante Daten zu Übergeben schaut folgendermaßen aus:

Quellcode 2.3: Attributübermittlung in einem return View Statement

```
1 return view('Viewname', ['AttributeName' => 'AttributWert']);
```

<sup>8</sup>Fake Models sind Models die nur temporär angelegt werden

<sup>9</sup>Rückgabewert einer Funktion

Diese Variante bietet den Vorteil, dass im Code gleich direkt auf das Attribut zugegriffen werden kann und keine Foreach Schleife nötig ist. Ein solcher Zugriff würde folgendermaßen aussehen :

Quellcode 2.4: Zugriff im Program

```
1  {{ $AttributName }}
```

Alle Informationen und Arten der Datenübergabe wurden aus dem Video aus der Quelle xy () entnommen.

## 2.4 Visuelle Darstellung der Energiesysteme und Energietechnologien

Von Seiten des Auftraggebers wurde von Anfang an der Wunsch geäußert, Energiesysteme und Energietechnologien, mithilfe eines Kartendienstes visuell darzustellen. Um andere Möglichkeiten, wie eine Karte auszuschließen, wurde auch zu Geoinformationssystem und CSS-System recherchiert. Die Ergebnisse dieser Recherchen werden in den nachfolgenden Kapiteln behandelt.

### 2.4.1 Kartendienste

Das Produkt stellt eine zentrale Verwaltungsmöglichkeit in Form einer Karte bereit. Um diese Karte auf dem Produkt zu verwenden, wird ein Kartenanbieter benötigt. Die Zwei Optionen Google Maps und OpenStreetMap wurden vom Projektteam analysiert. Auf diese zwei Anbieter wird in den Kapitel 2.4.1.2 und ?? genauer eingegangen. Von Seiten des Auftraggebers wurde, aufgrund einer gewissen Abneigung zu Diensten, die von Google bereitgestellt werden, eine Präferenz zu OpenStreetMap geäußert.

#### 2.4.1.1 Google Maps

Google Maps ist der klare Marktführer in der Sparte der Kartendienst Anbieter. Der Großteil aller Webanwendungen, welche Maps verwenden, werden mit Google Maps umgesetzt. Google Maps überzeugt nicht nur mit einer sehr hohen Individualisierungsmöglichkeiten, sondern auch mit einer sehr hohen Ausfallsicherheit. Der größte Vorteil ist jedoch, das große Angebot an vordefinierten Api Diensten. Diese können einfach über die Google Cloud Platform überwacht, bedient oder aktiviert werden. Genauere Infos zur Google Maps Plattform stehen im Kapitel 3.9. Ein weiterer Vorteil von Google Maps ist es, dass es wirklich weitflächig alles abdeckt, ganz im Gegensatz zu OpenStreetMap. Der einzige Nachteil ist, dass die Verwendung einer Google Maps Karte auf der Website kostenpflichtig ist. Es gibt jedoch auf der Google Cloud Seite eine genaue Kostenaufstellung. Bezahlt wird bei Verwendung der Api Dienste. Google verrechnet hier je nach Api Dienst einen gewissen Betrag pro Verwendung. Diese Kosten waren jedoch bei der Entwicklung kein Problem, da auf Google Cloud für die ersten 30 Tage ein Startkapital von 300 Dollar anbietet.



Abbildung 2.8: Google Maps Logo

#### 2.4.1.2 OpenStreetMap

OpenStreetMap ist ein 2004 gegründetes Projekt, welches das Ziel hat, eine frei verfügbare Weltkarte zu erstellen. Gegründet wurde dieses Projekt von Steve Coast und zählt mittlerweile um die 7.8 Millionen Nutzer. Das Hauptargument OpenStreetMap eher als Google Maps zu verwenden ist es, dass dieses kostenlos und lizenzfrei ist. Es gibt jedoch gravierende Nachteile gegenüber Google Maps. Openstreetmap bietet keine Api Dienste an. Es gibt zwar die Möglichkeit, mithilfe von Leaflet, welches eine JavaScript-Bibliothek ist, eigenen Api Dienste zu programmieren, dies ist jedoch mit einem sehr hohen Aufwand und einer sehr hohen Komplexität verbunden. Ein weiterer Nachteil von Openstreetmap ist es, dass es auf Benutzerdaten aufbaut. Das bedeutet, Nutzer sind dafür verantwortlich, die Karte immer auf dem aktuellsten Stand zu halten. Somit ist dieser Kartendienst wenig zuverlässig. Ein weiterer Nachteil wäre, dass Daten für alle User zugänglich sind.



Abbildung 2.9: Open Street Map Logo

Informationen über OpenStreetMap wurden aus der Quelle [xy \(\)](#) entnommen.



### 2.4.2 Geoinformationssystem

Die aufwendigste Option ist es, die Energiesysteme mittels GPS (Global Positioning System) zu orten und diese dann auf der Seite visuell darzustellen. In diesem Falle müsste das Galileo, welches das Europäische GPS System ist, genutzt werden. Galileo ist ein offenes, kostenloses Signal, welches es ermöglicht, den Sender auf bis zu zwei Meter genau zu orten. Ein solcher GPS Sender würde auf circa zehn bis zwanzig Euro pro System oder Technologie kommen. Die gesendeten Signale können mithilfe eines Empfängers ausgewertet werden.

### 2.4.3 CSS-System

Eine weitere Option ist es, die Energiesysteme und Technologien mittels CSS auf der Website darzustellen. Eine Möglichkeit wäre es, die Koordinaten der Systeme in die Datenbank zu schreiben, diese dann in Laravel auszulesen und auf einem selbst erstellten Hintergrund einzuzichnen. Dieses System könnte in Kombination mit einem Geoinformationssystem \* verwendet werden. Dies könnte dann folgendermaßen aussehen:

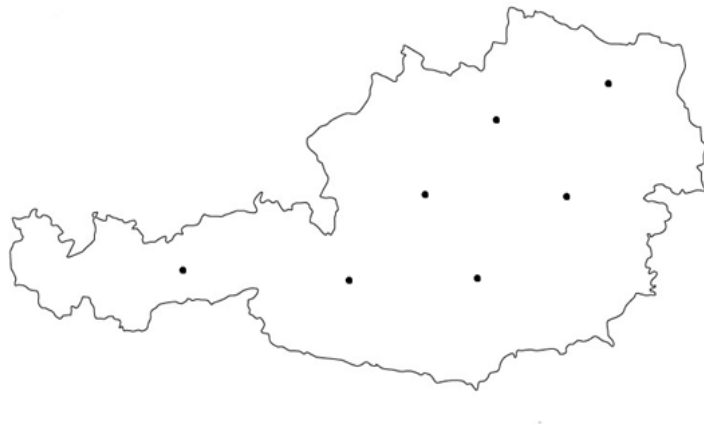


Abbildung 2.10: Beispiel eines CSS-Systems

### 2.4.4 Auswahl des Anbieters

Bei der Wahl des Anbieters, müsste der Auftraggeber Best GmbH berücksichtigt werden. Folgende Punkte muss der Anbieter erfüllen:

- leichte Implementierung
- es sollen alle gewünschten Funktionen umsetzbar sein
- es soll möglichst ohne viel extra Aufwand möglich sein
- es soll nicht Kostspielig sein

Um die Anbieter auf die gewünschten Anforderungen zu Überprüfen wurde die Liste angefertigt.

Anbieter	Kosten	Apis vorhanden	Umsetzbarkeit(Zeit)	Vorteile und Nachteile
Google Maps	gering	ja	ja	viele Apis, die die Arbeit erleichtern; einfache Einbindung auf der Website; performant; großflächig aufgeschlossen; Daten nur für den Betreiber selbst einsehbar
Open Street Map	keine	nein	nein	keine Apis vorhanden; viel mehr Aufwand, um das Gleiche wie Google Maps zu bieten; unperformant; keine großflächige Aufschlüsselung; Daten sind für Jeden einsehbar
Geoinformationssystem	sehr hohe	nein	nein	genaue Ortung der Systeme; sehr preisintensiv; sehr zeitintensiv und komplex
CSS-System	keine	nein	ja	keine Apis; sehr viel Aufwand; erfüllt den gleichen Zweck wie eine Karte

## 2.5 Berechtigungssystem Benutzer

Im folgenden Abschnitt wird näher auf das Benutzerverwaltungssystem eingegangen sowie auf die vorhandenen Benutzerrollen, die einem Benutzer zugeteilt werden können.

### 2.5.1 Benutzerrollen

Die Weboberfläche bietet für deren Besucher ein rollenbasiertes Benutzersystem an, um die Funktionen für jeden Benutzer zu deklarieren. Das rollenbasierte Benutzersystem unterscheidet zwischen folgenden Benutzern:

- Administrator
- Mitarbeiter
- Öffentlicher Benutzer

#### 2.5.1.1 Administrator

Der Administrator-Benutzer ist der höchste Benutzer von allen. Er darf Energiesysteme sowie Energietechnologien erstellen und alle anderen vorhandenen bearbeiten und löschen. Zusätzlich dazu hat er Einsicht in

sämtliche Grafana-Statistiken. Diese Benutzerrolle hat das Recht, neue Benutzer auf der Weboberfläche zu registrieren und vorhandene zu löschen, was für alle anderen Benutzer nicht möglich ist.

#### 2.5.1.2 Mitarbeiter

Die Rolle „Mitarbeiter“ darf auf der Weboberfläche neue Energiesysteme sowie Energietechnologien erstellen. Die von ihm erstellten Energiesysteme sowie Energietechnologien kann er bearbeiten oder löschen, und er hat auf diesen ebenso die Berechtigung auf Einsicht der Statistiken. Andere Mitarbeiter dürfen seine Energiesysteme und Energietechnologien nicht bearbeiten oder löschen und haben keinen Zugriff auf dessen Statistiken mit Ausnahme des Administrators. Ein Mitarbeiter-Benutzer darf somit nur seine selbst erstellten Energiesysteme und Energietechnologien verwalten. Der Mitarbeiter-Benutzer hat nicht die Berechtigung, neue Benutzer zu registrieren oder vorhandene zu löschen.

#### 2.5.1.3 öffentlicher Benutzer

Dieser Benutzer hat die geringste Berechtigung und tritt in Kraft, wenn man nicht auf der Weboberfläche angemeldet ist. Dieser Benutzer darf keine Energiesysteme sowie Energietechnologien erstellen, bearbeiten oder löschen. Zusätzlich hat der öffentliche Benutzer keine Einsicht auf sämtliche Grafana-Statistiken. Der Zugang zu der Benutzerverwaltung ist ebenso nicht erreichbar. Dieser Benutzer sieht lediglich die öffentlichen Daten, die für jedes Energiesystem und jede Energietechnologie preisgegeben werden.

### 2.5.2 Berechtigungen in Laravel

Im folgenden Abschnitt wird erklärt, wie in Laravel die zuvor genannten Benutzerrollen unterschieden werden. Dafür bietet Laravel Authentifizierungsrichtlinien. Dabei kann mit den Befehlen @auth und @guest überprüft werden, ob der aktuelle Benutzer authentifiziert oder ein Gast ist. Je nach Authentifizierung und Rolle hat der Benutzer unterschiedliche Rechte sowie Funktionen zur Verfügung. Für genauere Informationen, wie eine Berechtigungsüberprüfung in Laravel umgesetzt werden kann, siehe Quelle x.y.

## 2.6 Ui/Ux Design

Im Wort Ui/Ux Design steht das Ui für User Interface und Ux für User Experience. Es handelt davon, wie eine Website von einem Nutzer aufgefasst und wahrgenommen wird. Es beschreibt die Wirkung, die das Design auf einen Nutzer hat. Genauer beschreibt die User Experience wie sich ein Nutzer auf der Website fühlt oder wie er auf ihr navigieren kann. Ziel ist es, dem Benutzer ein möglichst entspanntes Gefühl zu geben und ihm bei Allem zu helfen. Das User Interface beschäftigt sich vielmehr mit der Website an sich. Man macht sich hierbei Gedanken, ob der Benutzer das Design ansprechend finden könnte, oder ob überall die idealen Elemente oder Elementgrößen gewählt wurden. Eine weitere Frage, die man sich im User Interface überlegt, ist was wirkt auf den Benutzer positiv was negativ. Einige Methoden das Ui/Ux Design umzusetzen sind :

- Erstellen eines Wireframes

- Umfragen an Usern durchführen
- Fragebögen auf der Seite selbst
- erstellen einer Übersicht mit den Anforderungen der Benutzer

Nützliche Programme zum Erstellen der besagten Dokumente sind folgende :

- Kissmetrics
- Figma
- SimilarWeb

### 2.6.1 Wireframe

Ein Wireframe hilft dem Design Team eine grobe Struktur der Website vorzudefinieren. Bei der Erstellung eines Wireframes wird der Visuelle Design Part komplett vernachlässigt. Bei einem Wireframe wird sich auf den Aufbau und die Position der Elemente konzentriert. Es ist meist in schwarz und weiß gehalten, da Farben bei diesem Aufbauprozess nur behindern würden.

### 2.6.2 Persona

Eine Persona dient dazu, die Bedürfnisse und Ziele der Zielgruppe zu erfassen. Mit Hilfe der Persona kann auf Wünsche und Bedürfnisse der Benutzer während der Entwicklung eingegangen werden. Empfohlen wird pro Zielgruppe um die fünf bis sechs Personen das Produkt testen zu lassen. Erstellt werden sie dann auf Grundlage von :

- Interviews mit dem Benutzer
- Benutzertests
- Umfragen

## 2.7 Template Layout

Laravel bietet die Möglichkeit, mithilfe einer Layout Datei, Codeabschnitte auf mehreren Seiten zu verwenden. Hauptsächlich wird diese Funktion genutzt, um statische Elemente, die auf jeder Unterseite gleich sind, nur einmal zu programmieren. Ein gutes Beispiel wäre der Header oder auch der Footer, da sich diese Elemente auf den Unterseiten meist nicht variieren.

### 2.7.1 Platzhalter Yield

In der Layout Datei kann `yield` verwendet werden, um Platz zu halten. Dies ermöglicht es, beispielsweise den Titel der Website individuell für jede Unterseite zu setzen. Der Platzhalter wird in der Layout Datei folgendermaßen konfiguriert: `<title>yield('title')</title>` . Möchte man diesen Platzhalter dann auf einer der Unterseiten, welche die Layout Datei verwenden, setzen, ist das folgendermaßen möglich : `section('title', 'titlename')`. Das Projektteam verwendet dieses Statement auf jeder Unterseite, um den Titel zu setzen.

### 2.7.2 Sections

Sections ermöglichen es, die Layout Datei in Abschnitte zu unterteilen. Der Beginn einer Section wird mit dem Befehl: `section('sectionname')` gekennzeichnet. Um die Section wieder zu schließen, muss folgendes Statement benutzt werden: `endsection`. Die Einteilung des Codes in Section erlaubt es, dieses Codestück auf einer anderen Seite mit nur einem Stichwort einzubinden.

### 2.7.3 Einbindung der definierten Sections

Um diese im Layout File definierten Sections im Code einbinden zu können, müssen einige Einstellungen vorgenommen werden. Am Anfang einer neuen Seite muss dieser Code eingefügt werden: `extends('layoutfilename')`. Damit wird der Seite mitgeteilt, dass die angegebene Seite als Layout Vorlage verwenden werden soll. Wenn diese Zeile vorhanden ist, können mittels `section('sectionname')` die im Layout File definierten Sections eingebunden werden. Wichtig ist, jedesmal nach dem `section` auch ein `endsection` einzugeben, sonst wird die Section nicht angezeigt. Zur Ergänzung der Section kann man den zu ergänzenden Code einfach zwischen dem `section` und dem `endsection` platzieren. Dieser wird dann zum bereits vordefinierten Code ergänzt.

## 2.8 Laravel Befehle

Laravel bietet die Möglichkeit, viele Dinge mithilfe der Konsole Models, Controller oder Views, automatisch generieren zu lassen. In den folgenden Unterkapiteln werden die Wichtigsten dieser Befehle aufgelistet und kurz erklärt.

### 2.8.1 Migration Befehle

Um eine neue Migration zu erstellen, muss folgendes Kommando in der Commandline ausgeführt werden:

Quellcode 2.5: Erstellen einer neuen Migration

```
1 php artisan make:migration migration_name
```

Wenn die Migration erfolgreich erstellt und initialisiert wurde, kann man mit folgendem Kommando die `up()` Methode aller Migrations aufrufen:

Quellcode 2.6: Erstellen einer neuen Migration

```
1 php artisan migrate
```

Nach Ausführung dieses Kommandos sind alle Tabellen in der Datenbank so wie gewünscht erstellt. Um alle ausgeführten Migrations anzeigen zu lassen, muss dieses Kommando verwendet werden:

Quellcode 2.7: Erstellen einer neuen Migration

```
1 php artisan migrate:status
```

Wenn die zuletzt erstellten Tabellen rückgängig gemacht werden sollen, gibt es die Möglichkeit, folgendes Kommando auszuführen:

Quellcode 2.8: Erstellen einer neuen Migration

```
1 php artisan migrate:rollback
```

Weitere nützliche Befehle sind unter der Quelle xy () zu finden.

## 2.8.2 Seeder und Factory Befehle

Um einen Datenbank Seeder zu erstellen, ist folgendes Kommando notwendig:

Quellcode 2.9: Erstellen einer neuen Migration

```
1 php artisan make:seeder seeder_name
```

Wenn eine Factory erstellt werden soll, gibt es folgende Möglichkeit:

Quellcode 2.10: Erstellen einer neuen Migration

```
1 php artisan make:factory factory_name
```

Um den Seeder nach Initialisierung auszuführen, gibt es folgende Möglichkeit:

Quellcode 2.11: Erstellen einer neuen Migration

```
1 php artisan db:seed
```

Weitere nützliche Befehle und Individualizations Möglichkeiten sind unter der Quelle xy() verzeichnet.

## 2.8.3 Model und Controller Befehle

## 2.8.4 Starten des Laravel Develop Servers

## 2.8.5 Befehle nach dem Git Pull

## Kapitel 3

# Ergebnisdokumentation

Dieser Abschnitt beinhaltet eine Dokumentation der verwendeten Werkzeuge zur Umsetzung der einzelnen Funktionen. Es werden alle behandelten Teilbereiche erläutert.

### 3.1 Laravel

Text

#### 3.1.1 Installation

Text

#### 3.1.2 Bootstrap Einbindung

Text

#### 3.1.3 Grafana Einbindung

Text

#### 3.1.4 MVC

Text

##### 3.1.4.1 Model

Text

### 3.1.4.2 View

Text

### 3.1.4.3 Controller

Text

## 3.2 Datenbankanbindung in Laravel

Um die Datenbank an das Programm anzubinden, müssen einige Änderungen in der `.env` Datei vorgenommen werden. Genauere Information zur `.env` Datei befinden sich im Kapitel 2.3.9.1. Welche Konfigurationen genau getätigt werden, können in den nachfolgenden Kapiteln nachgelesen werden.

### 3.2.1 Datenbank Anmeldeinformationen

Eine Änderung war es, die Datenbank Anmeldeinformationen auf die richtigen Einstellungen zu setzen. Es können folgende Parameter gesetzt werden :

- `DB_CONNECTION`
- `DB_HOST`
- `DB_PORT`
- `DB_DATABASE`
- `DB_USERNAME`
- `DB_PASSWORD`

`DB_CONNECTION` symbolisiert die Zugriffsvariante auf die Datenbank. `DB_HOST` bezeichnet die Domain des Datenbankservers, beispielsweise `127.0.0.1`, wenn der Datenbankserver über den Localhost bereitgestellt wird. `DB_PORT` ist dementsprechend der Port, auf welchem die Datenbank erreichbar ist. `DB_DATABASE` ist der Name der Datenbank, auf die zugegriffen werden soll. `DB_USERNAME` der Benutzername, mit welchem man auf die Datenbank zugreifen möchte. Diese Einstellungen sind vom Projekt Team vorgenommen worden, um eine Datenbank verwenden zu können und diese löschen beziehungsweise erstellen zu können.

### 3.2.2 Mail Server Konfigurationen

Um auf der Weboberfläche das Passwort Resetten mittels E-Mail zu ermöglichen, muss Laravel eine E-Mail versenden. Dazu braucht es die nötigen Konfigurationen in der `.env` Datei. Folgende Attribute müssen gesetzt werden:

- `MAIL_MAILER`



- MAIL\_HOST
- MAIL\_PORT
- MAIL\_USERNAME
- MAIL\_PASSWORD
- MAIL\_ENCRYPTION
- MAIL\_FROM\_ADDRESS
- MAIL\_FROM\_NAME

Bei MAIL\_MAILER wird das Protokoll eingetragen. Bei MAIL\_HOST wird die Host Domain eingegeben, also welcher SMTP Server verwendet werden soll. MAIL\_PORT gibt an, unter welchem Port der Server erreicht werden soll. MAIL\_USERNAME: hier muss die E-Mail Adresse, welche zum Absenden der Mail verwendet wird, eingetragen werden. Beim MAIL\_PASSWORD muss das dazugehörige Passwort eingetragen werden. MAIL\_ENCRYPTION gibt an, mit welchem Verfahren verschlüsselt werden soll. MAIL\_FROM\_ADDRESS ist die Adresse, welche beim Empfänger als Sender E-Mail angezeigt wird. MAIL\_FROM\_NAME ist der Name der beim Empfänger angezeigt wird. Das Produkt verwendet diese Einstellungen um eine Passwort Reset Email an den Benutzer zu senden. Alle geforderten Einstellungen wurden nach Übergabe an den Auftraggeber von diesem Konfiguriert.

### 3.2.3 Migrations

Die Migrations wurden dazu verwendet, um Tabellen zu gestalten. In ihnen wurden alle nötigen Attributnamen und deren Datentypen konfiguriert. Das Produkt verwendet folgende von Laravel automatisch erzeugte Migrations für die Benutzerauthentifizierung:

- 2014\_10\_12\_000000\_create\_users\_table.php
- 2014\_10\_12\_100000\_create\_password\_resets\_table.php
- 2019\_08\_19\_000000\_create\_failed\_jobs\_table.php
- 2019\_12\_14\_000001\_create\_personal\_access\_tokens\_table.php

Zusätzlich zu diesen bereits erstellten Migrations wurde für jede Energietechnologieart eine eigene Migration angelegt. Darüber hinaus wurden Templates<sup>1</sup> für ein Energiesystem und eine Energietechnologie erstellt. Im Energietechnologien Template sind alle Attribute, die bei jeder Energietechnologie gleich sind. Alle individuellen Attribute werden dann mithilfe der eigener Migrations abgebildet.

---

<sup>1</sup>ein Template ist ein Grundbauplan

### **3.3 Routen in Laravel**

Text

#### **3.3.1 Resource Routen**

Text

#### **3.3.2 GET Routen**

Text

#### **3.3.3 Auth Routen**

Text

### **3.4 Datenbankdesign**

Text

#### **3.4.1 Erstellen eines neuen Schemas**

Text

##### **3.4.1.1 ER-Model**

Text

##### **3.4.1.2 Fremdschlüssel**

Text

### **3.5 Corporate Design**

Vom Projektteam wurde ein Corporate Design Manual erstellt. Dies befindet sich auch auf der DVD der Diplomarbeit. In diesem Corporate Design Manual wurden folgende Themen bearbeitet:

- Oberfläche
- Definierte Farben
- Verwendung der definierten Farben

- Überschriften
- Buttons
- Interaktionsfarben
- Schriftarten
- Schriftgrade
- Logo
- Verwendete Icons
- Bedeutung der Icons
- Icons mit Funktionalitäten
- Website Design

Auf die wichtigsten Elemente wird in den nachfolgen Kapiteln eingegangen.

### **3.5.1 Vorschläge**

Text

### **3.5.2 Änderungsvorschläge**

Text

### **3.5.3 Finales Design**

Text

### **3.5.4 Definierte Farben**

Text

### **3.5.5 Überschriften**

Text

### **3.5.6 Interaktionsfarben**

Text

### **3.5.7 Schriftarten**

Text

### **3.5.8 Schriftgrade**

Text

### **3.5.9 Logo**

Text

### **3.5.10 Verwendete Icons und deren Bedeutungen**

Text

### **3.5.11 Map Icons**

Text

### **3.5.12 Icons in Formularen**

Text

### **3.5.13 Icons im DataTable**

Text

### **3.5.14 Buttons**

Text

### **3.5.15 Tabelle mit generellen Informationen über einzelne HTML Elemente**

Text

### **3.5.16 Datenformate**

Text

## 3.6 Weboberfläche

Im Abschnitt Weboberfläche wird speziell auf das Front-End sowie das Back-End des Produktes eingegangen. Bei dem Punkt Back-End vor allem auf die Funktionen, die für die Benutzer-Interaktionen zuständig sind, und bei dem Punkt Front-End auf das Layout sowie das Design der Weboberfläche.

### 3.6.1 Backend

In diesem Abschnitt wird genauer auf den Ablauf des Codes im Back-End, der für die Funktionen auf der Weboberfläche zuständig ist, eingegangen. Für jedes Energiesystem oder jede Energietechnologie stehen die Funktionen Erstellen, Bearbeiten und Löschen zur Verfügung. In den folgenden Abschnitten wird genauer auf diese Funktionen eingegangen.

#### 3.6.1.1 Energiesystem Erstellen

Für das Erstellen eines Energiesystems ist ein Mausklick auf der Karte notwendig. Daraufhin öffnet sich ein Pop-up Fenster, um die Kerndaten des Energiesystems einzugeben. Das Pop-up zum Erstellen eines Energiesystems ist in der Abbildung x.y ersichtlich. Beim Erstellen eines Energiesystems werden folgende Attribute in der Datenbank erfasst: Eingabe des Benutzers:

- Bezeichnung
- Katastralgemeinde
- Postleitzahl

Automatisch ausgefüllte Attribute, welche nicht im Pop-up dargestellt werden:

- Längengrad
- Breitengrad
- User\_ID

Die Attribute Längengrad und Breitengrad werden automatisch mithilfe des Mausklicks auf der Karte mit den entsprechenden Koordinaten beim Erstellen des Energiesystems befüllt. Das Attribut User\_ID wird automatisch mit der ID des gerade angemeldeten Benutzers ausgefüllt, um das Energiesystem einem Benutzer zuteilen zu können. Nachdem der Benutzer alle Daten eingegeben hat, werden diese Daten an den Controller übermittelt, wo anschließend das Energiesystem erstellt und in der Datenbank erfasst wird. Als nächstes werden die gespeicherten Informationen wieder zurück an die Weboberfläche übergeben, um die Energiesysteme-Marker auf der Karte zu platzieren sowie den Inhalt der Liste aller vorhandenen Energiesysteme zu aktualisieren. Die folgende Abbildung zeigt den Ablauf, um ein Energiesystem zu erstellen:

### 3.6.1.2 Energiesystem Bearbeiten

Mit dem Stift-Icon beim Energiesystem in der Liste ist es möglich, die Kerndaten eines Energiesystems zu bearbeiten. Die zu bearbeitenden Daten, welche einen weißen Hintergrund aufweisen, lauten auf „Bezeichnung“, „Katastralgemeinde“ und „Postleitzahl“. Die Koordinaten sowie die User\_ID des Energiesystems bleiben konstant und sind nicht änderbar und nicht im Pop-up ersichtlich. Folgende weitere Attribute werden bei jedem Energiesystem automatisch berechnet, angezeigt und sind nicht bearbeitbar, sondern dienen nur als weitere Information über das ausgewählte Energiesystem. Das Pop-up zum Bearbeiten eines Energiesystems ist in der Abbildung x.y ersichtlich.

- Anzahl-Erzeugungstechnologien
- Anzahl-Verbraucher
- Anzahl-Speicher
- Ges-Nennleistung [kW]
- Ges-Energie [kW/h]
- Ges-Verbraucher-Leistung [kW]
- Ges-Verbraucher-Energie [kW/h]
- Ges-Erzeuger-Leistung [kW]
- Ges-Erzeuger-Energie [kW/h]
- Ges-Speicher-Kapazität [kW/h]
- Aktueller Netzbezug [kW]

Für genauere Informationen zu den einzelnen Attributen siehe Kapitel 3.5.10 Mit dem Button „Mehr Details zu diesem Energiesystem“ ist es möglich, die weiteren Daten des Energiesystems ein- und auszuklappen. Nachdem der Benutzer die Attribute des Energiesystems bearbeitet hat, werden diese mithilfe des Controllers in der Datenbank aktualisiert. Anschließend werden die neuen Daten der Weboberfläche übergeben, um auf der Karte sowie in der Liste den aktuellen Stand der Energiesysteme darstellen zu können. In der Abbildung x.y ist der Ablauf „Energiesystem bearbeiten“ dargestellt:

### 3.6.1.3 Energiesystem Löschen

Um ein Energiesystem zu löschen, benötigt man das Mülleimer-Icon, welches sich neben jedem Energiesystem in der Liste befindet, sofern der angemeldete Benutzer die erforderliche Berechtigung dazu hat. Nachdem dieses Icon betätigt wurde, wird die Information, dass ein Energiesystem gelöscht wurde, an den Controller übermittelt. Dieser löscht anschließend mithilfe der ID des ausgewählten Energiesystems dieses aus der Datenbank und übergibt alle anderen Energiesysteme zurück auf die Weboberfläche, um die vorhandenen Energiesysteme auf der Karte zu platzieren sowie in der Liste anzuzeigen. Der Ablauf, um ein Energiesystem zu löschen, wird in folgendem Diagramm präsentiert:

#### 3.6.1.4 Energietechnologie Erstellen

Für das Erstellen einer Energietechnologie ist das Auswählen eines Energiesystems mit einem Doppelklick notwendig. Anschließend verwandelt sich der Cursor in das Energietechnologie-Icon, welches darauf hindeutet, dass jetzt das Hinzufügen einer Energietechnologie mittels eines Klicks auf der gewünschten Position auf der Karte möglich ist. Nach dem Klick auf der Karte öffnet sich ein Pop-up-Fenster, um die Kerndaten der Energietechnologie einzugeben. Dieses Pop-up ist in der Abbildung x.y ersichtlich.

Eingabe des Benutzers:

- Bezeichnung
- Typ
- Ort
- Bild einfügen
- Beschreibung

Automatisch ausgefüllt und nicht im Pop-up dargestellt:

- Längengrad
- Breitengrad
- Ensys-ID
- User\_ID

Nachdem der Benutzer die Daten im Pop-up ausgefüllt hat, werden diese Informationen über den Controller in der Datenbank erfasst. Anschließend werden die Energietechnologie-Daten an die Weboberfläche übermittelt, um die Energietechnologien auf der Karte darzustellen sowie in der Liste anzuzeigen. Der Ablauf, um eine Energietechnologie zu erstellen, wird in der Abbildung x.y dargestellt:

#### 3.6.1.5 Energietechnologie Bearbeiten

Um eine Energietechnologie zu bearbeiten, muss zuerst ein Energiesystem mit einem Doppelklick ausgewählt werden. Anschließend befinden sich rechts in der Liste alle dazugehörigen Energietechnologien, welche man mit dem Stift-Icon bearbeiten kann. Nach dem Betätigen des Stift-Icons öffnet sich ein Pop-up-Fenster, um die Attribute zu bearbeiten. Dieses Pop-up-Fenster ist in der Abbildung x.y ersichtlich. Dabei sind die Attribute „Bezeichnung“, „Ort“, „Bild“ und „Beschreibung“ bearbeitbar. Um dem Benutzer dies deutlich zu machen, weisen diese Attribute einen weißen Hintergrund auf. Die Attribute „Typ“, „Längengrad“, „Breitengrad“, „Ensys\_ID“ sowie „User\_ID“ sind nicht bearbeitbar. Nachdem der Benutzer die Daten bearbeitet hat, werden diese über den Controller in die Datenbank geschrieben. Anschließend werden die Energietechnologie-Daten an die Weboberfläche übergeben, um die Marker auf der Karte zu platzieren sowie den Inhalt der Liste zu aktualisieren. Das nachfolgende Diagramm beschreibt den Ablauf für das Bearbeiten einer Energietechnologie:

### 3.6.1.6 Energietechnologie Löschen

Um eine Energietechnologie zu löschen, benötigt man das Mülleimer-Icon, welches sich in der Liste neben den Energietechnologien befindet, sofern ein Energiesystem von dem Benutzer ausgewählt wurde. Sobald der Benutzer dieses Icon betätigt, wird mithilfe der ID der ausgewählten Energietechnologie dem Controller mitgeteilt, dass diese Energietechnologie gelöscht werden soll. Daraufhin löscht der Controller diese Energietechnologie aus der Datenbank und übergibt anschließend alle anderen vorhandenen Energietechnologie-Daten an die Weboberfläche, um dort die Marker zu platzieren sowie den Inhalt der Liste zu aktualisieren. Das nachfolgende Diagramm beschreibt den Ablauf für das Löschen einer Energietechnologie:

### 3.6.1.7 Benutzerverwaltung

Je nach Benutzerrolle des aktuell angemeldeten Benutzers stehen dem Benutzer unterschiedliche Funktionen auf der Weboberfläche zur Verfügung. Folgende drei Benutzer stehen zur Verfügung:

- Administrator
- Mitarbeiter
- Öffentlicher Benutzer

Wie eine generelle Überprüfung des Benutzers in Laravel umgesetzt werden kann, ist unter der Quelle x.y ersichtlich. Folgende Überprüfung wurde selbst vom Projektteam entwickelt und ist somit nicht unter der genannten Quelle auffindbar. Mit folgender Überprüfung wird kontrolliert, ob der gerade angemeldete Benutzer das Energiesystem oder die Energietechnologie selbst erstellt hat oder ob er die Rolle des Administrators aufweisen kann. Falls der Benutzer das Energiesystem oder die Energietechnologie selbst erstellt hat, ist der erste Teil der Überprüfung erfüllt und der Benutzer hat die Verwaltungsfunktionen zur Verfügung. Falls der Benutzer das Energiesystem oder die Energietechnologie nicht selbst erstellt hat, wird die zweite Überprüfung durchgeführt, welche überprüft, ob der Benutzer die Rolle des Administrators aufweisen kann. Falls er diese Rolle besitzt, hat er die Verwaltungsfunktionen zur Verfügung, andernfalls nicht.

### 3.6.1.8 Adresssuche

Mithilfe des Adresssuchfeldes ist es möglich, einen Standort einzugeben, um anschließend auf der Karte zu diesem zu gelangen. Dadurch ist das Auffinden von bestimmten Orten auf der Karte problemlos möglich. Dabei wird die eingegebene Adresse in geografische Koordinaten umgewandelt, zu welchen man anschließend navigiert wird. In der folgenden Abbildung ist das Adresssuchfeld ersichtlich.

Folgende Schreibweisen sind in der Suche möglich:

- Stadt
- Land
- Straße
- Postleitzahl



**Automatische Vervollständigung**

Um die Suche nach der richtigen Adresse zu vereinfachen, wird die Eingabe des Benutzers mit einer Auto-Complete-Funktion unterstützt. Diese Funktion bietet mögliche Ziel-Adressen anhand der bisher eingegebenen Daten an, welche vom Benutzer ausgewählt werden können. Ein Beispiel dieser Funktion ist in der Abbildung x.y dargestellt.

**Adresssuche durchführen**

Die Adresssuche kann auf zwei verschiedene Arten durchgeführt werden. Die erste Variante ist das Benutzen des dazugehörigen Buttons mit der Beschriftung „Suchen“. Die zweite Variante ist das Betätigen der Enter-Taste auf der Tastatur. Beide Varianten führen zu dem gleichen Ergebnis, und zwar, dass die Adresssuche durchgeführt wird. Sobald die Adresssuche durchgeführt wird, werden für die eingegebene Adresse die dazugehörigen Koordinaten berechnet. Sobald diese berechnet wurden, werden diese an die Karte übergeben, damit der Mittelpunkt der Karte auf diese Koordinaten gesetzt wird. Anschließend gelangt der Benutzer zu seiner eingegebenen Adresse auf der Karte und kann seine Interaktionen fortsetzen. Das folgende Diagramm beschreibt die Durchführung der Adresssuche:

**3.6.2 Front-End**

Text

**3.6.2.1 Home**

Text

**3.6.2.2 Energiesysteme**

Text

**3.6.2.3 Galerie**

Text

**3.6.2.4 Impressum**

Text

**3.6.2.5 Datenschutz**

Text

**3.6.2.6 Registrierungsseite**

Text

### 3.6.3 Login

Text

### 3.6.4 Registrierung

Text

### 3.6.5 Kartendienst Funktionalitäten

Neben den bekannten Kartendienst-Funktionen wie das Erstellen eines Energiesystems oder einer Energietechnologie bietet die Karte weitere Funktionen. Diese weiteren Funktionen sind das Auswählen und Abwählen eines Energiesystems.

#### 3.6.5.1 Auswählen eines Energiesystems

Das Auswählen eines Energiesystems ist notwendig, um die dazugehörigen Energietechnologien auf der Karte sowie in der Liste zu sehen. Ebenso ist es notwendig, um neue Energietechnologien erstellen zu können. Mit einem einfachen Klick auf das Energiesystem-Icon wird an dieses Energiesystem lediglich herangezoomt, jedoch noch nicht ausgewählt. Um ein Energiesystem auszuwählen, ist ein Doppelklick darauf notwendig. Dieser Doppelklick bewirkt, dass in der Liste sowie auf der Karte die dazugehörigen Energietechnologien angezeigt werden. Anschließend ist es für den Benutzer möglich, die Energietechnologien zu verwalten. Der Ablauf für das Auswählen eines Energiesystems wird in der folgenden Abbildung dargestellt:

#### 3.6.5.2 Abwählen eines Energiesystems

Nachdem ein Energiesystem ausgewählt wurde, ist mit einem einfachen Linksklick auf das Energiesystem-Icon möglich, das ausgewählte Energiesystem wieder abzuwählen. Nach Abwählen des Energiesystems werden die dazugehörigen Technologien wieder von der Karte sowie aus der Tabelle entfernt. Stattdessen werden in der Tabelle wieder alle vorhandenen Energiesysteme angezeigt. Der Ablauf für das Abwählen eines Energiesystems wird in der folgenden Abbildung dargestellt.

### 3.6.6 Anzeige von Energiesystemen und Energietechnologien auf der Karte

In diesem Abschnitt werden die Funktionen zum Darstellen und Entfernen der Marker auf der Karte erläutert.

#### 3.6.6.1 Energiesysteme Marker auf der Karte platzieren

Immer wenn die Map neu geladen wird, werden alle vorhandenen Energiesysteme in Form von Icons auf der Karte platziert. Dafür werden die Daten aller Energiesysteme aus der Datenbank gelesen, um anschließend die Marker auf der Karte platzieren zu können. Die dafür notwendigen Datenbank-Attribute, die ausgelesen werden müssen, sind „Bezeichnung“, „Breitengrad“ sowie „Längengrad“. Anschließend werden mit diesen

Informationen die Marker erstellt und auf der Karte dargestellt. Die Abbildung x.y stellt den Ablauf für das Platzieren der Energiesysteme-Marker auf der Karte dar.

### 3.6.6.2 Energietechnologien Marker auf der Karte platzieren

Sobald ein Energiesystem ausgewählt wurde, werden die dazugehörigen Energietechnologien angezeigt. Für das Anzeigen eines Energietechnologie-Markers werden die Attribute „Bezeichnung“, „Typ“, „Breitengrad“ sowie „Längengrad“ aus der Datenbank ausgelesen. Anschließend werden die Energietechnologie-Marker erstellt und auf der Karte präsentiert. Die Abbildung x.y stellt den Ablauf für das Platzieren der Energietechnologien-Marker auf der Karte dar.

### 3.6.7 Layoutvorlage der Website

Text

## 3.7 DataTable

Um die angezeigten Daten in der Tabelle zu sortieren, zu filtern oder deren Anzahl pro Seite zu begrenzen, wurde das Plug-in DataTable verwendet. DataTable ist ein Plug-In der JavaScript-Bibliothek jQuery. Dieses Tool bietet viele Funktionen wie die Suchfunktion, die Sortierfunktion und die Seitennummerierung. Für einen Standard-DataTable sind folgende Schritte notwendig:

Folgende JavaScripts und CSS-Files müssen für die Verwendung eines DataTable eingebunden werden.

Quellcode 3.1: Funktion Blade.php Zeile 1-2

```
101 <link rel="stylesheet"
    href="https://cdn.datatables.net/1.10.22/css/dataTables.bootstrap4.min.css">
102 <script src="https://cdn.datatables.net/1.10.22/js/jquery.dataTables.min.js"></script>
103 <script
    src="https://cdn.datatables.net/1.10.22/js/dataTables.bootstrap4.min.js"></script>
```

Anschließend muss folgende Funktion mit der ID des Tables eingebunden werden:

Anschließend sind alle DataTable-Funktionen an diesen Table gegeben. Links oben befindet sich ein Input-Feld, um die Anzahl der Datensätze pro Seite festzulegen. Die Suchfunktion, um nach einem bestimmtem Datensatz zu suchen, befindet sich rechts oben. Für jedes Attribut ist eine Sortierfunktion gegeben, welche an den kleinen Pfeilen neben dem Attributnamen erkennbar ist. Links unten sieht man, wieviele Datensätze gerade auf dieser Seite angezeigt werden und wieviele es insgesamt gibt. Rechts unten ist es möglich, mit den Buttons zwischen den Seiten hin und her zu wechseln, falls mehrere Seiten vorhanden sind. Die Sprache, die für den Standard-DataTable verwendet wird, ist Englisch. Mit den Standardeinstellungen würde die Liste folgendermaßen aussehen:

### 3.7.1 Individueller DataTable

Aufbauend auf den Standard-DataTable wurde der individuelle DataTable basierend auf den Anforderungen des Auftraggebers erstellt. Dabei ist die wichtigste Anforderung, dass pro Seite maximal 5 Datensätze dargestellt werden. Zuerst wurde die Sortierfunktion bei den Spalten drei, vier und fünf deaktiviert, da sich an diesen Stellen die Icons befinden und somit eine Sortierfunktion keinen Sinn macht. Die Auswahl für die Anzahl der Datensätze pro Seite wurde deaktiviert, da diese konstant auf den Wert fünf festgelegt wurde. Zum Schluss wurde die Sprache des Tables auf Deutsch geändert. Mit dieser DataTable Definition sieht die Liste anschließend folgendermaßen aus:

### 3.7.2 Sortierfunktion

Mit der Sortierfunktion ist es möglich, nach jedem einzelnen Attribut in der Liste zu sortieren. Die Sortierfunktion ist nur bei jenen Attributen aktiviert, wo es auch Sinn macht. Somit ist die Sortierfunktion bei den Icons nicht gegeben. Beim Laden des Tables ist der Inhalt automatisch alphabetisch nach dem ersten Attribut sortiert. Nach dem Betätigen der Sortierfunktion wird anschließend alphabetisch rückwärts sortiert.

### 3.7.3 Suchfunktion

Die Suchfunktion ermöglicht eine sofortige Textsuche des Inhaltes in der Liste. Dabei ist es möglich, mit Buchstaben oder mit Zahlen zu suchen. Jeder Datensatz, der den eingegebenen Buchstaben oder die eingegebene Zahl beinhaltet, wird angezeigt. Der Rest wird ausgeblendet und ist erst nach dem Beenden der Suche wieder sichtbar.

### 3.7.4 Seitenanzahl

Wenn auf einer Seite maximal fünf Datensätze angezeigt werden, entstehen bei einer großen Anzahl von Energiesystemen sowie Energietechnologien entsprechend viele Seiten. Diese Seiten sind mithilfe der Buttons rechts unten navigierbar. Links unten steht die Information darüber, wieviele Seiten es insgesamt gibt, und wieviele Einträge von allen vorhandenen gerade auf dieser Seite angezeigt werden.

### 3.7.5 Icons

Die Verwaltung der einzelnen Energiesysteme und der Energietechnologien ist mit den dazugehörigen Icons möglich. Welche Icons zu jedem Energiesystem oder jeder Energietechnologie zur Verfügung stehen, hängt von den Rechten des angemeldeten Benutzers ab. Es gibt zwei verschiedene Kombinationen von verfügbaren Icons. Entweder man hat nur das Icon „Auge“ zur Verfügung, welches bedeutet, dass man dieses Energiesystem oder diese Energietechnologie entweder nicht erstellt hat, nicht mit einem Administrator-Benutzer angemeldet ist oder gerade nicht auf der Weboberfläche angemeldet ist. Mit diesem Icon hat man die Möglichkeit, die Kerndaten eines Systems zu sehen, welche aber nicht bearbeitet werden können. Die zweite Variante ist, dass man die Icons „Mülleimer“, „Statistik“ und „Stift“ zur Verfügung hat, was bedeutet, dass der angemeldete Benutzer dieses System erstellt hat oder die Administrator-Berechtigungen besitzt. Auf Funktionen der einzelnen Icons wird im Kapitel 2.6.6.9 genauer eingegangen.

### 3.7.6 MoveToMarker

Um das Finden eines Energiesystems trotz des bereits vorhandenen Adress-Suchfeldes noch leichter zu ermöglichen, gibt es die Funktion MoveToMarker. Diese Funktion wird dann ausgeführt, wenn in der Liste auf die Bezeichnung, die Katastralgemeinde oder die Postleitzahl eines Energiesystems gedrückt wird. Das Ergebnis dieser Funktion ist, dass der Benutzer nach dem Klick auf ein Energiesystem in der Liste gleich zu dessen Position auf der Karte gelangt, um eine längere Suche danach zu ersparen. Der Ablauf der MoveToMarker-Funktion wird in folgender Abbildung nähergebracht.

Nachdem der Benutzer zum Standort des ausgewählten Energiesystems navigiert wurde, hat dieser dort wieder alle Funktionalitäten der Karte, wie das Hinzufügen eines neuen Energiesystems sowie einer Energietechnologie, gegeben.

## 3.8 Galerie Funktionen

Auf der Seite „Galerie“ ist es möglich, die Energietechnologien von einem ausgewählten Energiesystem anzeigen zu lassen. Dabei steht das ausgewählte Bild beim Erstellen einer Energietechnologie im Vordergrund. Dieses wird in Form einer Card mit deren Bezeichnung und Beschreibung präsentiert.

### 3.8.1 Auswahl eines Energiesystems

Die Auswahl eines Energiesystems ist mithilfe eines Drop-Down-Menüs möglich. Dieses Drop-Down-Menü beinhaltet alle vorhandenen Energiesysteme. In Abbildung x.y ist das Drop-Down-Menü mit allen vorhandenen Energiesystemen ersichtlich.

### 3.8.2 Energietechnologien des Energiesystems anzeigen

Wenn der Benutzer ein Energiesystem ausgewählt hat, werden die dazugehörigen Energietechnologien in Form von Cards dargestellt. Als Haupt-Überschrift über alle Energietechnologien dient die Bezeichnung des ausgewählten Energiesystems. Bei den einzelnen Energietechnologien, die dargestellt werden, wird zuerst das eingefügte Bild angezeigt, welches beim Erstellen einer Energietechnologie ausgewählt werden kann. Falls der Benutzer bei einer Energietechnologie kein Bild hinzugefügt hat, wird automatisch ein Standardbild eingefügt. Unter dem Bild dient die Bezeichnung sowie die Beschreibung der Energietechnologie als Bildunterschrift. In der Abbildung x.x sind Energietechnologien des Energiesystems MicroGridLab zu sehen. Dabei haben zwei Energietechnologien ein Bild beim Erstellen bekommen, und bei zwei weiteren namens PV-Dach2 und PV-Dach4 wurde das Standardbild eingefügt, da bei diesen Energietechnologien beim Erstellen kein Bild hinzugefügt wurde.

## 3.9 Grafana

Text

### **3.9.1 Automatisches Erstellen der Dashboards**

Text

### **3.9.2 Automatisches Erstellen der Panels**

Text

### **3.9.3 Energietechnologien Statistiken anzeigen**

Text

## **3.10 Einbindung von Google Maps**

Text

### **3.10.1 Google Cloud**

Text

### **3.10.2 Google Cloud Platform Account erstellen**

Text

### **3.10.3 Apis aktivieren und einbinden**

Text

### **3.10.4 Individuelle Map erstellen und einbinden**

Text

## Kapitel 4

# Resümee und Ausblick

Text

## Kapitel 5

# Quellen und Literatur

Text



# Abbildungsverzeichnis

1.1	Team . . . . .	4
2.1	vorhandeneSystemAuftraggeber . . . . .	6
2.2	Architektur . . . . .	9
2.3	clientseitig . . . . .	10
2.4	Laravel MVC . . . . .	12
2.5	Angular Logo . . . . .	12
2.6	ASP.net Logo . . . . .	13
2.7	React Logo . . . . .	14
2.8	Google Maps Logo . . . . .	21
2.9	Open Street Map Logo . . . . .	21
2.10	Beispiel eines CSS-Systems . . . . .	22

## Kapitel 6

# Tabellenverzeichnis

Text

## Kapitel 7

# Codeverzeichnis

Text

## Kapitel 8

# Begleitprotokoll gem. § 9 Abs. 2 PrO-BHS

Text

### 8.1 Begleitprotokoll David Pöchacker

Text

### 8.2 Begleitprotokoll Marcel Entner

Text

### 8.3 Begleitprotokoll Tobias Kronsteiner

Text

# Kapitel 9

## Anhang

Text

### 9.1 Verfasser der Kapitel

Text

#### 9.1.1 David Pöchacker

Text

#### 9.1.2 Marcel Entner

Text

#### 9.1.3 Tobias Kronsteiner

Text

### 9.2 Verwendete Software

Im folgenden Abschnitt wird die bei dieser Diplomarbeit verwendete Software präsentiert.

#### 9.2.1 Visual Studio Code

Visual Studio Code ist ein von Microsoft entwickelter Quelltext Editor. Dieser Editor bietet verschiedene Programmierhilfen wie Einfärbungen oder Autovervollständigungen. Dieser Editor unterstützt standardmäßig sehr viele Programmiersprachen, jedoch können jederzeit weitere Sprachen mittels Add-ons dazu installiert

werden, um das Programmieren für den Anwender zu erleichtern. Die von diesem Projektteam verwendeten Sprachen wie HTML, CSS, PHP und JavaScript werden alle von diesem Editor standardmäßig unterstützt.

### **9.2.2 Apache WebServer**

Text

### **9.2.3 Composer**

Text

### **9.2.4 Windows Eingabeaufforderung (CMD)**

Text

### **9.2.5 Github VCS und Github Desktop GUI**

Text

### **9.2.6 phpMyAdmin**

Text

### **9.2.7 Adobe XD**

Adobe XD ist eine von Adobe Systems entwickelte Grafik-Software zum Erstellen von grafischen Benutzeroberflächen für Web-Anwendungen. Verwendbar ist dieses Tool auf mehreren Betriebssystemen wie Windows, MacOS oder Linux. Diese Software wurde dem Projektteam von der Schule bereitgestellt, da Adobe-Programme nicht kostenfrei sind. Die erstellten Design-Vorschläge für die Weboberfläche wurden mit dieser Software erstellt.

### **9.2.8 Adobe Photoshop**

Adobe Photoshop ist ein von Adobe Inc. entwickeltes Bildbearbeitungsprogramm, welches bereits weltweit sehr verbreitet und beliebt ist. Dieses Programm ist im Jahre 1990 erschienen und ist nicht lizenzfrei, aufgrund dessen wurde uns dieses Programm ebenso von der Schule zur Verfügung gestellt. Mit dieser Software wurden alle Bilder auf der Weboberfläche auf die passende Größe skaliert und bearbeitet.

## **9.3 Projektplanung**

In diesem Kapitel wird näher auf das Projektmanagement eingegangen.

### **9.3.1 Projektkommunikation**

Im Laufe der Diplomarbeit wurden zahlreiche Besprechungen mit unserem Diplomarbeitsbetreuer Herrn Johann Burgstaller gehalten, um Maßnahmen sowie weitere Vorgehensweisen abzuklären. Ebenso waren unsere Kooperationspartner Stefan Aigenbauer, Jürgen Mitterlehner, Michael Zellinger und Armin Cosic bei diesen Besprechungen dabei, um deren Anforderungen sowie Wünsche besser umsetzen zu können. Diese Besprechungen fanden Online über Skype statt, weitere Kommunikation wurde über E-Mail fortgeführt.

### **9.3.2 Projektstrukturplan**

Text

### **9.3.3 Verantwortungsmatrix und Aufwandsschätzung**

Text

### **9.3.4 Meilensteinplan**

Text

### **9.3.5 Terminplan**

Text

## **9.4 Inhalt von GitHub**

Text