DHBW Karlsruhe

Projekthandbuch

ClimateSight

Leon Fertig (1142628)
Matteo Kosina (2912404)

18. Juli 2024

Inhaltsverzeichnis

1 Einführung **1**

2 Projektvorstellung **2**

3 Protokolle **8**

4 Projektauftrag **11**

5 Projektziele **14**

6 Phasen- und Meilensteinplan **15**

7 Projektorganigramm **16**

8 Offene-Punkte-Liste OPL **17**

9 Projektstrukturplan **17**

10 Arbeitspaketbeschreibungen **19**

11 Gantt-Diagramm **20**

12 Ressourcen- und Kostenplan **21**

13 Risikoanalyse **21**

14 Stakeholderanalyse **23**

15 Projektstatusbericht **24**

16 Product Backlog und Sprint Backlog **25**

1 Einführung

Wir - das Team des Projekts **ClimateSight** - haben im Rahmen der Vorlesung *Web-Engineering I* bei Herrn Jürgen Röthig eine Webseite entwickelt und begleitend zu dieser Webseite dieses Projekthandbuch über die Planung und organisatorische Steuerung unseres Projektes erstellt. Die meisten Planungsdokumente wurden erstellt, nachdem ein Projektmitglied die Gruppe verlassen hat, sodass dieser Verlust nur geringe Auswirkungen auf das Projekt hatte.

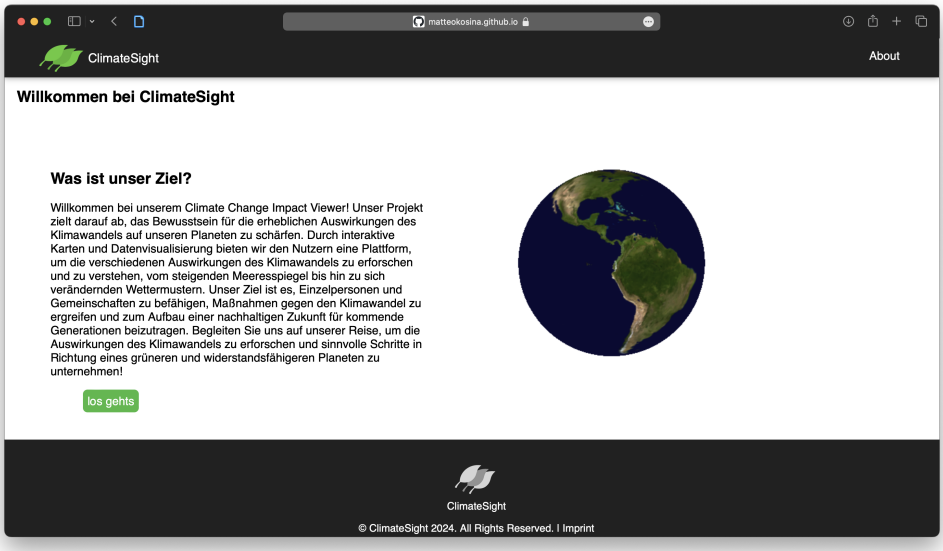
2 Projektvorstellung

Autor: Matteo Kosina

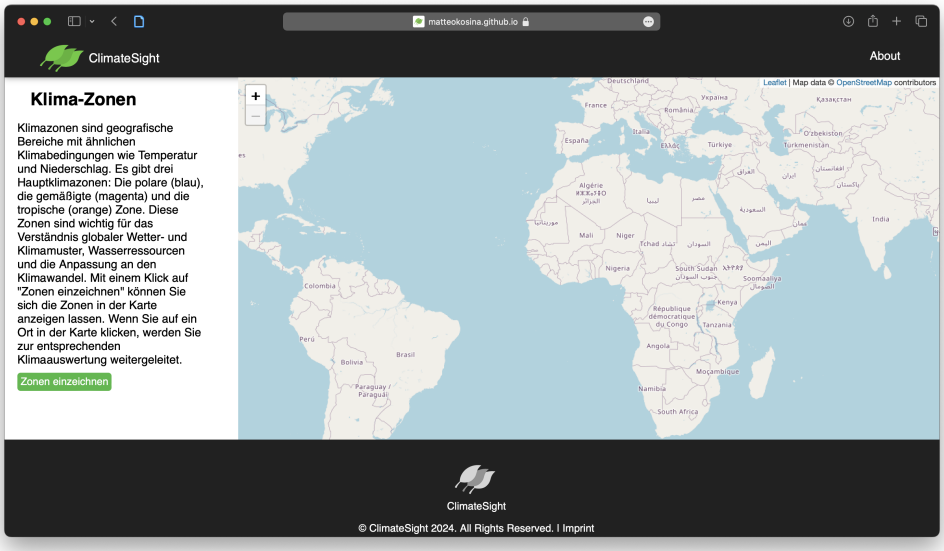
README.md 2024-07-18

ClimateSight

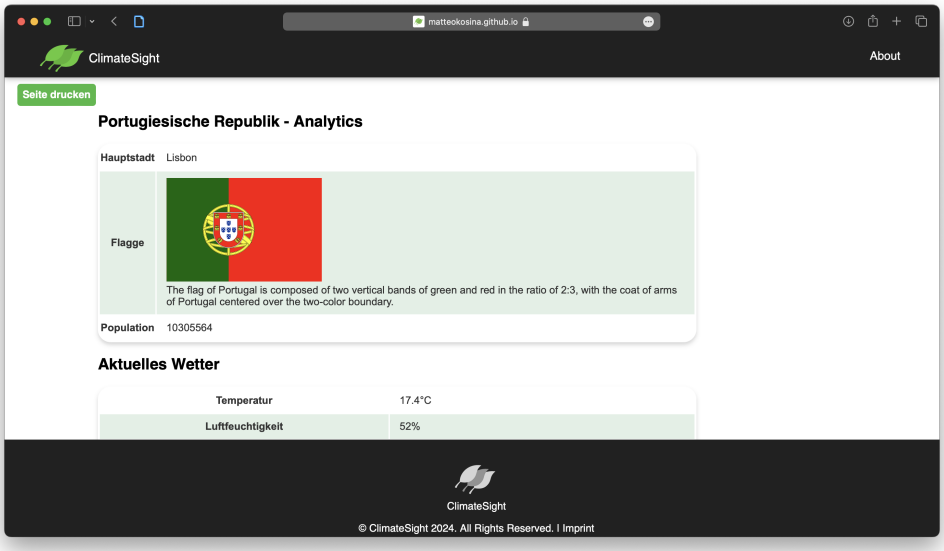
> Mit diesem Projekt wollen wir eine Möglichkeit schaffen, die Auswirkungen des Klimawandels auf eine neue, interaktive Art und Weise sichtbar zu machen. ClimateSight ist als Kartenanwendung gedacht und ermöglicht, etwas über Klimazonen zu lernen sowie auf ein beliebigen Ort auf der Welt zu klicken, um historische Wetterdaten zu betrachten.
>
> Eine Live-Demo der Website findet sich hier. Die Anwendung ist für die Nutzung an Desktop-Geräten ausgelgt und daher für dieses Bildschirmformat optimiert.
>
> Neben der Live-Demo ist es auch möglich, ClimateSight lokal zu betreiben. Eine Installationsanleitung findet sich ebenfalls in diesem README.



Startseite



Discover



Analytics

Umsetzung

Folgende Technologien wurden für die Realisierung des Projektes angewandt.

XML

Als Datenhaltungsformat dient eine XML-Struktur. Diese sind entweder statisch unter dem Pfad `./static/data/` zu finden oder werden dynamisch erzeugt. Da die von uns gewählten APIs zur Wetterdatenbereitstellung ausschließlich JSON-Formate unterstützen, haben wir mittels Javascript eine Konvertierung zu XML durchgeführt. Dabei findet das Abrufen jeder API in einer eigenen Javascript Datei statt, die die Daten in XML-Format zurückgibt. Der `Orchestrator` (orchestrator.js) ist dafür zuständig, alle Daten zusammenzuführen und ein gesamtheitliches, wohlgeformtes XML Dokument zu erstellen. Der Gedanke dahinter war, das Projekt modular aufzubauen und zukünftige Datenerweiterungen zu vereinfachen. Des Weiteren ist für jede statische aber auch für das dynamisch erzeugte XML-Dokument eine DTD erzeugt worden, somit ist nicht nur die Wohlgeformtheit, sondern auch die Gültigkeit gegeben.

XSLT

Zur Transformation in ein im Browser darstellbares Format haben wir XSLT genutzt. Dieses wird client-seitig verarbeitet. Für jede Seite wird hierfür Javascript genutzt, um XML Daten mithilfe einer XSL-Datei zu transformieren und in einen bereits vorhandenen HTML div-Element zu rendern. Bei den HTML-Dokumenten handelt es sich nur um Grundgerüste, die ausschließlich ein div-Element enthalten, in welches der komplette anzuzeigende Inhalt transformiert wird.

Karte

Für die Umsetzung der Karte haben wir auf die Bibliothek Leaflet und OpenStreetMap zurückgegriffen. Diese ermöglicht das laden einr KML-Datei (Keyhole Markup Language), um Markierungen vorzunehmen. Diese KML Datei wird wiederum mit XSLT auf Basis einer XML Datei erzeugt. Des Weiteren ist es möglich, auf einen beliebigen Punkt auf der Karte zu klicken, um auf eine Analytik-Seite zu gelangen, die historische und aktuelle Wetterdaten sowie allgemeine Informationen zu dem Ort an diesen Koordinaten bereitstellt.

Daten

Die Daten beziehen wir von zwei unterschiedlichen API-Anbietern:

- OpenMeteo (siehe hier)
- RestCountries (siehe hier)

Diese APIs kamen für uns in Frage, da sie die nötigen Daten bereitstellen und kostenfrei zu nutzen sind. OpenMeteo hat ein Rate-Limit von 10.000 API-Aufrufen pro Tag, ist aber für unseren Anwendungsfall ausreichend.

SVG

Zur Darstellung der zeitlichen Entwicklung der Daten erzeugen wir unsere eigenen SVG-Liniendiagramme mittels XSLT. Die Daten des Liniendiagramms liegen einer XML-Datei bereit, in der man neben der x- und y-Werten der Datenpunkte auch die Achsenbeschriftung und die Farbe der Linie definieren kann.

Struktur

Projektstruktur:

```
climatesight/
├── README.md
```

```
├── VERSION
├── about            : About Seite
│   ├── about.html
│   └── script.js
├── analytics        : Analitik-Seite mit Klima-/Wetterdaten
│   ├── analytics.html
│   └── script.js
├── data-orchestration        : dynamische Datenbeschaffung und XML-
Erzeugun
│   ├── climatesight.dtd
│   ├── currentWeather.js
│   ├── facts.js
│   ├── historicalWeather.js
│   └── orchestrator.js
├── imprint        : Impressum
│   ├── imprint.html
│   └── script.js
├── map        : Discover Seite (Kartenanwendung)
│   ├── discover.html
│   └── script.js
├── resources        : externe Resourcen für die Karte und den Globus auf
der Startseite
│   ├── leaflet.css
│   ├── leaflet.js
│   └── three.js
├── index.html        : Startseite
├── script.js
├── static        : Daten
│   ├── assets
│   │   ├── cookie.png
│   │   ├── favicon.ico
│   │   ├── logo-bw.png
│   │   ├── logo.png
│   │   └── texture.png
│   └── data
│       ├── about.dtd
│       ├── about.xml
│       ├── analytics.dtd
│       ├── analytics.xml
│       ├── discover.dtd
│       ├── discover.xml
│       ├── imprint.dtd
│       ├── imprint.xml
│       ├── index.dtd
│       ├── index.xml
│       ├── zones.dtd
│       └── zones.xml
├── styles        : Stylesheets
│   └── style.css
├── transformations        : XSL Dateien zur Transformation
│   ├── about.xsl
│   ├── analytics-data.xsl
│   ├── analytics.xsl
│   ├── discover.xsl
```

```
│   ├── imprint.xsl
│   ├── index.xsl
│   └── kml.xsl
└── webserver.py        : Python Webserver für lokale Inbetriebnahme
```

Features

Folgende Features umfasst ClimateSight:

- interaktive Karte
- Möglichkeit, die Klimazonen in der Karte einzeichnenzulassen
- mit dem Klick auf ein Land wird man zur Analytik Seite weitergeleitet, die Daten zum geklickten Punkt darstellt
- graphische Darstellung der Daten in Form von dynamisch generierten SVG-Liniendiagrammen
- Bereitstellung historischer und aktueller Wetter-/ und Klimadaten, sowie Informationen wie die Hauptstadt, Population und Flagge des ausgewählten Standortes
- Möglichkeit die Analytik Seite auszudrucken (hierfür werden bestimmte Teile der Seite ausgeblendet)
- About- und Impressum Seite
- Beachtung von Cookie-Regulierungen (Nutzer muss Cookies akzeptieren, bevor Daten von externen Quellen geladen werden. Will man eine Unterseite erreichen, ohne zuvor den Cookies zugestimmt zu haben, wird man auf die Startseite weitergeleitet)

lokale Installation

Für die lokale Inbetriebnahme genügt es, den im Projekt enthaltenen Python-Webserver mit folgendem Befehl zu starten:

```
python webserver.py
```

bzw.

```
python3 webserver.py
```

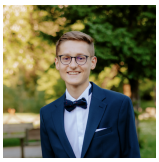
Alternativ lässt sich auch jeder andere Webserver (z.B. Apache) nutzen, solange dieser die Grundfunktionen unterstützt (HTML, Javascript, XSLT).

Browser Unterstützung

Die Website wurde für gängige Webbrowser getestet (Chrome, Firefox und Safari) und war auf allen funktionsfähig. Lediglich bei Firefox wurden die SVG-Liniendiagramme eingeschränkt dargestellt, jedoch noch in einem Rahmen, der die Verständlichkeit der dargestellten Daten nicht drastisch beeinflusst.

Contributor

Leon Fertig



Matteo Kosina

3 Protokolle

Autor: Leon Fertig

Für die Protokollierung ist eine Vorlage entworfen worden, nach der die Treffen jeweils protokolliert worden sind. Zwei Beispiele finden sich nachfolgend.

Protokoll „Kick-Off-Meeting“ vom 15.05.2024

Zeitraum:	08:30-09:00 Uhr
Ort:	DHBW Karlsruhe
Teilnehmer:	Leon Fertig, Matteo Kosina, Marius Kurth
Thema:	Kick-Off
Tagesordnung:	1. Begrüßung im Projekt 2. Beginn der Planungstätigkeit 3. Aufteilung von Aufgaben

Beschlüsse:

- Github-Strategie: Dev-, Feature-branches etc.
- Aufteilung des Teams:
  Projektmanager: Leon
  Developer: Marius, Matteo
- Logo-Erstellung: Marius
- Genereller Aufbau:
  - Strikte Trennung Backend-Frontend
  - Backend: Go
  - Frontend: XSLT -> HTML
  - Datenhaltung: XML
- Zeitplan:
  - Programmierende: 21.06.2024
  - Abschluss: 26.06.2024

Wichtige Fragen: Röthig:

- Abgabedatum Projekt?
- Frontend auch direkt in HTML (ohne XSLT) erlaubt?

Aufgaben: bis zum nächsten Meeting (24.05.2024):

- Proof of Concept (POC) der APIs (OpenWeatherMap, OpenStreetMap, OpenMeteo) -> Leon Fertig:
  - Welche Daten liefern sie?
  - Sind sie kostenlos nutzbar (wenn ja, in welchem Umfang)?
  - Sind sie passend für unsere Projektidee?
- Design-Entwurf + Entwurf des User-Flows -> Matteo Kosina

Protokollführer: Leon Fertig 24.05.2024

Protokoll „Planungsmeeting“ vom 24.05.2024

Zeitraum: 13:00-15:00 Uhr

Ort: DHBW Karlsruhe

Teilnehmer: Leon Fertig, Matteo Kosina, Marius Kurth

Thema: Planung des Projektes

Tagesordnung:

- kurze Besprechung, was die APIs (OpenWeatherMap, OpenStreetMap, OpenMeteo) leisten können
- Planung des weiteren Vorgehens
- "endgültige" Konkretisierung der Projektziele und des Umfangs des Projekts
- öffentliches Hosten des Projektes (Limitations)

Beschlüsse:

- Kein Geld für API-Keys ausgeben (kostenlose Varianten nutzen)

Themen:

APIs

- OpenStreetMap für Karte verwenden
- OpenMeteo für historische Wetterdaten verwenden
- OpenWeatherMap für aktuelle Temperatur
- Für weitere APIs: Recherche notwendig bzw. je nach Notwendigkeit hinzufügen

Öffentliches Hosten

- GitHub-Pages (alles muss im Frontend gemacht werden)
- Privat (auf Raspberry Pi -> dann läuft’s auch mit Backend)
  ➔ Entscheidung für GitHub-Pages (alle Berechnungen im Frontend!)
- Cookie-Banner(!), da öffentlich

General

- Review, wenn von dev-branch auf main-branch gepusht werden soll

Aufgaben:

Matteo: grobe Architektur-Skizze
Leon: API für Gletscher-Volumen
Marius: API für Ozon/CO_2-Gehalt
Matteo: API für grüne Energieversorgung
Matteo: API für Extremwetter
Leon: Recherche (Datenschutz, Cookie, Impressum)
(alles bis 28.05.2024)

4 Projektauftrag

Autor: Leon Fertig

Der Projektauftrag geht aus den Anforderungen des Auftraggebers Herrn Röthig hervor und gibt wieder, was der Auftraggeber von uns erwartet. Das Projektgesamtziel, die Teilziele und die Rahmenbedingungen und der Projektkontext sind direkt aus den Anforderungen von Herrn Röthig zitiert.

PROJEKTAUFTRAG

Projektname:	ClimateSight
Projekttyp:	Abschlussprojekt des Moduls Web Engineering
Projektleiter:	Leon Fertig
Projektauftraggeber:	Prof. Dr. Jürgen Röthig
Projektzeitraum:	Geplanter Beginn: 15.05.2024 Geplantes Ende: 21.07.2024
Ausgangslage:	Entwicklung einer Anwendung mit Hilfe der in der Vorlesung erlernten Techniken zur Webentwicklung als Abschlussprojekt zur Vorlesung.
Projektgesamtziel:	Interaktive Kartenanwendung Aufgabe ist, aus als Geokoordinaten vorliegenden Kartendaten (Laendergrenzen, Strassen, Fluesse, Kuesten, ...) eine graphische Darstellung im Web zu erstellen. Die Ausgangsdaten koennen aus OpenStreetMap oder anderen frei zugaenglichen Datenquellen gewonnen werden, die Zieldarstellung ist in der Sprache SVG zu realisieren. Mindestens eine interaktive Komponente muss enthalten sein, beispielsweise die Angabe von Kenndaten (Name, Lage, Einwohnerzahl, Bedeutung), wenn man auf ein Land klickt.
Teilziele:	eigene konkrete und genaue Zielaufgabenstellung selbst erstellen (!) Sichtung und Auswahl der geeigneten Datenquelle(n) Reduzierung des Umfangs der Geokoordinaten auf eine im Web (fuer eine SVG-Datei) praktikable Groesse unter Verringerung der Aufloesung Definition eines geeigneten XML-basierten Datenformats zur Speicherung der relevanten Daten Transformation der XML-Ausgangsdaten mit Hilfe von XSLT in eine SVG-Datei interaktive Veraenderung der SVG-Datei mit JavaScript Anforderung nachzuladender Teile vom Server und Einbau in die bestehende SVG-Datei (AJAX)

Rahmenbedingungen und Projektkontext:	Datenhaltungsformat: XML/XML-aehnlich, alternativ RDB Datentransferformat: XML Definition des Datenformats: DTD Eingabe der Daten: HTML-Formulare, JavaScript, einfache serverseitige Auswertung und Abspeicherung in beliebiger webserver-faehiger Programmiersprache (PHP, Perl, bash, ...). Ausgabe der Daten/Transfomation in darstellbare Form: mittels XSLT nach HTML und/oder (alternativ) SVG
Organisation:	**Kernteam**: • Leon Fertig • Matteo Kosina **Sonstige Beteiligte**: • Marius Kurth (nur Recherche)
Budget:	0€
Nutzen:	Abschlussprojekt zur Vorlesung Web Engineering
Unterschrift Auftraggeber:	
Unterschrift Projektleitung:	

5 Projektziele

Autor: Matteo Kosina

Muss-Kriterien

- Es werden verschiedene Datenquelle (APIs) für Klimadaten angeschaut und analysiert, welche nützlich für das Projekt sein können. Dabei muss beachtet werden, dass kein Budget für API-Keys zur Verfügung steht, die in Frage kommenden Datenquellen müssen also kostenlos sein.
- Es wird eine Web-Applikation für Desktop-PCs entwickelt mit folgenden Komponenten:
  - Eine Landing-Page, auf der ein kurzer Text mit Beschreibungen zu dem Projekt zu finden sind und ein Button, der den User auf die Discover-Page weiterleitet
  - Eine Discover-Page, auf der eine Karte zu sehen ist. Beim Klick auf ein Land wird der User auf die Analytics-Page des entsprechenden Landes weitergeleitet.
  - Eine Analytics-Page für jedes Land der Erde, die die Hauptstadt, die Flagge und die aktuelle Einwohnerzahl des Landes anzeigt. Außerdem zeigt diese Seite die aktuelle Temperatur und historische Wetterdaten für Temperatur und Niederschlag an.
- Es wird eine Installationsanleitung erstellt, die es einem Administrator erlaubt, den Webserver bei sich zu betreiben. Eine Demo-Installation mit einem einfachen Python-Webserver wird Schritt für Schritt beschrieben. Für die produktive Nutzung werden jedoch einige Kenntnisse benötigt, die die Installationsanleitung nicht bereitstellt.
- Für die Datenspeicherung wird XML verwendet (API-Daten müssen gegebenenfalls von JSON in XML umgewandelt werden). Mithilfe von XSLT werden diese Daten in HTML oder SVG umgewandelt. Hierfür werden geeignete Datenformate definiert.

Soll-Kriterien

- Die Sprache kann zwischen Deutsch und Englisch gewechselt werden.
- Auf der Analytics-Page werden lokale Nachrichten (mit Fokus auf Klima + Wetter) angezeigt.
- Auf der Discover-Page kann man mit einem Klimafilter (nach Temperatur und/oder Niederschlag) die Länder gruppieren, um eine bessere Übersichtlichkeit zu erreichen.

Kann-Kriterien

- Der User kann zwischen verschiedenen Kartenansichten auswählen: Satellit, geographisch, Heat-Map

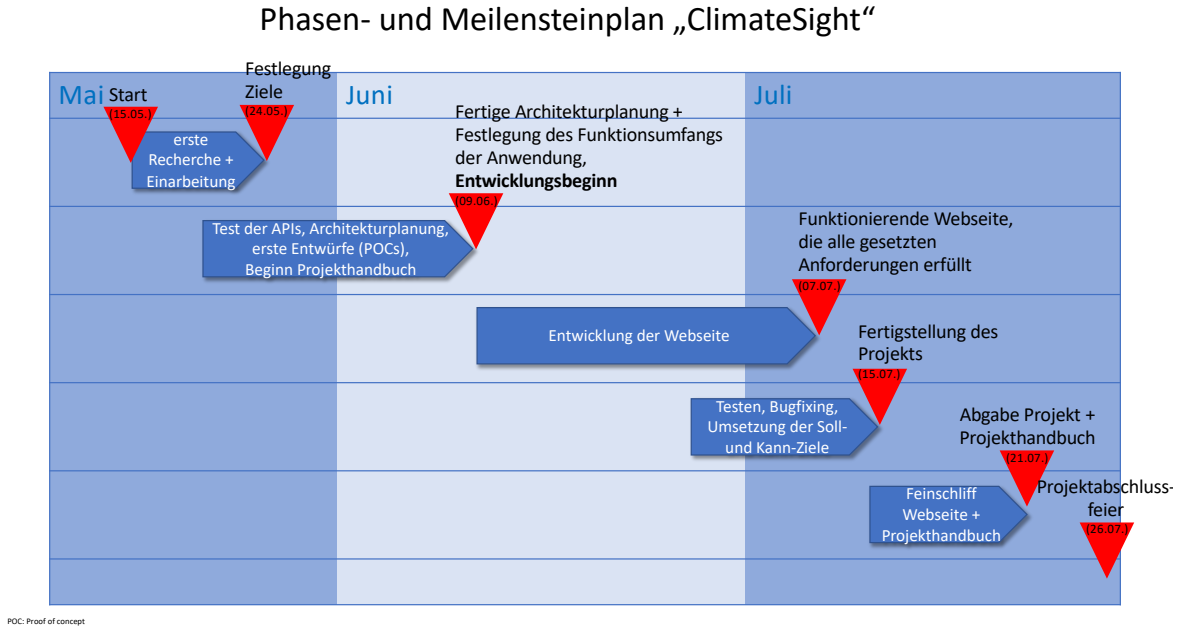
Abgrenzungskriterien

- kein Einsatz von KI
- keine Routenführung
- keine mobile Anwendung (nicht für Smartphone oder Tablets optimiert)

6 Phasen- und Meilensteinplan

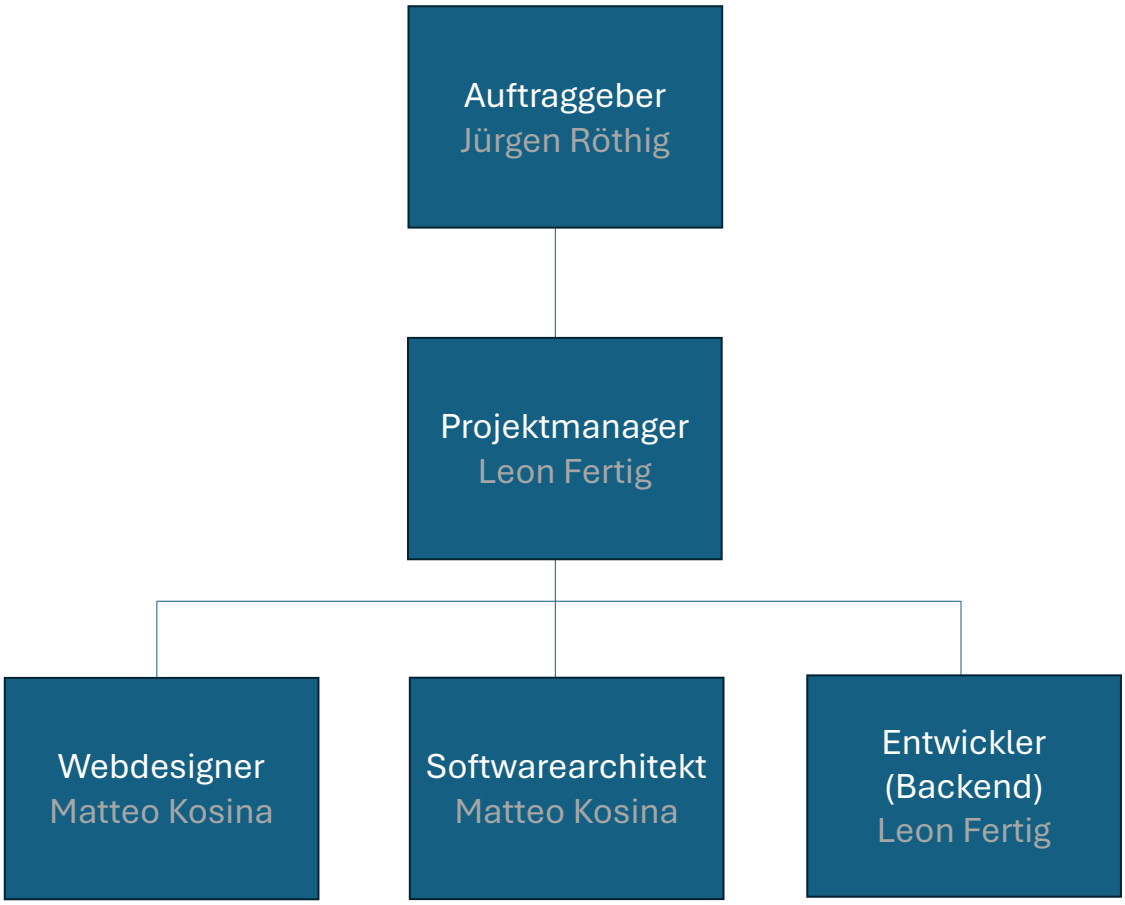
Autor: Leon Fertig

Der Phasen- und Meilensteinplan enthält **nicht** die Arbeitspakete aus Projektstrukturplan, Gantt-Diagramm etc., da die Arbeitspakete - der geringen Größe des Projekts geschuldet - relativ viele kleineren Aufgaben enthalten und daher über relativ große Zeiträume reichen, sodass ein übersichtlicher Phasen- und Meilensteinplan nicht möglich wäre. Daher wurde der Phasen- und Meilensteinplan als grobe Übersicht erstellt und das Gantt-Diagramm wurde zur Überprüfung des Zeitplans eingesetzt.



7 Projektorganigramm

Autor: Matteo Kosina



8 Offene-Punkte-Liste OPL

Autor: Leon Fertig

Offene Punkteliste zum 04.06.2024. Die Aufgaben aus der OPL stimmen nicht immer zu 100% mit denen der Protokolle überein, da sie teilweise nachträglich noch geändert (oder die Termine nach hinten verschoben) werden mussten, sodass die Protokolle keine aktuelle Version der Aufgaben darstellen, sondern **nur** die OPL!

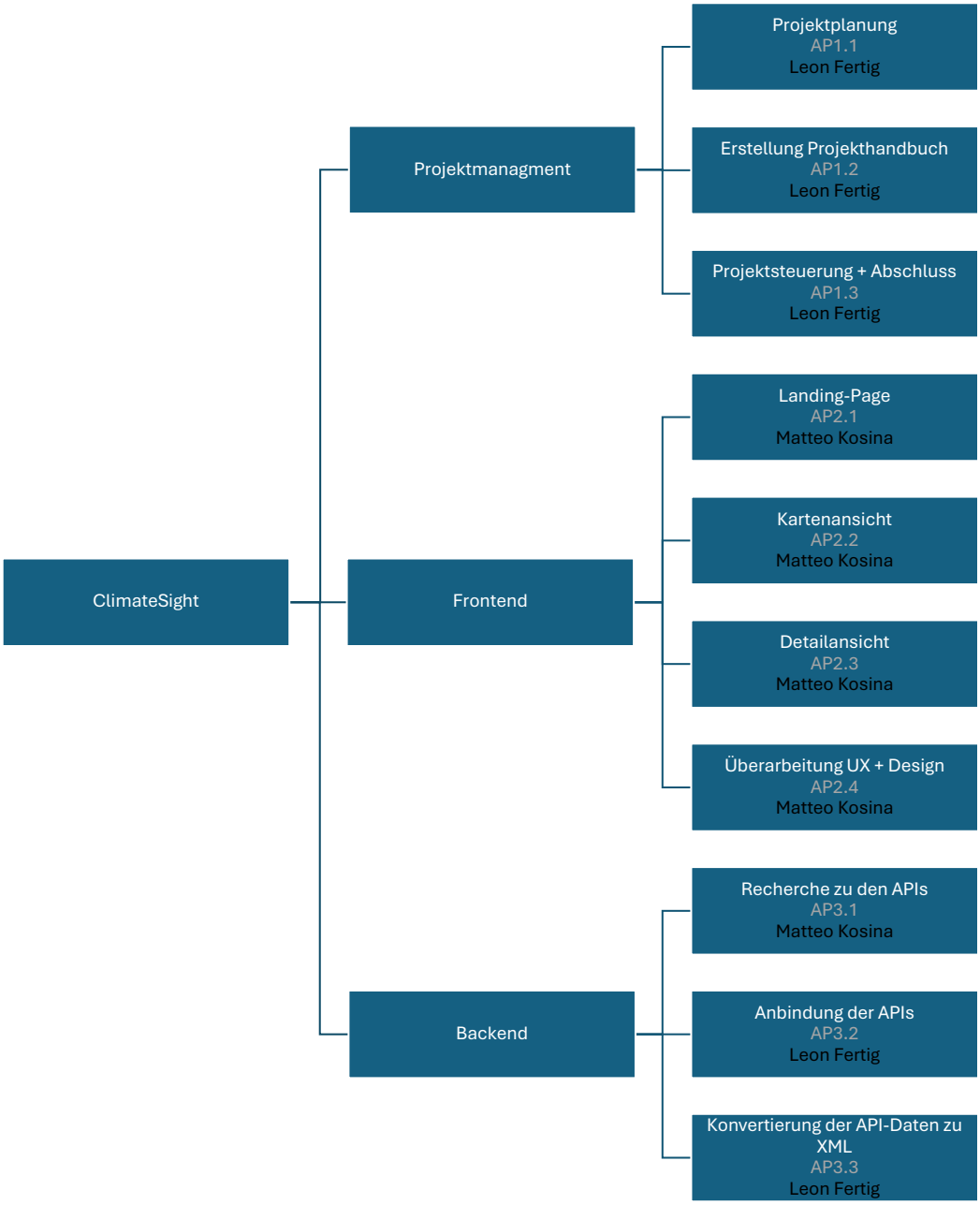
Nr.	Aktion	Verantwortlich	zu erledigen bis	Kommentar / Status	erledigt am	Typ	Status
1	**Kick-Off-Meeting**						
1.1	**POC der APIs**	Leon Fertig	24.05.24	Details s. Protokoll 15.05.2024		I	erl
1.2	**Design-Entwurf + Entwurf des User-Flows**	Matteo Kosina	09.06.24			W	erl
1.3	**Fragen an Röthig stellen**	Leon Fertig	27.05.24	Fragen s. Protokoll 15.05.2024		I	erl
2	**Planungsmeeting**						
2.1	**Recherche zu weiteren APIs**	Marius Kurth	01.06.24	Details s. Protokoll 24.05.2024		W	erl
2.2	**Architekturplanung (1)**	Leon Fertig	09.06.24			A	erl
2.2	**Architekturplanung (2)**	Matteo Kosina	09.06.24			A	erl
2.3	**Entscheidung, welche APIs verwendet werden**	Leon Fertig	09.06.24			B	ü
2.4	**Erstellung Projekthandbuch-Template**	Matteo Kosina	01.07.24			W	iB
3	**Start Entwicklung**						
3.1	**Implementierung Landing-Page**	Matteo Kosina	07.07.24				iB

9 Projektstrukturplan

Autor: Matteo Kosina

Es wurde eine Gemischter Projektstrukturplan erstellt:

- **Projektmanagement**: Mischung aus Objektorientierung und Phasenorientierung
- **Frontend**: Objektorientierung (wobei die Überarbeitung von UX + Design eher einem phasenorientierten Ansatz entspricht)
- **Backend**: phasenorientiert


ClimateSight
Projektmanagment
Projektplanung
AP1.1
Leon Fertig
Erstellung Projekthandbuch
AP1.2
Leon Fertig
Projektsteuerung + Abschluss
AP1.3
Leon Fertig
Frontend
Landing-Page
AP2.1
Matteo Kosina
Kartenansicht
AP2.2
Matteo Kosina
Detailansicht
AP2.3
Matteo Kosina
Überarbeitung UX + Design
AP2.4
Matteo Kosina
Backend
Recherche zu den APIs
AP3.1
Matteo Kosina
Anbindung der APIs
AP3.2
Leon Fertig
Konvertierung der API-Daten zu XML
AP3.3
Leon Fertig

10 Arbeitspaketbeschreibungen

Autor: jeweils AP-Verantwortlicher

Arbeitspaket: Landing-Page	**ID: AP2.1**

Ziele:	• Erstellen einer Startseite mit folgenden Eigenschaften:
	  ○ Auswahl zwischen Englisch und Deutsch
	  ○ Kurze Erklärung des Projektes
	  ○ Möglichkeit (Link/Button), um zur Kartenansicht zu gelangen
	  ○ Cookie-Banner mit Speicherung im local Cache
	  ○ Footer-Leiste mit Links: *Impressum* und *About us*

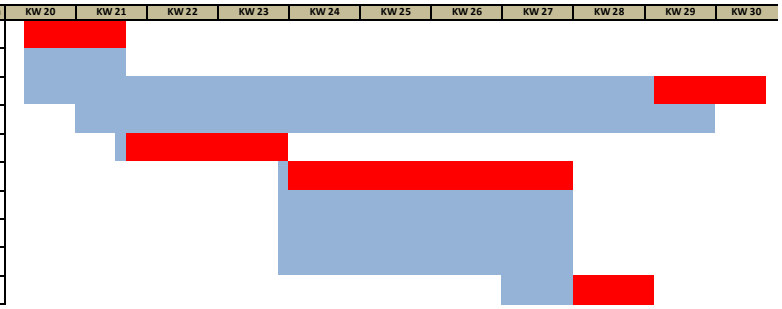
Verantwortung:	Matteo Kosina
Mitarbeit:	/
Aufwand (Ressourcen):	Zeit: 1h (Konzeption & Architektur) + 6h (Implementierung) = 7h Geld: 7h * 45€/h = 315€ Sonstiges: Laptop + Github Personen: 1
Abhängige Arbeitspakete:	AP1.1 (v. a. Ziele bezüglich Accessability & Design der Startseite)
Aufgaben:	• Konzeption + Architekturplanung • Implementierung Cookie-Banner • Formulierung einer Projektbeschreibung • Implementierung Startseite • Implementierung Footer + Links
Voraussetzung:	Ziele sind definiert, v. a. bez. Design und UX der Startseite
Zeitraum:	09.06.-07.07.
Abgenommen:	

Arbeitspaket: Konvertierung der API-Daten zu XML	**ID: AP3.3**
Ziele:	• Verarbeitung der API-Daten (größtenteils im JSON-Format) zu XML • Filtern der Daten-Attribute, sodass nur die in der Planung beschriebenen an das Frontend geliefert werden (s. AP1.1) • Keine Änderung der Daten
Verantwortung:	Leon Fertig
Mitarbeit:	Matteo Kosina (beratend)
Aufwand (Ressourcen):	Zeit: 0,5h (Planung) + 4h (Implementierung) = 4,5h Geld: 4,5h * 45€/h = 202,5€ Personen: 1 (+ 1 beratend) Sonstiges: Laptop + Github
Abhängige Arbeitspakete:	AP1.1, AP3.1, AP3.2
Aufgaben:	• Filtern der Daten • Umwandeln in XML-Format
Voraussetzung:	Funktionalität, die die Daten von der API abfragt, ist implementiert und liefert Daten (echt oder MOC) zum Testen
Zeitraum:	09.06.-07.07.
Abgenommen:	

11 Gantt-Diagramm

Autor: Leon Fertig

ID	Arbeitspaket	Verantwortlich	Startdatum	Enddatum
AP1.1	**Projektplanung**	Leon Fertig	15.05.24	24.05.24
AP3.1	**Recherche zu den APIs**	Matteo Kosina	15.05.24	24.05.24
AP1.3	**Projektsteuerung + Abschluss**	Leon Fertig	15.05.24	26.07.24
AP1.2	**Erstellung Projekthandbuch**	Leon Fertig	20.05.24	21.07.24
AP3.2	**Anbindung der APIs**	Leon Fertig	24.05.24	09.06.24
AP2.1	**Landing-Page**	Matteo Kosina	09.06.24	07.07.24
AP2.2	**Kartenansicht**	Matteo Kosina	09.06.24	07.07.24
AP2.3	**Detailansicht**	Matteo Kosina	09.06.24	07.07.24
AP3.3	**Konvertierung der API-Daten zu XML**	Leon Fertig	09.06.24	07.07.24
AP2.4	**Überarbeitung UX + Design**	Matteo Kosina	01.07.24	15.07.24



12 Ressourcen- und Kostenplan

Autor: Matteo Kosina

Ressourcen- und Kostenplan

ID	Arbeitspaket	Aufwand (in h)	Kosten (45€/h)	Eingeplante Mitarbeiter
AP1.1	Projektplanung	14	630 €	L. Fertig, M. Kosina
AP1.2	Erstellung Projekthandbuch	30	1.350 €	L. Fertig
AP1.3	Projektsteuerung + Abschluss	4	180 €	L. Fertig
AP2.1	Landing-Page	7	315 €	M. Kosina
AP2.2	Kartenansicht	9	405 €	M. Kosina
AP2.3	Detailansicht	15	675 €	M. Kosina
AP2.4	Überarbeitung UX + Design	8	360 €	L. Fertig, M. Kosina
AP3.1	Recherche zu den APIs	3	135 €	L. Fertig, M. Kosina
AP3.2	Anbindung der APIs	4	180 €	L. Fertig
AP3.3	Konvertierung der API-Daten zu XML	4,5	203 €	L. Fertig, M. Kosina
	Gesamt	98,5	4.433 €	

13 Risikoanalyse

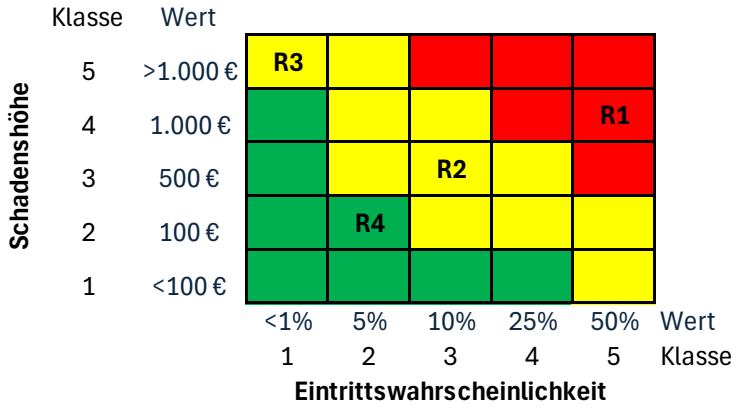
Autor: Matteo Kosina

Es wurde die qualitative Risikobehandlung gewählt, da sie einen besseren Überblick über die Gefahren/Auswirkungen der einzelnen Risiken geben. Außerdem fehlt dem Team die Erfahrung für akkurate Geldwert-Einschätzungen, weshalb der Fokus nicht so stark auf Geldbeträge gelegt werden soll.

Risikoanalyse

ID	Risiko	Ursache	Eintritts-klasse	Schadens-klasse	Risiko-Index	Gegen-maßnahme	Kosten der Maßnahme	Stakeholder
R1	Änderung der Anforderungen	Hr. Röthig erwähnt neue Anforderungen in Vorlesung	5 (50%)	4 (1 T€)	20	Fast nicht möglich; Einplanung eines Zeit-Puffers		Jürgen Röthig
R2	Verzögerungen im Zeitplan	Technische Schwierigkeiten mit XSLT	3 (10%)	3 (500€)	9	Sorgfältige Recherche zu allen eingesetzten Technologien; ordentliche Überwachung des Zeitplans	~3h => 135€	
R3	Ausfall eines weiteren Projektmitglieds	Abbruch des Studiums	1 (<1%)	5 (>1 T€)	5	Nicht möglich		
R4	Höherer Zeitaufwand durch Projektmanagment als erwartet	Fehleinschätzung des Projektmanagment-Aufwandes	2 (5%)	2 (100€)	4	Sorgfältige Planung der Projektmanagment-Aufgaben und ordentliche Berücksichtigung im Zeitplan		

Risikomatrix



14 Stakeholderanalyse

Autor: Matteo Kosina

Stakeholderanalyse

Stakeholder	Position	Klima/ Stimmung	Bedeutung/ Macht	Erwartungen/ Befürchtungen	Erwartungs-haltung	Maßnahmen
Leon Fertig	Projekt-leitung	++	3	+	Projektleitung, gute Arbeit der Mitarbeiter	Gute Arbeit aller Mitarbeiter
Matteo Kosina	Projekt-Mitarbeiter	++	3	++	Gute Projektplanung, gute Note in Web-Engineering und Projektmanag-ment	Gute Arbeit aller Mitarbeiter
Jürgen Röthig	Auftrag-geber	0	2	0	Teilweise unklar, möchte ein Projekt mit XSLT und in einem seiner Themenbereiche, den Rest lässt er offen	Regelmäßige Besprechungen über Projektstand

15 Projektstatusbericht

Autor: Leon Fertig

Projekt ClimateSight	**STATUSBERICHT 1 (SB1)**	**06.07.24**
Projektbeginn: 15.05.2024 Geplantes Ende: 21.07.2024 Projektleiter: Leon Fertig	Berichtszeitraum: 15.05.2024 – 01.07.2024	Anlass: Zwischenbericht

	Zeit	Kosten	Qualität
Status			
Trend	→	→	→

Im Berichtszeitraum erwartete Ergebnisse (was hätte erarbeitet werden sollen?)	erfüllt	nicht erfüllt
Kick-Off (T1)	☒	☐
Einarbeitung (T2.1)	☒	☐
Zieldefinition (T3.1)	☒	☐
Test der APIs (T2.1)	☒	☐
POCs (T2.3)	☒	☐
Architekturplanung (T4)	☒	☐
Projektplanung (T3.3)	☒	☐

Im Berichtszeitraum angefangene Themen (was sollte in Arbeit sein?)	Status
Projekthandbuch (T4)	
Entwicklung der Webseite (T5.1)	

Begründung von Abweichungen
Keine Abweichungen, alles im Zeitplan

Risiken/Proleme/Maßnahmen
/

Ausstehende Entscheidungen
/

Spezielle Anträge:
Aussteigen eines Projektmitglieds (Marius Kurth), aber Bekanntgabe des Ausstiegs schon während der Zieldefinition, daher ist eine rechtzeitige Anpassung der Planung möglich gewesen und kein Zeitverzug entstanden

Weiteres Vorgehen	Zuständig
Projekthandbuch (T4)	Leon Fertig
Entwicklung der Webseite (T5.1)	Matteo Kosina
Beginn von Umsetzung der Soll- und Kann-Ziele (T5.2)	Leon Fertig, Matteo Kosina

Sonstige Bemerkungen
/

Berichtverfasser:	Empfänger:
Leon Fertig (Projektleiter)	Jürgen Röthig (Auftraggeber)

Berichtverfasser (Datum, Unterschrift)

Bericht abgenommen
Empfänger (Datum, Unterschrift)

16 Product Backlog und Sprint Backlog

Autor: Leon Fertig

Product Backlog

ID	Epic	User Story	Priorität	Story Points
PB1	Filter	Als Anwender möchte ich die Länder nach ihrer Temperatur, Regenfall o. Ä. filtern können, weil mir das einen besseren Überblick gibt.	1	20
PB2	Statistiken	Als Host der Webseite möchte ich Statistiken generiert bekommen, die mir anzeigen, wie viele Anwender wie oft welche Seite besucht haben, weil das nützlich für die Verbesserung und Erweiterung der Seite ist.	2	30
PB3	Cookie-Banner	Als Projektleiter möchte ich, dass der Benutzer beim erstmaligen Betreten der Webseite ein Cookie-Banner angezeigt bekommt, weil das gesetzlich vorgeschrieben ist.	0	10
PB4	CO_2-Ausstoß-Anzeige		3	?

Sprint Backlog

ID	User Story	Tasks	Verantwortlich	Status
PB2	Als Host der Webseite möchte ich Statistiken generiert bekommen, die mir anzeigen, wie viele Anwender wie oft welche Seite besucht haben, weil das nützlich für die Verbesserung und Erweiterung der Seite ist.	Counter auf allen Seiten implementieren	Leon Fertig	fertig
		Daten speichern und für Frontend bereitstellen	Leon Fertig	i. A.
		Neue Admin-Seite zum Anzeigen der Daten entwerfen und an Daten anbinden	Matteo Kosina	fertig
PB3	Als Projektleiter möchte ich, dass der Benutzer beim erstmaligen Betreten der Webseite ein Cookie-Banner angezeigt bekommt, weil das gesetzlich vorgeschrieben ist.	Cookie-Banner funktional implementieren	Leon Fertig	fertig
		Cookie-Banner an Style der Webseite anpassen	Matteo Kosina	i. A.