Projekt Antrag

Team Mitglieder

MITGLIED	ORGAN
SIMEON STIX	Entwicklung

Kurzbeschreibung

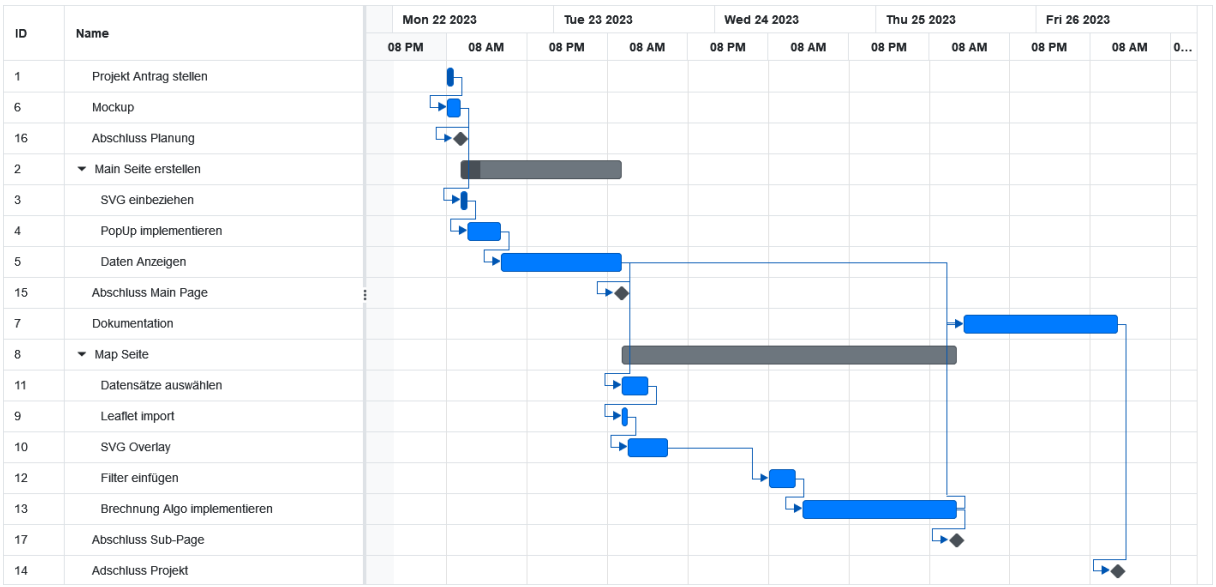
Ziel der Applikation ist es die Karte das Kanton Thurgaus, aufgeteilt in die Gemeinden, darzustellen und den entsprechenden Datensatz aus der Open Government Data Bibliothek den Gemeinden zuzuordnen und grafisch darzustellen. Dies wird erreicht, indem die entsprechenden Datensätze aus der Bibliothek per JSON ausgelesen werden (per fetchAPI angefragt) und danach pro Gemeinde verarbeitet. Dies zeigt sich in der Form, dass hinter dem Bild des Kanton Thurgau eine Funktion verborgen ist die über einen Popup, welches sein Erscheinungsbild bei einem Klick auf die entsprechende Gemeinde zeigt. Dieser Klick wird im Hintergrund verarbeitet. Dies bedeutet das die Daten aus der entsprechenden Datenbank (SQLite) gelesen, an die entsprechenden Stellen des HTML-Strukturcodierung eingefügt werden. Hierbei ließe sich auch noch ein detailliertes Diagramm der erneuerbaren Energiemischung der einzelnen Gemeinden darstellen. Im allgemeinem wird dieses Erscheinungsbild mithilfe von einem „*useState*“ geändert. Eine weiter eventuelle Erweiterung dieses Projekt wäre eine Farbcodierung der Karte, durch welche man die totale prozentuale Nutzung von erneuerbarem Energieträger im absoluten Energiemix darstellt werden und einen Slider mit welcher man die Karte durch die einzelnen Jahre der Datensätze bewegen werden kann.
Im Falle, dass die Applikation vor dem Abschluss der Entwicklungsdauer mit allen Erweiterungen zur Vollendung gebracht wurde und nach dem Ermessen des Entwicklers noch genügend Zeit zur Verfügung steht, wird eine zusätzliche unabhängige Funktionalität hinzugefügt. Hierbei handelt es sich um eine weitere Seite bei der über eine dynamische Karte mehrere Datensätze der Open Government Data Bibliothek dargestellt und zusammengefügt werden. Hierbei ist das Anwendungsziel die Suche der perfekten Gemeinde. Um dieses Ziel zu erreichen kann der Nutzer mithilfe mehreren Filtern Datensätze berücksichtigen und Eingrenzungsoptionen auswählen, welche ebenfalls wenn möglich grafisch in der dynamischen Karte angezeigt werden.
Implementierungsmöglichkeiten für eine solche dynamische Karte wird von dem ukrainischen Kartenframework Leaflet zur Verfügung gestellt.

Die gesamte Applikation wird, wie ersichtlich, mit „ReactJS“ aufgebaut. Da jedoch ein Backend gefordert wird, wird das Framework „NextJS“ zum Einsatz kommen. Dementsprechend wird es auch keine dezidierte Rest API entwickelt und/oder zur Verfügung gestellt.

Use Cases

USE CASE	BESCHEIBUNG	ERFÜLLT (PROBLEME)
1	Die Seite wird durch den Nutzer unter der Stamm-URL aufgerufen.	
2	Die Seite wird durch den Nutzer unter irgendeiner Sub-URL aufgerufen.	
3	Die Karte (Bild des Kantons Thurgau) wird angeklickt.	
4	Auf der Sub-Seite „Map“ wählt der Nutzer aus welche Datensätze für ihn interessant ist und kann diesen noch konfigurieren. Bsp.: kann eine maximal/minimal Distanz zum nächsten Polizeiposten einstellen	

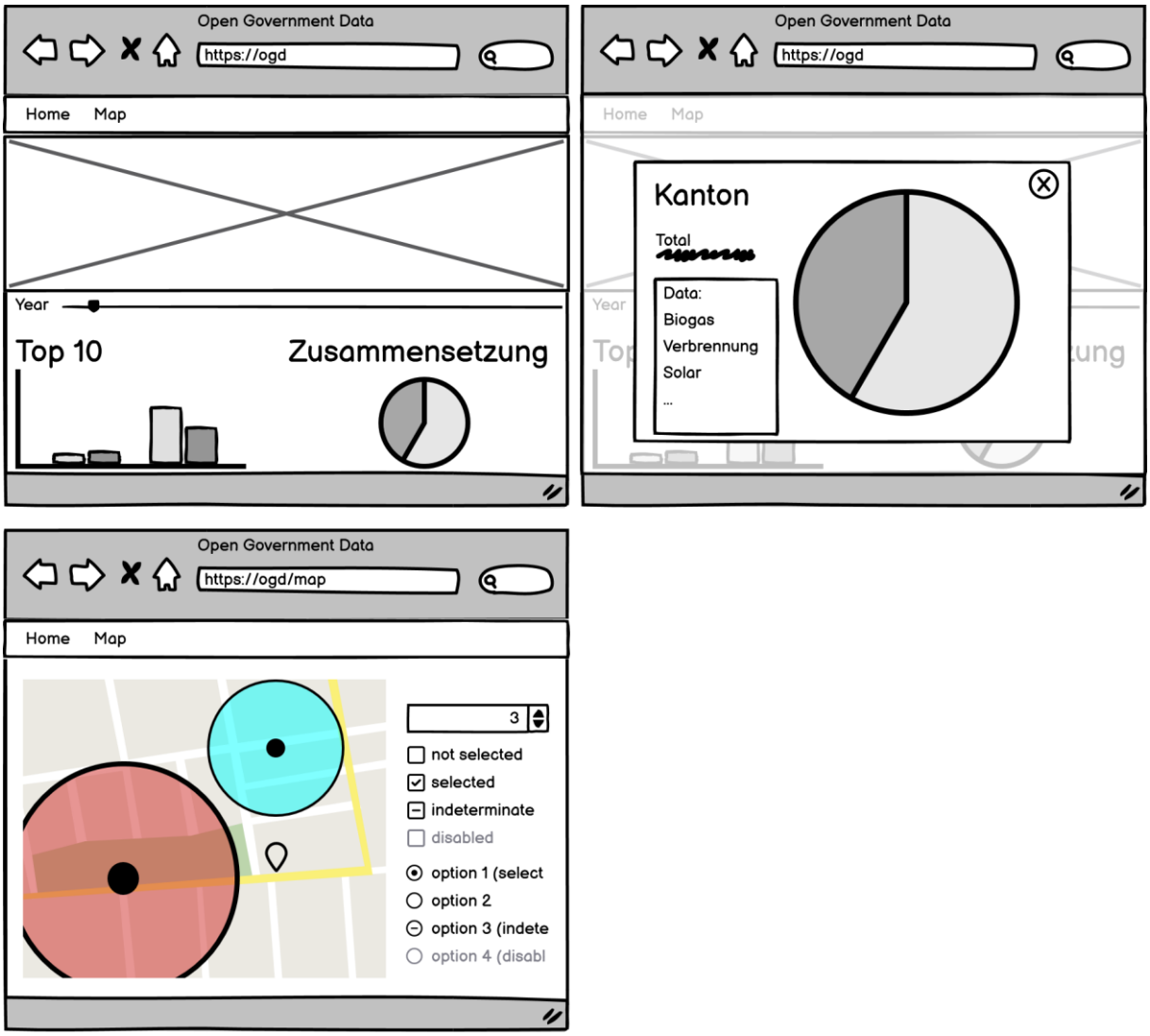
Zeitplanung



AUFGABE	ZEIT
PROJEKT ANTRAG STELLEN	1 Stunde
MOCKUP	2 Stunden
ABSCHLUSS PLANNUNG	Meilenstein
MAIN SEITE ERSTELLEN	8 Stunden = 1 Tag
↪ SVG EINBEZIEHEN	1 Stunde
↪ POPUP IMPLEMENTIEREN	4 Stunden
↪ DATEN ANZEIGEN	3 Stunden
ABSCHLUSS MAIN-PAGE	Meilenstein
DOKUMENTATION	1 Tag
MAP SEITE ERSTELLEN	21 Stunden = 2.6 Tage
↪ DATENSÄTZE AUSWÄHLEN	3 Stunden
↪ LEAFLET IMPORTIEREN	1 Stunde
↪ SVG OVERLAY	5 Stunden
↪ FILTER EINFÜGEN	4 Stunden
↪ ALGORITHMUS FÜR BERECHNUNG VON OPTIMALER POSITION	8 Stunden
ABSCHLUSS MAP-PAGE	Meilenstein
ABSCHLUSS PROJEKT	Meilenstein

Design Dokumente

Wireframe



Testfälle

NUMMER	BESCHREIBUNG	TYPE	ERGEBNIS	ERFÜLLT & PROBLEME
1	Der Nutzer ruft die Webseite normal auf.	Konstruktiv	Die Webseite erscheint, das SVG wird mit der richtigen Farbcodierung geladen, es ist auf das letzte Erfassungsjahr eingestellt	
2	Der Nutzer drückt auf die das Feld rechts und unterhalb des Bodensees.	Konstruktiv	Ein Popup erscheint mit allen Inhalten wie im Wireframe festgelegt.	
3	Der Nutzer drückt auf den „Schließen“-Button des Popups	Konstruktiv	Der Popup schließt sich.	

4	Der Nutzer macht das Browserfenster kleiner.	Konstruktiv	Die Webseite ändert ihre Größen. Die Diagramme, die bei einem Computer Screen (Fullscreen) nebeneinander liegen, sind nun übereinander.
5	Der Nutzer klickt in der Navigationsleiste auf den Punkt „Map“ (könnte in der Endfassung auch „Karte“ heißen).	Konstruktiv	Die Webseite ändert ihre Seite auf den Kartenansicht.
6	Der Nutzer bearbeitet die Filtereinstellungen.	Konstruktiv	Die Karte mit den Anzeigen ändert sich.
7	Der Nutzer drückt auf den Knopf, „Wohnort berechnen“	Konstruktiv	Ein Pointer wird auf der Karte gesetzt, wo der ideale Wohnort wäre.
8	Der Nutzer ruft die Sub-URL „./test“ auf.	Destruktiv	Der Nutzer bekommt eine personalisierte 404-Fehlerseite angezeigt.
9	Der Nutzer hat es auf die Hauptseite geschafft und klickt auf den Bodensee	Destruktiv	Der Nutzer erhält kein Popup, jedoch erscheint ein freundliches „Alert“, welches den Nutzer darauf aufmerksam macht, dass der Bodensee keine Gemeinde ist.
10	Der Nutzer schafft es zu einem Popup und zu öffnen und möchte es nun schließen. Dazu klickt es nicht auf das Kreuz, sondern direkt neben den Popup.		Auch wenn diese Handlung des Nutzers nicht ideal ist, reagiert das Programm korrekt und schließt den Popup.
11	Der Nutzer schafft es zur „Karten“-Seite und kommt nicht mit den selbsterklärenden Filtern zurecht.		Der Nutzer hat somit Pech. Evtl. wird es in Zukunft eine Hilfs-Seite geben worauf man unter einem Hilfsknopf „ ❓ “ weitergeleitet wird.

Rest API Rückgaben

Es gibt keine Rest API, da die „Index“ Seiten bei dieser Anwendung mit NextJS mit Static Site Generation gerendert werden.

Es könnte noch eine Rest API hinzukommen aber nur auf einer „Nice-To-Have“-Basis für die „Karten“-Seite, da diese wahrscheinlich Client-Side Rendering nutzt und nicht Static Site Generation oder Pre-Rendering.

Als Ergänzung sei erwähnt, dass mir beim Verfassen dieses Abschnittes nicht in Gedanken war, dass, um die lokalen gesicherten Zwischenspeicherdateien möglichst schmal zu halten, nicht den Datensatz in seiner kolossalischen Gesamtheit der angehäuften Aufzeichnungsjahre, sondern nur das gesuchte Jahr und bei potenziellen Optimierungen nur die allfällige Gemeinde einzulesen und anzeigen zu lassen, eine Static Site Generation nicht möglich oder zu mindestens nicht praktikable sei, da diese bei dies beim laden des Datensatz, um mit dem geplanten Slider annoncier bar auf der Karte zu sein, gänzlich und in summa im Stöberer zwischengespeichert werden müsse, welches den Nebeneffekt hätte, dass die gewonnene Geschwindigkeit, durch die sich nicht wiederholende Berechnung des Darstellungsbereiches infolge der Größe des einmalig gestalteten interaktiven Grafikendeskriptionsterritorium, aufgrund der Übertragungsrate des arrangierten Anzeigedokumentes, verloren gehen würde. In Angesicht dieser Tatsache kommt jedoch auch kein Server Side Rendering in Frage, da der Datensatz zum einen keine fluktuierenden Eigenschaften aufweist, der Vorteil der Zersplitterung des Datensatzes in subpartikulare Datensplitter durch regelmäßiges Verrichten des Gestaltungsprozesses nicht erreicht werden kann und dies ebenfalls für die gemischte Variant, Incremental Static Regeneration, zwischen Server Side Rendering und Static Site Generation gilt, bleibt nur die die Option des Client Side Rendering oder Client Side Fetching. Hierfür wiederum wird eine Rest API benötigt, infolgedessen veranlasse ich eine Revision meiner geistigen Einschätzung und Neubewertung des gewählten Ausgabemodels infolge einer gezielten Änderung der Parameter.

Dementsprechend sind die Entwürfe für die Rückgabewerte der Rest API im Still von klassischen Schemata zu finden.

Rest API Schemata

Bezüglich der Kodierung:

ZEICHEN	DEFINITION
?	Nicht sicher, ob dieses Attribut mitgeliefert wird
[…]	Array (alles, was sich darinnen befindet, befindet sich in Sub-Objekt)
ATTR: [TYPE]	Array vom Typ

./api/erneuerbareElektrizitatsproduktionNachEnergietragernUndGemeinden/[Jahr]

```
[
nr_gemeinde: String
gemeinde_name: String
einwohner: Int
?biogasanlagen_abwasser: Float
?biogasanlagen_industrie: Float
?biogasanlagen_landwirtschaft: Float
?biomasse_holz: Float
?kehricht: Float
?photovoltaik: Float
?wasserkraft: Float
?wind: Float
total: Float
```

```
]
./api/erneuerbareElektrizitatsproduktionNachEnergietragernUndGemeinden/[Jahr]/[GemeindeNR]
gemeinde_name: String
einwohner: Int
?biogasanlagen_abwasser: Float
?biogasanlagen_industrie: Float?
?biogasanlagen_landwirtschaft: Float?
?biomasse_holz: Float
?kehricht: Float
?photovoltaik: Float
?wasserkraft: Float
?wind: Float
total: Float
./api/erneuerbareElektrizitatsproduktionNachEnergietragernUndGemeinden/[Jahr]/[Energieträger]
gemeinde_name: String
nr_gemeinde: String
energieträger: Float
./api/erneuerbareElektrizitatsproduktionNachEnergietragernUndGemeinden/[GemeindeNummer]
gemeinde_name: String
nr_gemeinde: String
[
jahr: String
?biogasanlagen_abwasser: Float
?biogasanlagen_industrie: Float
?biogasanlagen_landwirtschaft: Float
?biomasse_holz: Float
?kehricht: Float
?photovoltaik: Float
?wasserkraft: Float
?wind: Float
total: Float
]
```