**Ständige ausländische Wohnbevölkerung Kanton Thurgau ab 2015 nach Gemeinden und Staatsangehörigkeit**

Ein Bild, das Gras, draußen, Himmel, Pflanze enthält.

Automatisch generierte Beschreibung

**Projektwoche 2i**

Verfasser: Bleron Regja, Yanis Bülow

Betreuende Lehrperson: Herr Hager, Herr Hofstetter

Informatikmittelschule

Abgabetermin: 26.05.2023

Inhaltsverzeichnis

[1 Analyse 4](#_Toc135995104)

[1.1 Aufgabenstellung 4](#_Toc135995105)

[1.2 Beschreibung 4](#_Toc135995106)

[1.3 Mitarbeiter 4](#_Toc135995107)

[1.4 Informationsbeschaffung 5](#_Toc135995108)

[1.5 Anforderungen 5](#_Toc135995109)

[1.6 Ziele 5](#_Toc135995110)

[1.7 Must-Haves 5](#_Toc135995111)

[1.8 Nice-to-Have 5](#_Toc135995112)

[1.9 Abgrenzungen 5](#_Toc135995113)

[1.10 Zielgruppe 5](#_Toc135995114)

[1.11 Use-Cases 6](#_Toc135995115)

[1.12 Umgebung 7](#_Toc135995116)

[1.13 Entwicklungstools 7](#_Toc135995117)

[1.14 Technologien 7](#_Toc135995118)

[1.15 Lösungsvarianten 7](#_Toc135995119)

[1.16 Zeiteinteilung 8](#_Toc135995120)

[1.17 Aufgabenteilung 8](#_Toc135995121)

[2 Design 9](#_Toc135995122)

[2.1 Wireframe 9](#_Toc135995123)

[2.2 Testfälle 10](#_Toc135995124)

[3 Code 11](#_Toc135995125)

[3.1 Navigation 11](#_Toc135995126)

[3.2 Karte 11](#_Toc135995127)

[3.3 Laden der Daten ins backend 12](#_Toc135995128)

[3.4 Diagramme 13](#_Toc135995129)

[3.5 Design 15](#_Toc135995130)

[4 Rest-API 16](#_Toc135995131)

[5 Testen 16](#_Toc135995132)

[6 Deployment 18](#_Toc135995133)

[6.1 Installationsanleitung 18](#_Toc135995134)

[7 Benutzeranleitung 18](#_Toc135995135)

[8 Abbildungsverzeichnis 18](#_Toc135995136)

[9 Quellenverzeichnis 18](#_Toc135995137)

# Analyse

## Aufgabenstellung

Wir müssen die Ständige ausländische Wohnbevölkerung im Kanton Thurgau ab 2015 nach Gemeinden und Staatsangehörigkeit in ein eine anschauliche Website verpacken, welche leicht bedient werden kann.

## Beschreibung

Wir wollen eine Single Page Webseite machen, die den Ausländeranteil in den jeweiligen Bezirken im Kanton Thurgau aufzeigt.

Wir benutzen dafür eine API von der Open Government Data, welche uns gratis Datensätze zur Verfügung stellt. Dabei haben wir uns vorgestellt, dass man, wenn man die Seite öffnet als erstes eine Karte des Kantons Thurgau sieht und auf jeden Bezirk mit der Maus manövrieren kann und dann die Einwohneranzahl und die jeweiligen Anzahlen von ausländischen Personen sieht.

Z.B:

Einwohner: 1200

Schweizer: 400

Portugiesen: 55

Italiener: 69

…

Darunter kann man Aus einem Dropdownmenü ein Jahr und eine Gemeinde auswählen, um dann in einem Kreisdiagramm einzusehen, wie die Unterschiede zur Hautstadt Frauenfeld aussehen.

Ein nice to have wäre in diesem Fall das man jeden beliebigen Bezirk miteinander vergleichen kann.

Als letztes ist noch ein Liniendiagramm zu sehen, auf welchem man auch einen Bezirk auswählen kann und auf dem die Entwicklung des Ausländeranteils über Jahre angezeigt wird.

## Mitarbeiter

Bleron:

Projektleiter,

Developer,

Designer,

Backend,

Frontend,

APIs,

Tester

Yanis

Projektleiter,

Developer,

Designer,

Backend,

Frontend,

APIs,

Tester

## Informationsbeschaffung

Die Daten, welche wir für unsere Webseite benötigt haben, bezogen wir über eine API von der Webseite opendata.swiss, welche gratis verschiedene Daten über alles Mögliche in der Schweiz liefert. In unserm Fall über die Ausländeranteile in jedem Bezirk im Kanton Thurgau aufgeteilt in Nationalitäten vom Jahr 2015 bis 2022. Da die Daten jedoch in einem sehr seltsamen Format waren, zwar haben sie für jede Nationalität ein eigenes Dokument erstellt, mussten wir diese noch im Backend (Applikations-Server) verarbeiten.

## Anforderungen

Wir mussten eine Webseite erstellen, auf der ersichtlich ist, wie und ob sich der Ausländeranteil im Kanton Thurgau über die Jahre 2015-2022 verändert.

Zur Hilfe haben wir und eine Grafik des Kantons genommen, bei der man oberhalb der Möglichkeit hat das Jahr aus einem Dropdownmenü auszuwählen und dann mit der Maus über die einzelnen Bezirke Hoover kann, um deren Anteil an ausländischen Nationalitäten für das gewählte Jahr zu sehen.

Auch haben wir ein Liniendiagramm, in welchem für jeden Bezirk das Wachstum jede Nationalität ersichtlich ist.

Zusätzlich kann man aus einem Dropdownmenü einmal die Gemeinde und das Jahr Wählen und direkt mit der Gemeinde Frauenfeld vergleichen, da nun 2 Kreisdiagramm erscheine

## Ziele

Unser Ziel war es, nochmals einen Einblick in die Webentwicklung zu bekommen, um React und vor allem nochmals backend mit express und node.js zu lernen.

Zudem ist es eine Wertvolle Erfahrung und macht viel Spass.

## Must-Haves

* Eine Karte mit der Möglichkeit, drüber zu fahren und Informationen zu erhalten.
* Zwei Kreisdiagramme, mit denen man die Entwickelung vergleichen kann.
* Ein Liniendiagramm, bei dem man die Entwickelung über den Zeitraum 2015-2022 sieht.
* Eine funktionierende Webseite, die schön und ansprechend ausseht.
* Mit den Filterfunktionen sollte man die Gemeinde und das Jahr verändern können.

## Nice-to-Have

* 2 beliebige Gemeinden miteinander vergleichen zu können.
* Eine Navigation für die einzelnen Komponenten

## Abgrenzungen

Wir sind auf die Daten beschränkt, die wir von opendata.swiss bekommen haben. Diese liegen nur zwischen dem Zeitraum von 2015-2022 und auch nur für den Kanton Thurgau.

## Zielgruppe

Die Webseite hat das Ziel, für jeden Person. Besonders die, welche sich für die Ausländeranteile im Kanton Thurgau interessieren.

Die Bedürfnisse sind dabei Wissbegierde und Recherche

## Use-Cases

Hier wird aufgezeigt, wie der Benutzer die Webseite benutzen kann.

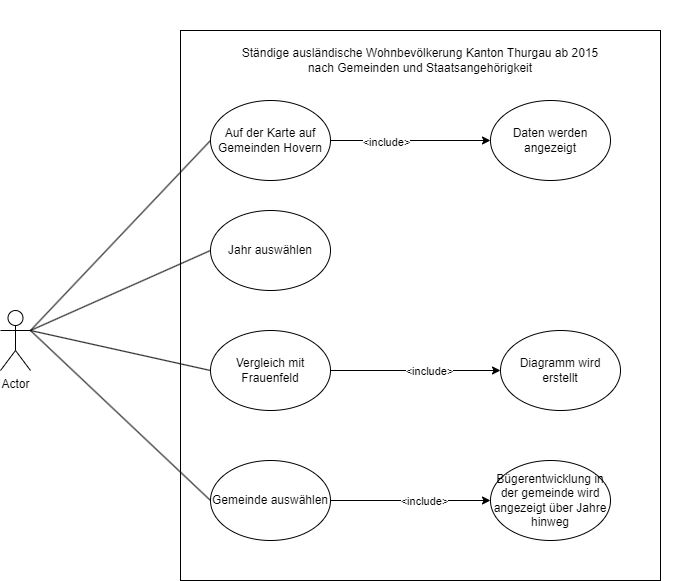


Abbildung Use-Cases

* Wenn der User auf Die Karte manövriert, dann werden in für die Gemeinde, auf der er mit der Maus ist Informationen angezeigt
* Dann kann er aus einem Dropdown-Menü das Jahr auswählen und die Informationen auf der Karte passen sich an das Ausgewählte Jahr an.
* Als nächstes kann er ein Jahr und eine Gemeinde auswählen und dann werden in einem Kreisdiagramm die Informationen über die Gemeinde aus dem Jahr angezeigt und daneben die Informationen von Kreuzlingen. Jetzt kann der User die Daten vergleichen
* Zuletzt kann der User wieder aus einem Dropdown eine Gemeinde auswählen und sieht das Wachstum der Nationalitäten über die Jahre 2015-2022

## Umgebung

Wir haben den Entschluss gefasst, für dieses Projekt Visual Studio Code zu verwenden, da dies eine einfache und gute Umgebung ist und wir schon früher damit Projekte gemacht haben und somit schon Erfahrung haben und keine Zeit aufwenden müssen, um uns erst an die neue Umgebung zu Gewöhnen.

Als Framework für das Frontend haben wir uns für React entschieden und für das Backend Node.

Für unsere Zusammenarbeit benötigten wir einen zentralen Ort, an dem der Sourcecode gespeichert wird. Dafür haben wir [Gitlab](https://gitlab.com/) benutzt.

## Entwicklungstools

Als Hilfe für die Entwicklung haben wir uns Axios express und svg-inject sowie Chartjs und Corse

## Technologien

Die Technologien, die wir benutzt haben, sind nodejs und React

## Lösungsvarianten

Wir hatten zwei mögliche Varianten wie wir das Projekt angehen können.

1. Wir erstellen eine Datenbank, bei der wir die Daten der API speichern und dann von der Datenbank in unsere Webseite laden
2. Wir müssen die Möglichkeit einbauen, die Informationen die wir Darstellen mit zum Beispiel einem Dropdown anzupassen.
3. Schlussendlich haben wir uns für die 2. Variante entschieden, da wir bisher noch nicht mit Grafiken gearbeitet haben und da unser Ziel daraus bestand, neue Sachen zu lernen, viel die Entscheidung nicht schwer.

## Zeiteinteilung

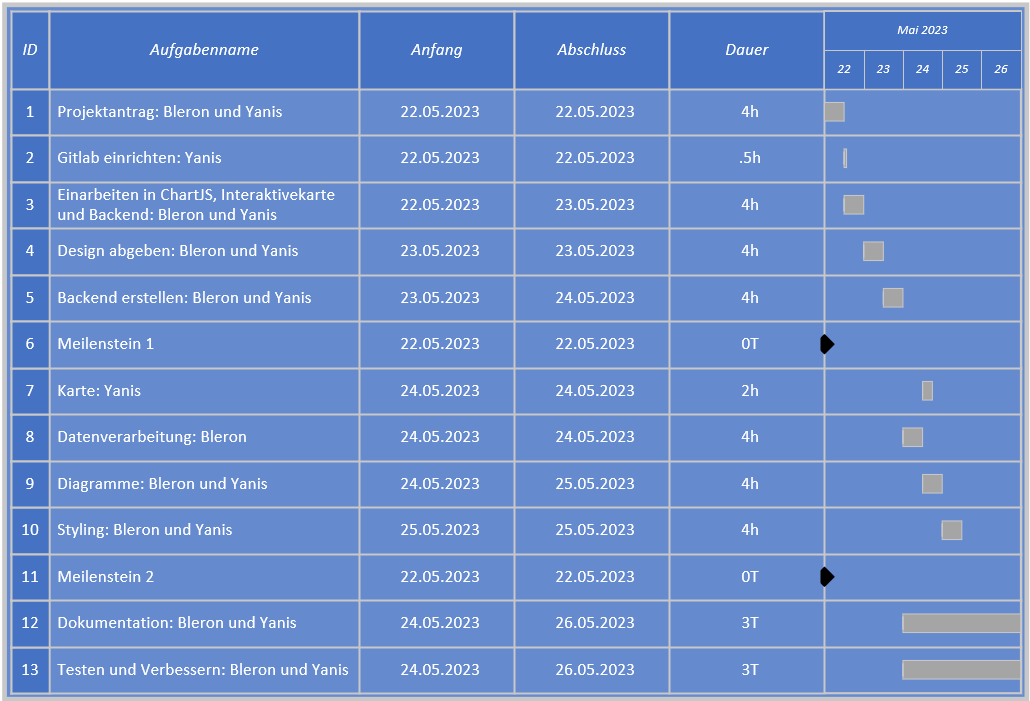


Abbildung Zeiteinteilung

## Aufgabenteilung

* Die Hauptaufgabe von Bleron war es die Daten von der API zu laden und im Backend zu verarbeiten. Zudem hat er auch bei den Diagrammen mitgeholfen.
* Die Aufgabe von Yanis war es im Frontend die Karte einzufügen und die Diagramme zu definieren auch war es Aufgabe von Yanis die Dokumentation zu schreiben.
* Das Design der Webseite haben wir uns geteilt.

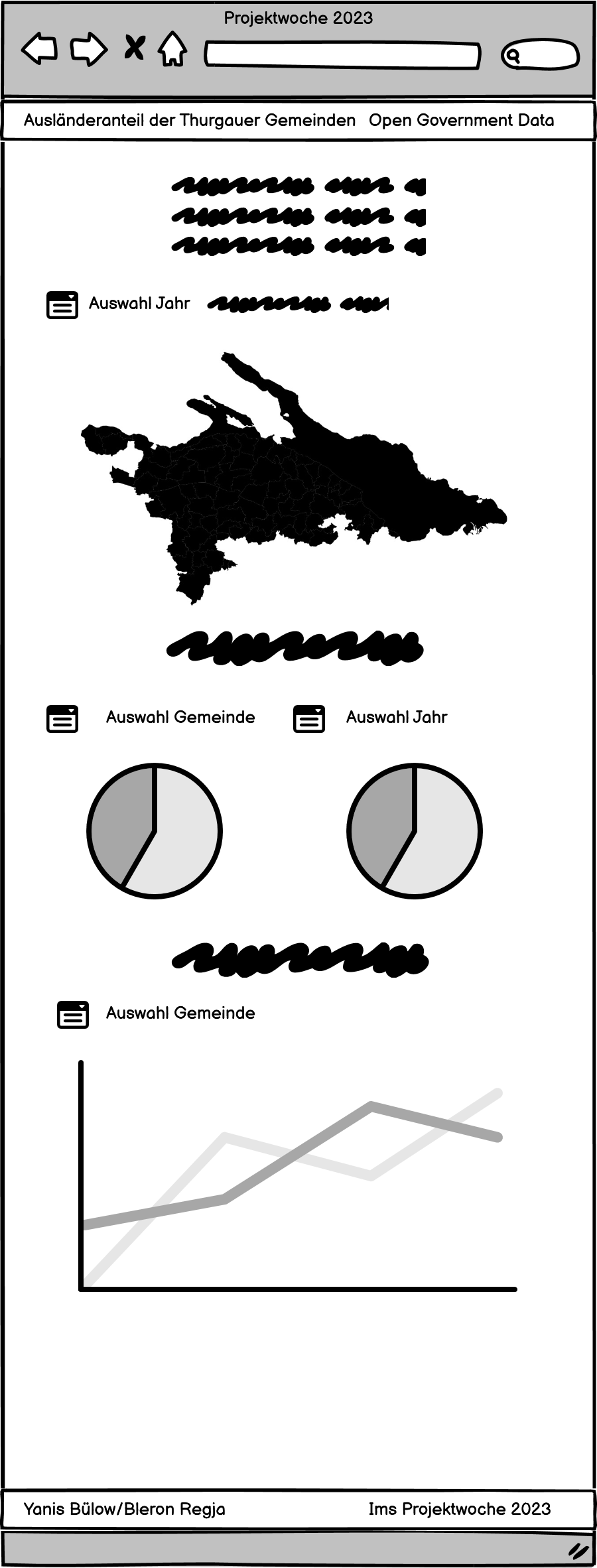


Abbildung Wireframe

# Design

Unser Ziel war es, eine möglichst einfache und ansprechende Seite für den Nutzer zu generieren.

Dabei haben wir den Fokus daraufgelegt, dass der User nicht von den ganzen Informationen erschlagen wird. Deswegen sind alle unsere Grafiken untereinander.

Auch haben wir Wert daraufgelegt, dass der User die Seite spielhaft erkunden kann. Dazu kann er die Maus benutzen und hat bei den Diagrammen die Möglichkeit, Sachen an- und abzuwählen.

## Wireframe

## Testfälle

Getestet von:

Getestete Version des Projekts:

Betriebssystem:

Browser:

Test-Datum:

|  |  |  |
| --- | --- | --- |
| Test | Ergebnis | Bestanden |
|  |  |  |
| Drückt man auf den ersten Redirect | Man wird zum ersten Abschnitt weitergeleitet |  |
| Drückt man auf den zweiten Redirect | Man wird zum zweiten Abschnitt weitergeleitet |  |
| Drückt man auf den letzten Redirect | Man wird zum letzten Abschnitt weitergeleitet |  |
|  |  |  |
| Über Kreuzlingen manövrieren | Der Name Kreuzlingen erscheint mit den Daten |  |
| Jahr im 1. Dropdown auswählen | Die Karte passt sich dem Jahr an |  |
| Danach über Berg manövrieren | Der Name Berg erscheint mit den angepassten Daten |  |
|  |  |  |
| Gemeinde auswählen im 2. Dropdown | Diagramm zeigt Daten in der gemeinde |  |
| Wählt man noch das Jahr | Diagramme passen sich an auf das entsprechende Jahr |  |
|  |  |  |
| Gemeinde im 3. Dropdown auswählen | Diagramm passt sich auf die Gemeinde an |  |
|  |  |  |
| Auf Yanis Bülow drücken | Man wird zu seinem Github weitergeleitet |  |
| Auf Bleron Regja drücken | Man wird zu seinem Github weitergeleitet |  |

# Code

Im folgenden Abschnitt wollen wir euch eine Art Führung durch unseren Code geben, in dem Sie Schritt für Schritt unsere Gedankengänge und Überlegungen mitverfolgen können, um zu sehen, wie eine Webseite von Grund auf aufgebaut wird.

## Navigation

Die Navigation ist keine herkömmliche Navigation, beider man von einer Seite auf die Nächste wechseln kann, sondern auf einer Seite, bei der man unter den verschiedenen Abschnitten wechseln kann.

Unsere Webseite ist nämlich in 3 Abschnitte aufgeteilt.

Die Karte, die Kreisdiagramme und das Liniendiagramm.

Hier sieht man das wir simple nur eine <a href >für jeden Komponenten erstellt haben, mit dem man direkt zur jeweiligen Klasse springen kann.

Das <a href> Element wird dazu benutzt, um zu dem Objekt mit der ID absatz1 weiterzuleiten.

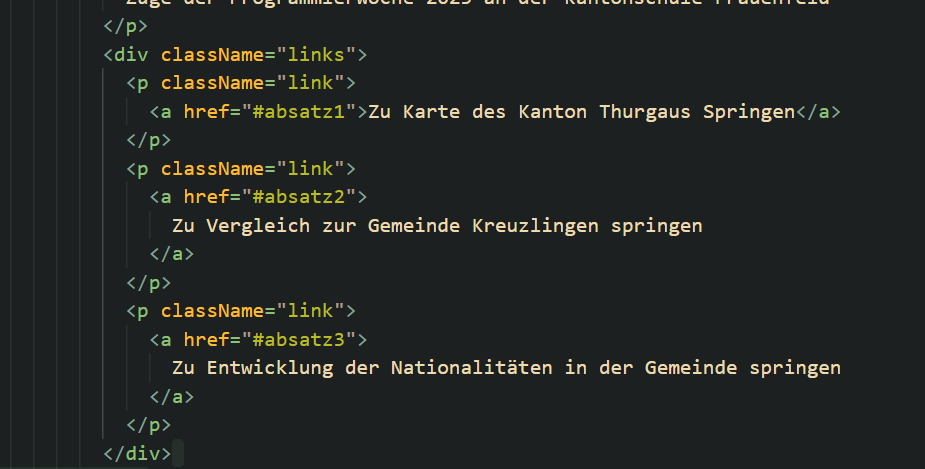


Abbildung Navigation Code

## Karte

Da die Karte in einem SVG-Format waren (ein SVG ist ein Bild von zum Beispiel einer Karte, welche direkt schon für jeden Bezirk jetzt in unserm Fall einen Path hat, also die genauen Koordinaten für das Bild sowie die BFS-Nummer. Dadurch konnten wir für den Hover- Effekt direkt die nötigen Tooltips auf den Bezirken platzieren) konnten wir mit Hilfe eines Svg-Injectes die Bildpfade direkt in den HTML code also, dem DOM platzieren.

Nun mussten wir nur noch auf jedem Bezirk ein Tooltip platzieren und darauf die nötigen Informationen einblenden.

Hier sieht man wie wir für jeden Path also für jeden Bezirk einen Inhalt hinzufügen.

Dabei mussten wir einen UseEffect benutzen, um direkt in den JavaScript code direkt einzufügen.

Wir benutzen hier ein querySelectorAll um von der Karte alle Objekte mit der ID #municipalities zu bekommen. Darunter wird das gleiche für den Tooltip gemacht. Nun wird mit Hilfe einer For-Schleife jedem Objekt die funktionalität hinzugefügt, dass man, wenn man drüber Hovert mit der Maus Informationen sieht.



Abbildung Karte

## Laden der Daten ins backend

Da wir uns dafür entschieden haben, bei dem Projekt keine Datenbank zu benutzen, mussten wir alle Daten in unserem Backend laden und für das Weiterschicken ans Frontend bearbeiten.

Dies entpuppte sich als ein wenig komplizierter als Anfangs gedacht, da die Daten wie schon beschrieben in einem sehr seltsamen Format waren.

Dies führet dazu, dass wir die Datensätze erst zusammenfügen mussten, um sie dann in unseren Diagrammen und Karten zu benutzen. Dies bewirkte jedoch auch, dass die Seite immer eine gewisse Zeit benötigt, um zu laden.

Dadurch das es auch so viele Datensätze waren, hatten wir generell Performance Probleme.

Im folgenden Bild ist ein teil des Codes ersichtlich, in dem wir die Daten von der API holen und für das Frontend aufbereiten

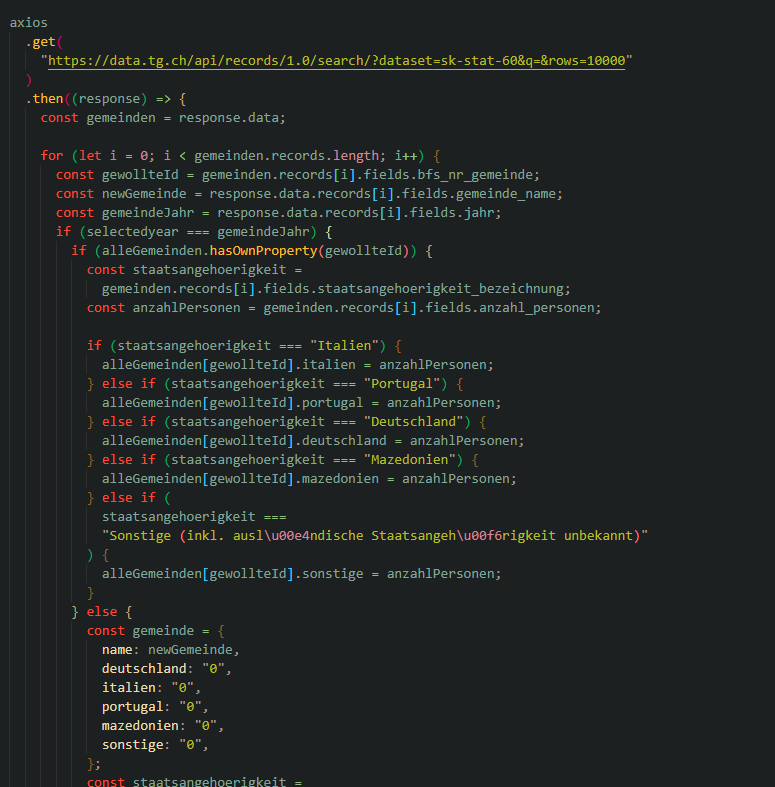


Abbildung Backedn

## Diagramme

Für die Diagramme haben wir JavaScript Bibliothek benutzt, die Chartjs heisst. Sie macht es einfach schnell, gute Diagramm anzeigen lassen zu können.

Damit wir unsere Hauptseite im Code etwas übersichtlicher halten konnten, haben wir die Diagramme auf einer anderen Seite geschrieben und dann mit Hilfe von React Komponente auf der Seite aufgerufen.

Hier sieht man wie wir ein Kuchendiagram mit Daten vom Backend erstellen:

Der Type setzt fest um was für eine Art von Diagramm es sich handelt.

In der Data sind die ganzen Informationen vorhanden. Die Labels sind die einzelnen Teile, die man im Kuchendiagramm sieht. Im dataset sind dann effektiv die ganzen Datensätze im Diagramm die angezeigt werden.



Abbildung Kuchendiagramm

## Design

Wir haben uns eigentlich von Beginn an unser Wireframe gehalten und unsere Webseite dementsprechend geplant und umgesetzt.

Beim Styling haben wir ganz standartgemäss CSS als Sprache gewählt, da wir zuvor bereits ein Modul in CSS hatten und bereits damit vertraut waren.

Wir suchten uns zudem bei der Planung noch gewisse Farbschemen für unsere Webseite aus, nachdem wir unsere Seite probierten zu gestalten

Hier ist ein Beispiel dafür:

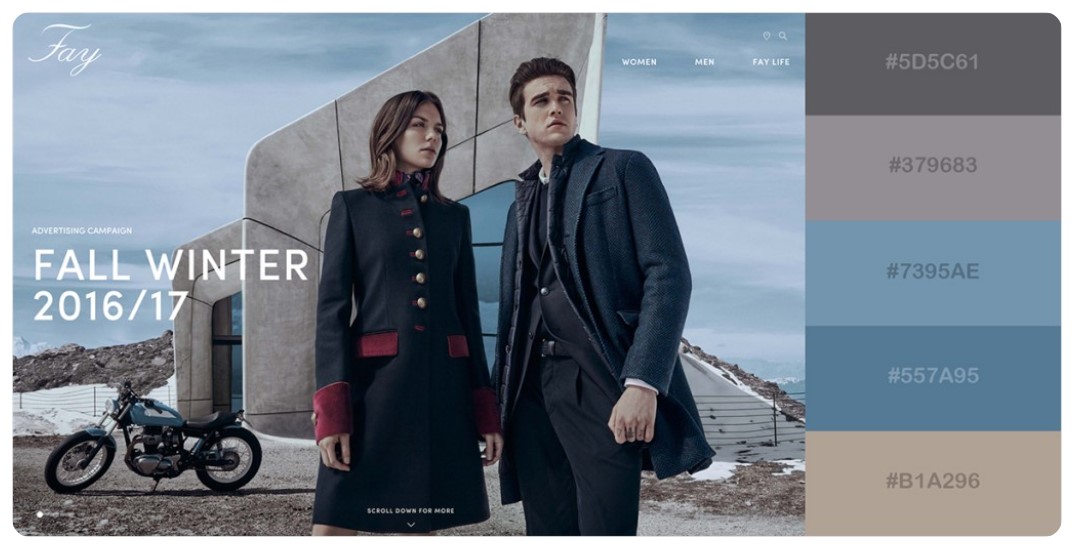


Abbildung Farbschema

# Rest-API

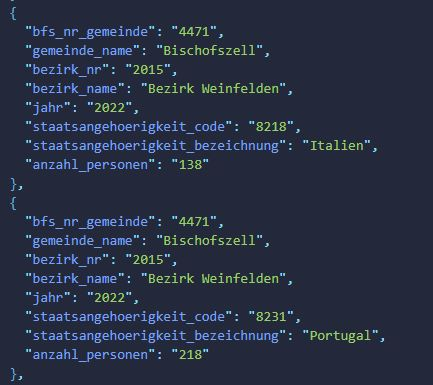


Abbildung Json File

# Testen

Getestet von: Yanis/Bleron

Getestete Version des Projekts: 3

Betriebssystem: Windows

Browser: Chrome

Test-Datum: 25.05.2023

|  |  |  |
| --- | --- | --- |
| Test | Ergebnis | Bestanden |
|  |  |  |
| Drückt man auf den ersten Redirect | Man wird zum ersten Abschnitt weitergeleitet | Bestanden |
| Drückt man auf den zweiten Redirect | Man wird zum zweiten Abschnitt weitergeleitet | Bestanden |
| Drückt man auf den letzten Redirect | Man wird zum letzten Abschnitt weitergeleitet | Bestanden |
|  |  |  |
| Über Kreuzlingen manövrieren | Der Name Kreuzlingen erscheint mit den Daten | Bestanden |
| Jahr im 1. Dropdown auswählen | Die Karte passt sich dem Jahr an | Bestanden |
| Danach über Berg manövrieren | Der Name Berg erscheint mit den angepassten Daten | Bestanden |
|  |  |  |
| Gemeinde auswählen im 2. Dropdown | Diagramm zeigt Daten in der gemeinde | Bestanden |
| Wählt man noch das Jahr | Diagramme passen sich an auf das entsprechende Jahr | Bestanden |
|  |  |  |
| Gemeinde im 3. Dropdown auswählen | Diagramm passt sich auf die Gemeinde an | bestanden |
|  |  |  |
| Auf Yanis Bülow drücken | Man wird zu seinem Github weitergeleitet | Bestanden |
| Auf Bleron Regja drücken | Man wird zu seinem Github weitergeleitet | Bestanden |

# Deployment

## Installationsanleitung

Lade das Projekt von [Gitlab](https://kftg-my.sharepoint.com/personal/bleron_regja_stud_kftg_ch/Documents/Lade%20das%20Projekt%20von%20GitLab%20herunter%20und%20entpacke%20es,%20falls%20erforderlich.) herunter und entpacke es, falls erforderlich.

Öffnen sie nun das Projekt.

Öffnen sie nun ein Terminal und geben Sie den Befehl «npm install» ein.

Nun geben sie den befehl «npm run start» in Terminal ein und voila, Geschafft

# Benutzeranleitung

Der Benutzer kann, wenn er die Seite öffnet über die Navigation aussuchen welchen Teil er anschauen möchte.

# Abbildungsverzeichnis

[Abbildung 1 Use-Cases 6](https://kftg-my.sharepoint.com/personal/bleron_regja_stud_kftg_ch/Documents/Dokumentation-Projektwoche.docx#_Toc135995138)

[Abbildung 2 Zeiteinteilung 8](https://kftg-my.sharepoint.com/personal/bleron_regja_stud_kftg_ch/Documents/Dokumentation-Projektwoche.docx#_Toc135995139)

[Abbildung 3 Wireframe 9](https://kftg-my.sharepoint.com/personal/bleron_regja_stud_kftg_ch/Documents/Dokumentation-Projektwoche.docx#_Toc135995140)

[Abbildung 4 Navigation Code 11](https://kftg-my.sharepoint.com/personal/bleron_regja_stud_kftg_ch/Documents/Dokumentation-Projektwoche.docx#_Toc135995141)

[Abbildung 5 Karte 12](https://kftg-my.sharepoint.com/personal/bleron_regja_stud_kftg_ch/Documents/Dokumentation-Projektwoche.docx#_Toc135995142)

[Abbildung 6 Backedn 13](https://kftg-my.sharepoint.com/personal/bleron_regja_stud_kftg_ch/Documents/Dokumentation-Projektwoche.docx#_Toc135995143)

[Abbildung 7 Kuchendiagramm 14](https://kftg-my.sharepoint.com/personal/bleron_regja_stud_kftg_ch/Documents/Dokumentation-Projektwoche.docx#_Toc135995144)

[Abbildung 8 Farbschema 15](https://kftg-my.sharepoint.com/personal/bleron_regja_stud_kftg_ch/Documents/Dokumentation-Projektwoche.docx#_Toc135995145)

[Abbildung 9Json File 16](https://kftg-my.sharepoint.com/personal/bleron_regja_stud_kftg_ch/Documents/Dokumentation-Projektwoche.docx#_Toc135995146)

# Quellenverzeichnis

Generelle Informationen:

<https://opendata.swiss/de>

Informationen über Chartjs und SVG:

<https://moodle.kftg.ch/course/view.php?id=358>

Spezifische Informationen über Chartjs

<https://www.chartjs.org/docs/latest/getting-started/installation.html>

Informationen über Git:

<https://rogerdudler.github.io/git-guide/index.de.html>