



Projektbericht:
Nationalparks in Österreich mit dem Fallbeispiel Nationalpark Hohe Tauern.

Geographie: Globaler Wandel - regionale Nachhaltigkeit

VU Geoinformatik: Webmapping

Kursleitung: Klaus Förster, Bernd Oegg

Dominic Kreisel
Matr.nr.: 12304670
E-Mail: dominic.kreisel@student.uibk.ac.at

Sebastian Spreitzer
Matr.nr.: 12018895
E-Mail: sebastian.spreitzer@student.uibk.ac.at

Madeleine Hill
Matr.nr.: 12238447
E-Mail: madeleine.hill@student.uibk.ac.at

Abgabe: 26.06.2024

Inhalt

1.	Projektbeschreibung.....	3
2.	Startseite.....	3
3.	Nationalpark Hohe Tauern – Ökologie und Geographie	9
4.	Tourismus Nationalpark Hohe Tauern.....	15
5.	Probleme und negative Aspekte.....	22
6.	Quellen	23

1. Projektbeschreibung

Unser Projekt besteht aus drei Webseiten. Dabei dient eine Seite als Übersicht, in welcher die Nationalparks in Österreich thematisiert werden. Diese ist verlinkt mit zwei thematischen Seiten zum Nationalpark Hohe Tauern. Dieser wurde aufgrund seiner Größe und Bedeutung ausgewählt. Wir haben uns entschieden zwei verschiedene Themenkomplexe anhand dieses Beispiels zu erarbeiten. Zum einen werden Themen der Umwelt behandelt, d.h. Gipfel, Zonierung, Moore, Gletscher. Zum anderen wird die touristische Aktivität des Raumes dargestellt. Dazu werden Wanderwege und Almen visualisiert.

2. Startseite

Die Startseite dient als Einstieg und gibt einen groben Überblick über die sechs Nationalparks Österreichs. Zu Beginn steht ein kleiner Einleitungstext, der von der offiziellen Seite der österreichischen Nationalparks stammt (<https://www.nationalparksaustria.at/de/>), welche auch verlinkt ist. Auf der Website sind die sechs Nationalparks mit einem Marker auf einer Übersichtskarte markiert. Links oben befinden sich die Zoom-Regler sowie der Button für den Vollscreen-Modus. Darunter findet man ein Lupen-Symbol, mit welchem man die jeweiligen Nationalparks suchen und zu dem dazugehörigen Marker zoomen kann. Leider funktioniert dieses Plugin nicht einwandfrei, dazu jedoch später mehr. In der rechten oberen Ecke kann man die Layer an- und ausschalten. Zur Auswahl stehen die Optionen Wettervorhersage, Windvorhersage und die Positionen der Nationalparks. Zur besseren Orientierung wurden die österreichischen Grenzen bzw. das Staatsgebiet rot hervorgehoben. Sind alle Layer aktiviert, sieht die Karte folgendermaßen aus:



Abbildung 1: Übersichtskarte der Startseite (eigene Darstellung 2024).

Diese Karte bietet sich ideal dafür an, den Ausflug in einen der Nationalparks zu planen, da neben dem Wind auch die Wettervorhersage angezeigt werden kann. Dafür klickt man auf eine beliebige Stelle in der Karte, woraufhin ein Popup erscheint, welches die Wettervorhersage für diesen Punkt anzeigt und die Daten zum Download anbietet:



Abbildung 2: Popup der Wettervorhersage (eigene Darstellung 2024).

Hat man sich für einen Nationalpark entschieden, klickt man ganz einfach auf den jeweiligen Marker, woraufhin ein Popup erscheint. Dieses beinhaltet den Namen des Nationalparks, sowie ein einführendes Bild, wie es in dem Nationalpark aussieht bzw. was einen erwartet:

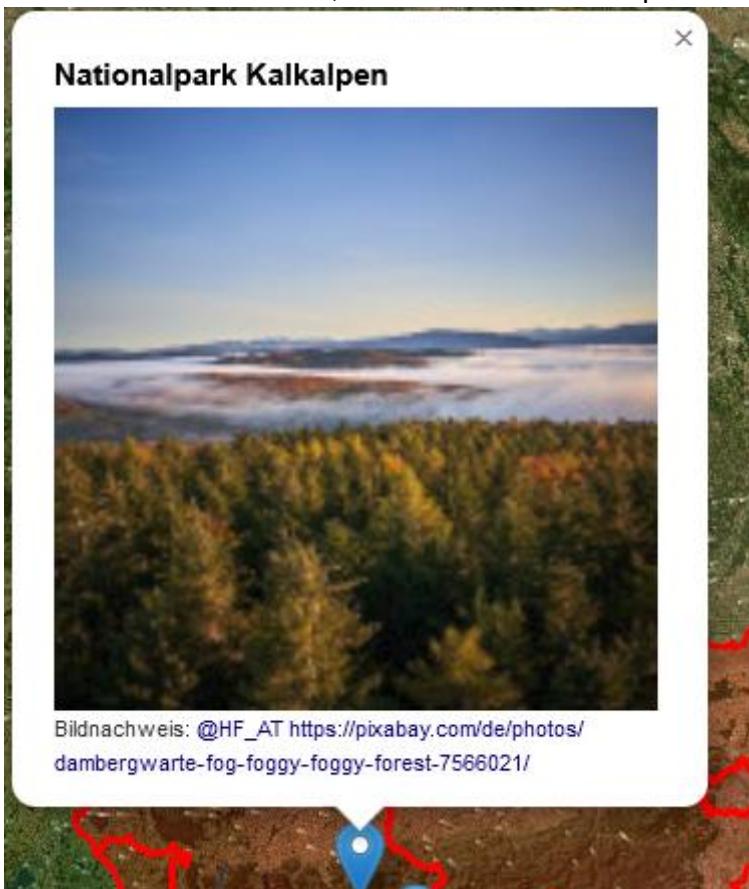
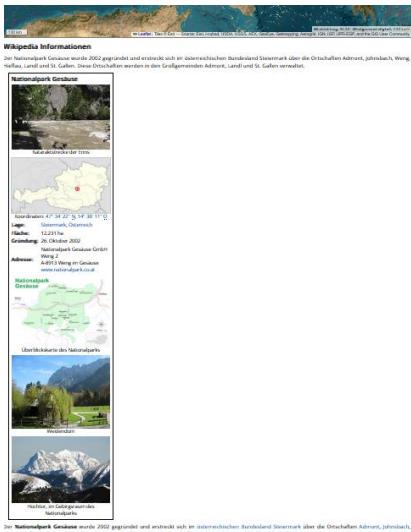


Abbildung 3: Popup der Nationalparks (eigene Darstellung 2024).

Zusätzlich wird bei dem Klick auf einen der Marker der jeweilige Wikipedia-Artikel unterhalb der Karte angezeigt, aus dem man alle weiteren Informationen beziehen kann:



Der Nationalpark Gerlos wurde 2002 gegründet und erstreckt sich im österreichischen Bundesland Tirolmark über die Ortschaften Admont, Jochberg, Weng, Zögg, Thallach, Landl und St. Gallen. Diese Ortschaften werden in den Gründenden Admont, Landl & St. Gallen verewelt.

Geographie

Das derzeit (Stand: 01.01.2018) aktuelle Projekt (200 km² große, Gelände, befindet sich westlich des Gerlospasses in der Obersteiermark. Das Planungsgebiet zu diesem Zeitpunkt umfasste eine Fläche von 122 km² bei einer Höhe von 950 m. Die heutige Fläche umfasst im Ergebnis des Staatsvertrahns erweiterte, nur gewisse Arten (und Differenzen) Weisengut oder Privatwald. Charakteristisch ist die weite Gestaltung mit ihrem markanten Natur- sowie die schlichte Tafelwirtschaft der Einwohner durch Gestaltung und Hofbau. Die Siedlungen liegen zwischen 400 m bis 2300 m Höhe).

Zonenierung

Der Nationalpark ist in eine Naturnutzung und eine Beweidungzone unterteilt. Die Naturnutzung wird 7% der Fläche und wird nach Abschluss von geplanten Managementmaßnahmen eingeführt bleiben. Die Beweidungszone erhält eingeschränkte Nutzungsformen (wie z. B. extensive Altmischwald) und besteht aus 93% der Fläche.

Zur sich dem Ziel der Umgestaltung zu richten werden nach umfangreichen Vorberatungen 2018 9% der Fläche als Brontschutzgebiet (S. M2a) (vorgestellt 1995) ausgewiesen. In diesem Fall ist die Grundsubstanz von der öffentlichen Maßnahmen gegen Brontschutzgebiet ausgenommen. Wärend in dieser Regionen die Brontschutzgebiete aufgrund der Brontenförderung eine wichtige Unterstützung wichtiger Maßnahmen, um angepasste Fortschreibung und Infrastruktur vor den Auswirkungen großflächiger Katastrophen zu schützen.[2]

Naturraum

Die Lebensräume Wasser, Wald, Ame und Fels sind die prägenden Elemente im Nationalpark Gerlos. Auf Grund dieser besonderen Bedeutung wurden sie in das Zuge des Nationalparks in Form des blauen, grünen und grauen Striftens voneinander getrennt.[2] Der Park befindet sich beim Naturhaften Gestaltungsweg toward West, liegt im Bereich der Steiermark und ist Teil des Nationalparks Gerlos. Der Nationalpark Gerlos ist ein Naturraum, der die gesamte Fläche des Parks abdeckt. Dieses ist ein Naturraum, der sich auf die Hälfte der Fläche des Nationalparks Gerlos erstreckt. Im Tal findet man Ame- und Ame-Flachmoor, Schneehaube-Hochmoor und andere Flachmoore. Die Flachmoore sind sehr wertvoll für die Natur und werden daher geschützt. Durch die Flachmoore kann man die Natur besser erkennen und sie kann auch einen Beitrag zur Erhaltung der Natur leisten.

Abbildung 4: Wikipedia-Auszug zum jeweiligen Nationalpark (eigene Darstellung 2024).

Da wir uns bei unserem Projekt jedoch speziell auf den Nationalpark Hohe Tauern festgelegt haben, gibt es nur für dieses Popup weiterführende Links, die zu unseren beiden vertiefenden Websites über den Nationalpark Hohe Tauern führen. Der erste Link führt zu einer Website, die Informationen über touristische Angebote im Nationalpark bereitstellt, während der zweite Link eine Website öffnet, die die landschaftlichen Highlights des Nationalparks beleuchtet.

Die Grundstruktur wurde von der Übung „forecast“ übernommen. Im Folgenden wird kurz erläutert, wie die jeweiligen Features umgesetzt wurden. Besonders auf neue Aspekte, die im Unterricht nicht behandelt wurden, soll hier detaillierter eingegangen werden.

Da ich die Wettervorhersage auf Österreich beschränken wollte, habe ich zuerst die Grenzen von Österreich durch die Koordinaten der südwestlichen und der nordöstlichen Ecke festgelegt.

```
let austriabounds = [[46.372276, 9.530952], [49.020608, 17.160776]]; // Bounding box for Austria
```

Abbildung 5: Bounding Box für Österreich (eigene Darstellung 2024).

Leider konnte ich nicht herausfinden, wie man die konkrete Grenze von Österreich darstellen kann. Daher wurde hier ein Rechteck erstellt, welches komplett Österreich enthält, wodurch leider auch Teile von Bayern und Italien miteingeschlossen werden. Die Grenzen von Österreich wurden daher rot hervorgehoben. Klickt man dennoch außerhalb von Österreich (bzw. außerhalb des definierten Rechtecks) kommt eine Meldung, dass man innerhalb Österreichs klicken soll, um eine Wettervorhersage zu erhalten. Dies wurde folgendermaßen umgesetzt:

```

// Karte auf Klick aktualisieren, nur wenn innerhalb der österreichischen Grenzen
map.on("click", function (evt) {
  if (isInAustria(evt.latlng)) {
    showForecast(evt.latlng.lat, evt.latlng.lng);
  } else {
    alert("Bitte innerhalb Österreichs klicken.");
  }
});

// Überprüfen, ob der Punkt in Österreich liegt
function isInAustria(latlng) {
  return latlng.lat >= austriaBounds[0][0] && latlng.lat <= austriaBounds[1][0] &&
    latlng.lng >= austriaBounds[0][1] && latlng.lng <= austriaBounds[1][1];
}

```

Abbildung 6: Räumliche Beschränkung der Wettervorhersage auf Österreich (eigene Darstellung 2024).

Außerdem wurde das Zentrum der Map sowie die Zoomstufe mit der bekannten Vorgehensweise festgelegt.

Im weiteren Verlauf wurde eine Liste angelegt, die den Namen des jeweiligen Nationalparks, dessen Koordinaten, die Bild-URL und die dazugehörige Quelle sowie den entsprechenden Wikipedia-Titel enthält.

```
{
  name: "Nationalpark Gesäuse",
  lat: 47.569953,
  lng: 14.615875,
  image: "images/gesäuse.jpg",
  credit: "@jplenio https://pixabay.com/de/photos/alpine-berge-sonnenuntergang-alpen-5630807/",
  wikiTitle: "Nationalpark_Gesäuse"
},
```

Abbildung 7: Struktur der Popups der Nationalparks (eigene Darstellung 2024).

Die Angabe des Wikipedia-Titels ist für den folgenden Schritt relevant.

Wie zuvor erwähnt soll beim Klicken auf einen Marker der Wikipedia-Artikel des jeweiligen Nationalparks unterhalb der Karte erscheinen. Die Funktion erhält einen Parameter (title), der den Titel des Wikipedia-Artikels nennt, der abgerufen werden soll. Anschließend wird eine URL erstellt, die den Titel enthält und zur Wikipedia-API führt, die den gesamten HTML-Inhalt des Artikels zurückgibt. Die Antwort wird in Text umgewandelt. Hierbei wurde Chat-GPT als Unterstützung herangezogen.

```

// Funktion, um den gesamten Wikipedia-Artikel abzurufen
async function fetchFullWikipediaContent(title) {
  let url = `https://de.wikipedia.org/api/rest_v1/page/html/${title}`;
  let response = await fetch(url);
  let data = await response.text();
  return data;
}

```

Abbildung 8: Abfrage der Wikipedia-Webseiten (eigene Darstellung 2024).

Um jedoch überhaupt auf einen Marker klicken zu können, müssen diese natürlich auch hinzugefügt werden. Der Inhalt des Popups soll abhängig vom Nationalpark sein. Dafür wird durch jedes Element in der zuvor angelegten Nationalpark-Liste gegangen und ein Marker an den entsprechenden Koordinaten platziert und dem Themalayer Parks hinzugefügt. Das Popup wird mit dem Marker verknüpft, sodass es angezeigt wird, wenn der Marker angeklickt wird. Zusätzlich wird innerhalb des Popups auf die beiden anderen Websites Tourismus und

Ökologie & Geographie verlinkt. Da dies jedoch nicht in jedem Popup passieren soll, sondern nur in dem des Nationalparks Hohe Tauern, wurde hier eine if-Funktion verwendet. Außerdem muss die zuvor beschriebene Funktion, die den Wikipedia-Artikel aufruft, angesprochen werden. Die Funktion ist asynchron, was bedeutet, dass sie auf die Ergebnisse von der fetch(Full)WikipediaContent-Funktion wartet, bevor sie fortfährt. Die fetch-Funktion wird aufgerufen und es wird gewartet, bis die Daten abgerufen sind:

```
// Wikipedia-Inhalt anzeigen, wenn auf den Marker geklickt wird
marker.on('click', async () => {
    let wikiText = await fetchWikipediaContent(park.wikiTitle);
    let wikiFull = await fetchFullWikipediaContent(park.wikiTitle);
    document.getElementById('wiki-text').innerText = wikiText;
    document.getElementById('wiki-full').innerHTML = wikiFull;
});
```

Abbildung 9: fetch-Funktion, welche die Wikipedia-Abfrage auslöst (eigene Darstellung 2024).

Als nächstes sollte die Search-Funktion von Leaflet eingebunden werden, um Nationalparks nach Namen zu suchen und die Karte auf den gefundenen Park zu zentrieren. Leider hat dies nicht fehlerfrei funktioniert und auch Chat-GPT konnte mir nicht helfen. Das Plugin zeigt nur die Meldung Location not found an. Dennoch möchte ich meine Versuche hier erläutern und den Code erklären, um zu schauen ob ich ihn überhaupt richtig verstanden habe (Vielleicht wisst ihr woran es scheitert😊). Zuerst wird ein neues Suchsteuerungselement erstellt. Dies ist eine Funktion von Leaflet, die es ermöglicht, nach bestimmten Orten auf der Karte zu suchen. Dieses wird konfiguriert und erhält ein Objekt mit mehreren Eigenschaften. Es wird angegeben in welchem Layer gesucht werden soll, nach welchem Attribut gesucht werden soll und auf welche Stufe gezoomt werden soll, wenn ein Objekt gefunden wird. Außerdem wird deaktiviert, dass ein neuer Marker an der gefundenen Stelle hinzugefügt wird.

```
// Leaflet Search Control für Nationalparks
let searchControl = new L.Control.Search({
    layer: themaLayer.parks,
    propertyName: 'name',
    zoom: 12,
    marker: false,
    textPlaceholder: "Suchen...",
    moveToLocation: function(latlng, title, map) {
        map.setView(latlng, initialZoom);
        let marker = L.marker(latlng).addTo(map).bindPopup(title).openPopup();
    }
}).addTo(map);
```

Abbildung 10: Leaflet-Search-Funktion (eigene Darstellung 2024).

Die Wettervorhersage für die nächsten 24 Stunden sowie das aktuelle Wetter wurden analog zu der Übung forecast erstellt. Der einzige Unterschied ist, dass auch hier eine asynchrone Funktion eingebaut wurde, die wartet und überprüft, ob sich der Punkt in Österreich befindet und erst falls dies bestätigt ist eine Wettervorhersage anzeigt. Gleiches wurde mit der Windkarte gemacht.

Die Grenzen für Österreich wurden folgendermaßen implementiert:

```
// Grenzen von Österreich einfügen
fetch('Daten/Oesterreich.geojson')
  .then(response => response.json())
  .then(data => {
    L.geoJSON(data, {
      style: {
        color: 'red'
      },
    }).addTo(map);
  })
  .catch(error => console.error('Error fetching data:', error));
```

Abbildung 11: Einbinden der Grenzen von Österreich (eigene Darstellung 2024).

Die Grenzen von Österreich befinden sich in der GeoJSON-Datei, an welche eine Anfrage durch fetch gestellt wird. Wenn die Datei erfolgreich heruntergeladen wurde, wird die Antwort in ein JSON-Objekt umgewandelt. Danach werden die Daten an eine Funktion übergeben, die die Daten verarbeitet und zur Karte hinzufügt.

Für diese Website wurde kein eigenes Stylesheet erstellt, sondern das in den Übungen erklärte Stylesheet benutzt.

Das index.html wurde nur durch einige Textbausteine und Leaflet-Plugins ergänzt. Weiterhin musste ein div-Element hinzugefügt werden, um zu definieren, wo die Inhalte von Wikipedia auf der Website angezeigt werden.

3. Nationalpark Hohe Tauern – Ökologie und Geographie

Folgt man dem Link zum Nationalpark Hohe Tauern – Ökologie und Geographie, öffnet sich eine weitere Website, welche den natürlichen Begebenheiten des Nationalpark Hohe Tauern (NPHT) gewidmet ist. Als Startansicht wird die Außengrenze des NPHT dargestellt, mit dem Großglockner als Mittelpunkt. In der linken oberen Ecke findet man die Zoom-Level-Regler sowie den Fullscreen-Modus. In der rechten oberen Ecke eine Übersicht über die Base-maps sowie die inkludierten Layer, welche im Folgenden genauer erklärt werden sollen. In der rechten unteren Ecke findet sich eine Minimap, welcher der besseren Orientierung in der Karte dient. Diese wurde über das MiniMap Leaflet Plugin nach der Vorlage des Bike Tirol Beispiels integriert. Die Außengrenze des NPHT wurde über ein Geojson in grün Symbolisiert. Die Daten wurden über Data.gv.at heruntergeladen und zeigen per Mausklick einen allgemeinen Text über den Nationalpark, welcher der Webseite des NPHT entnommen wurde. Integriert wurde das Popup über die onEachFeature-Funktion von L.geoJSON aus der Leaflet Bibliothek. Diese Funktion wurde ebenfalls für die übrigen Layer verwendet.

Der Layer „Zonierung“ zeigt die Einteilung des NPHT in Schutzzonen. Dieser gliedert sich in Kernzone, Außenzone und Sonderschutzgebiet. Die Popups zeigen Informationen zur jeweiligen Zone. Die Zonierung konnte ebenfalls über data.gv.at als GeoJSON heruntergeladen

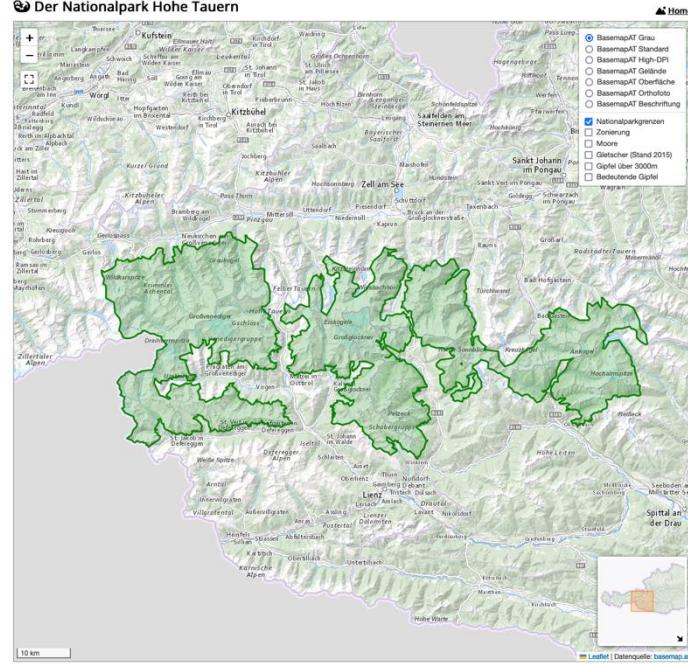
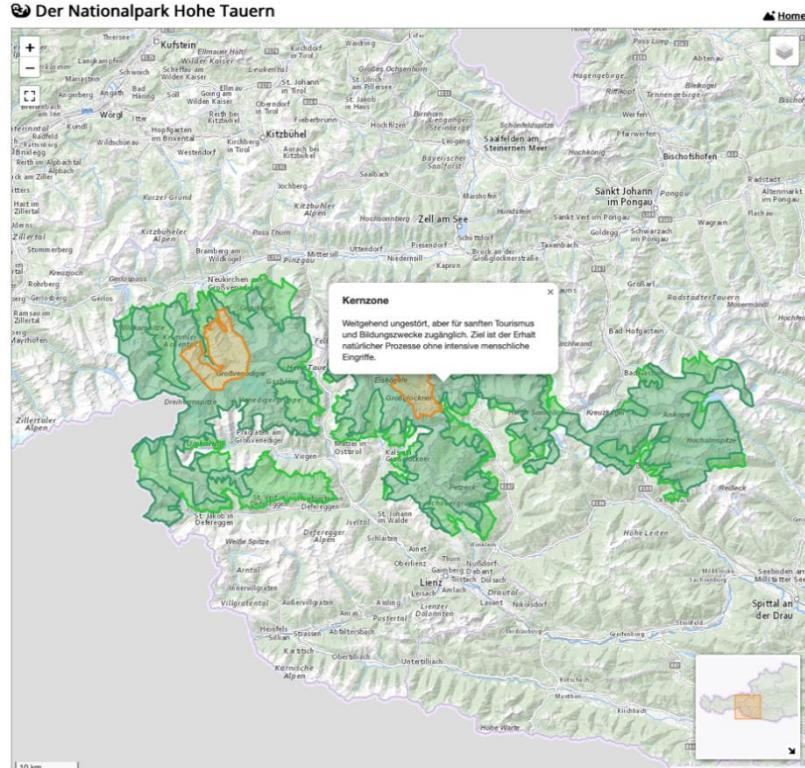


Abbildung 12: Übersicht der Detailkarte zum Nationalpark Hohe Tauern (eigene Darstellung 2024).



Datenquellen: Nationalpark Hohe Tauern parcs.at
Abbildung 13: Übersicht über die Zonierung des NPHT (eigene Darstellung 2024).

werden. Per Mausklick werden die Informationen über die jeweilige Zone angezeigt. Dies wurde analog zu Wien Übung über eine Abfrage integriert. Als Auswahlkriterium wurde der Zonenname festgelegt. Diese kann über feature.properties.ZONENAME abgefragt werden. Abhängig der Zone wurde der Infotext eingefügt. Dieser wurde ebenfalls der Webseite des NPHT entnommen.

Als nächstes wurden die im NPHT vorhandenen Moore integriert. Beim Datensatz handelt es sich um eine ESRI Geodatabase, welche mir von Florian Jurgeit vom Nationalpark Hohe Tauern für unser Projekt zur Verfügung gestellt wurde.

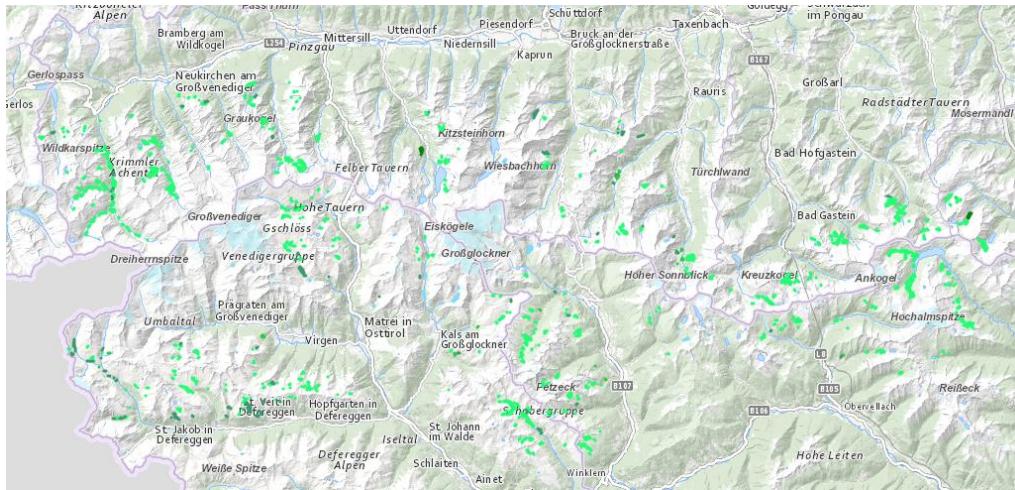


Abbildung 12: Moore im Nationalpark Hohe Tauern (eigene Darstellung 2024).

Die Moorinformationen wurde über einen Join in ArcGIS Pro an die Polygon-Shapefiles gehängt und anschließend als GeoJSON exportiert und in der Karte integriert. Für die Symbolisierung in der Karte wurde vier Moortypen unterschieden, welche in Abbildung 15 dargestellt sind. In den von Florian zu Verfügung gestellten Daten, war ebenfalls eine detaillierte Beschreibung des jeweiligen Moors, welche bei der Erhebung verfasst wurde. Da diese zu ausführlich sind, um sie als Popup darzustellen, wurde eine alternative Lösung entwickelt. Über die layer.on Funktion wird eine Abfrage des „feature-name“ sowie des „comment“-Felds gemacht. Die Informationen aus diesen Feldern werden über den Code in Abbildung 16 im index.html unter Karte dargestellt. Zusätzlich wird der Name des Moores sowie der Moortyp und die Moornutzung als Popup dargestellt. Diese werden aus den Properties des GeoJSON kopiert. Das Ergebnis dieser Funktion ist in Abbildung 18 abgebildet. Man erhält zu Beginn einen Infotext über das jeweilige

```
if (lineName.includes("Kalk-Niedermoor")) {
  lineColor = "#2ECC40";
} else if (lineName.includes("Kalk-Silikat-Niedermoor")) {
  lineColor = "#3D9970";
} else if (lineName.includes("Silikat-Niedermoor")) {
  lineColor = "#01FF70";
} else if (lineName.includes("Schwemmland")) {
  lineColor = "#7FDBFF";
```

Abbildung 15: Unterteilung der Moortypen (eigene Darstellung 2024).

```
onEachFeature: function (feature, layer) {
  if (feature.properties && feature.properties.KOMMENTAR) {
    layer.bindPopup(`

      <h4>${feature.properties.FONAME}</h4>
      <ul>
        <li>Moortyp: ${feature.properties.MOORTYP}</li>
        <li>Nutzung: ${feature.properties.NUTZUNG}</li>
      </ul>
    `);
  }
  layer.on('click', function () {
    if (feature.properties && feature.properties.KOMMENTAR && feature.properties.FONAME) {
      document.getElementById('feature-name').innerText = feature.properties.FONAME;
      document.getElementById('comment').innerText = feature.properties.KOMMENTAR;
    } else {
      document.getElementById('feature-name').innerText = 'No Feature Name';
      document.getElementById('comment').innerText = 'No comments available';
    }
  });
};
```

Abbildung 16: Layer.on Funktion zur Darstellung der Moorinformationen (eigene Darstellung 2024).

```
<article>
  <div id="map"></div>
  <div id="info" style="width: 100%; padding: 10px; overflow-y: auto;">
    <h2 id="feature-name"></h2>
    <p id="comment"></p>
```

Abbildung 17: Implementierung der layer.on Abfrage in das index.html (eigene Darstellung 2024).



Fusch an der Großglocknerstraße: Niedermoor bei der Walcher Hochalm

Der kartierte Lebensraum liegt unmittelbar westlich des Almgebäudes der Walcher Hochalm. Bei diesem Biotop handelt es sich um ein relativ großes, nur ganz flach geneigtes Niedermoor. Die Fläche wird von einem Hauptbach und mehreren kleinen Gerinnen durchzogen und ist gut wasserversorgt. Die Hauptmasse der Vegetation bilden Sauergräser, vor allem Carex nigra, Carex panicosa und Carex flava, bemerkenswert jedoch das relativ reichliche Auftreten von Molinia caerulea, die hier durchwegs rasig wächst. Es ist möglich, dass diese Fläche in früheren Zeiten einmal strewiesenähnlich bewirtschaftet wurde. Heute wird sie beweidet, wobei die Beweidung offensichtlich nur gering abrängliche Auswirkungen auf die Vegetationsausbildung nach sich zieht. Entlang der Bachläufe und in kleinen, etwas tiefer gelegenen Vernässungen tritt immer wieder Carex rostrata bestandsbildend auf. Im Umfeld des Niedermoors wechseln einander in kleineräumige Verzahnung Sievers-Nardete, Alneten viridis, Seslerio-Carecten semipervenitris und diverse Zwergrasformationen ab, die durch kleinräumige Lagerfuren und einige größere Felsbrocken mit Zwergstrauchvegetation gegliedert werden.

Im Zentralbereich des Niedermoors und zwar dort, wo der Bach von Norden in das Moor einströmt, ist der Kalkeinfluss deutlich größer, hier treten auch Carex davalliana und Eriophorum latifolium auf. Lokal kann von einem Carectum davallinae gesprochen werden. Auch kleinräumig schön ausgebildete Eriophorenscheuchen befinden sich in diesem Bereich. Aufgrund der generellen Niedermoortengarnitur, die zwischen Kalk- und Silikatzeigern steht, ist die Hauptmasse der Vegetation einem Amblystegio stellatae-Carectum diocaie zuzuordnen. Torfbildung ist in der relativ großen Weidefläche orographisch links vom Hauptbach in geringer Mächtigkeit gegeben. Verbundungstendenzen liegen nicht vor.

Prozentanteil der Moorfläche am Komplexbiotop: 80 %

Details der Vegetationsaufnahme:
Lage: großer Moorteil orographisch links vom Hauptbach
Flächengröße: 2 x 2 m
Deckung: 100 %
Exposition: -
Inklination: -

Abbildung 18: Ergebnisse des Moorlayers (eigene Darstellung 2024).

Dünn 0 %, Rest offene Wasserfläche

Abbildung 13: Fehlerhafte Beschreibungen der Moorkartierung (eigene Darstellung 2024).

Der nächste Layer in dieser Karte beschäftigt sich mit den Gletschern im NPHT. Hierfür wurde ein Datensatz von Fischer et al. (2015) verwendet, welcher das gesamte Gletscherinventar in Österreich mit Stand 2015 zeigt. Dieser Datensatz wurde auf das Gebiet des Nationalparks geclipped und dann als GeoJSON exportiert. In der Karte werden die Gletscherflächen in Blau symbolisiert (Abbildung 19).

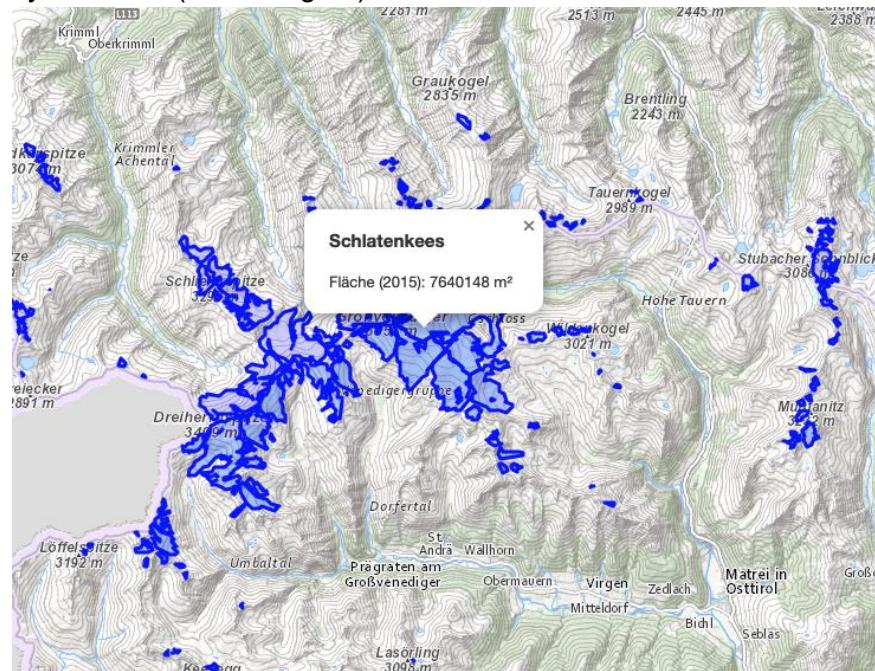


Abbildung 19: Gletscher im NPHT (eigene Darstellung 2024).

Moore ehe die Details der Mooraufnahme aufgelistet werden. Hierbei wurde aber bei der Datenaufbereitung festgestellt, dass nicht alle Beschreibungen korrekt aufgenommen wurden und es so zu fehlerhaften Darstellungen in der Moorinformation kommt. Diese Fehler sind in Abbildung XY dargestellt. Allerdings sind diese Fehler bereits in den Shapefiles zu finden, es kann sich somit um keine Fehler handeln, welche bei der Konvertierung in ein GeoJSON entstanden sind. Der Informationstext wurde korrekt eingetragen, was für diese Karte die wichtigeren Informationen sind.

Das dazugehörige Script ist vergleichsweise einfach und in Abbildung 20 dargestellt. Über die onEachFeature Funktion wird für jeden Gletscher zusätzlich zum Namen des Gletschers die dazugehörige Gletscherfläche aus dem Jahr 2015 dargestellt. Die Fläche wird über die feature properties berechnet und für die Darstellung im Popup auf m² gerundet.

```

fetch('Gletscherinventar2015.geojson')
  .then(response => response.json())
  .then(data => {
    // Create a GeoJSON layer and add it to the map
    L.geoJSON(data, {
      style: {
        color: 'blue'
      },
      onEachFeature: function(feature, layer) {
        // Check if the feature has properties and a name property
        if (feature.properties && feature.properties.name) {
          var areaInKM2 = feature.properties.Shape_Area.toFixed(0);
          layer.bindPopup(`

<h3>${feature.properties.name}</h3>
<p>Fläche (2015): ${areaInKM2} m2</p>
`);

        }
      }
    }).addTo(themaLayer.glaciers);
  })
  .catch(error => console.error('Error fetching GeoJSON data:', error));

```

Abbildung 20: Js-Script der Gletscherflächen (eigene Darstellung 2024).

Als nächster Layer werden die Berggipfel im Nationalpark mit einer Höhe von mehr als 3000 Metern dargestellt. Hierfür wurden die Gipfel aus dem Datensatz des digitalen Landschaftsmodells des BEV von Data.gv.at extrahiert.

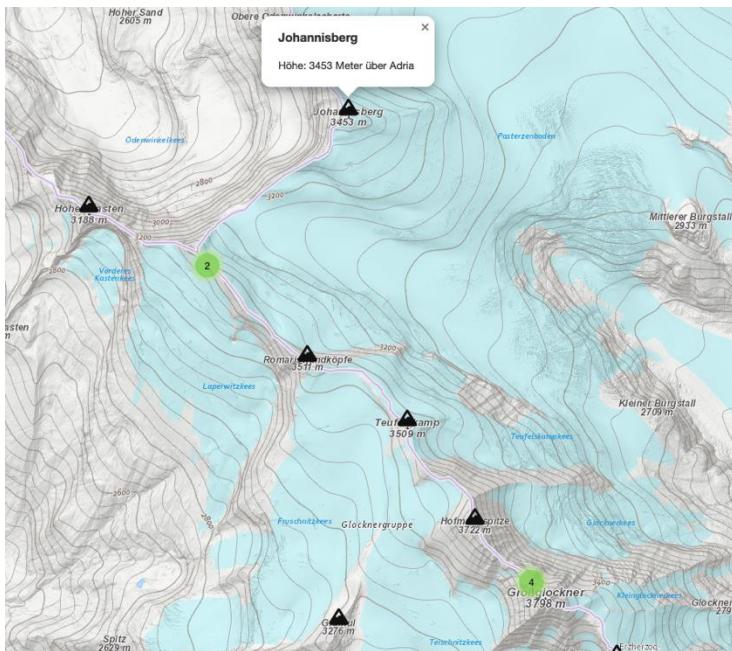


Abbildung 21: Übersicht über Gipfel im Bereich des Johannisbergs (eigene Darstellung 2024).

Dieser Datensatz enthält neben den Gipfel auch Siedlungsnamen, Gebietsnamen sowie Gewässernamen. Der Datensatz wird auf der Webseite als die umfassendste geographische Namensdaten für Österreich bezeichnet. Der Datensatz liegt im Shapefile-Format vor und wurde somit im Vorhinein mit GIS-Software bearbeitet. Aus dem Gesamtdatensatz wurden alle Gipfel entnommen und diese wiederum auf >3000m gefiltert, damit die Karte nicht überfüllt mit Gipfeln wird. Dieser modifizierte Datensatz wurde als GeoJSON exportiert und in das Projekt integriert. Wie bereits in Abbildung 21 ersichtlich ist, wurde als

Marker ein SVG-Berg verwendet. Ebenso wurde anstelle einer featureGroup eine markerClusterGroup für diesen Layer verwendet, was wiederum der Übersichtlichkeit in der Karte dient. In Abbildung 22 ist der dazugehörige Code dargestellt. Für die Symbolisierung wurde die Variable mountainIcon erstellt. Das dazugehörige Bild wurde von fontawesome integriert. Im Popup steht der Name des jeweiligen Gipfels sowie die Höhe.

```

//Gipfel über 3000m
fetch('Gipfel3000.geojson')
  .then(response => response.json())
  .then(data => {
    // Create a GeoJSON-layer and add it to the map
    L.geoJSON(data, {
      pointToLayer: function (feature, latlng) {
        // Create a marker with the Font Awesome mountain icon
        const mountainIcon = L.divIcon({
          html: '<i class="fa-solid fa-mountain" style="font-size:24px;color: #111111;"></i>',
          className: 'custom-div-icon', // class name for styling purposes
          iconSize: [24, 24], // size of the icon
          iconAnchor: [12, 24], // point of the icon which will correspond to marker's location
          popupAnchor: [0, -24] // point from which the popup should open relative to the iconAnchor
        });

        return L.marker(latlng, { icon: mountainIcon });
      },
      onEachFeature: function (feature, layer) {
        // Check if the feature has properties and a name property
        if (feature.properties && feature.properties.NAME) {
          layer.bindPopup(
            <h3>${feature.properties.NAME}</h3>
            <p>Höhe: ${feature.properties.HOEHE} Meter über Adria</p>
          );
        }
      }
    }).addTo(themaLayer.peaks);
  })
  .catch(error => console.error('Error fetching GeoJSON data:', error));
}

```

Abbildung 22: Js-Code der Gipfel (eigene Darstellung 2024).

Der letzte und aufwendigste Layer zeigt die bedeutenden Gipfel innerhalb des Nationalparks (siehe Abbildung 23). Diese Auswahl erfolgte allerdings nicht willkürlich, sondern nach dem oben bereits beschriebenen Datensatz. Diese klassifizierte 11 Gipfel innerhalb des Nationalparks als besonders bedeutend. Zu diesen Gipfel zählen:

1. Großglockner
2. Großvenediger
3. Großer Hafner
4. Kitzsteinhorn
5. Hoher Sonnblick
6. Große Wiesbachhorn
7. Ankogel
8. Hochschober
9. Hochalmspitze
10. Lasörling
11. Petzeck

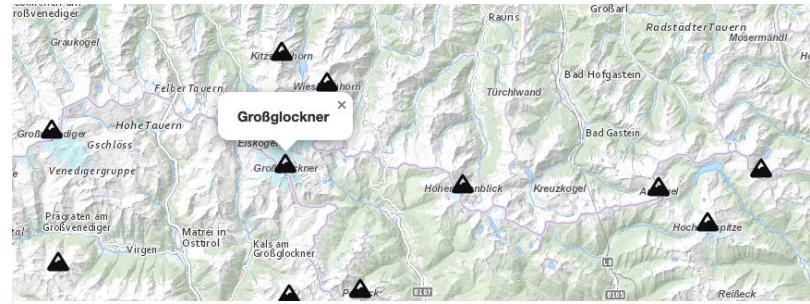


Abbildung 23: Bedeutende Gipfel im Nationalpark Hohe Tauern (eigene Darstellung 2024).

```

//Bedeutende Gipfel
fetch('Gipfel_Bedeutend.geojson')
  .then(response => response.json())
  .then(data => {
    // Create a GeoJSON-layer and add it to the map
    L.geoJSON(data, {
      pointToLayer: function (feature, latlng) {
        // Create a marker with the Font Awesome mountain icon
        const mountainIcon = L.divIcon({
          html: '<i class="fa-solid fa-mountain" style="font-size:24px;color:#111111;"></i>',
          className: 'custom-div-icon', // class name for styling purposes
          iconSize: [24, 24], // size of the icon
          iconAnchor: [12, 24], // point of the icon which will correspond to marker's location
          popupAnchor: [0, -24] // point from which the popup should open relative to the iconAnchor
        });

        return L.marker(latlng, { icon: mountainIcon });
      },
      onEachFeature: function (feature, layer) {
        // Check if the feature has properties and a name property
        if (feature.properties && feature.properties.NAME) {
          layer.bindPopup(
            <h3>${feature.properties.NAME}</h3>
            <p>Höhe: ${feature.properties.HOEHE} Meter über Adria</p>
          );
        }
      }
    }).addTo(themaLayer.peaks);
  })
  .catch(error => console.error('Error fetching GeoJSON data:', error));
}

```

Abbildung 24: Js-Code der bedeutenden Gipfel (eigene Darstellung 2024).

Warum genau diese Berge als besonders bedeutend klassifiziert wurden, geht aus dem Datensatz nicht hervor. Der erste Teil des Codes für diesen Layer ähnelt stark dem vorhergegangen. Mit der Ausnahme, dass ein anderes Datensatz als Input verwendet wurde. Er unterscheidet sich aber in der Hinsicht, als dass zusätzlich Information über den jeweiligen Gipfel dargestellt wird. Diese wird allerdings nicht im Popup dargestellt, sondern ähnlich dem Moor-Layer als Information unter der Karte. Diese Information wird allerdings nicht über ein

Feld im GeoJSON definiert, sondern wird über die Funktion `fetchWikipediaContent` direkt von der Wikipedia-Seite des jeweiligen Gipfel abgefragt. Der Code für diese Abfrage ist in Abbildung 25 dargestellt.

```

layer.on('click', function () {
  const peakName = feature.properties.NAME;
  fetchWikipediaContent(peakName);
});

}).addTo(themaLayer.peaks_important);

}).catch(error => console.error('Error fetching GeoJSON data:', error));

// Function to fetch Wikipedia content in German with text breaks
function fetchWikipediaContent(peakName) {
  const url = `https://de.wikipedia.org/w/api.php?action=query&origin=&format=json&prop=extracts&titles=${encodeURIComponent(peakName)}`;

  fetch(url)
    .then(response => response.json())
    .then(data => {
      const pages = data.query.pages;
      const pageId = Object.keys(pages)[0];
      const extract = pages[pageId].extract;
      const pageUrl = `https://de.wikipedia.org/wiki/${encodeURIComponent(peakName)}`;

      if (extract) {
        document.getElementById('wikipedia-content').innerHTML = `
          <h3>${peakName}</h3>
          ${extract}
          <p>Quelle: <a href="${pageUrl}" target="_blank">${pageUrl}</a></p>
        `;
      } else {
        document.getElementById('wikipedia-content').innerHTML = `
          <h3>${peakName}</h3>
          <p>Keine Wikipedia-Informationen verfügbar</p>
          <p>Quelle: <a href="${pageUrl}" target="_blank">${pageUrl}</a></p>
        `;
      }
    })
    .catch(error => console.error('Error fetching Wikipedia content:', error));
}

```

Abbildung 25: Funktion `fetchWikipediaContent` (eigene Darstellung 2024).

Hierfür wurde zuerst die Funktion als Zusatz zur `onEachFeature` Funktion die `layer.on` Funktion integriert. Diese speichert den Gipelnamen als Variable ab und gibt diesen als Input für die Funktion `fetchWikipediaContent` weiter. Diese Funktion nimmt den Text der Wikipedia Seite des Gipfels, falls vorhanden, und fügt in unterhalb der Karte ein. Zusätzlich fügt es am Ende des Textes einen Link zur jeweiligen Wikipedia-Seite ein, welcher in einem neuen Tab geöffnet wird. In Abbildung 26 ist das Resultat für den Hochschober dargestellt.



Hochschober

Der Hochschober ist mit einer Höhe von 3242 m ü. A. der vierthöchste und namensgebende Berg der Schobergruppe in den Hohen Tauern. Seine vergletscherte Nordflanke bietet von Kals einen beeindruckenden Anblick. Als 1845 der Geograf Adolf Schaubach die Schobergruppe als Gebirgsgruppe einführte, wählte er die Benennung nach dem Hochschober, da er ihn fälschlicherweise für den höchsten Berg südlich des Peischlachtörls hielt.

Der leichteste Anstieg erfolgt über die Hochschoberhütte (2322 m) von Süden über die Stanikascharte (2936 m) und den Westgrat. Eine von der Schwierigkeit vergleichbare Anstiegsmöglichkeit bietet sich von Osten und dem oberen Teil des Südostgrats an. Stützpunkt ist dabei die Lienzer Hütte (1977 m). Beide Anstiegswege sind markiert und im Hochsommer im Gegensatz zu früheren Jahren in der Regel schneefrei, weisen leichte Kletterschwierigkeiten (I) auf und sind stellenweise etwas ausgesetzt. Eine weitere Anstiegsmöglichkeit bietet der nur teilweise markierte Steig vom Schobertörl (2898 m) über den Nordostgrat, dabei wird der Kleinschober (3125 m) überschritten.

- Alpenvereinskarte Blatt 41, 1:25.000, *Schobergruppe*, ISBN 3-928777-12-2
- Richard Goedeke: *3000er in den Nordanalpen*, Bruckmann Verlag, München 2004, ISBN 3-7654-3930-4
- Walter Mair: *Alpenvereinführer Schobergruppe*. Bergverlag Rother, München 1979. ISBN 3-7633-1222-6

Quelle: <https://de.wikipedia.org/wiki/Hochschober>

Abbildung 26: Infotext von Wikipedia für den Hochschober (eigene Darstellung 2024).

4. Tourismus Nationalpark Hohe Tauern

Auf der Webseite zum Tourismus sollen die Almen bzw. Hütten und mögliche Wanderwege in der Region dargestellt werden. Dazu habe ich zunächst einen neuen Unterordner für meine Seite angelegt mit einem icons und Daten Ordner. Nachdem alle files angelegt wurden, wurde die Standardstruktur für index.html verwendet und die eigenen Hauptskripte wurden verknüpft.

Innerhalb des index.html wurden die klassischen Plug-Ins wie Leaflet, Leaflet-Providers, Font Awesome und die eigenen Stile eingefügt.

Daraufhin wurde mit dem Ausdruck L.map eine Karte auf der Seite initialisiert. Dazu wurde ein Punkt verwendet, der relativ zentral in unserem Untersuchungsausschnitt liegt. Dies war der Großglockner. Nachdem die Koordinaten dazu ausgelesen wurden, konnte ein Punkt dort gesetzt werden. Dieser wurde als Standardpunkt der Karte gesetzt, um immer zentral in der Karte landen zu können, der zoom wurde auf 10 gestellt. Zudem wurde die Skala der Karte eingestellt (Abbildung 27).

```
💡 Karte initialisieren
var map = L.map('map').setView([grossglockner.lat, grossglockner.lng], 10);
grossglockner.addTo(map)
```

Abbildung 27: Karte initialisieren

Danach wurden zudem mehrere Basemaps unter L.control hereingeladen, um mehr Variation bei der Darstellung zu haben und falls Leute unterschiedliche Eindrücke der Wanderwege erhalten wollen (blauer Bereich Abbildung 28). Die Auswirkungen können auf der rechten Seite gesehen werden (Abbildung 29).

```
L.control
  .layers({
    "BasemapAT Grau": startLayer,
    "BasemapAT Standard": L.tileLayer.provider("BasemapAT.basemap"),
    "BasemapAT High-DPI": L.tileLayer.provider("BasemapAT.highdpi"),
    "BasemapAT Gelände": L.tileLayer.provider("BasemapAT.terrain"),
    "BasemapAT Orthofoto": L.tileLayer.provider("BasemapAT.orthofoto"),
    "BasemapAT Beschriftung": L.tileLayer.provider("BasemapAT.overlay"),
  },
  {
    "Nationalparkgrenze": themaLayer.borders.addTo(map),
    "Almen": themaLayer.hut.addTo(map),
  })
  .addTo(map);
```

- BasemapAT Grau
- BasemapAT Standard
- BasemapAT High-DPI
- BasemapAT Gelände
- BasemapAT Orthofoto
- BasemapAT Beschriftung

Abbildung 29: Basemap Menü

Abbildung 28: Basemaps reinladen

Anschließend wurde ein weiteres Plug-In mit dem Namen Leaflet.Fullscreen implementiert. Dabei wurde folgender Code in das index.html geladen (Abbildung 30).

```
<!--Leaflet Fullscreen-->
<script src='https://api.mapbox.com/mapbox.js/plugins/leaflet-fullscreen/v1.0.1/Leaflet.fullscreen.min.js'></script>
<link href='https://api.mapbox.com/mapbox.js/plugins/leaflet-fullscreen/v1.0.1/leaflet.fullscreen.css'
      rel='stylesheet' />
```

Abbildung 30: HTML Fullscreen

Dazu musste der Befehl im main.js folgendermaßen in die map mit eingebunden werden (Abbildung 31).

```
// Karte initialisieren
var map = L.map('map', {
  fullscreenControl: true,
}).setView([großglockner.lat, großglockner.lng], 10);
gorßglockner.addTo(map)
```



Abbildung 31: Main.js Fullscreen

Des Weiteren wurde eine miniMap eingebaut, welche rechts unten in der Ecke platziert wurde. Diese wurde mit dem folgenden Plugin eingebaut (Abbildung 32).

```
<!-- Leaflet Minimap -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet-minimap/3.6.1/Control.MiniMap.css"
  integrity="sha512-efbAfGnrnjA+hLwOLu91W034fBGPsMwZMVCTwLUI2PDX/m7rOiuYZ+D2mZ8rKcpC/I/7pdgoL8T4eYvMHNQg==">
<script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet-minimap/3.6.1/Control.MiniMap.js"
  integrity="sha512-ceQPs2CHke3gSINLT/JV37W1rfJOM64yuH999hnRhtP7tNtcSBp5h1TKhn8CEIhsFweSBrZMPVotAKjoyxGWNg==">
  crossorigin="anonymous" referrerpolicy="no-referrer"></script>
```

Abbildung 32: Leaflet Minimap HTML

Zudem musste auch diese wiederum in main.js eingebaut werden (Abbildung 33). Daraus ergibt sich die Übersichtskarte in Abbildung 34.

```
new L.Control.MiniMap(
  L.tileLayer("https://wmts.kartetirol.at/gdi_summer/{z}/{x}/{y}.png", {
    attribution: `Datenquelle: <a href="https://www.data.gv.at/katalog/dataset/land-tirol_elektronische-karte-tirol">eGrundkarte Tirol</a>`}),
  {
    toggleDisplay: true,
  }
).addTo(map);
```

Abbildung 33: Leaflet Minimap main.js



Abbildung 34: Minimap in der Karte

Nachdem diese Plug-ins eingebaut waren, habe ich die ersten (Geo)Json Dateien eingeladen. Zunächst wurde ein Themenlayer in Form einer feature Group angelegt für die Grenzen des Nationalparks Hohe Tauern (Abbildung 35).

```
let themalayer = {
  borders: L.featureGroup().addTo(map),
  hut: L.markerClusterGroup({ disableClusteringAtZoom: 17 }),
}
```

Abbildung 35: Themalayer einfügen

Diese wurde dann wiederum an L.control weitergegeben, wie in Abbildung 33 zu sehen ist. Anschließend wurde das GeoJSON File eingelesen nach folgendem Code:

```

/*let polygons = new Array();*/
fetch("../NationalparkHoheTauern/npht_agrenze_new.geojson")
  .then(response => response.json())
  .then(data => {
    /* for(var i = 0; i< data.features[0].geometry.coordinates.length;i++){
      for(var j=0; j< data.features[0].geometry.coordinates[i].length;j++){
        polygons.push(data.features[0].geometry.coordinates[i][j])
        console.log("edit new polygon")
      }
    }*/
    L.geoJSON(data, [
      style: {
        color: 'green'
      },
      ]).addTo(themaLayer.borders);
  })
  .catch(error => console.error('Error fetching data:', error));

```

Abbildung 36: Nationalparksgrenzen laden

Ein ähnliches Prozedere wurde für die Almen verwendet. Der Unterschied besteht allerdings darin, dass im Themalayer keine featureGroup sondern ein Markercluster verwendet wurde, um die vielen Punkte innerhalb des Jsons beim rauszoomen zusammenzufassen, um eine bessere Übersichtlichkeit zu erhalten (Bsp. Abbildung 35).

Nachdem diese Schritte durchgeführt wurden, sieht der L.Control in der Mappe nun aus wie in Abbildung 37 abgebildet.

- BasemapAT Grau
- BasemapAT Standard
- BasemapAT High-DPI
- BasemapAT Gelände
- BasemapAT Orthofoto
- BasemapAT Beschriftung

- Nationalparkgrenze
- Almen

Abbildung 37: Kontrolle Layer

Zudem wurden nun im Almen-Json pop-ups mit den Namen und der Wirtschaftsform der Hütten eingebaut (Abbildung 38) und auf mapicons.mapsmaker wurden icons mit einer Hütte heruntergeladen, welche ebenfalls implementiert wurden. Das Ergebnis dieses Codes kann in Abbildung 39 nachvollzogen werden.

```

onEachFeature: function (feature, layer) {
  layer.bindPopup(` 
    <h4>${feature.properties.NAME}</h4>
    <ul>
      <li>Wirtschaftsform: ${feature.properties.OBJEKTBESCHREIBUNG} | 'keine Angabe'</li>
    </ul>
  `);
}
).addTo(themaLayer.hut);
})
.catch(error => {
  console.error('Error loading the JSON data:', error);
});

```

Abbildung 38: Pop-up bei Almen einbauen

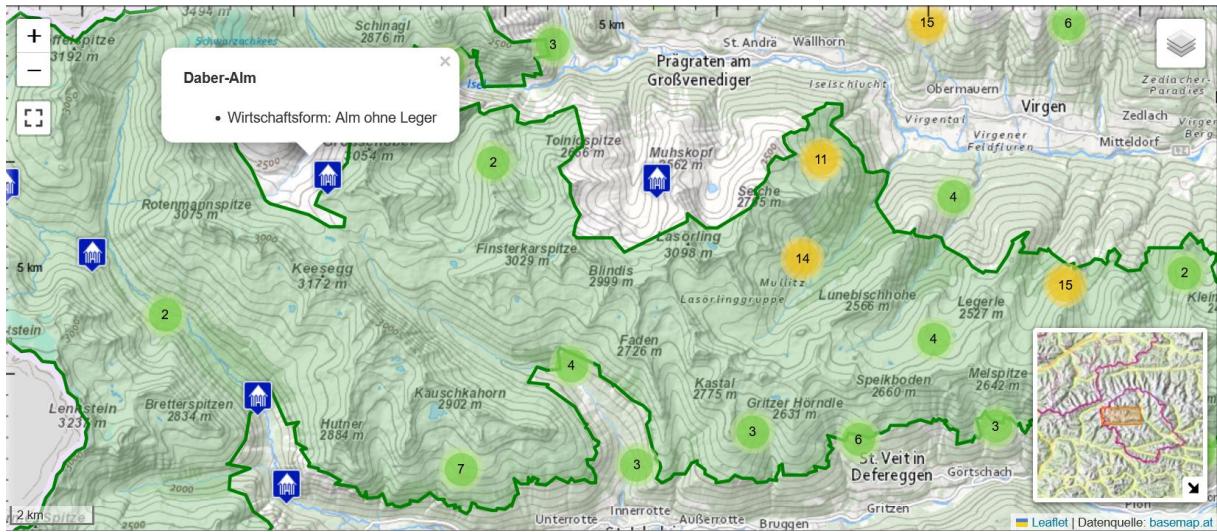


Abbildung 39: Beispiel für das Aussehen nach all den Schritten

Neben Almen sind allerdings auch viele Wanderwege für den Wandertourismus von Bedeutung, weshalb ich mir im nächsten Schritt vier voneinander unabhängige Wanderrouten ausgesucht habe. Diese wurden im folgenden Stil eingeladen (Abbildung 40).

```
let controlElevation_5 = L.control.elevation({
    time: false,
    elevationDiv: "#profile2",
    toggleDisplay: false,
    theme: "tracks",
    height: 200,
    collapsed:true,
}).addTo(map);
controlElevation_5.load("Daten/track1.gpx");
```

Abbildung 40: Einbinden GPX Daten

Nachdem wir im main.js von einem div profile2 reden, müssen wir dieses im index.html wieder einbinden (Abbildung 41).

```
<div id="profile2"></div>
```

Abbildung 41: div profil einbauen

Des Weiteren geben wir einen Stil an, indem wir theme: „tracks“ sagen. Dieser Stil musste auch spezifisch für die Profile erstellt werden (Abbildung 42).

```
.tracks{
    --ele-area: #78dcefa8;
    --ele-bg: white;
}
```

Abbildung 42: Stil für tracks

Nachdem dies erledigt ist, wird eine Linie innerhalb der Karte sichtbar mit einem Anfangs- und Endpunkt, sowie farblicher Einzeichnungen, teilweise Infos zu Hütten oder Stopps auf dem Weg. Zudem kann ein Profil gesehen werden, an welchem mit Hilfe der Maus vorbeigefahren werden kann, um zu sehen, wo man sich befindet (Vgl. Abbildung 43, Abbildung 44).

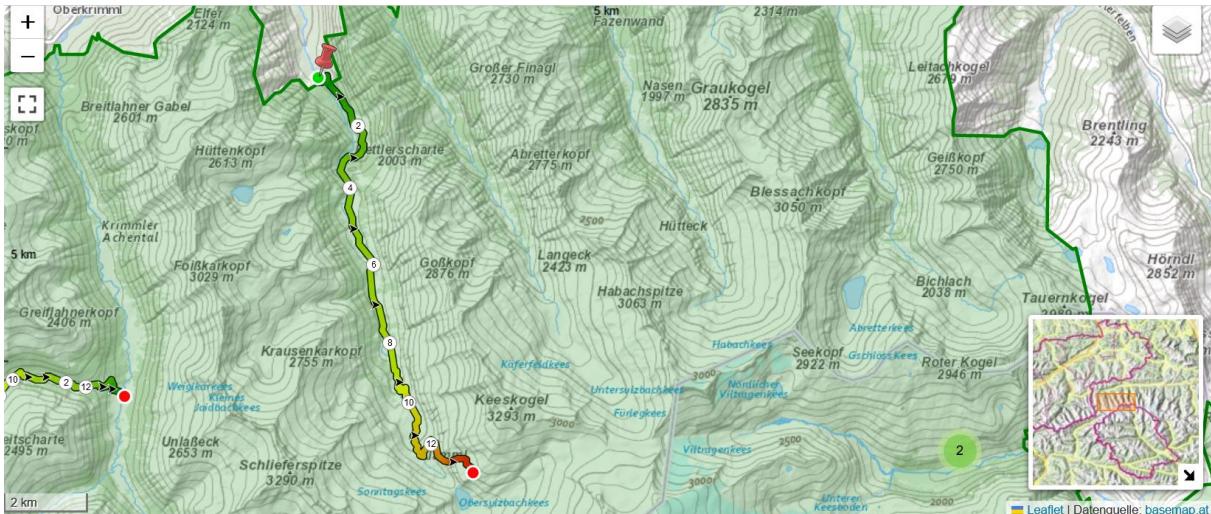


Abbildung 43: Übersicht Track 1

Höhenprofile Wanderwege Hohe Tauern

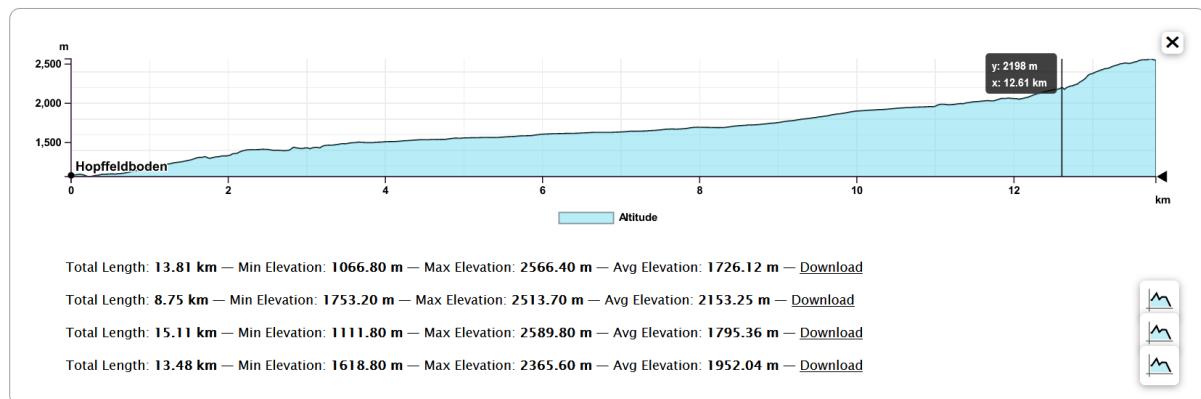


Abbildung 44: Höhenprofil Track 1

Beim Laden der Seite wurden festgelegt, dass die Höhenprofile alle geschlossen sind, damit die Seite nicht zu überladen ist. Dies wurde durch den Befehl collapsed: true (Abbildung 40) erreicht. Der Kasten wurde um die Höhenprofile aus ästhetischen Gründen gezogen und für die Übersicht eine Überschrift im index.html eingefügt.

Nachdem die vier unabhängigen Wanderwege eingeladen wurden, wurden auch die Etappen einer Wanderung verwendet. Diese wurden genauso wie die Daten in Abbildung 40 eingeladen und auch mit denselben Vorgaben im main.css gestylt, allerdings wurde für die bessere Übersicht ebenfalls ein eigener Block für die Profile eingerichtet und eine Überschrift hinzugefügt. Aufgrund der Schwierigkeit und Distanz fand ich den Kreuzeckhöhenweg besonders

```
<details closed>
  <summary>Weitere Infos</summary>
  <br>
  <div class="summary-badge">
    <i class="fas fa-walking"></i> 11,3 km
  </div>
  <div class="summary-badge">
    <i class="fas fa-clock"></i> 7 h
  </div>
  <div class="summary-badge">
    <i class="fas fa-sort-amount-up-alt"></i> 1649 m
  </div>
  <div class="summary-badge">
    <i class="fas fa-mountain"></i> 2006 m
  </div>
  <div class="summary-badge">
    <i class="fas fa-sort-amount-down-alt"></i> 18 m
  </div>
  <div class="summary-badge">
    <i class="fas fa-download"></i> <a href="Daten/kreuzeckhoehenweg_etappel.gpx">Download GPX</a>
  </div>
  <div class="summary-badge">
    <i class="fa-solid fa-globe"></i> <a href="https://www.outdooractive.com/de/route/hertour/nationalpark-region-hohe-tauern/kreuzeckhoehenweg-1-tag-moellbruecke-55m-salzkofelhuetten">Details</a>
  </div>
  <div>
    Gegenüber vom Bahnhofes von Möllbrücke, vorbei an der Kirche und unmittelbar danach links beginnt unsere Tour. Ein gut markierter und landschaftlich schön integrierter Zustieg über den sogenannten Sachsenweg führt uns über den Amaliaboden und der Harrachalm zur Rosssebe, weiter zur Salzkofelhütte, der Ausgangspunkt unserer Kreuzeck-Höhenwanderung. Unsere Hütte liegt direkt unterhalb des Salzkofelmassivs. Herrlich blauer Himmel und angenehme Temperaturen waren von Anfang an unsere Begleiter.
  </div>
</details>
```

```
/* CSS für HTML Text aus- einklappen */

details {
  border: thin solid;
  background: #78dcefa8;
  padding: 0.4em 1em;
  margin-bottom: 15px;
  border-radius: 10px;
}

summary {
  position: relative;
  padding-left: 1em;
  font-weight: 300;
}

details[open] summary {
  margin-bottom: 0.4em;
}
```

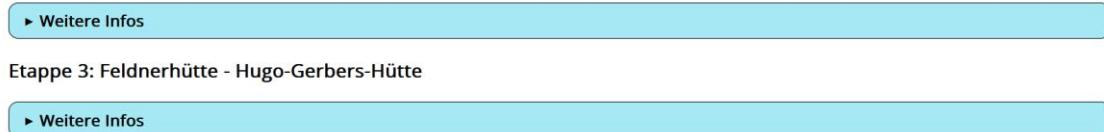
Abbildung 45: Zusammenstellung Code (links html, rechts css)

Dieser ausklappbare Text ist zum einen ästhetisch und zum anderen auch sehr praktisch, da nicht zu viel Text auf einmal zu sehen ist (vgl. Abbildung 45). Die Kurzinfos wurden jeweils mit badges dargestellt, in welchen Text und icons von Font-Awesome vorhanden sind. Das Ergebnis ist in Abbildung 46 dargestellt.

Etappe 1: Möllbrücke - Salzkofelhütte



Etappe 2: Salzkofelhütte - Feldnerhütte



Etappe 3: Feldnerhütte - Hugo-Gerbers-Hütte



Abbildung 46: Textdarstellung

Anschließend wollte ich buttons integrieren, die zwischen dem Nationalpark und dem nahegelegenen Kreuzeckhöhenweg hin und her zoomen. Dazu wurden im main.js zunächst Funktionen für die jeweilige Etappe formuliert (vgl. Abbildung 47).

```
function zoomToKreuzeckhoehenweg (){
    var bounds1 = controlElevation_1.getBounds();
    var bounds2 = controlElevation_2.getBounds();
    var bounds3 = controlElevation_3.getBounds();
    var bounds4 = controlElevation_4.getBounds();
    map.fitBounds(bounds1.extend(bounds2.extend(bounds3.extend(bounds4))));
}
function zoomToEtappe1 (){
    map.fitBounds(controlElevation_1.getBounds());
}
function zoomToEtappe2 (){
    map.fitBounds(controlElevation_2.getBounds());
}
function zoomToEtappe3 (){
    map.fitBounds(controlElevation_3.getBounds());
}
function zoomToEtappe4 (){
    map.fitBounds(controlElevation_4.getBounds());
}
function zoomToNP (){
    map.flyTo(gorßglockner.getLatLng(), 10);
}
```

Abbildung 47: Funktionen für Zoom on click

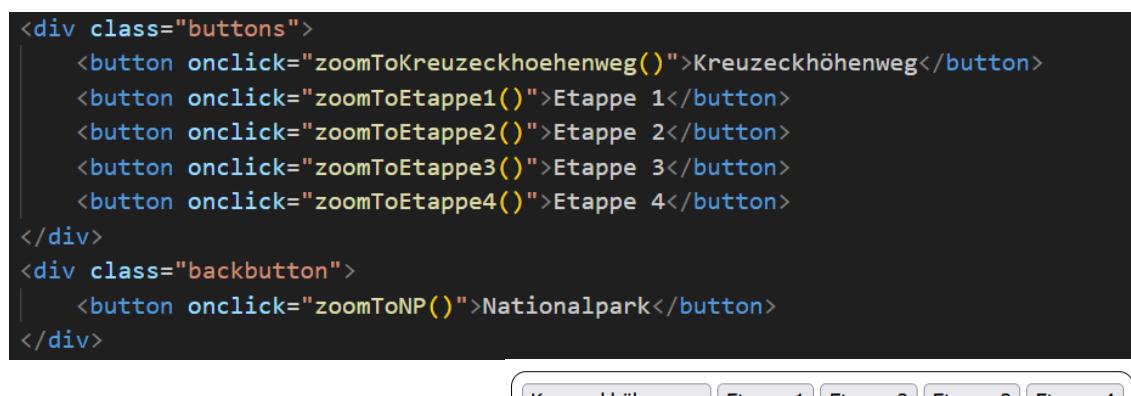


Abbildung 48: html Code zoom beim clicken und Darstellung auf der Webseite

Daraufhin habe ich im html buttons integriert, die dann entsprechend auch im css gestylt wurden (Abbildung 48).

Auffällig ist, dass in Abbildung 47 die Codes von den einzelnen Etappen zu der gesamten Übersicht der Wanderung und dem Zoom wieder auf den Nationalpark anders sind.

Dies kommt zustande, da die Funktion fit.Bounds nicht für Punkte gedacht ist. Da hier die anfangs verwendete Variable des Großglockners verwendet wurde, musste daher auf die Funktion flyTo zurückgegriffen werden. Des Weiteren kann in der Funktion fit.Bounds kein Zoom eingestellt werden, wodurch die Gesamtheit der Wanderung kaum abgebildet werden kann. Daher wurde hier die Etappen als einzelne Variablen definiert und dann wieder zusammen in das fit.Bounds gegeben. Durch dieses hin und her Zoomen und springen ist eine einfachere Navigation innerhalb der Karte möglich.

Am Ende wurden die Texte noch einmal ausformuliert und überarbeitet, die Quellen in den Footer geschrieben, ein paar zusätzliche Icons von FontAwesome verwendet und die Links zur Organisation in Github sowie zu den zwei Partnerseiten hergestellt. Daraus ergibt sich das Endergebnis in Abbildung 49.

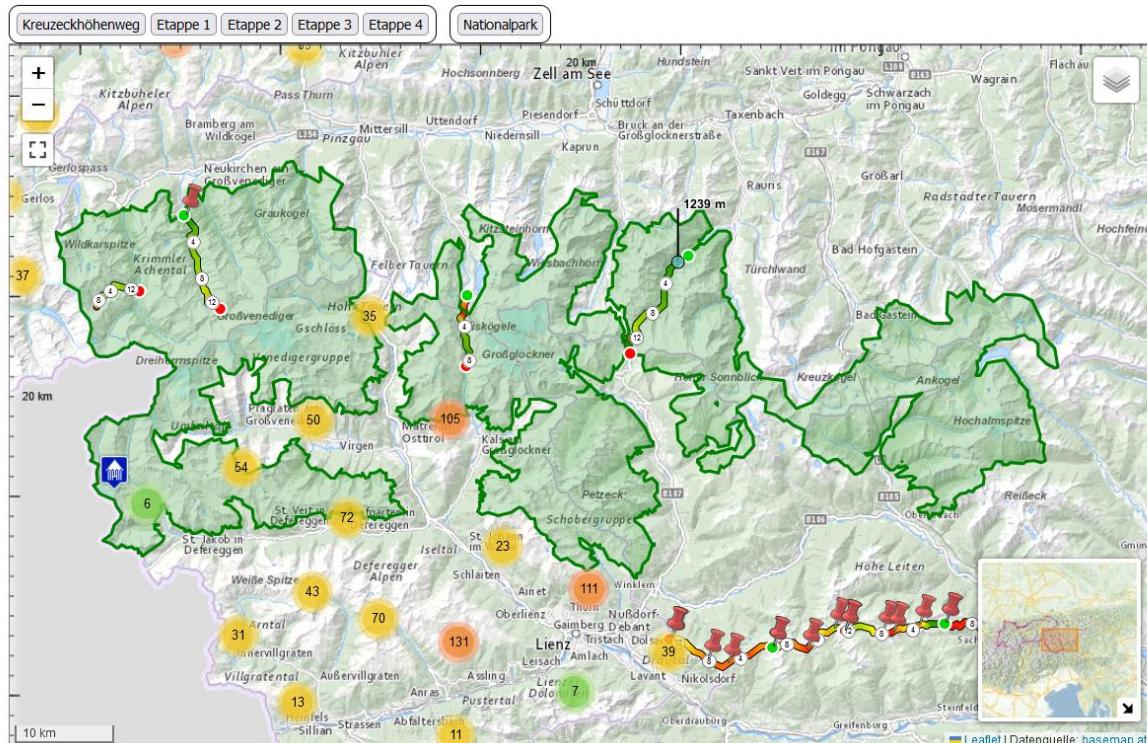


Abbildung 49: Endergebnis der Karte

5. Probleme und negative Aspekte

Während der Bearbeitung des Projektes ergaben sich für uns einige Probleme. Zum Ersten war die Datenverfügbarkeit oft sehr spärlich. Um an die gesuchten Datensätze zu gelangen musste oft in Kontakt mit den Organisationen getreten werden. Die Antworten blieben oft aus bzw. durften die Daten nicht verwendet werden. Hier möchten wir uns noch einmal bei Florian Jurgeit bedanken, welcher uns die Daten zum Nationalpark Hohe Tauern freundlicherweise zur Verfügung gestellt hat.

Ein weiteres Problem sind die Daten an sich. Hierbei hatten wir oft fehlerhafte oder unvollständige Datensätze zur Verfügung. Das macht es schwierig die Daten in unseren Karten zu integrieren. Als dritter Punkt ist, dass die Bedienung in Github an sich nicht intuitiv ist. Wenn drei Personen zur gleichen Zeit am gleichen Projekt arbeiten, schleichen sich immer kleine Fehler ein, welche dann das gesamte Projekt lahmlegen können. Diese Probleme lassen sich allerdings nie komplett ausschließen, da beim gemeinsamen Arbeiten in einem Projekt zwangsläufig immer Fehler entstehen werden.

Fragen die offen geblieben sind, sind beispielsweise das Zoomen der Karte immer auf die .gpx des Kreuzeckhöhenweg, obwohl die Karte auf setview des Großglockners gesetzt wurde.

Des Weiteren haben Plug-Ins nicht funktioniert wie Leaflet Search. Dieses konnte implementiert werden, aber erfüllt seinen Zweck leider nicht.

6. Quellen

Gletscherflächen:

[Fischer, Andrea; Seiser, Bernd; Stocker-Waldhuber, Martin; Abermann, Jakob \(2015\): The Austrian Glacier Inventory GI 3, 2006, in ArcGIS \(shapefile\) format \[dataset\]. PANGAEA, https://doi.org/10.1594/PANGAEA.844985,](https://doi.org/10.1594/PANGAEA.844985)

Zonierung:

<https://www.data.gv.at/katalog/dataset/BB946718-58AA-515B-B0A8-7640828D2C5A#resources>

Gipfel – Digitales Landschaftsmodell:

<https://www.data.gv.at/katalog/dataset/458469ad-b310-445a-ac64-d6e5b2a7db65>

Moore:

Zur Verfügung gestellt von Florian Jurgeit vom Nationalpark Hohe Tauern

GPX-Daten:

<https://www.outdooractive.com/de/>
<https://www.nationalpark.at/de/nationalpark/tourenportal#bm=osm%3Asummer&cat=Wander-main,Wanderung,Fernwanderweg,Pilgerweg,Themenweg&filter=r-fullyTranslatedLangus-,r-onlyOpened-,r-openState-,sb-sortedBy-0&fu=1&ov=hiking&zc=10,12.69539,47.21724>

Almen:

<https://www.data.gv.at/katalog/dataset/41760DF5-1EBF-424C-8251-EA0466BC2EE5>