

## **Bericht Webmapping**

### **Team Freizeitpass**

VU Geoinformatik Webmapping

MSc Geographie: Globaler Wandel – regionale Nachhaltigkeit

Sommersemester 2024, eingereicht am 25.06.2024

Teammitglieder:

Laura Eddeibüttel, Matrikelnummer: 52010234

Merit Koch, Matrikelnummer: 12345017

Ylenia Salucci, Matrikelnummer: 12022426

Lehrveranstaltungsleitung: Klaus Förster und Bernd Öggl

## Inhaltsverzeichnis

1.	Einleitung .....	1
2.	Beschreibungen der Seiten.....	2
2.1	Startseite .....	2
2.1.1	Aufbau und Inhalt .....	3
2.1.2	Benutzer:innenfreundlichkeit .....	9
2.1.3	Probleme und Herausforderungen.....	10
2.2	Wandern .....	11
2.2.1	Aufbau und Inhalt .....	11
2.2.2	Benutzer:innenfreundlichkeit .....	15
2.2.3	Probleme und Herausforderungen.....	16
2.3	Radfahren.....	17
2.3.1	Aufbau und Inhalt .....	17
2.3.2	Aufbau Javascript.....	20
2.3.3.	Benutzer:innenfreundlichkeit .....	21
2.3.3.	Probleme und Herausforderungen.....	22
3.	Fazit und Ausblick .....	23
	Quellenverzeichnis .....	24

## Abbildungsverzeichnis

Abbildung 1: Seitenübersicht / Navigation .....	1
Abbildung 2: Aufbau Visual Code Studio.....	2
Abbildung 3: Dropdown Startseite .....	3
Abbildung 4: Wettervorhersage Startseite .....	3
Abbildung 5: Div Container Videos .....	3
Abbildung 6: Slideshow Container .....	4
Abbildung 7: Slideshow .....	4
Abbildung 8: Slideshow Navigation.....	4
Abbildung 9: Slideshow .....	5
Abbildung 10: Links in einer unordered list; Screenshot aus Code und Endergebnis..	6
Abbildung 11: Übersichtskarte Startseite.....	6
Abbildung 12: Reachability Plugin.....	7
Abbildung 13: Reachability Plugin.....	7
Abbildung 14: Bounding Box .....	9
Abbildung 15: Scrollposition beim Neu laden.....	9
Abbildung 16: Popup Startseite.....	10
Abbildung 17: Wander index.html, main.css sowie Navigationsleiste der Homepage	11
Abbildung 18: index.html und Videos auf der Homepage .....	11
Abbildung 19: Karte und Höhenprofil Wandern.....	12
Abbildung 20: Screenshot aus data/gps-daten-karwendel-hoehenweg.gpx.....	13
Abbildung 21: Screenshot aus dem Wander index.html sowie der Seite Wandern...	13
Abbildung 22: Screenshots aus index_en.html und main.css zu den Slideshows ....	14
Abbildung 23: Screenshot der mobilen Version der Seite Wandern sowie dem Ausschnitt aus main.css.....	15
Abbildung 24: Navigationsleiste der Radseite (Website-Ansicht und HTML) .....	17
Abbildung 25: Übersicht Radseite - Highlight2 (Website-Ansicht).....	18
Abbildung 26: Beispiel ausgeklapptes Höhenprofil, Routenüberschrift inkl. Schwierigkeit und Beschreibung; Highlight zur Abgrenzung .....	19
Abbildung 27: Beispiel einer Radkarte mit Themalayern und Popup.....	20
Abbildung 28: Struktur der Radseite (Ordner für Bilder, Data + Icons; main.js und sechs map.js, index.html) .....	20
Abbildung 29: Beispiel Themalayer eines map.js (Konstante COFFEE und Popups) .	21
Abbildung 30: Scroll-to-top Button (Website-Ansicht) .....	21
Abbildung 31: Beispielkarte Rad mit Gesture Handling Plugin.....	22

## 1. Einleitung

Im Rahmen dieses Projekts wurde eine interaktive Webseite entwickelt, die Freizeitaktivitäten in und um Innsbruck präsentiert. Dabei ist das Ziel, eine benutzer:innenfreundliche und informationsreiche Plattform zu gestalten, welche sowohl für Einheimische als auch für Tourist:innen von Nutzen ist. Die Webseite gliedert sich in vier Hauptabschnitte (siehe Abb. 1):

Der erste Abschnitt ist die Startseite. Diese dient als zentrale Übersicht und stellt die beiden ausgewählten Freizeitaktivitäten sowie die geografische und kulturelle Region Innsbruck vor. Zusätzlich wird eine Wettervorhersage für die kommenden Tage mithilfe einer interaktiven Karte visualisiert.

Dank der Links auf der Startseite, kommt man auf drei weiterführende Seiten. Eine erste thematisiert die Wanderwege. Die Seite widmet sich den zwei bekanntesten Hütentouren im Karwendel und im Inntal. Sie stellt wesentliche Kartendaten und Informationen bereit, die für Wanderinnen und Wanderer von Bedeutung sind.

Mit dem zweiten Link gelangt man auf die Seite, die die Rennradwege in und um Innsbruck darstellt. Spezifische Rennradrouten in der Umgebung Innsbrucks werden kartografisch dargestellt und ausführlich beschrieben, um Radsportbegeisterten detaillierte Informationen und Planungsmöglichkeiten zu bieten.

Über den letzten Link gelangt man auf die Seite „Über dieses Projekt“, in dem über die Hintergründe des Projekts nachgelesen werden kann.



**Abbildung 1: Seitenübersicht / Navigation**

Zur Steigerung der Nutzer:innenfreundlichkeit und zur Erweiterung der Funktionalitäten wurden mehrere spezialisierte Leaflet-Plugins integriert. Diese Plugins ermöglichen eine visuell ansprechende und informationsreiche Darstellung der geografischen Daten. Beispielsweise wird das Leaflet.Fullscreen-Plugin (Leaflet 2018) genutzt welches

die Möglichkeit bietet, die Kartenansicht zu maximieren, wodurch die Benutzerfreundlichkeit erhöht wird.

Das übergeordnete Ziel dieses Projekts besteht darin, eine Plattform zu schaffen, die nicht nur detaillierte Informationen über Freizeitmöglichkeiten bereitstellt, sondern auch durch den Einsatz interaktiver Karten und Echtzeitdaten eine umfassende und dynamische Planung der Aktivitäten ermöglicht. Durch die Integration moderner Webtechnologien und Geoinformationssysteme soll die Webseite einen signifikanten Mehrwert für die Nutzer:innen darstellen.

## 2. Beschreibungen der Seiten

Nach der Erstellung der Github Organisation "Freizeitspass" und dem Hinzufügen aller Teilnehmerinnen wurde das Repository <https://freizeitspass.github.io> eingerichtet. Diese Seite dient als Startseite, während die Unterseiten ".../wandern" und ".../rad" sowie ".../ueber" das Gruppenprojekt vervollständigen. Um ein einheitliches Design zu ermöglichen, beziehen sich alle Seiten auf ein main.css. Dort wurden mit Hilfe einer class „highlight“ Padding-Boxen um alle gewünschten Bausteine gelegt. In Abb. 2 sieht man die Ordnerstruktur des Projekts. Für die beiden Unterseiten Wandern und Rad wurde je ein Ordner mit Bildern und einer mit Daten angelegt. Im Folgenden werden die drei Hauptseiten mit ihren Inhalten, verwendeten Plugins und Details beschrieben.

### 2.1 Startseite

Die Startseite dieses Webmapping-Projekts bietet eine zentrale Plattform zur Visualisierung und Planung von Freizeitaktivitäten in der Region Innsbruck. Ziel dieses Projekts ist es, durch interaktive Karten und präzise Wettervorhersagen eine umfassende Informationsquelle zu schaffen, die Nutzer:innen bei der Planung ihrer Unternehmungen unterstützt. Die Darstellung ausgewählter Aktivitäten sowie die Integration aktueller Wetterdaten sollen dazu beitragen, eine fundierte Entscheidungsgrundlage für Einwohner:innen und Besucher:innen zu schaffen. Diese Webseite dient nicht nur der Information, sondern auch der Inspiration, indem sie die Vielfalt der Freizeitmöglichkeiten in Innsbruck hervorhebt und erlebbar macht.

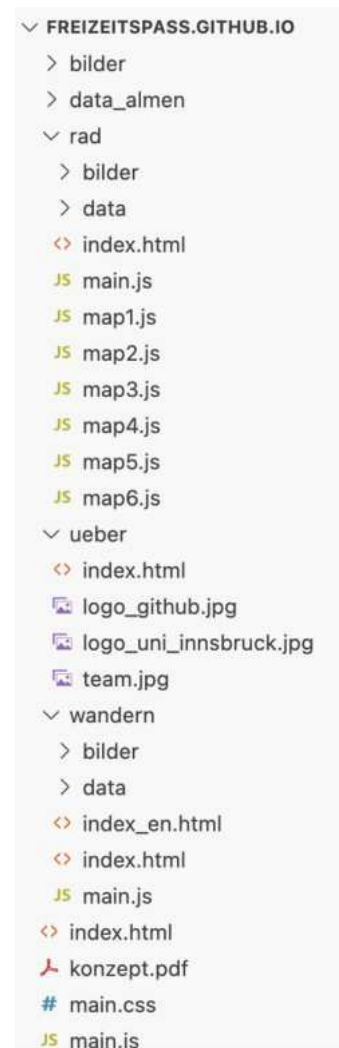
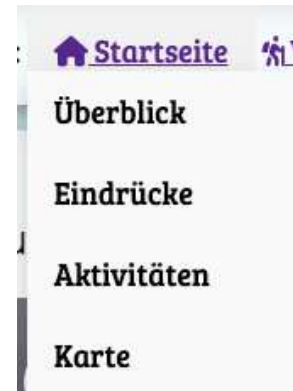


Abbildung 2: Aufbau Visual Code Studio

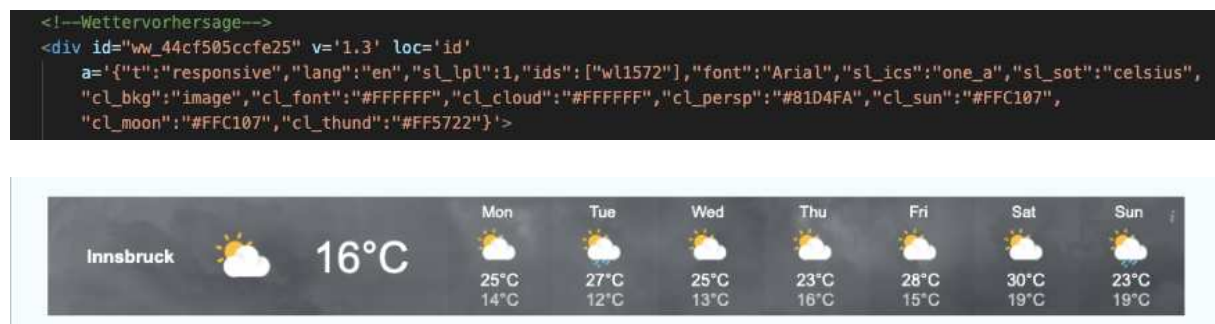
### 2.1.1 Aufbau und Inhalt

Das erste Element der Seite ist das Header. Für die Startseite wurde ein Foto von Pixabay benutzt (Rei 2020). Die Navigationsleiste, die sich sofort drunter befindet erlaubt es einem auf die anderen Seiten zu gelangen. Wie auch bei den anderen Seiten wurde hier ein Dropdown Menü integriert (siehe Abb. 3). Dieses erlaubt einem schnell zum gewünschten Abschnitt zu gelangen.



**Abbildung 3:**  
Dropdown Startseite

Im Hauptteil der Seite findet man als erstes eine Wettervorhersage für die ganze Woche. Diese wurde mit Hilfe eines frei verfügbaren Widgets hinzugefügt. Über die Internetseite Weather Widget (2024), kann man sich das gewünschte Design auswählen und kleine Änderungen vornehmen wie beispielsweise, in welcher Einheit die Temperaturen angezeigt werden sollen. Anschließend braucht man nur einen Code zu kopieren, der basierend auf den gewünschten Einstellungen erstellt wird. Dieser muss dann nur noch an der gewünschten Stelle im HTML-Code hinzugefügt werden, und schon hat man eine Wettervorhersage eingebaut (siehe Abb. 4).



**Abbildung 4:** Wettervorhersage Startseite

Als nächstes kommt eine Kurzbeschreibung zu Innsbruck und ein kurzes Video zu der Stadt. Diese wurden, genau wie bei den Videos zum Wandern und zum Radfahren, mit einer `<div class>` in die HTML-Datei eingefügt (siehe Abb. 5).



**Abbildung 5:** Div Container Videos

Weiter geht es dann mit einer Slideshow. Diese besteht aus Fotos von Innsbruck die selbst aufgenommen wurden. Sie ist dazu da, um den Nutzer:innenn Eindrücke von Innsbruck zu zeigen. Man kann die Bilder manuell durchblättern. Die Slideshow wurde mit Hilfe eines Tutorials von w3schools (o.D.) erstellt.

Die Slideshow befindet sich auch hier wieder in einem `<div class>` Container. Die Bilder an sich befinden sich dann auch nochmal in einem eigenen Container `<div class="mySlides fade">`. Hier beinhalten die Container eine Bildnummer und eine Beschreibung. Für jedes Foto muss ein eigener Container erstellt werden. Die aktuelle Position des Fotos sieht man bei `<div class="numbertext">`. Die Beschreibung des Fotos findet man bei `<div class="text">` (siehe Abb. 6).

```
<!-- Slideshow-->
<div class="slideshow-container">

  <!-- Bilder mit Nummer und Text -->
  <div class="mySlides fade">
    <div class="numbertext">1 / 4</div>
    
    <div class="text">Sicht auf die Nordkette (Copyright: Ylenia Salucci)</div>
  </div>
```

**Abbildung 6: Slideshow Container**

Um die Bilder navigieren zu können wurden 2 Pfeiltasten eingefügt. Der Code hierzu sieht man auf dem Screenshot. Die `onclick="plusSlides(-1)/(1)"` Funktion ist dafür zuständig, dass die Bilder in die gewünschte Richtung ändern (siehe Abb. 7).

```
<!-- Vorhäriges/nächstes Foto -->
<a class="prev" onclick="plusSlides(-1)">#10094;</a>
<a class="next" onclick="plusSlides(1)">#10095;</a>
</div>
<br>
```

**Abbildung 7: Slideshow Navigation**

```
<!-- Punkt unten zur Angabe welches Foto es ist -->
<div style="text-align:center">
  <span class="dot" onclick="currentSlide(1)"></span>
  <span class="dot" onclick="currentSlide(2)"></span>
  <span class="dot" onclick="currentSlide(3)"></span>
  <span class="dot" onclick="currentSlide(4)"></span>
</div>
```

**Abbildung 8: Slideshow**

Der nächste Teil des Codes ist dazu da, um eine Reihe von Punkten zur Slideshow einzufügen. Die Punkte zeigen den Nutzer:innen an, bei welchem Foto sie sich gerade befinden. Außerdem, durch die `onclick="currentSlide(x)"`-Funktion, kann man auf die einzelnen Punkte klicken, und die Slideshow springt sofort zum gewünschten Foto (siehe Abb. 8). Die Funktionalität der Slideshow wird durch JavaScript gewährleistet. Hier wird sichergestellt, dass die Bilder automatisch wechseln und die Benutzer:innen die Bilder manuell durchblättern können. Die Funktion `showSlides(n)` steuert die Anzeige der Bilder, wobei nur das aktuelle Bild sichtbar ist und die anderen ausgeblendet werden. Die `plusSlides(n)`-Funktion ermöglicht das Vor- und Zurückblättern, während die `currentSlide(n)`-Funktion direkt zu einem bestimmten Bild springt (siehe Abb. 9).



```
//Slideshow
let slideIndex = 1;
showSlides(slideIndex);

// Vorhäriges/nächstes Foto
function plusSlides(n) {
    showSlides(slideIndex += n);
}

// Thumbnail image controls
function currentSlide(n) {
    showSlides(slideIndex = n);
};

function showSlides(n) {
    let i;
    let slides = document.getElementsByClassName("mySlides");
    let dots = document.getElementsByClassName("dot");
    if (n > slides.length) { slideIndex = 1 }
    if (n < 1) { slideIndex = slides.length }
    for (i = 0; i < slides.length; i++) {
        slides[i].style.display = "none";
    }
    for (i = 0; i < dots.length; i++) {
        dots[i].className = dots[i].className.replace(" active", "");
    }
    slides[slideIndex - 1].style.display = "block";
    dots[slideIndex - 1].className += " active";
};
```

**Abbildung 7: Slideshow**

Durch die Implementierung dieser Slideshow auf der Startseite wird den Nutzer:innen eine dynamische und visuell ansprechende Möglichkeit geboten, die Region Innsbruck und ihre Freizeitaktivitäten kennenzulernen. Die Kombination aus interaktiven Elementen, Bildbeschreibungen und einfacher Navigation trägt wesentlich zur Benutzer:innenfreundlichkeit der Webseite bei.

Im nächsten Abschnitt werden noch alternative Aktivitäten aufgezählt, die man in Innsbruck machen kann. Hierfür wurde eine `<ul>` angelegt, also eine unordered list. Eine ungeordnete Liste, im HTML als `<ul>`-Element bekannt, ist eine häufig verwendete Methode, um eine Sammlung von Elementen darzustellen, die keine spezifische Reihenfolge erfordern. Jedes Element innerhalb der Liste wird durch das `<li>`-Element (list item) repräsentiert. In diesem Beispiel wurde noch ein Link hinzugefügt wie man am `<a href="">` Element erkennt. Wenn man also auf den Text klickt, wird man zu



einer Website weitergeleitet, auf der man mehr Infos zu der jeweiligen Aktivität findet (siehe Abb. 10).

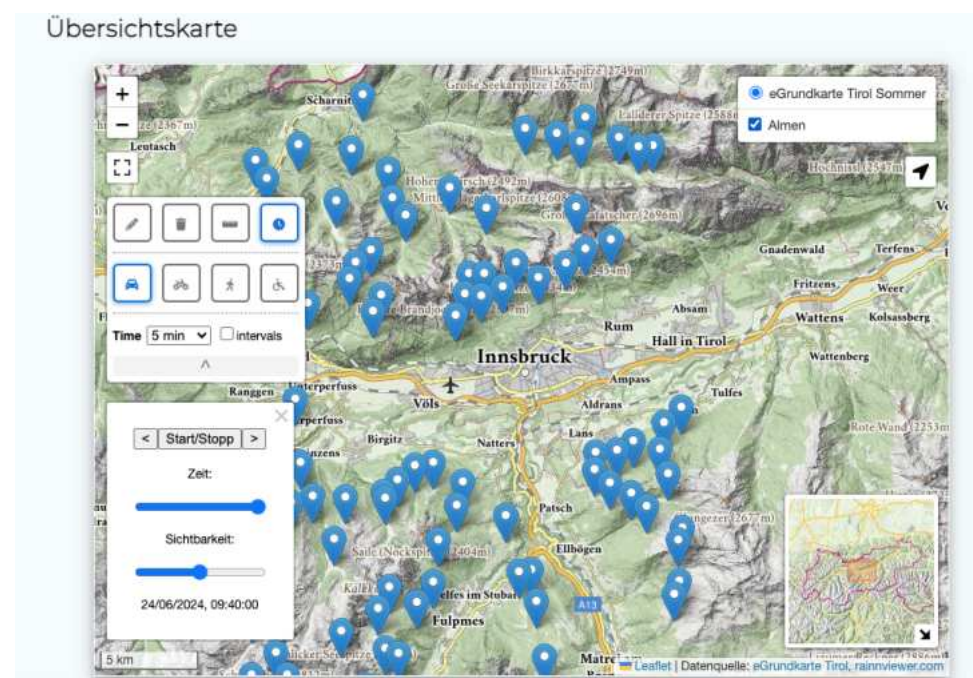
```
<ul>
<li><a href="https://www.tirol.at/reisefuehrer/ausflugstipps/ausflugsziele/a-alpenzoo-innsbruck"> Alpenzoo, in dem
man die heimische Tierwelt entdecken kann</a></li>
<li><a href="https://www.tirol.at/reisefuehrer/ausflugstipps/ausflugsziele/a-audioversum-abenteuerhoeren">
interaktive Erlebnis im Audioversum, das die Welt des Hörens erforscht,</a></li>
<li><a href="https://www.tirol.at/reisefuehrer/ausflugstipps/ausflugsziele/a-aut-architektur-innsbruck">
architektur und tirol, das spannende Einblicke in Architektur und Geschichte bietet,</a></li>
<li><a href="https://www.tirol.at/reisefuehrer/ausflugstipps/badeseen/badegewaesser/a-baggersee-rossau"> Baggersee
für entspannte Stunden am Wasser,</li></a>
<li><a href="https://www.tirol.at/reisefuehrer/ausflugstipps/ausflugsziele/a-bergisel-schanze-innsbruck">
Bergisel-Schanze mit ihrem historischen und sportlichen Erbe,</li></a>
<li><a href="https://www.tirol.at/reisefuehrer/ausflugstipps/badeseen/badegewaesser/
a-erlebnisoase-olympisches-dorf">Erlebnisoase Olympisches Dorf, ideal für Sport und Entspannung.</li></a>
</ul>
```

- [Alpenzoo, in dem man die heimische Tierwelt entdecken kann](https://www.tirol.at/reisefuehrer/ausflugstipps/ausflugsziele/a-alpenzoo-innsbruck)
- [interaktive Erlebnis im Audioversum, das die Welt des Hörens erforscht,](https://www.tirol.at/reisefuehrer/ausflugstipps/ausflugsziele/a-audioversum-abenteuerhoeren)
- [architektur und tirol, das spannende Einblicke in Architektur und Geschichte bietet,](https://www.tirol.at/reisefuehrer/ausflugstipps/ausflugsziele/a-aut-architektur-innsbruck)
- [Baggersee für entspannte Stunden am Wasser,](https://www.tirol.at/reisefuehrer/ausflugstipps/badeseen/badegewaesser/a-baggersee-rossau)
- [Bergisel-Schanze mit ihrem historischen und sportlichen Erbe,](https://www.tirol.at/reisefuehrer/ausflugstipps/ausflugsziele/a-bergisel-schanze-innsbruck)
- [Erlebnisoase Olympisches Dorf, ideal für Sport und Entspannung.](https://www.tirol.at/reisefuehrer/ausflugstipps/badeseen/badegewaesser/a-erlebnisoase-olympisches-dorf)

**Abbildung 8: Links in einer unordered list; Screenshot aus Code und Endergebnis**

Am Schluss der Seite findet man eine Karte (siehe Abb. 11). Diese wurde mit Hilfe von Leaflet implementiert. Die Karte enthält einige Plugins. Wie auch bei den anderen Karten, enthält diese die Minimap (Norkart 2018), die Fullscreen controle (B. 2023), das Locate Plugin (Moritz 2024) sowie einen Rainviewer (Marcin 2023).

Ein weiteres Plugin, das hier verwendet wurde ist das Reachability Plugin (traffordDataLab 2020). Dieses erkennt man in Abb. 11 mittig links. Das Leaflet Reachability Plugin ist ein leistungsstarkes Werkzeug zur Darstellung von Erreichbarkeitszonen auf Karten. Es ermöglicht Benutzer:innen, verschiedene Reiseentfernungen oder -zeiten



**Abbildung 9: Übersichtskarte Startseite**

visuell zu erkunden und zu analysieren, indem es auf OpenRouteService (ORS) zugreift. Zu den Hauptfunktionsweisen des Plugins gehören die Reichweite eines bestimmten Punktes innerhalb eines bestimmten Zeit- oder Entfernungsrahmens zu visualisieren und das Auswählen verschiedener Fortbewegungsmöglichkeiten wie beispielsweise Auto oder zu Fuß.

Der erste Schritt zum Einbinden des Plugins ist wie immer der gleiche. Man muss zuerst die Leaflet Bibliothek im <head> Teil der HTML-Datei einbinden (siehe Abb. 12).

```
<!--Reachability-->
<link rel="stylesheet"
  href="https://cdn.jsdelivr.net/gh/trafforddatalab/leaflet.reachability@v2.0.1/leaflet.reachability.css" />
<script
  src="https://cdn.jsdelivr.net/gh/trafforddatalab/leaflet.reachability@v2.0.1/leaflet.reachability.js"></script>
```

**Abbildung 10: Reachability Plugin**

In Javascript muss man dann das Plugin initialisieren. Damit das Plugin auch funktioniert, braucht man einen API-Schlüssel von OpenRouteService. Die OpenRouteService API (ORS API) ist ein leistungsstarker und vielseitiger Webdienst, der geografische Daten und Routinginformationen bereitstellt. Entwickelt von der Universität Heidelberg, ermöglicht es Entwickler:innen, verschiedene Routing- und Geoinformationsdienste in ihre Anwendungen zu integrieren. ORS bietet viele verschiedene Funktionsweisen wie beispielsweise das Routing. Die ORS API bietet verschiedene Routingdienste, einschließlich Auto, Fahrrad, Fußgänger:innen, Schwerlastverkehr und Rollstuhl. Sie unterstützt unterschiedliche Routingprofile, wie schnellste Route, kürzeste Route oder Route mit den wenigsten Höhenmetern. Ein weitere Funktion, die vor allem für dieses Projekt wichtig ist, ist das der Isochronen und Isodistanzen. Diese Funktionen ermöglichen es, Zonen gleicher Reisezeit (Isochronen) oder gleicher Reiseentfernung (Isodistanzen) um einen bestimmten Punkt zu berechnen. Dies ist nützlich für die Analyse der Erreichbarkeit und Reichweite (openrouteservie, 2024).

Um nun einen eigenen API-Schlüssel zu erhalten, muss man nur ein Account erstellen, oder sich mit Github anmelden. Anschließend wird einem ein Schlüssel generiert, den man dann in seinen Projekten nutzen kann. Wenn man diesen Schritt gemacht hat, kann man dann den nötigen Code des Plugins auch in Javascript einfügen (siehe Abb. 13).

```
//Reachability
var reachability = L.control.reachability({
  apiKey: '5b3ce3597851110001cf62483132be05085146fa9c513961e12c7208',
  opacity: 0.6,
  range: [5, 10, 15],
  label: 'Erreichbarkeit',
  serviceUrl: 'https://api.openrouteservice.org/v2/isochrones',
  travelModeButton1Content: '',
  travelModeButton1StyleClass: 'fa fa-car',
  travelModeButton2Content: '',
  travelModeButton2StyleClass: 'fa fa-bicycle',
  travelModeButton3Content: '',
  travelModeButton3StyleClass: 'fa fa-person-walking',
  travelModeButton4Content: '',
  travelModeButton4StyleClass: 'fa fa-wheelchair',
  drawButtonContent: '',
  drawButtonStyleClass: 'fa fa-pencil',
  deleteButtonContent: '',
  deleteButtonStyleClass: 'fa fa-trash',
  distanceButtonContent: '',
  distanceButtonStyleClass: 'fa fa-ruler-horizontal',
  timeButtonContent: '',
  timeButtonStyleClass: 'fa fa-clock',
}).addTo(map);

map.on('click', function (e) {
  reachability.addTo(map).setLatLng(e.latlng);
  reachability.queryService({
    locations: [[e.latlng.lng, e.latlng.lat]]
  });
});
```

**Abbildung 11: Reachability Plugin**

Im Rahmen eines Projekts zur Darstellung von Freizeitaktivitäten in und um Innsbruck könnte das Reachability Plugin verwendet werden, um die Erreichbarkeit von bestimmten Punkten wie Wanderwegen, Radrouten oder Sehenswürdigkeiten zu visualisieren. Benutzer:innen können durch Klicken auf die Karte sofort sehen, wie weit sie in einer bestimmten Zeit oder Entfernung von einem Punkt aus gelangen können, abhängig von der gewählten Reiseart.

Außerdem wurden in die Karte noch Standorte von Almen rund um Innsbruck hinzugefügt. Die nötigen Daten stammen von Open Government Data vom Land Tirol. Es besteht die Möglichkeit, die Daten in verschiedenen Formaten herunterzuladen. In diesem Beispiel wurden die Daten im GeoJSON Format genutzt.

In einem nächsten Schritt wurden die Daten in einen Ordner gepackt und dann in das Projekt in Visual Studio Code eingefügt. Der Befehl `fetch('data_almen/Almenzentren.geojson')` ist ein Aufruf, um die Datei `Almenzentren.geojson` aus dem Verzeichnis `data_almen` zu laden. Der `fetch` Befehl ist eine Methode, um Daten asynchron zu laden. Die nächste Etappe im Code ist `.then(response => response.json())`. Sobald die Datei geladen ist, wird die Antwort (`response`) in das JSON-Format umgewandelt. Dies geschieht, weil die Antwort ursprünglich als Response-Objekt vorliegt und in eine lesbare JSON-Struktur umgewandelt werden muss. In einem letzten Schritt benötigt man `.then(data => { ... })`. Sobald die JSON-Daten vorliegen, wird der nächste Schritt eingeleitet, nämlich das Verarbeiten dieser Daten. In einem nächsten Schritt wurde eine *Bounding Box* implementiert. Diese hat dabei geholfen, die Daten zu filtern und nur die Almen in einem bestimmten Perimeter anzuzeigen. Die 4 Zeilen mit den Koordinaten geben eine Grenze des Bereich ein in den 4 Himmelsrichtungen, also Süden, Norden, Westen und Osten. Der 2. Teil des Codes ist dafür da die Features im Array zu durchlaufen und herauszufinden welche in die Bedingungen passen. Genauer werden die Koordinaten der Daten analysiert. Die Bedingung innerhalb der `filter`-Methode prüft, ob die Koordinaten des Features innerhalb der definierten Bounding Box liegen (siehe Abb. 14). Im letzten Teil des Codes werden nur die Features im `filteredFeatures`-Array gespeichert, die innerhalb der Bounding Box liegen. Der letzte Schritt ist wie immer im Javascript, dass man den Layer zur Map hinzufügt.

### 2.1.2 Benutzer:innenfreundlichkeit

Die Hauptfunktionen die eingebaut wurden, sind auf den drei Seiten die gleichen. Der Button, der am Rand der Seite erscheint, bringt einen wieder an den Anfang der Seite und die Seite wurde für die Anwendung auf dem Handy angepasst. Wie diese zwei Einstellungen funktionieren wird in Kapitel 2.3.3 näher erklärt.

```
// Bounding Box für Innsbruck
let bbox = {
  minLat: 47.1000, // Geänderte Min-Latitude
  maxLat: 47.4000, // Geänderte Max-Latitude
  minLng: 11.2000, // Geänderte Min-Longitude
  maxLng: 11.5000 // Geänderte Max-Longitude
};

let filteredFeatures = data.features.filter(feature => {
  let [lng, lat] = feature.geometry.coordinates;
  return lat >= bbox.minLat && lat <= bbox.maxLat && lng >= bbox.minLng && lng <= bbox.maxLng;
});

let filteredData = {
  type: 'FeatureCollection',
  features: filteredFeatures
};
```

Abbildung 12: Bounding Box

Ein weitere Funktion die auf der Startseite eingebaut wurde ist, dass wenn die Seite neu-geladen wird, dann wird man nicht wieder an den Anfang gebracht sondern die Seite geht wieder da auf wo man vorher war. Hierbei war die Idee, dass wenn beispielsweise durch eine schlechte Internetverbindung die Seite neu geladen wird, man wieder dort weiter machen kann, wo man auf der Seite stehen geblieben ist (siehe Abb. 15).

```
// Speichern der Scrollposition vor dem Neuladen der Seite
window.addEventListener('beforeunload', function() {
  localStorage.setItem('scrollTop', window.scrollY);
});

// Wiederherstellen der Scrollposition nach dem Neuladen der Seite
window.addEventListener('load', function() {
  if (localStorage.getItem('scrollTop') !== null) {
    window.scrollTo(0, parseInt(localStorage.getItem('scrollTop')));
  }
});
```

Abbildung 13: Scrollposition beim Neu laden



### 2.1.3 Probleme und Herausforderungen

Auch hier kam es zu Problemen und auch Funktionen, die am Ende leider nicht funktioniert haben. Mit Hilfe von ChatGPT und einigen Google Recherchen konnten ein paar Probleme behoben, jedoch leider nicht alle.

Eine erste Funktion die leider wieder entfernt werden musste war die, von einem Popup Fenster. Ursprünglich war geplant, dass über der Karte ein Popup Fenster erscheinen sollte, das dazu da war die Reachability Funktion zu erklären. Dabei ist jedoch das Problem aufgetreten, dass das Fenster beim Neu laden der Seite nie an der richtigen Stelle war. Entweder befand es sich ganz unten rechts in der Ecke oder irgendwo in der Mitte der Seite (siehe Abb. 16).



**Abbildung 14: Popup Startseite**

Ein weiteres Problem, das aufgekommen ist, ist im Zusammenhang mit dem Rainviewer Plugin. Dadurch, dass die Seite jedes Mal neu laden muss, wenn das Plugin benutzt wird, springt nach dem Aktivieren die Seite immer wieder an den Anfang. Hierfür konnte leider keine Lösung gefunden werden, sodass man dies nun in Kauf nehmen muss.

Auch das Einfügen der Almen hat sich am Anfang schwieriger erwiesen als gedacht. Docht trat immer wieder das Problem auf, dass die Plugins nicht mehr funktioniert haben, sobald ein extra Layer für die Almen hinzugefügt wurden. Jedoch hat dies dann mit Hilfe der Session Notes und auch ChatGPT irgendwann funktioniert.

## 2.2 Wandern

Die Wander-Seite besteht aus einer deutschen Version (index.html) und einer englischen (index\_en.html). Im Folgenden wird zunächst der Inhalt mit den verwendeten Plugins beschrieben. Im Anschluss wird die Benutzer:innenfreundlichkeit diskutiert sowie Probleme und Herausforderungen erläutert.

### 2.2.1 Aufbau und Inhalt

Wie auf jeder der vier Seiten ist oben ein Header-Bild (Koch 2024) mit der Navigationsleiste zwischen den verschiedenen Seiten. Mit Hilfe eines Dropdown Menüs kann zwischen den verschiedenen Abschnitten der Seite gesprungen werden. Um die verschiedenen Dropdown-Menüs der Seite getrennt ansteuern zu können, wurde im main.css die zwei classes „dropdown“ und „dropdown1“ erstellt (siehe Abb. 17). Ohne diesen Zwischenschritt wurden beide Menüs immer gleichzeitig geöffnet.

```
<nav>
FREIZEITPASS IN INNSBRUCK:
<a href="https://freizeitspass.github.io"><i class="fa-solid fa-house"></i>
  Startseite</a>
<div class="dropdown1">
  <a href="https://freizeitspass.github.io/wandern" class="dropdown1-toggle"><i
  class="fa-solid fa-person-hiking"></i> Wanderwege</a>
  <div class="dropdown1-content">
    <a href="#ueberblick">Überblick</a>
    <a href="#map">Karte</a>
    <a href="#karwendel">Karwendel Höhenweg</a>
    <a href="#inntal">Inntaler Höhenweg</a>
  </div>
</div>
281 /* Dropdown-Stile für Rad und Wandern (1) */
282 .dropdown,
283 .dropdown1 {
284   position: relative;
285   display: inline-block;
286 }
287
288 .dropdown-content,
289 .dropdown1-content {
290   display: none;
291   position: absolute;
292   background-color: #f9f9f9;
```



Abbildung 15: Wander index.html, main.css sowie Navigationsleiste der Homepage

Darunter beginnt der <main> Teil der Seite mit einer Kurzbeschreibung der beiden Weitwanderwege sowie zwei Videos. Diese sind von Youtube mit Hilfe eines „embed“ Links in einer <div> class mit <iframe> eingebunden (siehe Abb. 18). Im main.css

```
<div class="video-container">
  <iframe width="450" height="300" src="https://www.youtube.com/embed/VH3fwl1NzY" allowfullscreen></iframe>
  <iframe width="450" height="300" src="https://www.youtube.com/embed/0aKnbnjHxFQ" allowfullscreen></iframe>
</div>
```

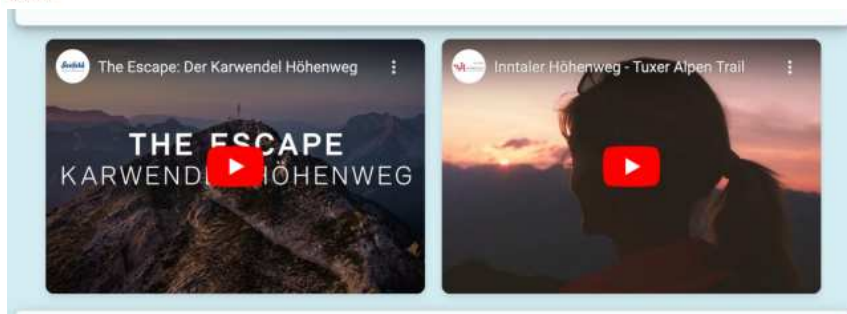
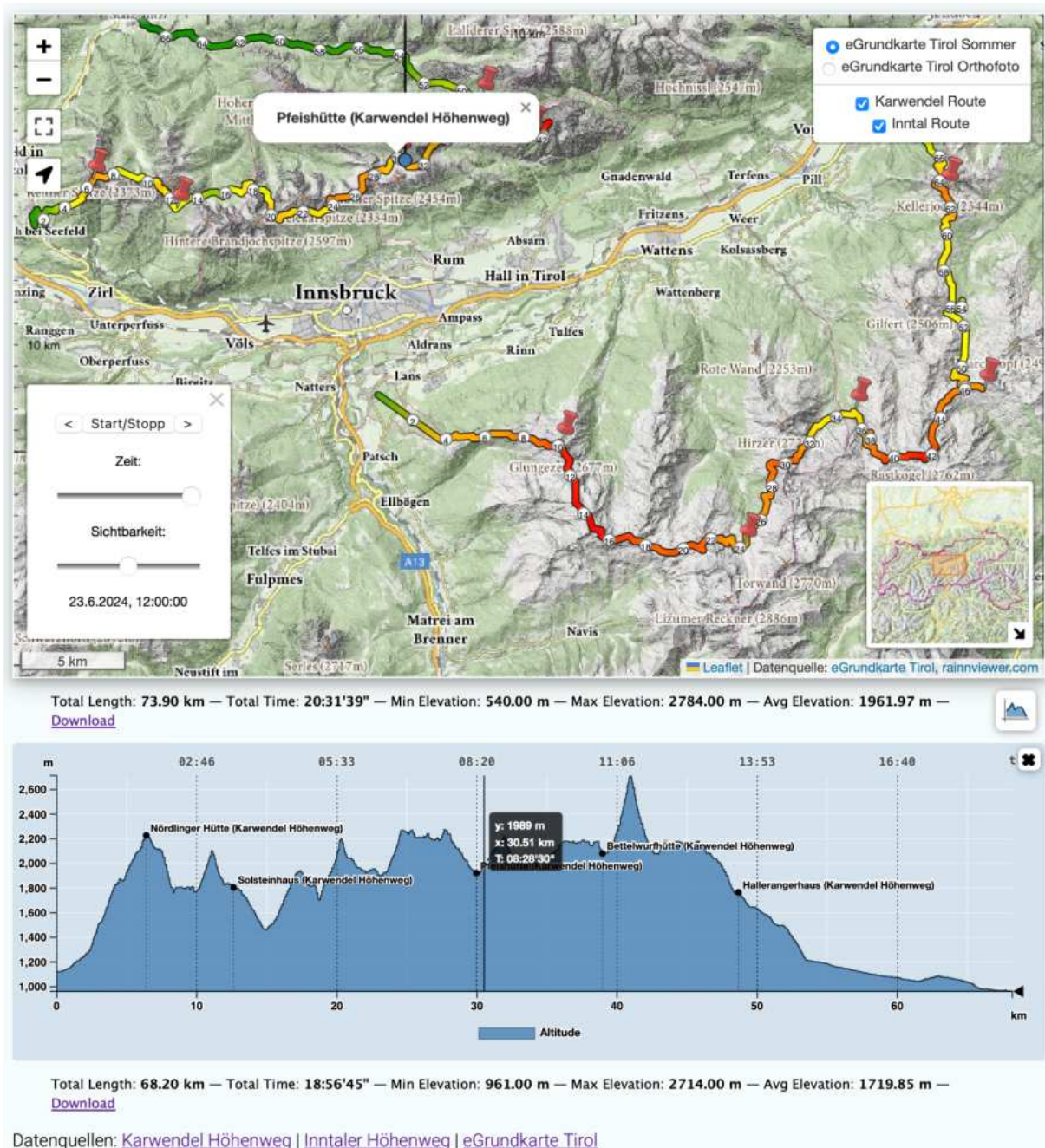


Abbildung 16: index.html und Videos auf der Homepage

wurde das Design ansprechend und interaktiv gestaltet (bspw. vergrößert sich die Videoansicht beim Darüberfahren mit der Maus).

Unter den Videos werden die beiden Höhenwege in einer Karte dargestellt (siehe Abb. 19). Rechts oben in der Layercontrol kann eine andere Grundkarte (Orthofoto) ausgewählt werden. Rechts unten befindet sich die Minimap, welche mit Hilfe des *Leaflet Plugins MiniMap* (Norkart 2018) eingebaut wurde. Die Minimap ist eine Miniaturansicht der Hauptkarte und hilft Benutzer:innen dabei, sich zu orientieren. Links unten befindet sich der *Leaflet Plugin Rainviewer*, durch den man Niederschlagsradardaten über eine Zeitleiste anzeigen lassen kann (Marcin 2023). Darüber kann durch das *Locate-Control Plugin* (Moritz 2024) der eigene Standort in der Karte angezeigt und durch das *FullScreen Plugin* die Karte in den Vollbildmodus geschaltet werden (B. 2023).



**Abbildung 17: Karte und Höhenprofil Wandern**



Die GPX-Dateien konnten mit Hilfe des *Leaflet-elevation Plugins* dargestellt werden (Raruto 2023). Die Daten der beiden Höhenwege stammen von der Seite Tirol.at (Tirol Werbung GmbH 2024a & Tirol Werbung GmbH 2024b). In den GPX-Dateien des Inntaler Höhenweges waren bereits Waypoints der Hütten vorhanden, allerdings anders benannt. Um die Karte so übersichtlich wie möglich zu gestalten, wurden die Beschriftungen zu den Hüttennamen und dem Höhenweg geändert. Für den Karwendel-Höhenweg lagen diese `<wpt>` nicht vor, weswegen sie manuell hinzugefügt wurden (siehe Abb. 20). Fährt man nun über die Stecknadel oder klickt darauf, erscheint der Name der Hütte als Popup. Die Hüttennamen sind auch in dem Höhenprofil eine Orientierungshilfe. Das Höhenprofil kann mit einem Klick auf das Icon rechts unter der Karte aktiviert und deaktiviert werden. Fährt man mit der Maus über das Höhenprofil, kann man den genauen Ort in der Karte sehen und auch andersherum.

```

2 <wpt lon="11.236" lat="47.32"><ele><![CDATA[2605]]></ele><name><![CDATA[Nördlinger Hütte (Karwendel Höhenweg)]]></name></wpt>
3 <wpt lon="11.288375" lat="47.308153"><ele><![CDATA[2017]]></ele><name><![CDATA[Solsteinhaus (Karwendel Höhenweg)]]></name></wpt>
4 <wpt lon="11.42548" lat="47.3300"><ele><![CDATA[1802]]></ele><name><![CDATA[Pfeishütte (Karwendel Höhenweg)]]></name></wpt>
5 <wpt lon="11.507488" lat="47.338551"><ele><![CDATA[2117]]></ele><name><![CDATA[Bettelwurfhütte (Karwendel Höhenweg)]]></name></wpt>
6 <wpt lon="11.477216" lat="47.354716"><ele><![CDATA[2234]]></ele><name><![CDATA[Hallerangerhaus (Karwendel Höhenweg)]]></name></wpt>

```

Abbildung 18: Screenshot aus data/gps-daten-karwendel-hoehenweg.gpx

Unterhalb der Karte beginnen die Detailbeschreibungen der einzelnen Etappen der Höhenwege. In den Überschriften zu den Etappen wurden alle Hütten verlinkt. Außerdem wurden Googlemaps-Navigationslinks zu den Start- und Endpunkten hinzugefügt (siehe Abb. 21). Mit Hilfe eines *Fontawesome-Plugins* konnte die Icon-Bibliothek, welche als Schriftart formatiert ist, integriert werden (Fontawesome 2024).

```

<div class="highlight">
  <h2 id="karwendel">Karwendel Höhenweg</h2>
  <p>Quelle des folgenden Textes: <a href="https://www.tirol.at/reisefuehrer/sport/wandern/wandertouren/a-karwendel-hoehenweg">tirol.at</a></p>
  <h3>
    Etappe 1: <a href="https://www.google.com/maps/dir/?api=1&origin=My+Location&destination=Reith+bei+Seefeld" target="_blank"><i class="fa-solid fa-route"></i> Reith bei Seefeld</a> – <a href="https://www.noerdlingerhuette.at/"><i class="fa-solid fa-arrow-up-right-from-square"></i> Nördlinger Hütte</a>
  </h3>
  <p><em>
    Höhenmeter bergauf: 1.150 Meter, Dauer: 3 Stunden, Länge: 6,5 Kilometer </em></p>
  <p>Die Tour beginnt mit einem Aufstieg zu einer gemütlichen AV-Schutzhütte mit grandiosem Aussichts panorama: Am

```



Abbildung 19: Screenshot aus dem Wander index.html sowie der Seite Wandern

Den Abschluss der beiden Textblöcke bildet jeweils eine Slideshow mit eigenen Fotos der fünf Hütten (meist Symbolbilder, alle Koch M. 2012-2024). Die Slideshows wurden mit einem Tutorial von w3schools implementiert (w3schools o.D.). Doch diese Anleitung ermöglicht nur eine Slideshow pro Seite. Der von S. Walker in einem stack overflow Forum vorgeschlagene Weg war die Lösung und so konnten doch zwei Slideshows auf der Seite angezeigt werden (Walker 2017). Die Slideshows wurden mit je einer ID (slideskarwendel und slidesinntal) versehen. Die ID definiert den spezifischen Container, der die Slideshow umfasst und im main.css wird die Ausrichtung (mittig) festgelegt. Innerhalb dieses Containers sind die einzelnen Bilder jeweils in separaten <div> Elementen mit der class „mySlides fade“ enthalten. Außerdem sind die Bildnummern und Bildunterschriften festgehalten. Mit Hilfe eines Navigationselements sind Vor- und Zurückpfeile sowie Navigationspunkte definiert (siehe Abb. 22).

```

153 <!-- Slideshow für Karwendel Weg -->
154 <div id="slideskarwendel" class="slideshow-container">
155
156 <!-- Bilder mit Nummer und Text -->
157 <div class="mySlides fade">
158 <div class="numbertext">1 / 5</div>
159 
160 <div class="text">Nördlinger hut <a href="https://www.noerdlingerhuette.at/"><i class="fa-solid fa-arrow-up-right-from-square"></i>
Homepage</a> (Copyright: Merit Koch)</div>
161 </div>
162
163 <div class="mySlides fade">--
164 </div>
165
166 <div class="mySlides fade">--
167 </div>
168
169 <div class="mySlides fade">--
170 </div>
171
172 <div class="mySlides fade">--
173 </div>
174
175 <div class="mySlides fade">--
176 </div>
177
178 <div class="mySlides fade">--
179 </div>
180
181 <div class="mySlides fade">--
182 </div>
183
184 <!-- Vorheriges/nächstes Foto -->
185 <a class="prev" onclick="plusSlides(-1, slideskarwendel)"></a>
186 <a class="next" onclick="plusSlides(1, slideskarwendel)"></a>
187
188 <br>
189
190 <!-- Punkt unten zur Angabe welches Foto es ist -->
191 <div style="text-align:center">
192 <span class="dot" onclick="currentSlide(1, slideskarwendel)"></span>
193 <span class="dot" onclick="currentSlide(2, slideskarwendel)"></span>
194 <span class="dot" onclick="currentSlide(3, slideskarwendel)"></span>
195 <span class="dot" onclick="currentSlide(4, slideskarwendel)"></span>
196 <span class="dot" onclick="currentSlide(5, slideskarwendel)"></span>
197
198 </div>
199
200 </div>
201 </div>
202
366 /* Slideshow container */
367 .slideshow-container {
368 max-width: 1000px;
369 position: relative;
370 margin: auto;
371 }
372
373 /* Hide the images by default */
374 .slides-karwendel {
375 display: none;
376 }
377
378 .slides-inntal {
379 display: none;
380 }
381
382 /* Vor und zurück */
383 .prev,
384 .next {
385 cursor: pointer;
386 position: absolute;
387 top: 50%;
388 width: auto;
389 margin-top: -22px;
390 padding: 16px;
391 color: white;
392 font-weight: bold;
393 font-size: 18px;
394 transition: 0.6s ease;
395 border-radius: 0 3px 3px 0;
396 user-select: none;
397 }
398
399 /* "Weiter" auf der rechten Seite*/
400 .next {

```

Abbildung 20: Screenshots aus index\_en.html und main.css zu den Slideshows

## 2.2.2 Benutzer:innenfreundlichkeit

Es wurden einige Maßnahmen ergriffen, um die Wander-Seite so benutzer:innenfreundlich und barrierefrei wie möglich zu gestalten. Die Seite lässt sich sowohl auf dem Computer als auch auf dem Handy komfortabel öffnen. Die beiden Videos auf der Wander- und Radseite werden nun untereinander statt nebeneinander dargestellt und die Karte im horizontalen Format (siehe Abb. 23). Dank der auf „true“ gestellte „gestureHandling“ aus dem *Leaflet.GestureHandling Plugin* kann man auch auf der Karte weiterhin nach unten scrollen und bedient sie mit zwei Fingern (Elmarquis 2021). Dies ist auch auf der Computer-Version der Seite der Fall. Mit Hilfe des abgebildeten Codes @media konnte das Design der Seite für das Handy angepasst werden. Darunter fällt die maximale Breite, die Anpassung der Navigationsleisten sowie die Dropdownlinks .

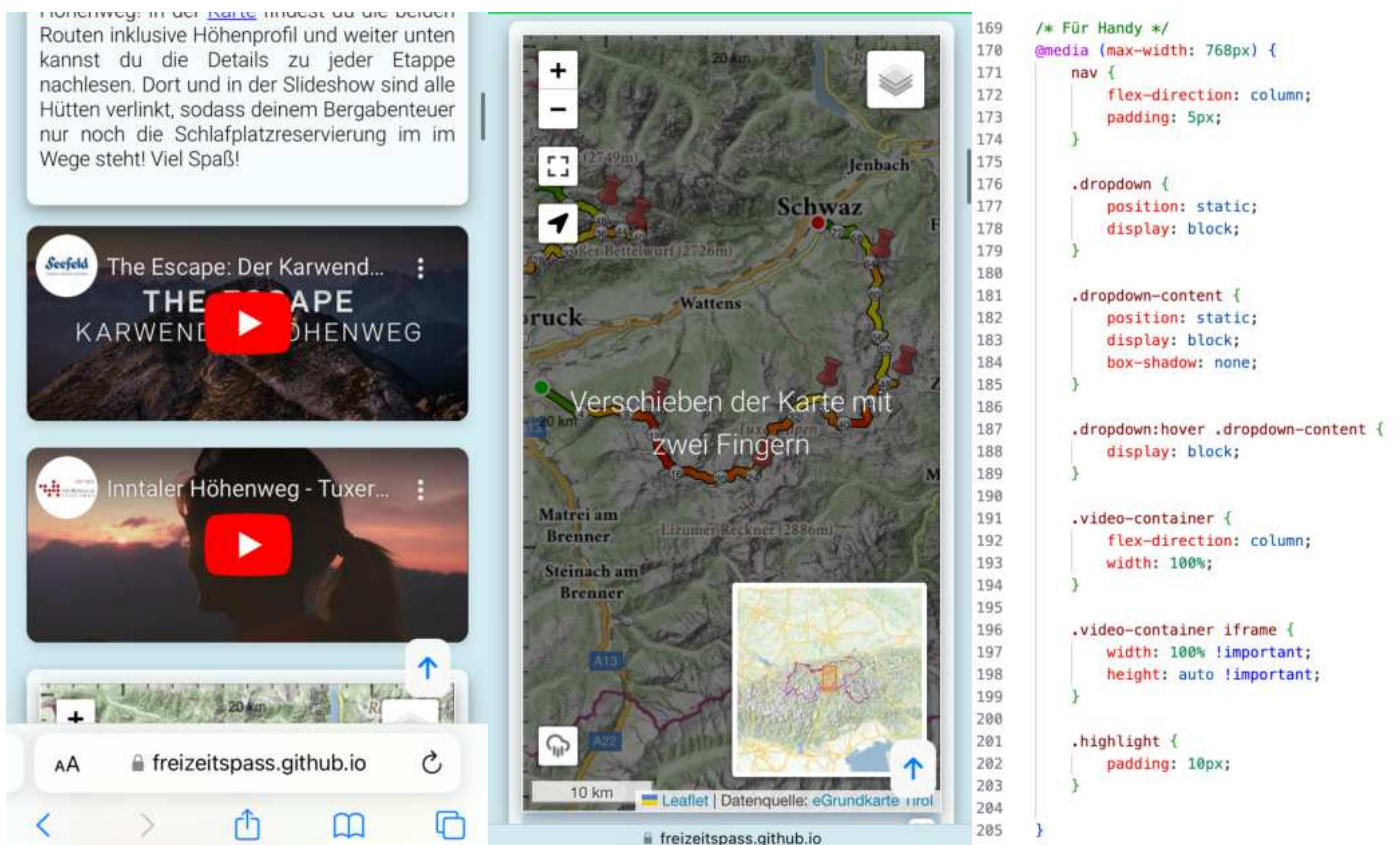


Abbildung 21: Screenshot der mobilen Version der Seite Wandern sowie dem Ausschnitt aus main.css

Um auch nicht-deutschsprachige Menschen zu erreichen, wurde eine englische Version dieser Seite erstellt. Diese erreicht man über einen Button in der Navigationsleiste, über den man auf die `index_en.html` Seite gelangt. Es wurden alle Texte und Überschriften übersetzt.

Die Bilder haben entweder eine Textalternative („alt“-Attribut) oder aber eine Bildunterschrift, wodurch die Inhalte auch für Menschen, die auf einen Screenreader angewiesen sind, erreichbar sind. Außerdem wurde die Farbgestaltung der gesamten

Webseite so gewählt, dass sie einen hohen Kontrast aufweist, um die Lesbarkeit zu optimieren.

Es befinden sich Links zu einer Google Maps Routensuche von dem aktuellen Standort zu den Ausgangspunkten der Wanderungen sowie von den Endpunkten zurück zum aktuellen Standort. Dies soll die Orientierung und Wanderplanung erleichtern. Gemeinsam mit dem Wetter-Widget auf der Startseite können Touren optimal geplant werden.

Außerdem wurde zur Verbesserung der Benutzer:innenerfahrung ein Button integriert, der es den Nutzer:innen ermöglicht, schnell zum Anfang der Seite zurückzukehren. Dieser "Scroll to Top"-Button wird in Kapitel 2.3.3 detailliert beschrieben.

### **2.2.3 Probleme und Herausforderungen**

Insgesamt konnten viele kleinere Probleme mit Hilfe von gezielten Google-Suchen sowie ChatGPT gelöst werden. Dabei war es sehr wichtig, Fragen präzise zu formulieren und die vorgeschlagenen Wege zu hinterfragen.

Eine große Schwierigkeit für die Seite Wandern war die Einbindung zweier Slideshows. Eine Slideshow mit Fotos hat problemlos funktioniert, doch die zweite Slideshow wurde nie korrekt angezeigt. Gemeinsam mit der Tutorin Laura wurde intensiv nach Lösungen gesucht, die in der Anleitung von einem User in einem stack overflow Forum lag (Walker 2017).

Auch die Einbindung der beiden .gpx-Dateien hat nicht auf den ersten Weg funktioniert. Leider lassen sie sich bis jetzt nicht in der Layercontrol ausschalten. Die Implementierung der .gpx Dateien war angelehnt an das biketirol Beispiel aus dem Kurs. Doch auch dort hat bei niemandem das An- und Ausschalten der Routen funktioniert.

Die Übersetzung der Seite hat nicht auf einen eleganten Weg (bspw. Über dynamische Sprachumschaltung mit Javascript) funktioniert. Deswegen wurde für die "Wander" Seite eine zweite html Seite angelegt, in der alle Texte übersetzt eingefügt wurden.

Ursprünglich war geplant, die Details der Routen in einer Sidebar darzustellen. Die ausgesuchte Anleitung (leaflet-sidebar-v2; siehe Konzept) hat aber leider nicht funktioniert. Außerdem war der Textumfang sehr groß, weswegen ein anderer Weg gewählt wurde und die Inhalte in größeren Textabschnitten zu lesen sind.



## 2.3 Radfahren

Die Seite <https://freizeitspass.github.io/rad/> widmet sich den besten Rennradstrecken rund um Innsbruck. Es werden sechs ausgewählte Routen kartografisch und schriftlich vorgestellt.

### 2.3.1 Aufbau und Inhalt

Der Header der Rennrad-Seite ist ähnlich wie bei den anderen Seiten aufgebaut. Ein Bild von Pixabay (TF3000 2019) vermittelt den Nutzer:innen sofort das Thema der Seite. In der Navigation wurde ebenfalls ein Dropdown-Menü integriert, über das man direkt zu den verschiedenen Routen gelangen kann. Dafür wurden im Body mehrere IDs vergeben, zu denen im Dropdown-Menü verlinkt wird. Dies ermöglicht eine schnelle und einfache Navigation zu den spezifischen Routenabschnitten der Webseite (siehe Abb. 24).

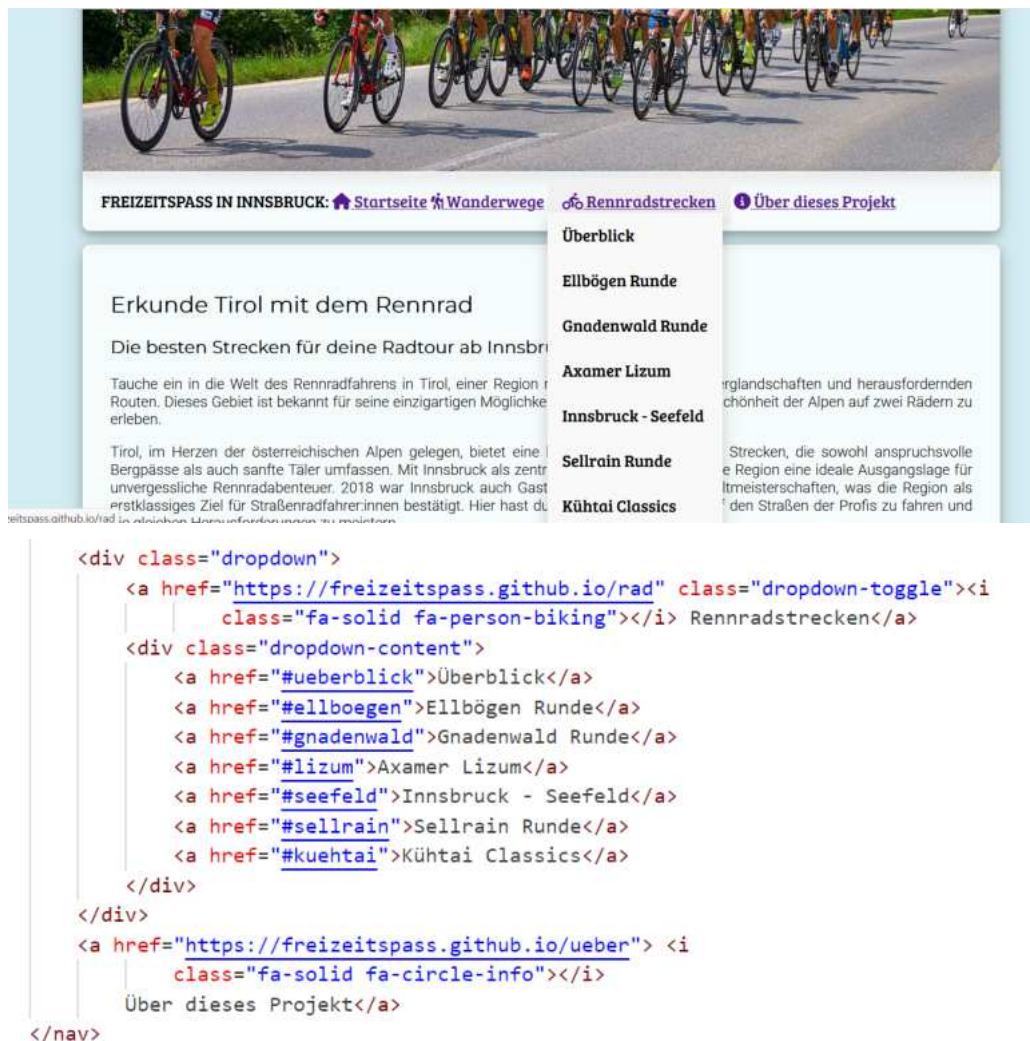


Abbildung 22: Navigationsleiste der Radseite (Website-Ansicht und HTML)

Diese Abschnitte umfassen eine Einleitung, die sowohl eine Übersicht als auch eingebettete Videos enthält, sowie sechs verschiedene Karten, die jeweils mit einer eigenen Beschreibung versehen sind. Um diese Abschnitte optisch und strukturell voneinander zu trennen, wird eine im CSS definierte Klasse namens `highlight` verwendet. Diese Klasse sorgt dafür, dass jeder Bereich der Webseite durch entsprechende Stilelemente hervorgehoben und voneinander abgegrenzt wird, was zur besseren Lesbarkeit und Navigation beiträgt.

Die Übersicht in der Einleitung wurde zudem durch eine weitere Klasse namens `highlight2` in einer anderen Farbe hervorgehoben. Eine `ordered list` verlinkt hier nochmals zu den einzelnen Routen (siehe Abb. 25).

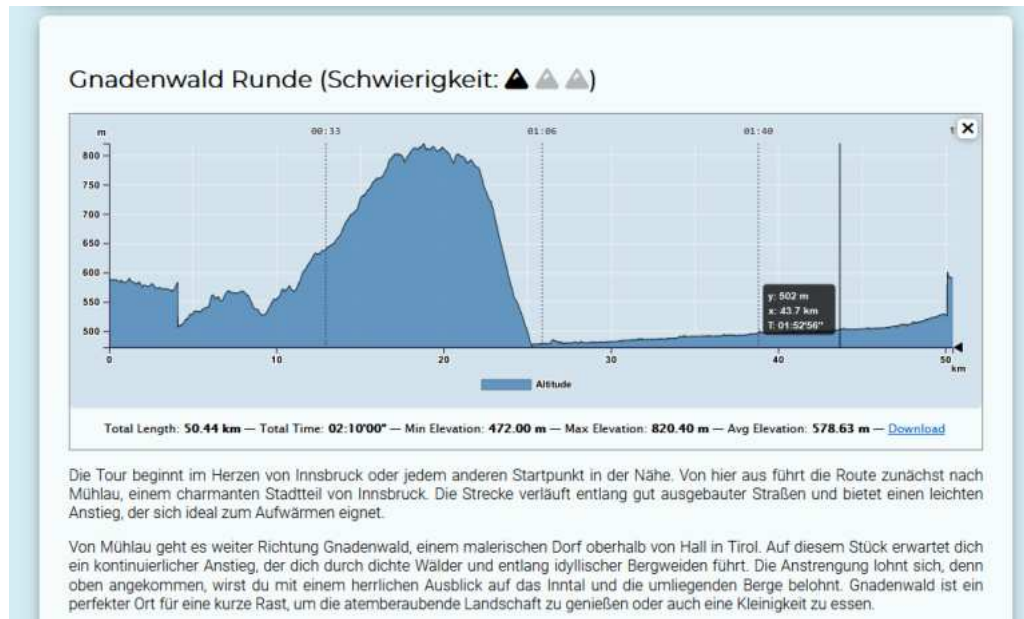


**Abbildung 23: Übersicht Radseite - Highlight2 (Website-Ansicht)**

Wie auch bei den anderen Seiten wurden Video-Container eingefügt, die zwei Youtube-Videos über das Rennradfahren in Innsbruck wiedergeben.

Unterhalb der Videos beginnen dann die einzelnen Abschnitte für die jeweiligen Routen. Jeder Abschnitt ist gleich aufgebaut:

In der `h2` Überschrift wird Auskunft über die Schwierigkeit der Route gegeben (max. 3 Berge (Fontawesome 2024)). Unter der Überschrift gibt das `div` Element „profile“ Auskunft über die Streckeneigenschaften und es lässt sich ein Höhenprofil öffnen und auf Wunsch auch wieder schließen. Dann folgt eine kurze Beschreibung der Strecke und ihrer Highlights, die selbst basierend auf eigenen Erfahrungswerten verfasst wurde (siehe Abb. 26).



**Abbildung 24: Beispiel ausgeklapptes Höhenprofil, Routenüberschrift inkl. Schwierigkeit und Beschreibung; Highlight zur Abgrenzung**

Das letzte und größte Element jedes Abschnittes bildet eine Karte. Hier wird der GPX-Track der Route mithilfe des Leaflet Elevation Plugins (Raruto 2023) dargestellt. Über die Layercontrol kann zwischen drei verschiedenen Hintergrundkarten gewählt werden. Entlang jeder Route sind Fahrradwerkstätten und ausgewählte Cafés/Gasthöfe mit Markern und Popups markiert, die auf Wunsch in der Layercontrol auch ausgeblendet werden können.

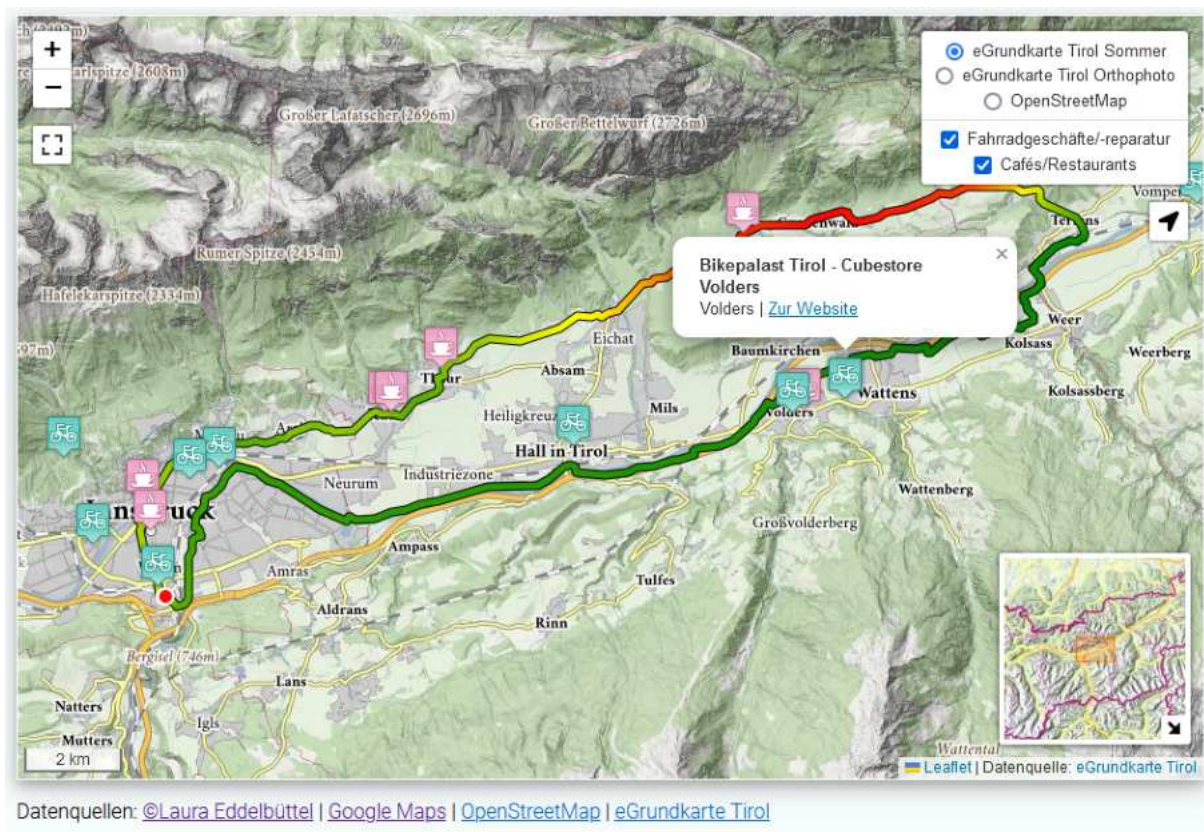
Es wurden zudem einige nützliche Plugins integriert:

- FullScreen: Ermöglicht es, die Karte im Vollbildmodus anzuschauen (B. 2023).
- Locate Plugin: Sucht den aktuellen Standort, sodass man sich auch während der Radtour orientieren kann (Moritz 2024).
- MiniMap: Bietet eine Übersicht über die Ausdehnung der Route und kann bei Bedarf eingeklappt werden (Norkart 2018).
- Gesture Handling: Verbessert die Gestensteuerung auf Touchscreen-Geräten für eine natürlichere und flüssigere Navigation durch die Karte (Elmarquis 2021).

Unterhalb der Karte werden noch die Datenquellen verlinkt. Die GPX-Tracks sind von Laura selbst mit ihrer Garmin Forerunner 30 bzw. Ihrem Garmin Edge 810 aufgezeichnet wurden. Damit der Wohnort geheim bleibt und um die Datengröße zu reduzieren wurden alle GPX-Tracks im GPX-Studio (Gpx.studio 2021) bearbeitet und schließlich in einem eigenen Ordner innerhalb der Radseite gespeichert.



Für die Fahrradwerkstätten bzw. die Cafes entlang der Route gab es leider keine öffentliche Datengrundlage in Tirol, sodass hier einige über GoogleMaps recherchiert wurden (siehe Abb. 27).

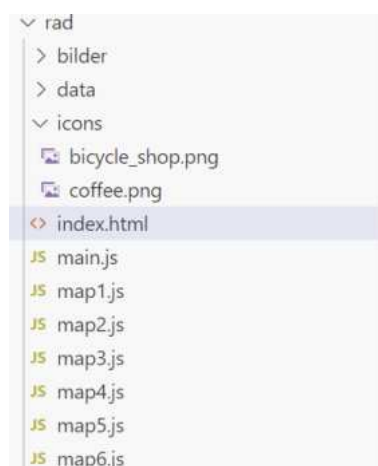


**Abbildung 25: Beispiel einer Radkarte mit Themalayern und Popup**

### 2.3.2 Aufbau Javascript

Um die einzelnen Karten gut auf einer Seite darstellen zu können, wurde für jede Karte eine eigene .js-Datei erstellt. Die Karten wurden von 1 bis 6 nummeriert. Außerdem gibt es eine main.js, in der das Dropdown-Menü und der ScrollToTop-Button gespeichert sind (siehe Abb. 28).

Die einzelnen map.js-Dateien sind alle gleich aufgebaut, wobei jeweils nur die ID und der GPX-Track ausgetauscht wurden. Nach der Initialisierung der Karte wurden die zwei thematischen Layer definiert, die Hintergrundkarten eingebunden und die LayerControl hinzugefügt. Anschließend folgt das Elevation Plugin, bei dem einige Stilelemente festgelegt wurden (zum Beispiel das Color-Theme, das standardmäßig eingeklappte Höhenprofil und die Trennung von Höhenprofil und Karte), welches ebenfalls zur jeweiligen Karte hinzugefügt wurde.



**Abbildung 26: Struktur der Radseite (Ordner für Bilder, Data + Icons; main.js und sechs map.js, index.html)**

Alle map.js-Dateien enthalten zudem nur die relevanten Fahrradwerkstätten und Cafés, die in eigenen Konstanten `const COFFEE` bzw. `const BIKESHOP` definiert sind. Für jeden Punkt sind der Name („title“), der Link zur Website, der Ort und die Koordinaten (lat und lng) angegeben. Die Icons, die für alle Karten gleich sind (Mollet 2024) wurden zum jeweiligen Thematischen Layer hinzugefügt. Das Popup wurde der Übersichtlichkeit halber nur mit den wichtigsten Informationen ausgestattet (siehe Abb. 29).



```
//Cafes Liste
const COFFEE = [
  {
    title: "manni.coffee",
    link: "-",
    ort: "Innsbruck",
    lat: 47.26537316391705,
    lng: 11.394063993012983,
  },
  {
    title: "haepinest",
    link: "https://haepinest.at/",
    ort: "Innsbruck",
    lat: 47.271126885551034,
    lng: 11.391991823597085,
  },
  {
    title: "Cafe Platzl",
    link: "-",
    ort: "Rum",
    lat: 47.2872244805296,
    lng: 11.457385753772916,
  },
  {
    title: "Forum Cafe",
    link: "-",
    ort: "Innsbruck",
    lat: 47.26537316391705,
    lng: 11.394063993012983,
  }
];

//Cafes Popups
for (let coffee of COFFEE) {
  L.marker([coffee.lat, coffee.lng], {
    icon: L.icon({
      iconUrl: 'icons/coffee.png',
      popupAnchor: [0, -37],
      iconAnchor: [16, 37],
    })
  })
  .addTo(themaLayer.coffee)
  .bindPopup(`<b>${coffee.title}</b><br> ${coffee.ort} | <a href="${coffee.link}">Zur Website</a>`);
}
```

**Abbildung 27: Beispiel Themalayer eines map.js (Konstante COFFEE und Popups)**

Am Ende jeder map.js-Datei werden der Maßstab, die MiniMap und die Locate Control hinzugefügt. Im index.html der Radseite wird dann jeweils über die ID die entsprechende Karte eingefügt. Die unterschiedlichen .js-Dateien wurden natürlich alle im `<head>` der Seite verlinkt. Dort sind außerdem die main.css und alle Leaflet Plugins integriert.

### 2.3.3. Benutzer:innenfreundlichkeit

Für die Benutzerfreundlichkeit wurden, wie schon bei den anderen Seiten, folgende Elemente eingefügt:

Ein "Nach oben scrollen"-Button erscheint, sobald die Seite mehr als 20 Pixel nach unten gescrollt wurde (siehe Abb. 30). Die Funktionalität dieses Buttons ist in der main.js der Radseite definiert, wobei der Button ab 20 Pixel sichtbar wird und bei einem Klick ein sanftes Scrollen nach oben auslöst. Die Gestaltung des Buttons, wie Schriftgröße, Farbe und Padding, ist im Allgemeinen main.css festgelegt. Der Button selbst wird im Footer der Radseite über die ID `scrollToTopBtn` aufgerufen, wobei der Pfeil erneut von Fontawesome (Fontawesome 2024) eingefügt wurde.

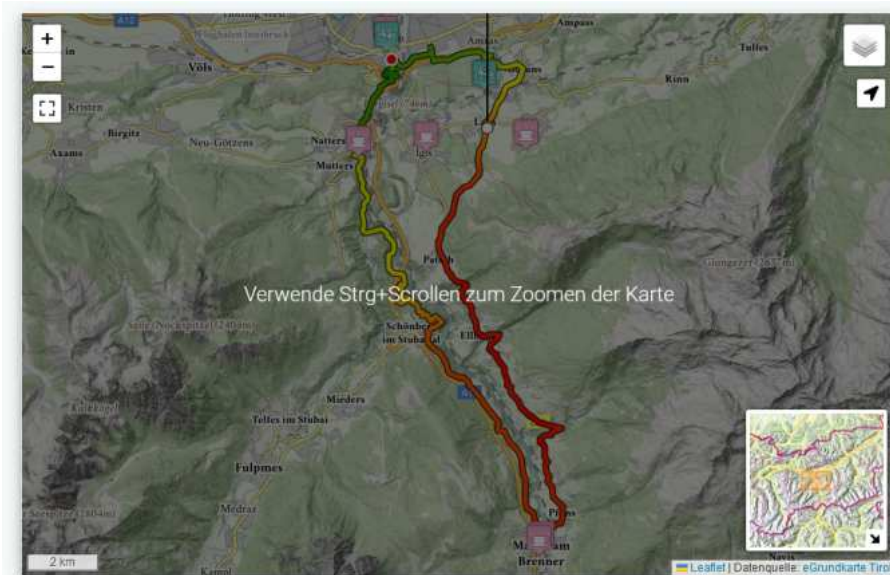


**Abbildung 28: Scroll-to-top Button (Website-Ansicht)**

Die Navigationsleiste wurde für alle Seiten im main.css gestaltet. Das Dropdown-Menü ist als Event in der main.js der Radseite festgelegt, während die Hover-Effekte ebenfalls im main.css definiert sind.

Für eine bessere Darstellung der Seite auf mobilen Geräten wurden im main.css einige Anpassungen vorgenommen. Unter @media wurde eine maximale Breite festgelegt, und für die Navigation wurde eine flex-direction definiert. Das Dropdown-Menü funktioniert auch auf Smartphones, wobei hier einige Stilelemente, wie der Schatten des Dropdown-Inhalts, reduziert wurden. Die Videocontainer wurden zudem als Spalten (column) und iFrames (iframe) festgelegt, sodass sie nicht mehr abgeschnitten werden.

Das Leaflet Plugin "Gesture Handling" verbessert zusätzlich die Interaktion mit Karten auf mobilen Geräten, indem es die Handhabung von Gesten wie Pinch-Zoom und Wischen optimiert. Es ermöglicht eine flüssigere und natürlichere Navigation über die komplette Seite sowohl am Desktop als auch am Smartphone, indem es Konflikte zwischen Karteninteraktionen und Standard-Scroll Bewegungen reduziert oder vermeidet. Dies ist besonders bei den vielen Karten der Rad Seite sehr nützlich (siehe Abb. 31).



**Abbildung 29: Beispielkarte Rad mit Gesture Handling Plugin**

### 2.3.3. Probleme und Herausforderungen

Die Erstellung dieser Seite brachte zunächst Herausforderungen bei der Entwicklung der Navigation mit sich. Es dauerte eine Weile, die richtige Größe und Position zu finden. Besonders der Umgang mit dem Padding-Stil war neu und es wurde zunächst aufwendig versucht, ein Rechteck einzufügen, über das der Text gelegt werden sollte. Nach mehreren Versuchen gelang es schließlich, eine Darstellung zu finden, die alle Teammitglieder zufriedenstellte.

Ursprünglich war für diese Seite geplant eine große Karte mit allen GPX-Tracks zu entwerfen, deren unterschiedliche Beschreibung dann in einer Sidebar angezeigt werden sollte. Jedoch war das bei der Vielzahl der Tracks und Informationen sehr unübersichtlich, sodass sich für Darstellung einzelner Karten entschieden wurde.

Ein noch ungelöstes Problem betrifft die Initialisierung der Kartenansicht in Verbindung mit dem Elevation Plugin. Trotz der Festlegung eines bestimmten Zoomlevels sowohl bei der Karteninitialisierung als auch durch die Verwendung von `.setView` ändert sich der Zoom nach dem Laden des Elevation Plugins unerwartet. Dies stellt insbesondere dann eine Herausforderung dar, wenn für bestimmte Karten eine geringere Zoomstufe bevorzugt wird.

### **3. Fazit und Ausblick**

Insgesamt hat die Umsetzung dieses Projekts sehr gut funktioniert. Es hat sehr geholfen, sich zu treffen und jeweils an den eigenen Seiten zu arbeiten. So konnten schnell kleinere Fragen und Probleme aus dem Weg geräumt werden.

Die Seite könnte in Zukunft noch mit weiteren Freizeitaktivitäten in Innsbruck erweitert werden. Außerdem könnten auch für die Rad- und Wanderseite noch mehr Routen eingefügt werden. Dafür könnte man die Benutzer:innen auffordern, eigene Routen und Tipps einzusenden, über ein angepasstes Kontaktformular, in dem zum Beispiel GPX-Tracks direkt mitgeschickt werden können.

Weitere Ideen für die Gestaltung der Seite sind:

- Benutzer:innenbewertungen und Kommentare, die den Austausch fördern und die Seite noch interaktiver gestalten würde.
- Ein Veranstaltungskalender, der aktuelle und lokale Veranstaltungen in Innsbruck anzeigt.
- Eigene Tools zum Planen von Routen mit dem ÖPNV.

## Quellenverzeichnis

### Plugins:

- B., B. (2023). Leaflet.Control.FullScreen. <https://github.com/brunob/leaflet.fullscreen> (Erste Veröffentlichung 2012).
- Elmarquis (2021). Leaflet.GestureHandling. <https://github.com/elmarquis/Leaflet.GestureHandling> (Erste Veröffentlichung 2018).
- Fontawesome (2024). Fontawesome. [https://github.com/FortAwesome/Font-Awesome?utm\\_source=cdnjs&utm\\_medium=cdnjs\\_link&utm\\_campaign=cdnjs\\_library](https://github.com/FortAwesome/Font-Awesome?utm_source=cdnjs&utm_medium=cdnjs_link&utm_campaign=cdnjs_library) (Erste Veröffentlichung 2018)
- Gpx.studio (2021). Gpx.studio. <https://gpx.studio/>
- Leaflet (2018). Leaflet Fullscreen. <https://github.com/Leaflet/Leaflet.fullscreen> (Erste Veröffentlichung 2016).
- Marcin, W. (2023). Leaflet.Rainviewer. <https://github.com/mwasil/Leaflet.Rainviewer> (Erste Veröffentlichung 2019).
- Mollet, N. (2024). Map Icons Collection <https://mapicons.mapsmarker.com> (Erste Veröffentlichung 2009)
- Moritz, D. (2024). Leaflet-locatecontrol. <https://github.com/domoritz/leaflet-locate-control>.
- Norkart (2018). Leaflet MiniMap. <https://github.com/Norkart/Leaflet-MiniMap> (Erste Veröffentlichung 2015).
- Raruto (2023) Leaflet-elevation. <https://github.com/Raruto/leaflet-elevation> (Erste Veröffentlichung 2018).
- traffordDataLab (2020). Leaflet Reachability. <https://github.com/traffordDataLab/leaflet.reachability> (Erste Veröffentlichung 2018).
- Walker S. (2017) Antwort in stack overflow Forum "How do I make multiple slideshows in the same html document?". <https://stackoverflow.com/questions/43299759/how-do-i-make-multiple-slideshows-in-the-same-html-document> (Letzter Zugriff: 23.06.24).
- Weather Widget (2024). weatherwidget.org. <https://weatherwidget.org/> (Letzter Zugriff: 24.06.24)



W3schools (o.D.) How TO – Slideshow.

[https://www.w3schools.com/howto/howto\\_js\\_slideshow.asp](https://www.w3schools.com/howto/howto_js_slideshow.asp) (Letzter Zugriff: 23.06.24).

#### *Daten und Textinhalte:*

Open Data Gouvernement Land Tirol (2024). Almzentren in Tirol. <https://data-tiris.opendata.arcgis.com/datasets/almzentren-1/explore> (Letzter Zugriff 23.06.24).

OpenRouteService (2024). Openrouteservice API services: <https://openrouteservice.org/> (Letzter Zugriff 24.06.2024)

Tirol Werbung GmbH (2024a). Weitwanderung Karwendel Höhenweg. <https://www.tirol.at/reisefuehrer/sport/wandern/wandertouren/a-karwendel-hoehenweg> (Letzter Zugriff: 23.06.24).

Tirol Werbung GmbH (2024b). Weitwanderung Inntaler Höhenweg. <https://www.tirol.at/reisefuehrer/sport/wandern/wandertouren/a-inntaler-hoehenweg> (Letzter Zugriff: 23.06.24).

#### *Fotos:*

Freizeitspass (2024). Headerbild Teamvorstellung. Selbst aufgenommen.

Koch M. (2012-2024). Fotos der Hütten. Alle selbst aufgenommen.

Koch M. (2024). Foto vom Sonnenaufgang an der Reither Spitze oberhalb der Nördlinger Hütte, 14.06.24.

Rei Simon (2020). Innsbruck, Tirol, Österreich. <https://pixabay.com/de/photos/innsbruck-tirol-%C3%B6sterreich-4761198/> (Letzter Zugriff 24.06.24)

Salucci Y. (2020-2024). Fotos von Innsbruck. Alle selbst aufgenommen.

TF3000 (2019). Sports Cycling Races Cycles. <https://pixabay.com/photos/sports-cycling-races-cycles-move-4663417/> (Letzter Zugriff 25.06.2024)