

Leopold - Franzens - Universität Innsbruck

PROJEKTBERICHT

Erstellung eines Online-Wanderführers

Im Rahmen des Seminars: VU Geoinformatik - Webmapping

Verfasser_innen

Laura Brandstetter Matrikel-Nr. 11904610
Alexandra Buck Matrikel-Nr. 11936154
Leon Bührle Matrikel-Nr. 11939582

Betreuer

Klaus Förster
Bernhard Öggl

Eingereicht am:

17.06.2020

Institut für Geographie, Fakultät für Geo- und
Atmosphärenwissenschaften, Universität Innsbruck

Inhalt

Abbildungsverzeichnis	2
1. Einleitung	3
2. Ausgangsdaten.....	3
3. Aufbau und Funktionen der Website	4
3.1 Aufbau der Website	5
3.2 Einzeichnen der Tracks.....	9
3.3 Fotospots	10
3.4 Tourinformationen.....	11
3.5 Navigation	13
3.6 Anklickbare Länder auf der Startseite.....	14
4. Verwendete Plugins	16
4.1 Rainviewer	16
4.2 Standort	16
4.3 Maßstab	18
4.4 Leaflet Providers	18
5. Verbesserungsvorschläge	18
6. Fazit.....	20
7. Quellenverzeichnis.....	21

Abbildungsverzeichnis

Abbildung 1: Aufbau der Website.....	5
Abbildung 2: Startseite (index.html).....	6
Abbildung 3: Übersichtskarte Österreich (wege.html)	7
Abbildung 4: Tourenseite Olpererhuette.html	8
Abbildung 5: Fotospot in Zell am See	11
Abbildung 6: Lokalisierung eines möglichen Treffpunkts über die GPS-Funktion	17
Abbildung 7: L.ControlScale	18
Abbildung 8: Eigene Mapbox-Karte der Tour Dachstein.....	19

1. Einleitung

Dieses Projekt beschäftigt sich mit der Darstellung und Erläuterung von berühmten Wandertouren in den Alpen. Hierbei wurde sich auf Wanderoutes fokussiert, welche durch einzigartige Fotomotive charakterisiert sind. Die Firma Mountain Moments, welche ihren Sitz in Innsbruck hat und durch Marius Schwager und Matthias Mayer geführt wird, hat sich auf die Vorstellung berühmter Fotospots in den Alpen spezialisiert. Unter anderem publizierte Mountain Moments das gleichnamige Buch Mountain Moments, welches 30 Wanderoutes in den Alpen beinhaltet. Aufgrund der steigenden Nachfrage von Wanderbüchern in digitaler Form steigt ebenfalls das Interesse an interaktiven Onlinebüchern. Durch die seit längerem bestehende Kooperation mit Mountain Moments entstand daher die Idee, die vorhandenen Bücher als Onlineressource zu erstellen. Das hier vorgestellte Projekt dient als Skizze und soll bei der späteren Umsetzung eines webbasierten Foto- und Tourenbuches helfen. Da es sich lediglich um eine Vorlage handelt und aufgrund des gegebenen zeitlichen Rahmens, wurden lediglich die Touren für Österreich in das Programm implementiert.

Das Ziel dieses Projekts ist daher, die erhaltenen Daten mittels frei verfügbaren Tools möglichst benutzerfreundlich zu verarbeiten und darzustellen. Hierfür wurde von Mountain Moments die notwendigen Daten als Open Source Ressource auf Github unter folgendem Link zur Verfügung gestellt: <https://github.com/MountainMoments/Alpen>. Der Bericht nennt zunächst die Ausgangsdaten und erläutert anschließend den Aufbau der Seiten. Des Weiteren wurden einige Leaflet-Plugins verwendet, welche nachfolgend vorgestellt werden. Abschließend erfolgt eine kurze Diskussion, welche zusätzliche Tools bei einer weiteren Umsetzung denkbar sind.

2. Ausgangsdaten

Wie bereits beschrieben, wurde ein Großteil der Ausgangsdaten von Mountain Moments als open-source Ressource zur Verfügung gestellt. Unter anderem stellte Marius Schwager verschiedene GPX-Daten von verschiedenen Wanderungen bereit. Die GPX-Daten wurden mithilfe des Webtools <https://mygeodata.cloud/converter/gpx-to-geojson> in GeoJson umformatiert, um das Einzeichnen der einzelnen Routen auf der Website zu erleichtern. Weitere

Erläuterungen dazu folgen in Kapitel 3.1. Die GPX-Tracks sind trotzdem als Download auf der Website im .gpx-Format verfügbar, da dies für den Wanderer das wichtigste Format ist. Die verwendeten Fotos sind von Marius Schwager und dürfen für die Zwecke dieses Projektes verwendet werden. Marius Schwager behält hierbei weiterhin die Rechte an den Bildern. Das Bildmaterial ist unter „<https://github.com/ibkdrei/ibkdrei.github.io/tree/master/fotospots>“ zugänglich. Marius Schwager hat der Verwendung der Daten ausdrücklich zugestimmt und ist auf jeder HTML-Seite als Urheber gekennzeichnet (@Mountain Moments). Die Daten der administrativen Grenzen Europas im GeoJson-Format stammen von Common-Data (Carto 2020).

3. Aufbau und Funktionen der Website

Eine wichtige Vorüberlegung beim Erstellen von Websites ist die Strukturierung des Projektes und die Zusammensetzung aus den einzelnen Seiten, um alle Inhalte darstellen zu können. Die einzelnen HTML-Seiten müssen sinnvoll verknüpft sein. Um dynamische Inhalte zu einer Webseite hinzuzufügen, sind JavaScript Files geeignet. Die Programmiersprache erlaubt Interaktionen mit den Nutzern. Designanpassungen erfolgen über CSS-Files. In den folgenden werden die Funktionen der Website, die sich aus den verschiedenen Files ergeben, vorgestellt. Vorab wird der Aufbau des Projekts erläutert.

3.1 Aufbau der Website

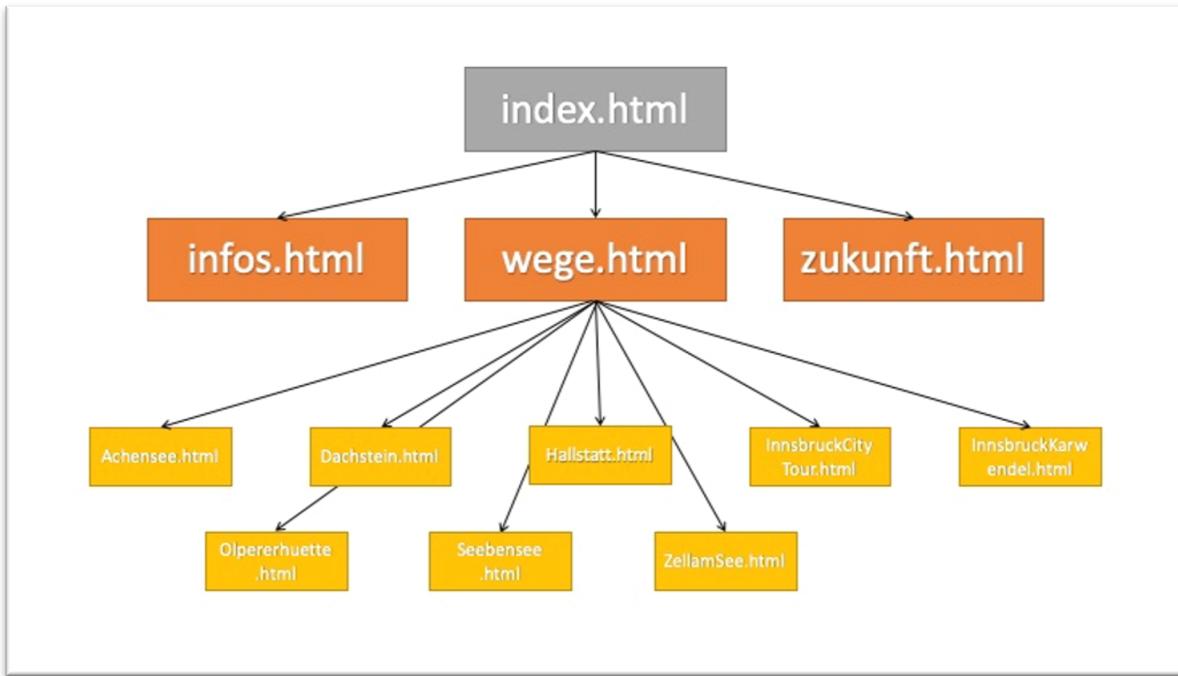


Abbildung 1: Aufbau der Website

Abbildung 1 zeigt den Aufbau der Website <https://www.ibkdrei.io>. Die `index.html` (siehe Abbildung 2) fungiert als Startseite. Diese ist verknüpft mit der `main.js`. Auf dieser Seite ist das Logo der Mountain Moments, ein Navigationsmenü und eine allgemeine Übersicht über die Website untergebracht. Die Textbausteine sind als horizontale Liste angeordnet und diesen zur Orientierung. Es wird auf die `infos.html` verlinkt, der Rainviewer erwähnt, sodass der Nutzer diesen in der Karte nicht übersieht und diesen direkt anwenden kann. Als dritter Punkt wird die Handhabung der Website erklärt.

Zusätzlich ist eine Übersichtskarte eingebaut, welche Boundaries für die Wanderregionen zeigt. Mit einem Mausklick auf die jeweilige Wanderregion gelangt der Nutzer direkt auf die `wege.html`, auf der es dann genauere Informationen zu den Touren gibt.

Wie erwähnt wird die Seite `infos.html` sowohl im Navigationsmenü als auch im Textbaustein verlinkt. Diese Seite ist eine reine Textseite mit zwei Bildern, welche viele wichtige allgemeinen Anmerkungen bezüglich der Wandertouren bieten. Die Texte hieraus stammen aus dem Buch Mountain Moments und die Quellen der Bilder sind angegeben.



Die schönsten Fotospots und Wanderungen in den Alpen

HOME WANDERREGIONEN INFOS MOUNTAIN MOMENTS

Übersicht der zahlreichen Wanderregionen in Europa

Jede neue Bergtour in den Alpen ist eine Entdeckungsreise, hier findet ihr Inspiration für eure nächsten Touren. Mountain Moments präsentiert euch lohnende Wanderungen zu den eindrucksvollsten Fotospots der Alpen. Von den bayerischen Alpen bis zu den Dolomiten, von Slowenien bis Frankreich werden die schönsten Gebirgsgruppen der Alpen carto- und fotografisch präsentiert.

Sicher unterwegs im Gelände

Bei einer Wanderung in alpinem Gelände sind viele Dinge zu beachten. Hier findest du mehr Informationen zur Tourenplanung, Erläuterungen zu den Karten und zur allgemeinen Sicherheit in alpinen Gelände.

Unterstützung bei der Tourenplanung

Um die Entscheidung für das Ziel zu erleichtern kann der Rainviewer auf der Übersichtskarte zu Hilfe genommen werden. Egal ob strahlender Sonnenschein für eine Wanderung abgewartet oder die nächste Regenfront für eine mystische Stimmung beim Fotoshoot abgepasst werden soll.

Alle Details zu den Touren

Im Menü stehen mehrere Wanderregionen zur Auswahl, die einzelnen Touren können auf der Karte über die Icons angewählt werden. Über "Zur Tour" gelangt ihr zur visualisierten Route inklusive Fotospots und ihr erhaltet Details wie Dauer, Länge und Höhenmeter

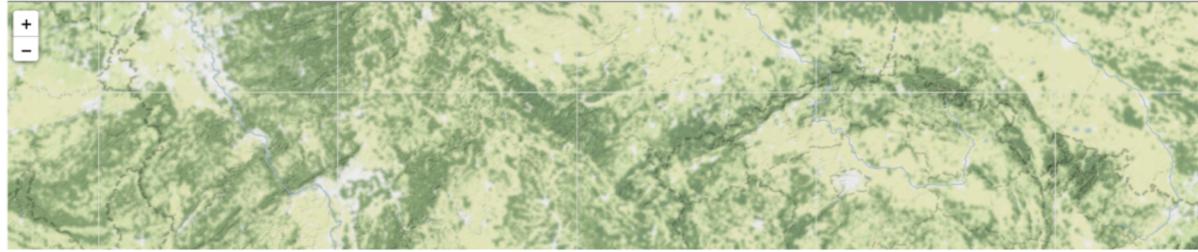


Abbildung 2: Startseite (index.html)

Sie fungiert auch als eine Art Legende, da die Symbole, welche später unter den Wanderkarten erscheinen, definiert werden. Dies hilft, beispielsweise den Schwierigkeitsgrad besser einzuschätzen. Unter der Erklärung der Symbole sind weitere Hinweise und Informationen zum Fall eines Notfalls aufgelistet.

Über die Navigation gelangt der Nutzer zum eigentlichen Kern der Website, die wege.html, beziehungsweise die Unterseiten für die jeweiligen Touren. Jede Tour hat eine eigene HTML Seite. Es wurde bewusst auf eine Dropdown-Auswahl verzichtet, damit sich für jede Tour eine neue Seite öffnet und der Benutzer dann zwischen den verschiedenen Tabs hin- und herwechseln kann. In der Navigation gelangt der Besucher mit einem Klick auf die Boundaries der Landesgrenzen Österreichs auf das wege.html. Bei einem Klick auf eine andere Wanderregion findet eine Weiterleitung zu der Seite zukunft.html statt, da es im Rahmen dieses Projekts nicht möglich war, auch diese Regionen zu berücksichtigen. Allerdings gehören diese Regionen ebenfalls zu dem Mountain Moment Projekt dazu und könnten nachfolgen

nach dem gleichen Muster wie bei wege.html implementiert werden. Auf Abbildung 3 ist der Aufbau von wege.html dargestellt.

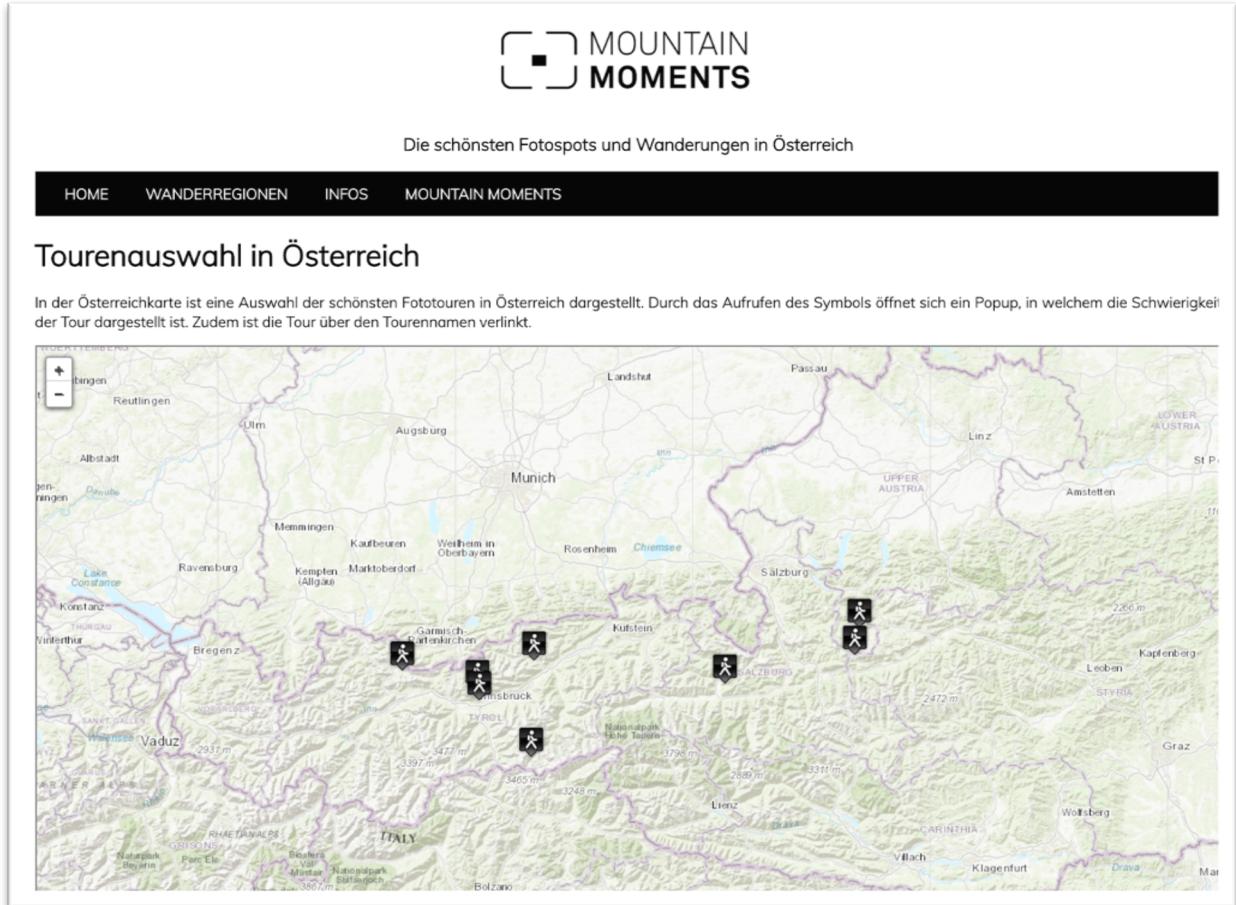


Abbildung 3: Übersichtskarte Österreich (wege.html)

Unter einem kurzen Textbaustein erscheint eine weitere Karte, welche direkt auf den Ausschnitt von Österreich zoomt. Auf der Karte sind Marker eingebaut, welche die Startpunkte der verschiedenen Wanderrouten zeigen. Klickt der Benutzer auf den Marker, erscheint ein Popup mit dem Namen der jeweiligen Tour und dem dazugehörigen Schwierigkeitsgrad. Mit einem Klick auf den Namen wird der Benutzer direkt auf die jeweilige Unterseite der Tour weitergeleitet. Diese öffnet sich dann in einem neuen Tab, es kann also jederzeit zu der Übersichtsseite zurückgekehrt und eine andere Tour ausgewählt werden. Die htmls für die Touren sind gleich aufgebaut und greifen auf dasselbe wege.js zu, also auf dieselbe Karte. Durch einen dynamischen Zoom gelangt der Nutzer direkt auf die gewünschte Tour. Je nach Javaversion in dem Browser gibt es leichte Probleme bei dem Zoom. Ein

reibungsloses Benutzen gelingt im Firefox Browser. Der Routenverlauf wird nur für die jeweilige Route angezeigt. Auf Abbildung 4 ist eine exemplarische Detailroute dargestellt.

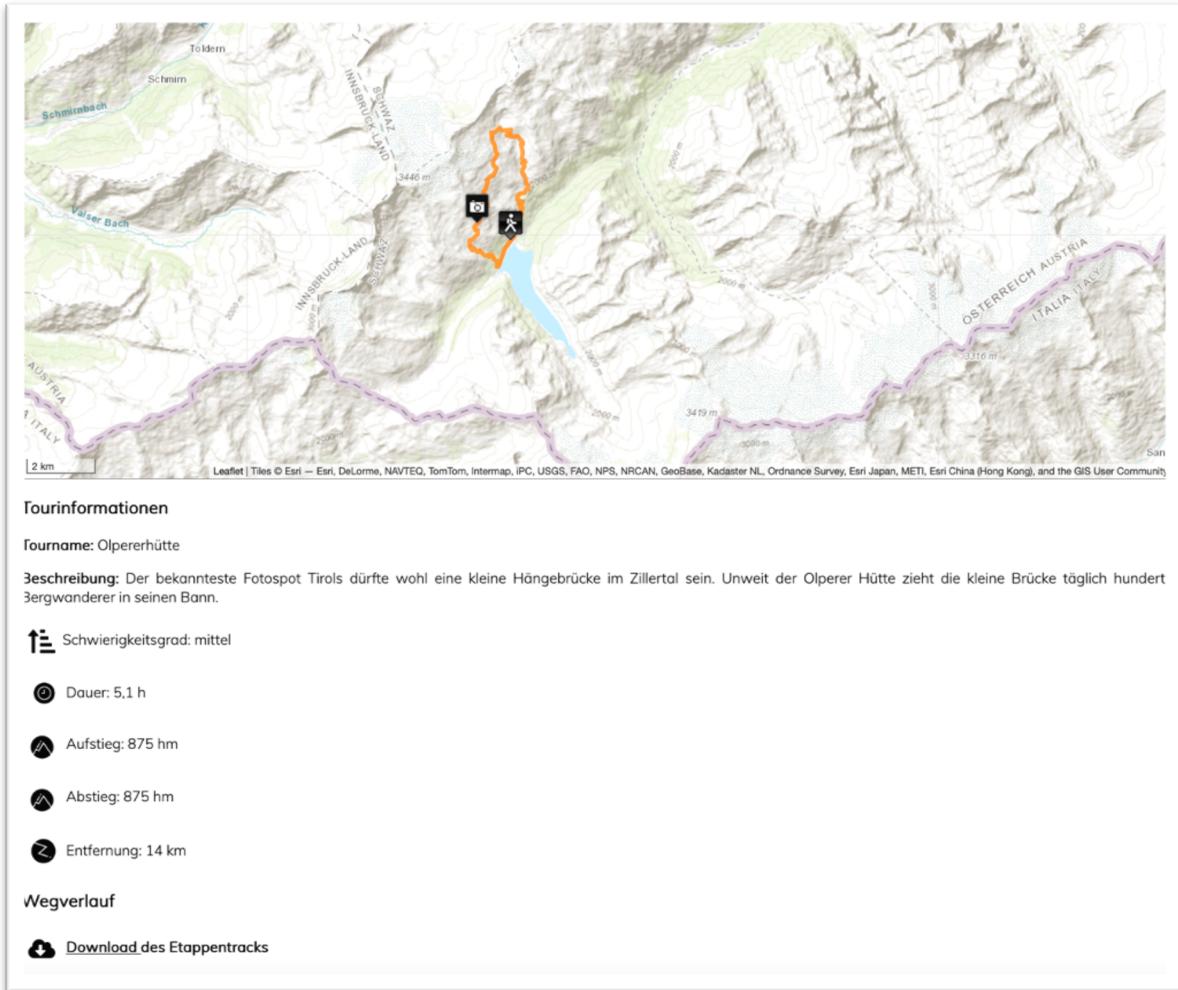


Abbildung 4: Tourenseite Olpererhütte.html

Im oberen Bereich befindet sich die Karte mit dem eingezeichneten Track und zwei Markern. Ein Marker symbolisiert den Startpunkt der Route und ein weiterer den Fotospot. Für jede Tour wurde ein Fotospot ausgewählt, welcher sich besonders gut für einzigartige Fotomotive eignet. Mit einem Klick auf den Marker erscheint ein Foto aus der Perspektive. Unterhalb der Karte sind individuelle Tourinformationen erläutert. Diese Informationen kommen aus der JavaScript.Datei weginfos.js und können mit Hilfe einer dynamischen Abfrage von jeder HTML-Seite abgefragt und eingefügt werden. Im unteren Bereich ist es viabel, den Etappentrack als .gpx Datei herunterzuladen. Die aufgeführten Erklärungen beschreiben den allgemeinen Aufbau der Webseite. Es folgt eine detaillierte Auseinandersetzung an ausgewählten Codebereichen, welche für die Funktionalitäten eminent sind.

3.2 Einzeichnen der Tracks

Nachdem der allgemeine Aufbau der Webseiten ausführlich erläutert wurde, werden in den folgenden Kapiteln spezielle Codeschnipsel erläutert, um die Funktionsweise nachzuvollziehen. Ein wichtiger Punkt beschäftigt sich mit dem Einzeichnen der Wanderrouten in den jeweiligen Karten. Die Schwierigkeit bestand darin, dass in der jeweiligen html-Seite nur die dazugehörige Tour zu sehen sein soll. Des Weiteren gibt es aufgrund der Übersichtlichkeit und der Datenmengen nur eine js für alle Detailtouren. Die Touren wurden in dem wege.js über die Funktion L.geoJSON(myLines) aufgerufen und unter einem davor festgelegten Style in die Karte eingezeichnet (siehe Abbildung 5).

```
var myStyle = {
    "color": "#ff7800",
    "weight": 5,
    "opacity": 0.65
};

L.geoJSON(myLines, {
    style: myStyle
}).addTo(map);
```

In der Funktion wird die Variable myLines aufgerufen, welche in jeder individuellen Routen-JavaScript-Datei die FeatureCollection enthält. Zudem beinhaltet die JavaScript-Datei die Koordinaten von jedem Punkt des Tracks. Dadurch, dass die Variable myLines in allen Routen.Javascript Dateien vorhanden ist, könnten theoretisch alle Routen gleichzeitig aufgerufen werden. Allerdings ist die Webseite so strukturiert, dass auf wege.js immer nur über eine der zahlreichen Routen.html Seiten wie beispielsweise Dachstein.html zugegriffen wird. Auf dieser html-Seite ist jeweils nur das dazugehörige Routen.Javascript mit den Koordinaten im Header als Quelle verlinkt, sodass nur ein Routen.Javascript gefunden wird und aus dieser die entsprechenden Koordinaten entnommen und nach dem erläuterten Verfahren eingezeichnet werden.

```
var myLines = [
    {
        "type": "FeatureCollection",
        "name": "tracks",
        "features": [{}]
```

```

    "type": "Feature",
    "properties": {
        "name": "Fototour Zell am See by Mountain Moments"
    }

    ,
    "geometry": {
        "type": "MultiLineString",
        "coordinates": [
            [
                [12.846814, 47.334159, 917.2],
                [12.846914, 47.334214, 917.3]...

```

3.3 Fotospots

Wie bereits erwähnt, ist auf jeder Tour ein Fotospot vorhanden, welcher sich als Fotomotiv besonders gut eignet. Zudem ist an dem Fotospot ein exemplarisches Foto eingefügt. Der Standort des Fotos ist durch die jeweiligen Koordinaten definiert. Da es für alle Routen nur eine JavaScript-Datei gibt, mussten die Daten der Koordinaten und die Bilder von der jeweiligen HTML-Seite an die Javascript-Seite übergeben werden. Dazu erhält jeder Wert im div-Element eine ID. Das sieht exemplarisch für die Zell am See Route folgendermaßen aus:

```

<div id="map" center-lat="47.334159" center-lng="12.846814" photo_lat="47.339"
photo_lng="12.8281389" zoom="14">
    

```

Photo_lat und *photo_lng* beschreiben hierbei die Koordinaten des Fotospots. Der Quellpfad des Fotos ist angegeben und eine ID vergeben, welche für jede Route dieselbe ist, sodass das JS jeweils auf diese zugreifen kann. Zusätzlich ist die Größe des Bildes angepasst, um eine gute Übersicht zu ermöglichen. Mit der *.getAttribute* Methode werden die Daten dann vom HTML in das JavaScript übertragen:

```

const photo_lat = div.getAttribute("photo_lat");
const photo_lng = div.getAttribute("photo_lng");
const img = document.getElementById("myfoto");

```

Der Marker kann dann mit L.marker an der richtigen Stelle gesetzt werden. Der Marker hat zunächst kein Pop-Up, dieser erscheint mithilfe des Bausteines *.bindPopup*.

```

let fotospot = L.marker([photo_lat, photo_lng], {
    icon: L.icon({
        iconSize: [32, 37],
        iconAnchor: [16, 37],
        popupAnchor: [0, -37],
        iconUrl: "icons/photo.png"
    })
}).addTo(map);
fotospot.bindPopup(
    img, {
        maxWidth: "auto"
    }
);

```

Als Inhalt erhält der Pop-Up das jeweilige Bild des Fotospots. Hierbei ist es bedeutend, die maximale Breite dynamisch an das Popup anzupassen. Ansonsten ist es möglich, dass das Foto die Grenzen des Popups überschreitet.

Das Ergebnis sieht dann wie folgt aus:

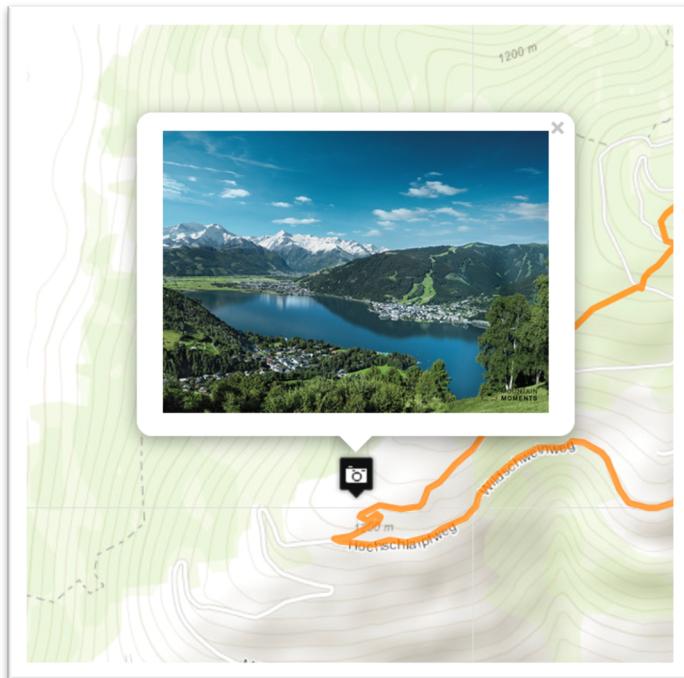


Abbildung 5: Fotospot in Zell am See

3.4 Tourinformationen

Unterhalb jeder Karte werden die Tourinformationen eingeblendet. Diese stammen aus einem ehemaligem JSON Array, welches als JavaScript mit dem Namen `weginfos.js` abgespeichert ist. Diese müssen dynamisch im `wege.js` abgefragt und an die jeweilige HTML

Seite übergeben werden, damit diese unterhalb der Karte dargestellt werden. Das Array mit den Objekten ist wie folgt aufgebaut:

```
const ETAPPEN = [
    "Nummer": "Tournummer",
    "Land": "Wanderregion",
    "Berg": "Attraktion",
    "Beschreibung": "Tourbeschreibung",
    "Tourname": "Tourname",
    "Schwierigkeit": "Schwierigkeitsgrad",
    "Dauer": "Dauer",
    "KM": "Länge",
    "Aufstieg": "Aufstieg",
    "Abstieg": "Abstieg",
}
```

Im HTML wurde dann ein Textbaustein mit einer jeweiligen ID erstellt, in das die Informationen dann eingetragen werden soll. Die IDs sind als et-key definiert.

```
<h3 id="NUM" data-xyz="5">Tourinformationen</h3>

<p> <b>Tourname: </b><span id=et-Tourname></span> </p>
<p><b>Beschreibung: </b><span id=et-Beschreibung></span></p>
<p> Schwierigkeitsgrad:
<span id=et-Schwierigkeit></span>
</p>
<p>Dauer: <span id=et-Dauer></span> h
</p>
<p>Aufstieg: <span id=et-Aufstieg></span>
          hm</p>
<p>Abstieg: <span id=et-Abstieg></span>
</p>
<p>Entfernung: <span
           id=et-KM></span> km</p>
```

Außerdem wurde im h3 eine weitere ID vergeben und die Nummer der Tour bestimmt, damit das JavaScript weiß, wo die Informationen platziert werden. Die Variable data-xyz beschreibt die Objektnummer in dem Array von weginfo.js. Die Informationen wurden wie folgt aufgerufen und wieder ins HTML übertragen:

```
let xyc = document.querySelector("#NUM");
let Nummera = xyc.dataset.xyz
```

```

let val = ETAPPEN[Nummera].key;

for (const key in ETAPPEN[Nummera]) {
    let val = ETAPPEN[Nummera][key]
    let elem = document.querySelector(`#et-${key}`);
    if (elem) {
        elem.innerHTML = val;
        console.log(val);
    };
}

```

Die Variable xyz in dem xyc.dataset.xyz Objekt entspricht einem key, in diesem Fall die dafür in dem jeweiligen Routen.html definierter Zahl. Diese Zahl wird anschließend als Variable Nummera definiert. Mit Hilfe einer for Schleife kann dann der jeweilige key aus dem Array aufgerufen werden und unter Verwendung des querySelectors und der jeweiligen ID an die richtige Stelle eingefügt werden.

3.5 Navigation

Die Navigation durch die einzelnen HTML-Seiten erfolgt mit Hilfe einer Menüleiste. Diese besteht aus Listenelementen, welche im main.css gestylt sind. Die Liste mit der ID „Navigation“ befindet sich im main-Absatz jeder HTML-Seite, um eine gute Orientierung zu gewährleisten ist es auf allen Seiten einheitlich. Über die einzelnen „href=“-Elemente wurden jeweils die passenden HTML-Seiten untereinander verlinkt. Entscheidend für die Benutzerfreundlichkeit ist es, dass der Zurück-Button im Browser-Fenster funktioniert.

```

<ul id="navigation">
    <li><a href="#">Home</a></li>
    <li class="sub">
        <a href="#">Wanderregionen</a>
        <ul id="laender">
            <li><a href="wege.html"> Österreich</a></li>
            <li><a href="zukunft.html">Deutschland</a></li>
            <li><a href="zukunft.html">Italien</a></li>
            <li><a href="zukunft.html">Frankreich</a></li>
            <li><a href="zukunft.html">Slowenien</a></li>
        </ul>
    </li>
    <li>
        <a href="infos.html">Infos</a>
    </li>
    <li>

```

```

        <a href="https://mountainmoments.de/">Mountain Moments</a>
    </li>
</ul>
```

Die Menüleiste verfügt über ein Dropdown Menü, über welches die Länder (Wanderregionen) ausgewählt werden können. Dabei war es wichtig, im CSS-File mittels Z-Wert und overflow die Hierarchie so anzupassen, dass das Dropdown-Menü nicht unter der Karte verschwindet.

```
#navigation>li.sub ul {
    width: 10em;
    margin: 0;
    padding: 0.5em 0;
    list-style: none;
    background: grey;
    position: absolute;
    top: -1000em;
    z-index: 999;
    overflow: visible;
}
```

Die einzelnen Touren werden jeweils über den Link im Popup aufgerufen und öffnen sich in einem eigenen Tab. Diese Variante ist praktisch für die Tourenplanung, da so schnell einzelne Touren miteinander verglichen werden können.

3.6 Anklickbare Länder auf der Startseite

Um die Navigation noch intuitiver zu gestalten, wurde auf der Startseite zur Map ein Layer hinzugefügt, bei dem die Features anklickbar sind. Dazu wurde aus dem Datensatz der administrativen Grenzen das Element Österreich herausgesucht und im JS-File „boundaries“ gespeichert. Das Feature in Form des Landes ist durch die Funktion onClick(e) anklickbar und mit der wege.html verlinkt. Aufgerufen wird sie in der Funktion onEachFeature.

```
function highlightFeature(e) {
    var layer = e.target;

    layer.setStyle({
        weight: 5,
        color: '#666',
        dashArray: '',
        fillOpacity: 0.7
    });

    e.source.setFeatureState(e.target, {highlighted: true});
}

function resetHighlight(e) {
    var layer = e.target;

    if (!layer.get('highlighted')) {
        return;
    }

    layer.setStyle(style);
    e.source.setFeatureState(layer, {highlighted: false});
}
```

```
        });
    };

function resetHighlight(e) {
    geojson.resetStyle(e.target);
};

function onClick(e) {
    window.open("https://ibkdrei.github.io/wege.html");
}

function onEachFeature(feature, layer) {
    layer.on({
        mouseover: highlightFeature,
        mouseout: resetHighlight,
        click: onClick
    });
};
```

Im weiteren Projektverlauf sollen weitere Länder hinzugefügt werden, welche jeweils zu der passenden länder.html führen.

4. Verwendete Plugins

In Kombination mit Leaflet können verschiedene Plugins verwendet werden, um die Funktionsauswahl auszuweiten. Diese werden in der Regel durch die Community oder Github-Nutzer erstellt und zur Verfügung gestellt (Leaflet 2020). Auf der offiziellen Leaflet-Seite sind diverse Möglichkeiten aufgelistet, wobei sich in diesem Projekt auf folgende Plugins beschränkt wurde: Der Basemap Provider, der Rainviewer, der Live-Standort und der Maßstab.

4.1 Rainviewer

Auf der index.html wurde das Rainviewer-Plugin von dem Nutzer mwasil eingebunden. Dem Projekt wurden die dafür notwendigen Dateien LeafletLocationShare.js, die leaflet.rainviewer.js und leaflet.rainviewer.css hinzugefügt. Aufgrund der unterschiedlichen Benennung der Karten war es essentiell, den Kartennamen in dem vorhandenen Plugin „map“ durch „minimap“ zu ersetzen. Das Plugin ist per Mausklick über den Wolken-Button in der linken unteren Ecke aktivierbar. Anschließend wird das Niederschlagsradar mit zahlreichen Features wie unterschiedlicher Niederschlagsintensität dargestellt (Wasilewski 2019).

4.2 Standort

Aufgrund der steigenden Aufrufzahlen von Webseiten mit Hilfe mobiler Endgeräte, erhalten bestimmte Features eine zunehmende Bedeutung. Insbesondere der aktuelle Standort und die Weitergabe dieses Standortes erhält in der ständigen Kommunikation über mobile Geräte eine eminente Bedeutung im Zusammenhang der Tourenplanung. Daher ist ein wichtiger Bauteil für die Umsetzung dieses Projekts die Implantation eines möglich genauen Standorts. Dieser ist bei der Detailansicht der jeweiligen Tour über das dafür typische Symbol  aktivierbar. Durch die Aktivierung des Standorts wird der Nutzer gefragt, ob er seinen Standort freigeben will. Bei aktiver Standortfunktion am Endgerät und Zustimmung der Abfrage wird der tatsächliche Standort des Nutzers mit einer hohen Genauigkeit ermittelt und dargestellt. Zusätzlich entsteht ein Feld, in welchem der Nutzer eine Wortgruppe eingeben kann und aus welchem anschließend eine URL erstellt wird. Diese URL kann weiterverschickt

werden, wodurch der Empfänger dieser URL die Karte und den Standort der anderen Person erhält.

Wenn der Standort des Endgeräts nicht aktiviert ist oder die Nachfrage abgelehnt wird, kommt es zu einer fehlerhaften Lokalisierung des Standorts. Die Idee hierbei ist, dass der Standort verschoben werden kann, um einen beliebigen Standort weiterzuleiten. Diese Funktion kann insbesondere bei der zukünftigen Tourenplanung bedeutend sein, um beispielsweise einen Treffpunkt für den nächsten Tag mitzuteilen (siehe Abbildung 6).

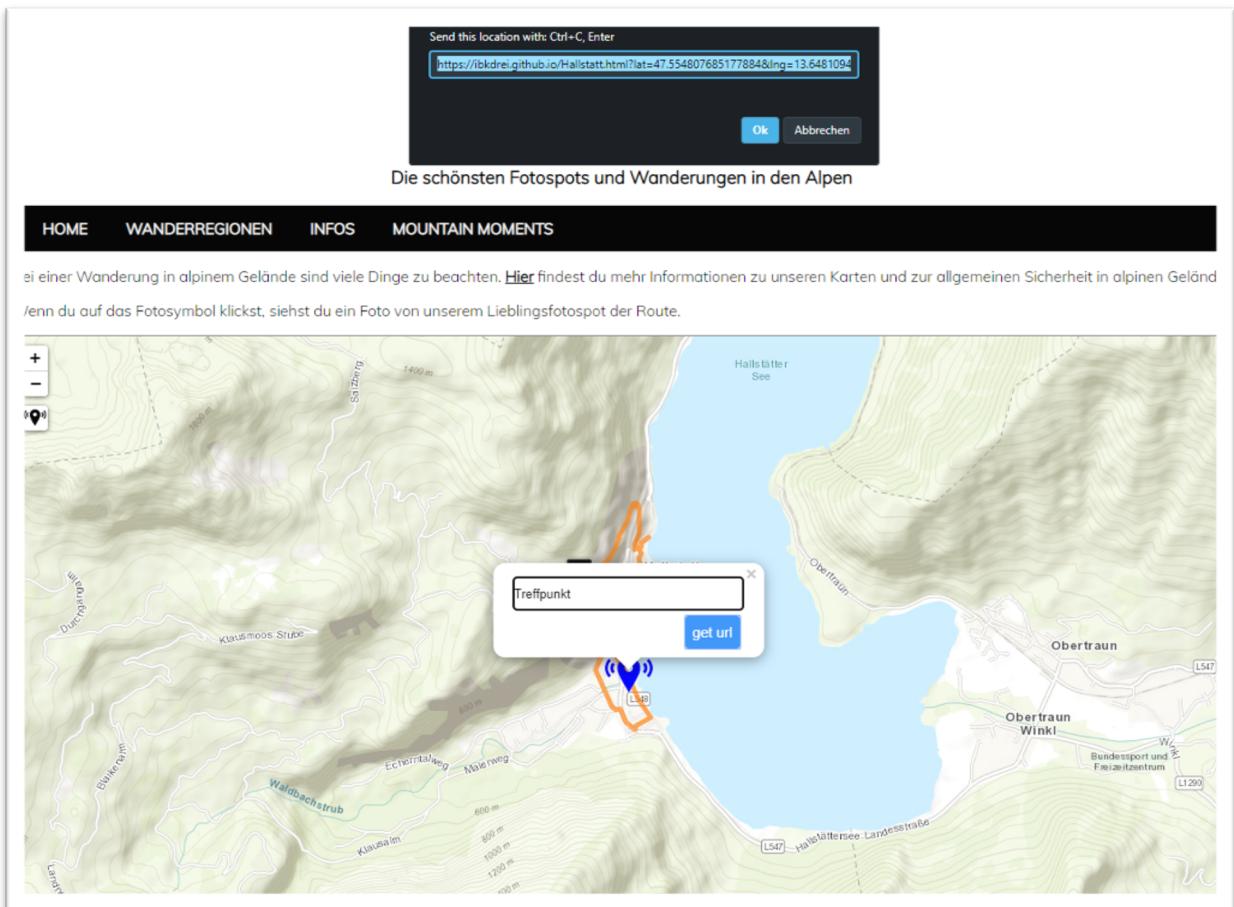


Abbildung 6: Lokalisierung eines möglichen Treffpunkts über die GPS-Funktion

Das Tool stammt vom Github Nutzer CliffCloud, wobei dessen Tool leicht angepasst und verändert wurde. Der Code ist in der JavaScript-Datei Leaflet.LocationShare.js enthalten und wird über das wege.js abgerufen.

4.3 Maßstab

In die Wanderkarten wurde mit Hilfe des Control Scale Plugins ein Maßstab eingefügt:

```
L.control.scale({  
    imperial: false  
}).addTo(map);
```

Um lediglich relevanten metrischen Werte zu erhalten, werden die englischen Maßeinheiten ausgeschaltet. Der Maßstab fungiert dynamisch und passt sich je nach Zoomfaktor automatisch an (siehe Abbildung 8).



Abbildung 7: L.ControlScale

4.4 Leaflet Providers

Mit Unterstützung des Leaflet Providers Plugins ist es viabel, verschiedene Kachel-Layer in die Map einzubinden. Abhängig nach Inhalt und weiteren Kriterien können verschiedene Basemaps ausgewählt werden. Für das Projekt wurde sich für die Startseite für den Layer „Stamen Terrain Background“ und für die Wanderkarten für den Layer „Esri World Topo Map“ entschieden. Der Link zu den Providern muss in den HTML-Seiten eingebunden sein und in den Javascript über L.tilelayer.provider („Name der Karte“) aufgerufen werden (Leaflet Providers 2020).

5. Verbesserungsvorschläge

Die Umsetzung der Seite erfüllt einen Großteil der Features, welche im Vornherein als Ziel festgelegt wurde. Trotzdem gibt es einige Verbesserungsmöglichkeiten, welche im Folgenden vorgestellt und bei weiterer Umsetzung umgesetzt werden können. Wie bereits apostrophiert, nimmt die Anzahl der Nutzer mit mobilen Endgeräten in den letzten Jahren stetig zu. Daher ist es bedeutend, dass die heutigen Webseiten ebenfalls auf diesen Geräten einwandfrei funktionieren. Hierzu gibt es Anleitungen von Leaflet, welche in diesem Projekt nicht umgesetzt wurden.

Des Weiteren gibt es eine starke Abhängigkeit der Funktionalitäten von einem Browser mit einem aktuellen Java Plugin.

Zusätzlich gibt es, wie bereits erwähnt, einen fehlerhaften Standort bei deaktiviertem Standort. Trotzdem kann das Standortsymbol für weitere Verwendungen eingesetzt werden, wobei es bei dem Nutzer hierbei zu Missverständnissen kommen könnte. Dem hinzuzufügen ist, dass die Qualität einer Webanwendung in der Regel durch die Benutzerfreundlichkeit charakterisiert ist. Auf diesen Punkt wurde viel Wert gelegt, wobei es an einigen Stellen weiterhin Verbesserungsbedarf gibt.

Des Weiteren sehen einige Nutzer möglicherweise Verbesserungsansätze in der verwendeten Leaflet-Karte in dem wege.js. Für die meisten Applikationen und eine Übersicht ist diese vollkommen ausreichend und wurde daher in diesem Projekt verwendet. Zunächst wurde das Projekt mit einer aufwendig erstellten und selbst gestylten Mapbox-Karte durchgeführt. Allerdings entschieden sich die Autoren nach Rücksprache mit den Dozenten, dass Mapbox vor allem für privatwirtschaftliche Zwecke genutzt wird und für das Projekt zunächst eine Leaflet-Karte bevorzugt wird. Trotzdem wird sie hier kurz erwähnt, um die Karten gegenüberzustellen (siehe Abbildung 8).

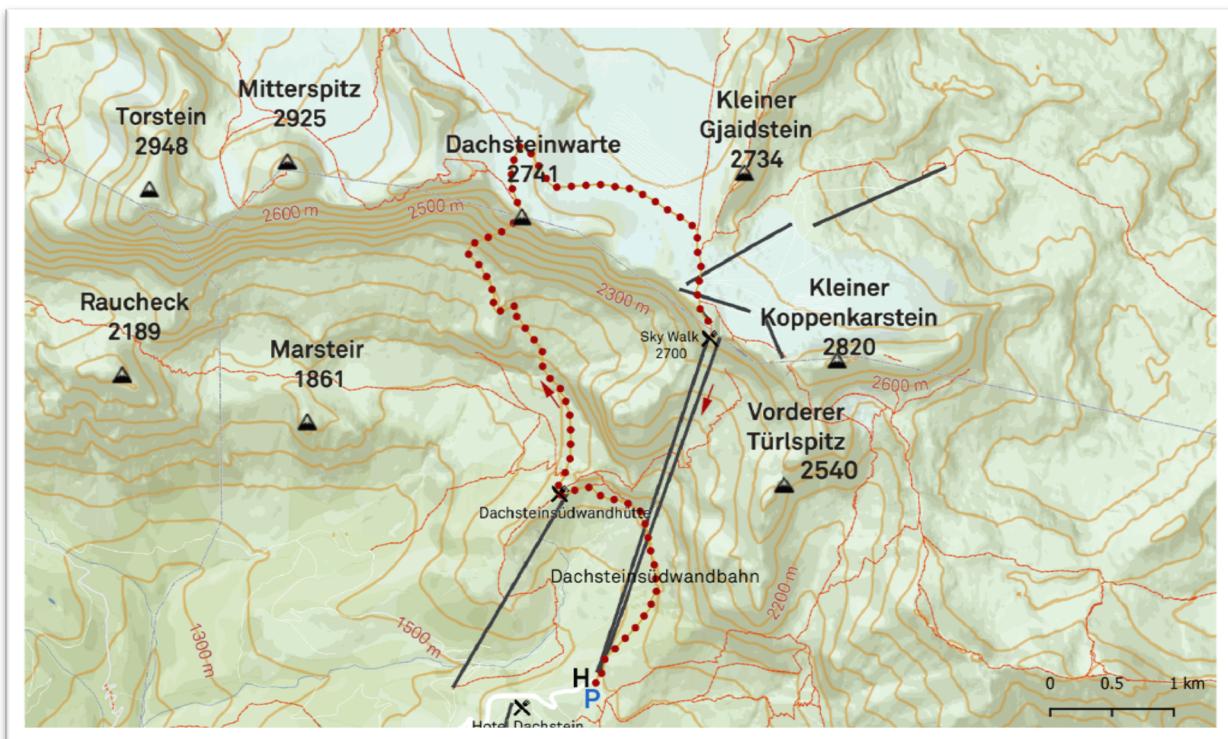


Abbildung 8: Eigene Mapbox-Karte der Tour Dachstein

6. Fazit

Abschließend soll ein kurzes Fazit der Webseite erfolgen. Zusammenfassend war die Umsetzung der Aufgabenstellung und des Konzepts möglich. Es ist gelungen, die von Mountain Moments zur Verfügung gestellten gpx Tracks und Fotografien auf einer Website aufzubereiten, welche benutzerfreundlich und intuitiv zu bedienen ist. Die Seiten der Wanderrouten enthalten alle notwendigen Informationen, die ein Nutzer von einem Wanderführer erwarten würde. Durch den intuitiven Aufbau der Seite, sollte sich der Benutzer schnell zurechtfinden und an die Infos gelangen, welche er benötigt. Dazu beiträgt die Strukturierung der Routen in einzelne HTMLs, die erlauben immer wieder in den Tab der Übersichtskarte zu wechseln, um eine neue Route auszuwählen. Die Plugins des Standorts oder des Maßstabs helfen bei der Orientierung und haben daher einen positiven Einfluss auf die Funktionalität der Seite. Das Einbauen der Fotospots und das Zeigen der jeweiligen Bilder geben dem Benutzer die Möglichkeit, im Vorhinein einen Eindruck der Route zu erhalten. Für eine höhere Funktionalität der Webseite in der Zukunft können dann noch weitere Länder ergänzt werden, von denen es auch Datensätze zu den Tracks und weitere Fotos vorhanden sind. Zusätzlich gibt es noch weitere Leaflet Plugins, wie das Höhenprofil, welche die Seite weiterhin verbessern könnten. Auch das Einfügen der Mapbox Basemap hätte einen positiven Einfluss, da diese für die Nutzung zugeschnittene Informationen beinhaltet.

Das Projekt hat einen Einblick in die Erstellung von Online-Wanderführer erlaubt, welche in der aktuellen Zeit zunehmend an Attraktivität und Relevanz gewinnen. Zudem wurde ein tieferes Verständnis für den Entwicklungsprozess einer Webseite mit den verschiedenen Bestandteilen entwickelt. Die Verfasser sind daher weiterhin motiviert, das Projekt zu verbessern und zu ergänzen.

7. Quellenverzeichnis

Carto (2020): Boundries of European Countries.

URL: https://common-data.carto.com/tables/ne_adm0_europe/public.

Zuletzt geöffnet: 17.07.2020.

CliffCloud (2016): Leaflet.Locationshare.

URL: <https://github.com/CliffCloud/Leaflet.LocationShare>.

Zuletzt geöffnet: 17.07.2020.

FontAwesome (2020): Icons.

URL: <https://fontawesome.com>.

Zuletzt geöffnet: 17.07.2020.

Leaflet Providers (2020): Leaflet-providers.

URL: <https://github.com/leaflet-extras/leaflet-providers>.

Zuletzt geöffnet: 17.07.2020.

MapIcons (2020): Icons. URL: <https://mapicons.mapsmarker.com/>.

Zuletzt geöffnet: 17.07.2020.

Marcin Wasilewski (2019): Leaflet.Rainviewer.

URL: <https://github.com/mwasil/Leaflet.Rainviewer>.

Zuletzt geöffnet: 17.07.2020.

Schwager, M. (2020): Mountain Moments. Die schönsten Fotospots und Wanderungen in den Alpen. Innsbruck.