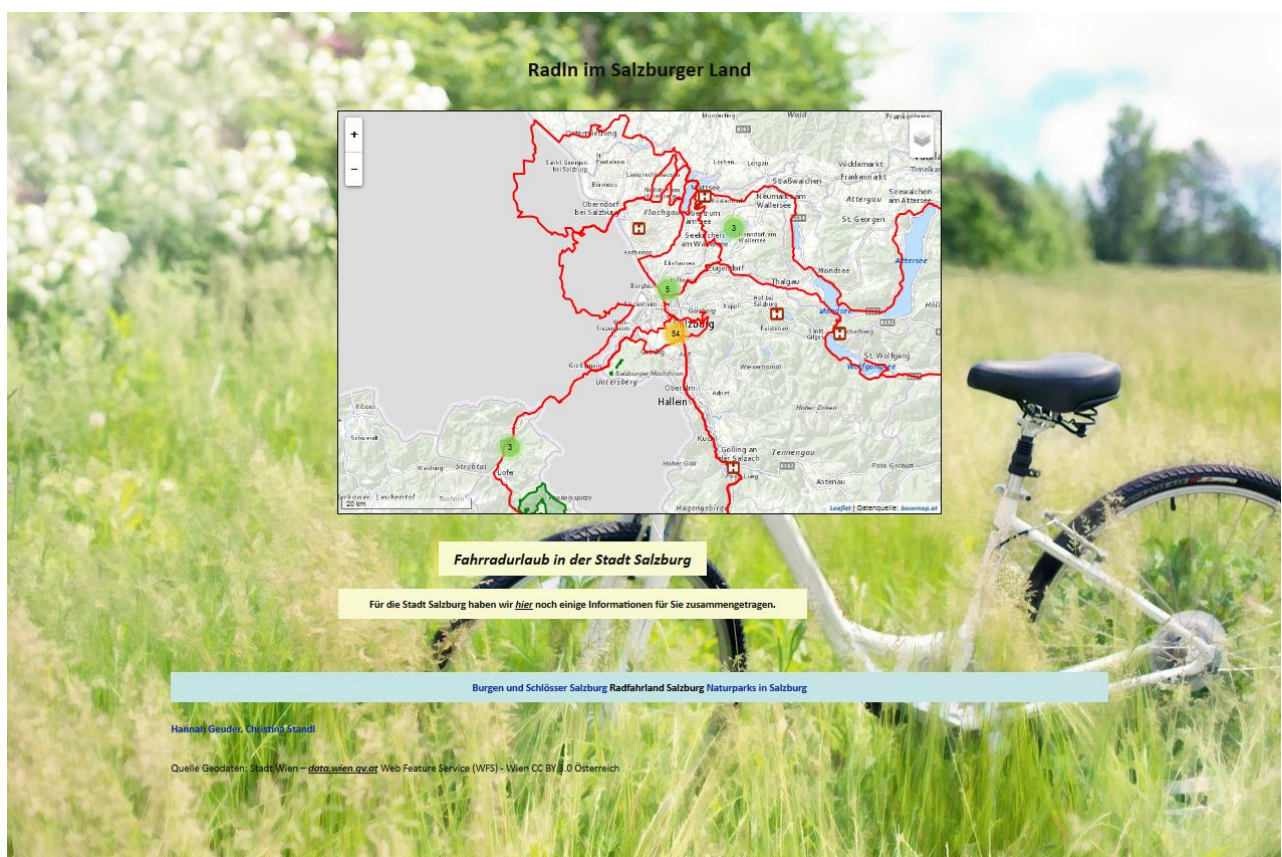


## Projektbericht “Radfahrland Salzburg”



## Inhalt

1	Kurzbeschreibung: .....	3
2	Übersicht über die Seiten .....	3
3	Erstellung der HTML-Seiten .....	4
3.1	Erstellung der Seite "Radfahren im Salzburger Land" .....	4
3.1.1	HEAD .....	4
3.1.2	SCRIPT .....	5
3.2	Erstellung der Seite "Radfahren in der Stadt Salzburg" .....	8
3.3	Erstellung der HTML-Seiten Naturparks und Burgen und Schlösser .....	8
4	Auslagern der Skripts .....	9
5	Erstellung der CSS-Files .....	10
6	Zusammenfassung und Fazit .....	11
7	Verwendete Datensätze/Datengrundlagen .....	12

# 1 Kurzbeschreibung:

Mittelpunkt des Projektes “Radfahrland Salzburg” stellt eine HTML-Seite mit einer Übersichtskarte dar, die interessante touristische Radwege im Bundesland Salzburg zeigt. Zusätzlich zu den Radwegen werden touristisch interessante Orte - Burgen und Schlösser sowie Naturparks -, die mit den ebenfalls mit dem Rad gut erreichbar sind, dargestellt. Als Ziel hinter diesem Projekt könnte das Bestreben stehen in Salzburg auch für touristische Aktivitäten künftig das Fahrrad als Transportmittel zu promoten. Eine weitere Seite mit einer Karte für das Salzburger Stadtgebiet mit eingezeichneten Radwegen, Kinderspielplätzen und Wifi-Standorten bietet Informationen zur Benutzung des Rades in der Stadt Salzburg - auch für Nicht-Locals. Zwei zusätzliche HTML-Seiten bieten weitere Informationen zu den dargestellten Naturparks sowie ausgewählten Burgen und Schlössern.

## 2 Übersicht über die Seiten

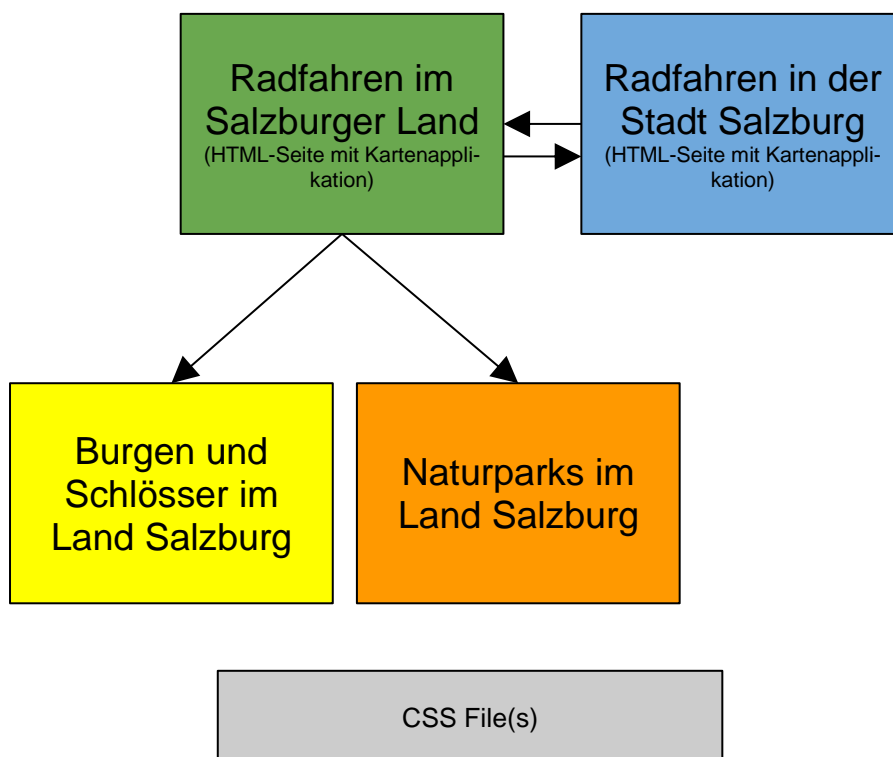


Abb. 1: Übersicht und Zusammenhang zwischen den Seiten

Die Seiten “Radfahren im Salzburger Land” und “Radfahren in der Stadt Salzburg” sind jeweils mit einer Kartenapplikation ausgestattet und verweisen aufeinander. Für weitere Informationen zu den Karteninhalten der Seite “Radfahren im Salzburger Land” wurden zwei weitere HTML-Seiten zu den Themen Naturparks in Salzburg und Burgen und Schlösser in Salzburg (Auswahl) erstellt. Das Design aller HTML-Seiten wurde in CSS-Dateien ausgelagert.

## 3 Erstellung der HTML-Seiten

### 3.1 Erstellung der Seite "Radfahren im Salzburger Land"

#### 3.1.1 HEAD

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
7   <title>Radfahrland Salzburg</title>
8
9   <link rel="stylesheet" href="https://unpkg.com/leaflet@1.0.3/dist/leaflet.css" />
10  <script src="https://unpkg.com/leaflet@1.0.3/dist/leaflet.js"></script>
11
12  <!-- GeoJSON Datenskripts -->
13  <script src="Naturpark.json.js"></script>
14  <script src="Burgen.json.js"></script>
15  <script src="radwegLinien.json.js"></script>
16
```

Screenshot 1 head der index Datei

Zuerst wurde in einem HTML-Dokument der <head> definiert. Darin befindet sich die Angabe zur Codierung der Unicode-Zeichen, der Titel, welcher im Reiter des Internet Explorers sichtbar ist, die zu verwendenden GeoJSON Skripte, die eingebundenen CSS-Files für die Leaflet-Applikation und die HTML-Seite sowie in einem <style>-tag die Größe und Platzierung der Karte.

```
] <style>
  #sMap {
    width: 900px;
    height: 600px;
    border: 1px solid black;
    margin: auto;
  }
</style>
```

Screenshot 2 style im head der index Datei

### 3.1.2 SCRIPT

```
<script>
// WMTS-Layer definieren - Quelle: http://www.basemap.at/wmts/1.0.0/WMTSCapabilities.xml
var layers = {
  geolandbasemap: L.tileLayer("https://{s}.wien.gv.at/basemap/geolandbasemap/normal/google3857/{z}/{y}/{x}.png", {
    subdomains: ['maps', 'maps1', 'maps2', 'maps3', 'maps4'],
    attribution: 'Datenquelle: <a href="http://www.basemap.at/">basemap.at</a>'
  }),
  bmapgrau: L.tileLayer("https://{s}.wien.gv.at/basemap/bmapgrau/normal/google3857/{z}/{y}/{x}.png", {
    subdomains: ['maps', 'maps1', 'maps2', 'maps3', 'maps4'],
    attribution: 'Datenquelle: <a href="http://www.basemap.at/">basemap.at</a>'
  }),
  bmapoverlay: L.tileLayer("https://{s}.wien.gv.at/basemap/bmapoverlay/normal/google3857/{z}/{y}/{x}.png", {
    subdomains: ['maps', 'maps1', 'maps2', 'maps3', 'maps4'],
    attribution: 'Datenquelle: <a href="http://www.basemap.at/">basemap.at</a>'
  }),
  bmaphidpi: L.tileLayer("https://{s}.wien.gv.at/basemap/bmaphidpi/normal/google3857/{z}/{y}/{x}.png", {
    subdomains: ['maps', 'maps1', 'maps2', 'maps3', 'maps4'],
    attribution: 'Datenquelle: <a href="http://www.basemap.at/">basemap.at</a>'
  }),
  bmaporthofoto30cm: L.tileLayer("https://{s}.wien.gv.at/basemap/bmaporthofoto30cm/normal/google3857/{z}/{y}/{x}.png", {
    subdomains: ['maps', 'maps1', 'maps2', 'maps3', 'maps4'],
    attribution: 'Datenquelle: <a href="http://www.basemap.at/">basemap.at</a>'
  }),
  osm: L.tileLayer("http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png", {
    subdomains: ['a', 'b', 'c'],
    attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>'
  })
};
```

Screenshot 3 Kartengrundlage einbinden

Im Script-Teil wurden zuerst die Kartenlayer eingefügt. Dazu wurden die Variable 'layers' definiert, um anschließend die verschiedenen Layer einzubinden. In einem weiteren Schritt wurde die Map definiert (Variable map) und ein Mittelpunkt in Koordinaten sowie ein Zoom-Level angegeben. Mit der Leaflet Applikation L.control.scale wurde zum Weiteren eine Maßstabsleiste hinzugefügt. Mit dem Befehl addTo (map) werden die Elemente (in weiterer Folge auch die JSON-Layer zur Karte hinzugefügt.

```
// Karte definieren und Ausschnitt setzen
var map = L.map('sMap', {
  layers: [layers.bmapgrau],
  center: [47.463351, 13.256514],
  zoom: 9
});

// Maßstab hinzufügen
L.control.scale({
  maxWidth: 200,
  metric: true,
  imperial: false
}).addTo(map);
```

Screenshot 4 Karte definieren und Maßstab

Anschließend wurden die verschiedenen (bereits im head eingebundenen) JSON-Layer visualisiert. Bei den Radwegen wurden hierfür eine kräftige Farbe und ein Pop-Up mit der vorhandenen Information in den Geo-JSON-Dateien verwendet. Einige Radwege hatten auch einen eingetragenen Wert in der Klassifikation nach der Schwierigkeit, leider besaßen viele

andere aber nur Null-Werte weswegen es uns nicht sinnvoll erschien, die Radwege nach der Schwierigkeit einzufärben.

```
var Linienradwege = L.geoJSON(window.radwegLinien, {
  onEachFeature : function (feature, layer) {
    var markup = '<h2>' + feature.properties.BEZEICHNUNG + '</h2>';
    markup += '<p>' + 'Streckenverlauf: ' + feature.properties.Bedeutung + '</p>';
    markup += '<p>' + feature.properties.Homepage + '</p>';
    layer.bindPopup(markup);
  },
  style : function (feature) {
    var fill = 'red';
    return {
      color : fill,
    }
  }
}).addTo(map);
```

Screenshot 5 Variable Radwege definieren

Mit der Visualisierung der Naturparke wurde ähnlich verfahren, auch hier wurde eine Farbe für alle fünf Naturparks verwendet und ein Pop-Up eingefügt, das den Link zur Homepage der Salzburger Naturparke enthält.

```
// Farbige Flächen der Naturparke mit Popup hinzufügen
var Flaechennaturpark = L.geoJSON(window.naturparkFlaeche, {
  onEachFeature : function (feature, layer) {
    var markup = '<h2>' + feature.properties.NAME + '</h2>';
    markup += '<p>' + '<a href="https://www.naturparke.at/naturparke/salzburg/" >a>' + '</p>';
    layer.bindPopup(markup);
  },
  style : function (feature) {
    var fill = 'green';
    return {
      color : fill,
    }
  }
}).addTo(map);
```

Screenshot 6 Variable Naturparks definieren

Die Burgen und Schlösser wurden wie die Naturparke und Radwege als Variable definiert, jedoch unterscheiden sie sich von diesen, da es Punktdaten sind, wodurch die Einbindung durch `point.To.Feature` erfolgen muss. Außerdem wurde aufgrund der hohen Anzahl und der räumlichen Dichte um die Stadt Salzburg ein Cluster erstellt. Dieses wird als Variable definiert (hier: `clusterGruppeburg`) und durch die Leaflet-Anwendung `L.marker.ClusterGroup` eingebunden. Das Cluster wird direkt mit dem Befehl `.addTo(map)` zur Karte hinzugefügt, was zur Folge hat, dass sobald das Cluster `clusterGruppeburg` und der Befehl `.addTo(clusterGruppe)` verwendet wird, die Inhalte automatisch in die Karte integriert werden.

Zudem wurden Icons statt Marker zur Darstellung verwendet, die noch mit einem Pop-Up Fenster versehen wurden. Die Kombination aus Pop-Up Fenster und Darstellung mit Icon führt dazu, dass das Pop-Up Fenster mit der Funktion `onEachFeature` eingebunden werden muss. So wird das Pop-Up mit einer Variable (`markup`) definiert und die jeweiligen Informationen werden durch den Befehl `feature.properties.xx` dargestellt, wobei `xx` die Bezeichnung für die jeweiligen properties in der JSON-Datei der Burgen und Schlösser darstellt. Durch



den Befehl `layer.bindPopup(markup)` wird die Variable `markup` in die Variable der Burgen und Schlösser eingebunden, wodurch das Pop-Up Fenster für die jeweilige Burg bzw. das jeweilige Schloss erscheint.

Im Anschluss wird der Icon mit der Funktion `pointToLayer` in das Skript implementiert. Dabei wird der aus Leaflet stammende `L.marker` mit der Option `Icon` verwendet, der den jeweiligen Punkt auf Basis der geographischen Koordinaten setzt. Außerdem wird die Größe (`iconSize`), (`iconAnchor`), (`popupAnchor`) und die Linkadresse des Icons (`iconUrl`) festgelegt. Nach korrekter Klammersetzung wird die Definition der Variable Burgen und Schlösser mit dem Hinzufügen der Variable zur Karte durch `.addTo(clusterGruppeburg)` beendet. Der folgende Screenshot zeigt die Einbindung der Burgen und Clusterbildung in der Index Datei.

```
//Cluster für Burgen definieren
var clusterGruppeburg = L.markerClusterGroup().addTo(map);

var Punkteburgen = L.geoJSON(window.burgenPunkte, {
  onEachFeature: function(feature, layer) {
    var markup = '<h2>' + feature.properties.Name + '</h2>';
    markup += '<li>' + 'Adresse: ' + feature.properties.Adresse + '</li>';
    markup += '<li>' + 'Ort: ' + feature.properties.Ort + '</li>';
    markup += '<li>' + 'Homepage: ' + feature.properties.DBLink + '</li>';

    layer.bindPopup(markup);
  },

  pointToLayer: function(feature, latlng) {
    return L.marker(latlng, {
      icon: L.icon({
        iconSize: [20, 20],
        iconAnchor: [1, 1],
        popupAnchor: [1, 1],
        iconUrl: 'http://www.data.wien.gv.at/icons/burgschlossogdsichtbar.png'
      })
    });
  }
}).addTo(clusterGruppeburg);
```

Screenshot 7 Variable Burgen und Schlösser und Cluster einfügen

Abschließend wurde mit `L.control.layers` noch eine Schaltfläche am Kartenrand eingefügt, mit der die verschiedenen Layer ein- bzw. ausgeschaltet werden können. In Anführungszeichen sind die Namen angegeben, die im Menü aufscheinen. Eingebunden werden die Layer hier mit den im Vorhinein definierten Variablen ( z.B. `var = Punkteburgen`)

```
L.control.layers({
  "basemap.at - STANDARD": layers.geolandbasemap,
  "basemap.at - GRAU": layers.bmapgrau,
  "basemap.at - OVERLAY": layers.bmapoverlay,
  "basemap.at - HIGH-DPI": layers.bmaphidpi,
  "basemap.at - ORTHOFOTO": layers.bmaporthofoto30cm,
  "OpenStreetMap": layers.osm
}, {
  "Touristische Radwege": Linienradwege,
  "Burgen und Schlösser": Punkteburgen,
  "Naturparks": Flaechnaturpark
}).addTo(map)
```

Screenshot 8 Schaltfläche

## 3.2 Erstellung der Seite “Radfahren in der Stadt Salzburg”

Die HTML-Seite “Radfahren in der Stadt Salzburg” wurde quasi analog zur Seite “Radfahren im Salzburger Land” erstellt. Weil es sich hier um einen kleineren Kartenausschnitt und damit um einen größeren Maßstab handelt, haben wir uns dafür entschieden, hier kein Cluster für die Icons zu erstellen, um eine bessere Übersichtlichkeit der Karte bzw. leichtere Verständlichkeit zu ermöglichen.

## 3.3 Erstellung der HTML-Seiten Naturparks und Burgen und Schlösser

Die HTML-Seiten mit den Inhalten zu ausgewählten Burgen und Schlössern sowie zu den Naturparks in Salzburg wurden von der Struktur und des Designs (CSS-Files) gleich erstellt, um eine Zusammengehörigkeit - einen durchgehenden Faden - somit eine leichtere Orientierung für den/ die Benutzer/in zu schaffen.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Burgen und Schlösser in Salzburg </title>
6      <link type="text/css" rel="stylesheet" href="styles.css" />
7
8  </head>
```

Screenshot 9 head der HTML-Seite Burgen und Schlösser

Ebenso wie bei den HTML-Seiten mit Kartenapplikation besteht der erste Teil des HTML-Dokuments aus einem <head>, welcher Informationen zur Codierung, den Titel des Reiters und die Einbindung des für das Design ‘verantwortliche’ CSS-File enthält.

Anschließend erfolgt bei diesem HTML-Dokument sofort der <body>, der die Texte, Bilder und Links - also den “eigentlichen” Seiteninhalt enthält.

```
10 <body>
11 <h1> Burgen und Schlösser in Salzburg
12 </h1>
13
14 <h2> Zu Gast bei Salzburgs Adel
15 </h2>
16 <h3> Festung Hohensalzburg
17 </h3>
18 <a href="https://cdn.pixabay.com/photo/2015/07/12/17/23/salzburg-842215_960_720.jpg">
19 
20 </a>
21 <p class="info"> Die Festung Hohensalzburg ist ein echter Blickfang hoch über den barocken Türmen der Stadt. Als weithin sichtbares Wahrzeichen ist
22 die Burg unverkennbarer Teil der weltberühmten Silhouette Salzburgs. Mächtig erscheint sie dem Besucher aus der Ferne, hautnah wird hinter ihren
23 dicken Gemäuern Geschichte erlebbar.
24 </p>
25 <a href="http://www.salzburg-burgen.at/de/hohensalzburg/">http://www.salzburg-burgen.at/de/hohensalzburg/</a>
```

Screenshot 10 body der HTML-Seite Burgen und Schlösser

Im <body> wurden verschiedene Überschriften (z.B. <h2>...) eingefügt. Weiters wurden Texte mittels eines <p>-tags eingefügt. Mithilfe des Befehls class= z.B. “info” können “Klassen”



also Gruppennamen vergeben werden, die im CSS-File dann als “Klasse” formatiert werden können. Alle Absätze mit der Klasse “info” können also im CSS-File ausgelagert gleich formatiert werden. Damit ist es möglich, unterschiedliche Absätze (Gruppen) gleich zu formatieren. (z.B. class=“info”, class=“beschreibung”, class=“kommentare”...)

Zusätzlich wurden links mit dem <a href=“betreffenderlink”> linktext </a> eingefügt. In Anführungszeichen ist dabei der Link anzugeben, zwischen dem Ende des Anfags-<a>-tags und dem End-tag ist dann jener Text anzugeben, der angezeigt werden soll. Bei Klicken auf den Text wird der Benutzer/die Benutzerin zur Website, die in Anführungszeichen angegeben wurde, weitergeleitet.

Für das Einfügen von Bildern wurden der <img>- tag verwendet. Dabei muss darauf geachtet werden, dass beim verlinken des Bildes der korrekte Pfad angegeben wird. Liegt das Bild im gleichen Ordner wie die HTML-Seite, reicht der Name der Datei, ansonsten muss die Ordnerstruktur angegeben werden. Sollte das Bild aufgrund von Problemen mit dem Internetexplorer nicht angezeigt werden, kann ein Alternativtext “alt=” innerhalb des img-tags angegeben werden.

```
97 <nav>
98 <p> Navigieren zu: </p>
99 <a class="previous" href="naturschutzgebiete.html">Naturschutzgebiete in Salzburg</a>
100 <span class="current">Burgen und Schlösser in Salzburg</span>
101 <a class="next" href="index.html">Radfahren in Salzburg</a>
102 </nav>
103 <p class="author">Hannah Geuder, Christina Standl</p>
104 </body>
105 </html>
```

Screenshot 11 Navigationsleiste

Am Ende der HTML-Seiten wurde jeweils ein Navigationselement mittels des <nav>-tags angebracht. Hier gilt wiederum der link in Anführungszeichen als jener, zu dem die/der Benutzer/in bei klicken auf den (hier schwarz geschriebenen) angezeigten Text weitergeleitet wird. Auch hier wurden die links Klassen mittels “class=” zugeordnet, um im CSS-File eine gesonderte Schriftart, Hingergrundfarbe etc. einstellen zu können.

Bei den Namen der Autorinnen wurde dies ebenfalls so gemacht.

## 4 Auslagern der Skripts

Die Javascript files für beide Karten wurden zur besseren Übersichtlichkeit in eigene Javascript-Dateien ausgelagert (radelInland.js und radelInstadt.js). Dafür muss am Anfang der js-Datei der Befehl window.onload () { stehen, nach dem Einfügen des Inhalts wird dieser mit }; beendet. Die Struktur für unser Projekt ergibt sich also wie folgt:

Name	Änderungsdatum	Typ	Größe
bilder	19.06.2017 14:41	Dateiordner	
css	19.06.2017 14:41	Dateiordner	
icons	19.06.2017 14:40	Dateiordner	
js	19.06.2017 14:40	Dateiordner	
burgen	19.06.2017 13:39	HTML-Dokument	7 KB
index	19.06.2017 14:35	HTML-Dokument	2 KB
indexstadt	19.06.2017 14:35	HTML-Dokument	2 KB
Naturparks	19.06.2017 13:38	HTML-Dokument	4 KB
radelnland	19.06.2017 13:45	JScript-Skriptdatei	5 KB
radelnstadt	19.06.2017 14:36	JScript-Skriptdatei	5 KB

Screenshot 12 Ordnerstruktur

## 5 Erstellung der CSS-Files

Es wurden zwei verschiedene CSS-Files erstellt: Zum einen die CSS Datei für die HTML-Seiten des index und indexstadt und zum anderen die Datei für die HTML-Seiten der Naturparks sowie der Burgen und Schlösser. Anzumerken ist, dass zwar zwei verschiedene CSS Dateien für Burgen und Schlösser und Naturparks (style\_b und style\_n) vorhanden sind, jedoch sind diese identisch bis auf die Einbindung des Hintergrundbilds durch die Eigenschaft background-image.

Allgemein wurde für die Überschriften (h1-h4), den Fließtext (p), die Einbindung der Links (a), die Bilder (img), die Navigationsleiste (nav) und die unterschiedlichen Klassen ein bestimmte Eigenschaften definiert, allerdings ähnelt sich das Erscheinungsbild. Des Weiteren wurde für die HTML Datei sowie dem body Eigenschaften (Farbe, Abstand zum Rand, etc.) definiert.

Folgender Ausschnitt zeigt den standardmäßigen Aufbau mit den verwendeten Eigenschaften für Textbereiche.

```
h2{
    font-family: Calibri;
    font-weight: lighter;
    text-align: center;
    font-style: italic;
    font-size: 20px;
    padding-top: 16px;
    padding-bottom: 16px;
    color: #0F1010;
}
```

Screenshot 13 CSS Eigenschaften Textbereich

font-family gibt die Schriftart an, font-weight die Schriftstärke, text-align die Ausrichtung, font-style den Schriftstil, font-size die Schriftgröße, color die Schriftfarbe und mit padding-top bzw. padding-bottom werden die Zwischenräume zwischen den Elementinhalten und der oberen bzw. unteren Elementgrenze.

Wie schon erwähnt wurden auch Eigenschaften für Klassen definiert. Dies wurde zur besseren Differenzierung der einzelnen Inhalte der HTML-Seite und der einheitlichen Darstellung, wie zum Beispiel der Navigationsleiste, durchgeführt. Dies wird an den Klassen `previous` und `next` für die Navigationsleiste und den Klassen `navigation` und `author` aufgezeigt.

```
.previous, .next{
    font-family: Calibri;
    font-weight: bold;
    font-style: normal;
    text-decoration: none;
    text-align: center;
    padding-top: 12px;
    padding-bottom: 12px;
    color: #0A2F9F;
}
.navigation, .author{
    font-family: Calibri;
    font-weight: bold;
    font-style: normal;
    text-decoration: none;
    text-align: center;
    padding-top: 12px;
    padding-bottom: 12px;
    color: #FFFFFF;
}
```

Screenshot 14 Klassen im CSS-file

## 6 Zusammenfassung und Fazit

Die erstellten HTML-Seiten geben eine Übersicht über mögliche Radrouten und Informationen über kulturelle Aktivitäten (Burgen und Schlösser) sowie Aktivitäten in der Natur (Naturparks) im Salzburger Land. Außerdem gibt es eine detaillierte Ansicht zur Stadt Salzburg mit weiteren Radwegen sowie relevante Informationen wie Wifi Standorte und Kinderspielplätzen. Durch die interaktive Karte werden recht viele Informationen wiedergegeben, die in den einzelnen Markern und deren Pop-Ups verarbeitet sind.

Es ist zu beachten, dass die ursprünglichen GeoJSON Dateien, die vom Land Salzburg zur Verfügung gestellt werden, noch modifiziert werden müssen. Die angegebenen Koordinaten sind nicht als geographische Koordinaten, sondern im Referenzsystem MGI mit Gauß-Krüger Koordinaten dargestellt. Dadurch werden diese bei der Umwandlung in eine GeoJSON Datei nicht erkannt. Aus diesem Grund sollten die vom Land Salzburg zur Verfügung gestellten Shapefiles verwendet werden. Im QGIS können sie als GeoJSON exportiert werden, wodurch die Koordinaten als geographische Koordinaten wiedergegeben werden, die dann in die HTML-Skripte eingebunden werden können.

Zusammenfassend ist zu sagen, dass die erstellten HTML-Seiten einen guten Überblick über die Freizeitmöglichkeiten liefern und durch recht einfache Strukturen gebildet wurden. Dies hat den Vorteil, dass auch mit geringen Kenntnissen in HTML und CSS interaktive Karten erstellt werden können, die einen hohen Informationsgehalt aufweisen. Des Weiteren können die Skripte als Basis für weitere Modifizierungen gesehen werden, auf die immer wieder zurückgegriffen werden kann.

## 7 Verwendete Datensätze/Datengrundlagen

### **Kartengrundlage:**

Stadt Wien Basemap "<http://data.wien.gv.at/>",  
<http://www.basemap.at/wmts/1.0.0/WMTSCapabilities.xml>

### **Geodatensätze:**

*Touristische Radrouten Salzburg Land (JSON&Shapefile)*

<https://www.data.gv.at/katalog/dataset/328eea39-089a-46fb-a18c-fb40d83e6b16>

*Burgen und Schlösser Land Salzburg (JSON&Shapefile)*

<https://www.data.gv.at/katalog/dataset/7b5df951-b8b0-4153-b599-490b4fe39d8b>

*Naturparks Land Salzburg (JSON&Shapefile)*

<https://www.data.gv.at/katalog/dataset/f035e1ef-9b98-4d77-b2ad-1daf6013e6b3>

*Themenradrouten Stadt Salzburg (JSON)*

<https://www.data.gv.at/katalog/dataset/6b2d0c3c-10cc-4cc0-a1fd-c07ab1bff086>

*Spielplätze Stadt Salzburg (JSON)* <https://www.data.gv.at/katalog/dataset/a2aacf9c-c06b-49ac-b196-0eb22fbc6e84>

*„Salzburg Surft“ Wifi-Standorte (JSON)*

<https://www.data.gv.at/katalog/dataset/4b901b6bf131-49f7-ba8b-2e3b7bfd4ea6>

### **Bildmaterial und Icons:**

[pixabay.com](http://pixabay.com)

### **Literatur:**

*Burgen und Schlösser*

<http://www.burg-finstergruen.at/die-burg.html>

<http://www.salzburg-burgen.at/de/hohensalzburg/>

<https://www.salzburg.info/de/sehenswertes/festung-schloesser/schloss-leopoldskron>

<https://www.salzburg.info/de/sehenswertes/museen/volkskunde-museum>

<https://www.salzburg.info/de/sehenswertes/museen/volkskunde-museum>

<http://www.salzburg-burgen.at/de/mauterdorf/>

<http://www.salzburg-burgen.at/de/werfen/>

<https://www.salzburg.gv.at/kultur/Seiten/hochneukirchen.aspx>

<http://www.schlossmoosham.at/>

<http://www.schlossgoldegg.at/>

*Naturparks*

<http://www.groedig.net/untersberg-seilbahn>

<https://www.naturparke.at/naturparke/salzburg/naturpark-buchberg/>

<http://www.naturpark-riedingtal.at/>

<http://www.naturpark-weissbach.at/>

*Leaflet*

<http://leafletjs.com/2012/08/20/guest-post-markerclusterer-0-1-released.html>

<http://leafletjs.com/reference-1.0.3.html#marker>

<http://leafletjs.com/reference-1.0.3.html#control-layers>

*Wiki selfhtml*

<https://wiki.selfhtml.org/wiki/Schnell-Index/CSS>