

Luca Vinti

Richard Matalewski

Joel Robert

Projektbericht

1 Einleitung

Die Europameisterschaft 2020 (2021) wird den meisten Fußballfans in Erinnerung bleiben. Das liegt daran, dass das Turnier zum ersten Male, in vielen verschiedenen Ländern und Städten stattgefunden hat. Des Weiteren, handelte es sich hierbei, um das erste größere Turnier, welches während der Corona-Pandemie stattgefunden hat.

Insgesamt, wurden elf Spielorte, als Austragungsort der Spiele auserwählt. Dieser Tatbestand wird in der ersten Seite des Projekts dargestellt. Die Seite enthält eine Einleitungspassage und eine Leaflet-Karte, welche die verschiedenen Austragungsorte aufzeigt.

Vierundzwanzig Mannschaften haben sich um den Titel gestritten. Dieser Punkt ist Gegenstand der zweiten Projektseite. Hier, werden die verschiedenen Teilnehmer aufgelistet und anschließend, auf einer Leaflet-Karte visualisiert.

Europameister wurde schließlich Italien, gegen England im Wembley Stadion, London. Weitere Statistiken, werden auf der dritten Projektseite vorgestellt. Des Weiteren, präsentieren wir eine Übersichtskarte, die die verschiedenen Ränge der teilnehmenden Länder visualisiert.

Wir haben uns darum bemüht, möglichst einheitliche Designs für die verschiedenen Seiten zu verwenden. Die einzig wirklichen Unterschiede, sind auf Seiten der Banner zu vermerken. Es bleibt zu präzisieren, dass es Schade ist, dass die von der UEFA benutzten Logos und Designs, nicht verwendet werden durften. Wir haben ausschließlich, mit Open Source Daten gearbeitet, was das Stylen der Seiten erschwert hat.

2 Arbeitsschritte

2.1 Stadionübersicht

Um die verschiedenen Austragungsorte der Fußballeuropameisterschaft abbilden zu können, habe ich mich dazu entschlossen, eine Leaflet-Karte in das Projekt zu implementieren. Das Arbeiten mit Leaflet ermöglicht es mir, eine leicht zu bearbeitende Karte zu erstellen, welche als Hauptaugenmerkmal der Seite fungiert.

Der erste Schritt bestand daraus, eine HTML-Seite zu erstellen, die den Namen “index.html” trägt. Diese Seite wurde so genannt, da sie als Startseite des Projekts in Erscheinung tritt. Des Weiteren, wurden ein Java-File (stadien.js) und ein CSS-File (stadien.css) erstellt, die auf der einen Seite, zum Bearbeiten der Leaflet-Karte (Java) und auf der anderen Seite, zum Stylen der Seite benutzt werden.

Da, ich mir im Vorfeld bereits Gedanken darüber gemacht habe, wie meine Leaflet-Karte aussehen soll, konnte ich sofort damit beginnen, die benötigten Leaflet Plugins in die index.html (im Head-Bereich) zu implementieren (Leaflet Karte, Leaflet Providers, Leaflet Minimap, Leaflet Fullscreen).

Nachdem die Plugins eingefügt wurden, konnte ich mit dem weiteren Aufbau der Seite fortfahren. Zuerst wurden die üblichen Elemente der Seite implementiert (Body-, Main-, Header-, Article- und Footer-Bereich). Um die Karte starten zu können, wurde im Article-Bereich, dieser Code eingefügt: “`<div id="map"></div>`”. Nach diesem Schritt, konnte ich mit dem Bearbeiten der Karte (im stadien.js) beginnen.

```
let Prag = {lat: 50.088611, lng: 14.421389,};  
let startLayer = L.tileLayer.provider("Esri.WorldImagery");  
let overlays = {stadien: L.featureGroup({})  
let map = L.map("map", {center: [Prag.lat, Prag.lng], zoom: 4, layers: [startLayer],});
```

Als Center der Karte wurde die Stadt Prag definiert (zentrale Lage in Europa). Als Startlayer der Karte, sollte die “Esri World Imagery” fungieren. Der anfängliche Zoom der Karte, wurde nach mehreren Versuchen auf den Wert “4” bestimmt. Nun war der Moment gekommen, in dem die Karte gestartet werden konnte.

```
let layerControl = L.control.layers({"Orthofoto": startLayer, "World Street Map":
L.tileLayer.provider("Esri.WorldStreetMap"), "Topographische Karte":
L.tileLayer.provider("OpenTopoMap"), "Schwarz-Weiß Karte":
L.tileLayer.provider("Stamen.Toner"),}).addTo(map);

layerControl.expand();
```

Nachdem die Karte aufgerufen wurde, war es nun an der Zeit, weitere Layer hinzuzufügen. Durch das Leaflet Providers Plugin, war dieser Schritt mühelos zu bewerkstelligen. Auf der Internet-Seite <https://leaflet-extras.github.io/leaflet-providers/preview/>, kann man sich beliebige Kartenlayer aussuchen und in das Projekt einfügen. Dieser Schritt wurde in den vier ersten Zeilen vom Code dargestellt. Der Startlayer war bereits vordefiniert, die drei weiteren Layer stellen sich aus der Esri World Street Map, der Open Topo Map und der Stamen Toner Map zusammen. Die drei ersten Layer, zeigen geographische Gegebenheiten, rund um die Stadien auf. Der letztere Layer, verfolgt einen ästhetischen Hintergrund. Der letzte Schritt bestand daraus, die Layer zur Layer Control hinzuzufügen.

```
.control.fullscreen().addTo(map);

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

let miniMap = new L.Control.MiniMap(L.tileLayer.provider("Esri.WorldImagery"),
{toggleDisplay: true}).addTo(map);
```

Nachdem, die verschiedenen Layer implementiert wurden, wurden weitere Kartenelemente hinzugefügt. Zuerst, wurde das Leaflet Fullscreen Plugin erstellt. Anschließend, wurde die Karte mit einem Maßstab versehen. Schlussendlich, wurde das Minimap Plugin eingefügt.

```
"features": [{ "id": "1", "geometry": { "coordinates": [11.624372, 48.218807], "type":
"Point"}, "properties": { "Land": "Deutschland", "Stadt": "München", "Stadio_Name":
"Allianz Arena", "Spiele_Num": "4", "Kapazität": 75024}, "type": "Feature"}, ]
```

Nach tiefgründigen Recherchen ist mir aufgefallen, dass es keine passende GeoJson-Datei gab, die mir bei der Abbildung der Stadien behilflich sein könnte. Kurzerhand, habe ich mich dazu entschlossen, eine eigene, lokale Datei zu erstellen, die die für das Projekt relevante Daten beinhaltet. Im obenstehenden Code, ist ein Beispiel abgebildet, welches darlegt, wie ich in diesem Prozess vorgegangen bin. Die erstellte GeoJson hat mir anschließend ermöglicht, die Stadien leichter auf der Karte abbilden zu können.

```
async function loadStadien(url) {let response = await fetch(url);let geojson = await response.json();
```

```
let overlay = L.featureGroup();layerControl.addOverlay(overlay, "Stadien");overlay.addTo(map)
```

```
let stadionLayer = L.geoJSON(geojson, {pointToLayer: function (geoJsonPoint, latlng) {let popup = `<strong>${geoJsonPoint.properties.Stadio_Name}</strong> <br><br>${geoJsonPoint.properties.Land},    ${geoJsonPoint.properties.Stadt}    <br>Stadionkapazität: ${geoJsonPoint.properties.Kapazität} Zuschauer <br> Anzahl der Spiele:  ${geoJsonPoint.properties.Spiele_Num}` ;return L.marker(latlng, {icon: L.icon({iconUrl: `icons/stadium.png`,iconAnchor: [17, 36], })}).bindPopup(popup);}}).addTo(overlay);
```

Mittels einer Async Function war ich jetzt in der Lage, die lokalen GeoJson-Daten aufzurufen und in meiner Karte zu implementieren. Zuerst, wurden die die Daten an das Overlay angehängt. Anschließend, wurden das Popup und der Marker hinzugefügt. Das Popup beinhaltet Infos zu den verschiedenen Stadien, wie die Stadt und das Land, in dem es sich befindet, die Stadionkapazität und die Anzahl der abgehaltenen Spiele an dem Standort. Ich hatte zusätzlich die Idee, Fotos in das Popup einzufügen, allerdings habe ich mich damit schwergetan, Bilder ohne Copyright zu finden. Deswegen, habe ich diesen Schritt auf der Seite gelassen. Nachdem, das Popup fertiggestellt wurde, wurden die Marker mit den dazu passenden GeoJson-Koordinaten versehen. Des Weiteren, habe ich mich dazu entschlossen, andere Icons zur Abbildung der Stadien zu benutzen. Diese habe ich auf der Seite <https://mapicons.mapsmarker.com/> gefunden, diese dürfen ohne Einschränkungen weiterverwendet werden. Im einer letzten Etappe, wurden die Position der Icons, mittels dem "Icon Anchor Tool" von Leaflet verbessert.

Der letzte Schritt beim Bearbeiten der Karte bestand daraus, eine Suchleiste zu implementieren. Der Vorgang hat damit begonnen, die Suchleiste im index.html einzufügen und die verschiedenen Parameter (name, placeholder, list, usw.) zu benennen. Nachdem dies erfolgte, wurde in der Java-Datei damit fortgeführt, die Daten der Stadien mit der Suchleiste zu verbinden:

Schritt 1: Die Value (Stadionnamen) mit der Searchlist verbinden

```
searchList.innerHTML += `<option value= "${geoJsonPoint.properties.Stadio_Name}"></option`;
```

Schritt 2: Die Searchlist Funktion erstellen

```
let form = document.querySelector("#searchForm");
```

```
form.suchen.onclick = function () {
console.log(form.stadion.value);
stadionLayer.eachLayer(function (marker) {
```

Schritt 3: Das Zoomen auf das Popup einstellen (Zoom = 16)

```
if (form.stadion.value == marker.feature.properties.Stadio_Name) {
map.setView(marker.getLatLng(), 16);
marker.openPopup();
```

Somit, war das Arbeiten an der Karte abgeschlossen und ich konnte mich auf das Stylen meiner Seite konzentrieren. Auf der Seite <https://pixabay.com/>, habe ich mir zwei Bilder ausgesucht, die auf der einen Seite als Favicon und auf der anderen Seite, als Banner fungieren sollen. Auf der Seite <https://redketchup.io/image-resizer>, habe ich die Bilder so zugeschnitten, dass ihre Dimensionen auf die zutreffende Funktion angepasst sind. Nachdem, ich die Bilder in das index.html eingebunden habe, wurden sie in der CSS-Datei (vor allem das Banner) bearbeitet, um auf die Ausrichtung der Seite zu passen.

```
body {background-color: rgb(154, 154, 154); font-family: 'Open Sans', sans-serif;
main {background-color: white; max-width: 1024px; margin: auto;
```

Die Hintergrundfarben der HTML-Seite, wurden ebenfalls in der CSS-Datei bearbeitet. Das Body-Element erscheint in einem Grauton, das Main-Element in Weiß. Zusätzlich, wurde ein Shadow hinzugefügt, der es so aussehen lässt, als würde das Main-Element, oberhalb des Body-Elements schweben.

```
@import url('https://fonts.googleapis.com/css2?family=Montserrat:wght@300&family=Open
+Sans:ital@0;1&family=Roboto:wght@300&display=swap');
```

Um das Stylen der Schriftelemente zu vereinfachen, wurde die Google Font Erweiterung eingeladen. Anschließend, habe ich mich darangesetzt, die Überschriften, die Textpassagen und die Datenquellen zu verfassen. Schlussendlich, wurden die Textelemente in der CSS-Datei gestyled. Die Icons, die neben den Stadionnamen stehen, habe ich von meinem Projektpartner Joel bezogen.

```

<a                                     class="back"
href="https://euro2021projekt.github.io/Statistik.html"><i class="fa-solid fa-
circle-arrow-left"></i> Statistiken </a>

<a      class="next"      href="https://euro2021projekt.github.io/mannschaft.html">
Mannschaften <i class="fa-solid fa-circle-arrow-right"></i></a>

```

Das letzte Element meiner Arbeit bestand daraus, die Navigation zu den anderen Projektseiten herzustellen. Hierfür, wurde ein Nav-Element im index.html aufgerufen. Die Links zu den anderen Seiten, wurden mit den passenden Adressen ausgefüllt und anschließend, mit einer Icon versehen. Diese Icon, stammt aus der Font Awesome Bibliothek, die ein Plugin für HTML bereitstellt. Bei Abschluss dieses Schrittes, war meine Seite fertig und ich habe noch ein letztes Mal, die Dateien mit einem Beautify verschönert.

2.2 Statistikübersicht

Website zur Darstellung statistischer Daten der Europameisterschaft 2020

Diese Website soll nochmals einige interessante Fakten und Statistiken zu der EURO 2020 auszeigen und durch eine interaktive Karte untermauern. Wie schon erwähnt besteht die Website aus zwei Teilen. Der erste Teil ist eine Zusammenfassung nennenswerter Statistiken und der zweite Teil besteht aus einer Karte zu den jeweiligen Platzierungen der Mannschaften im Turnier.

Aufbau der Website

Als erster Schritt in der Erstellung der Website, wurde das html Grundgerüstes, wie der Titel der Website, die Struktur (header, body, footer etc.) in die Statistik.html geschrieben. Anschließend wurde als nächster Schritt ein Tabvorschau bild entworfen, dazu wurde eine Verlinkung zu dem „icons“ Ordner erstellt, wo sich alle Abbildungen und Icons der Website befinden. Anschließend kommen einige notwendige, aber fundamentale Arbeitsschritte, wie das Stylesheet verlinken, die Leaflet Bibliothek einbinden und die font awesome einbauen. Zudem wurden gleich auch noch Vorbereitungen für die Kartenapplikation der Website getroffen. Hierzu gehören, das fullscreen plugin und die Leaflet MiniMap. Diese werden mit Verlinkungen in das html Skript eingebaut, damit sie anschließend durch die Befehle im Javaskript abrufbar sind. Danach folgt der erste ersichtliche Schritt auf der Website, nämlich das Hinzufügen des Banners. Dieser wurde auch mit einem Verweis auf den „icons“ Ordner eingebaut. Das Bild von dem leeren Stadium wurde von pixabay.com heruntergeladen und ist copyright free und somit frei verwendbar. Das Bild wurde vorher noch auf die gebrauchten Dimensionen mit Hilfe des Webtools redketchup.io gebracht.

Danach wurde der Titel der Website eingebaut jeweils mit den Befehlen `<h></h>` und in drei Abstufungen unterteilt. Die Titel wurde nochmals in h1, h2 und h3 gegliedert, was auch ihrer jeweiligen Hierarchie entspricht, somit wird h1 am größten geschrieben und h3 am kleinsten.

Darauf folgten nun die ersten Statistiken in Form der Rangliste, der Platzierungen der Mannschaften. Diese, sowie auch alle anderen Daten von den Statistiken wurden von der offiziellen UEFA Website abgerufen und sind somit für jeden öffentlich zugänglich. Um diese Statistiken anschaulicher darzustellen und auch unterschiedlich zu designen, wurden den Paragraphen (`<p></p>`) unterschiedliche id's zugeteilt (Abb. 1). Dies hat den Vorteil, dass diese im Stylesheet anschließend individuell bearbeitet werden können. Somit konnte für Italien, die Farbe Gold und für England die Farbe Silber eingestellt werden, um nochmals den Sieger und Zweitplatzierten hervorzuheben.

```
<h1><i class="fa-solid fa-ranking-star"></i> Rangliste</h1>
<p id="one">1. Italien</p>
<p id="two">2. England</p>
<p id="three">Gruppensieger: Belgien, Italien, Niederlande, England, Schweden, Frankreich </p>
<p id="three">Gruppensiebzweite: Wales, Dänemark, Österreich, Kroatien, Spanien, Deutschland.</p>
<p id="three">Gruppensiebdritte: Portugal, Tschechische Republik, Schweiz, Ukraine.</p>
<p id="three">Gruppensiebvierthe: Türkei, Schottland, Nordmezedonien, Ungarn</p>
<p id="three">(Quelle: <a
  href="https://www.uefa.com/uefaeuro/history/seasons/2020/statistics/">UEFA.com</a></p>

<h1>UEFA EURO 2021 Statistiken</h1>
```

Abbildung 1: Unterteilung der Paragraphen in verschiedene id's

Um die Statistiken in den Abschnitten „Übersicht“ und „Tore“ optisch ansprechender zu platzieren, wurde mit dem Befehl „splitt-para“ und einem „``“ die beiden Paragraphen nebeneinander platziert (Abb. 2). Zudem wurden noch fontawesome icons vor die Überschriften „Übersicht“ und „Tore“ gesetzt.

```
<p class="split-para"> <i class="fa-solid fa-ranking-star"></i> Übersicht <span><i class="fa-solid fa-futbol"></i> Tore</span><
<p class="split-para">51 Spiele gespielt <span>142 Tore insgesamt</span></p>
<p class="split-para">24 Teams haben teilgenommen <span>2,79 Tore pro Spiel</span></p>
<p class="split-para">55 mit den Qualifizierungsrunden <span>32 Tore pro Minute</span></p>
<p class="split-para">(Quelle: <a
  href="https://www.uefa.com/uefaeuro/history/seasons/2020/statistics/">UEFA.com</a>) <span>(Quelle: <a
  href="https://www.uefa.com/uefaeuro/history/seasons/2020/statistics/">UEFA.com</a></span></p>
```

Abbildung 2: Paragraf unterteilt in links und rechts

Die Abstände zwischen den Paragraphen, konnte anschließend im statistik.css eingestellt werden. Die Übersicht enthält Statistiken zu dem wichtigsten Turnier gesamt, mit gespielten Spielen, teilgenommenen Team und Anzahl der Qualifizierungsrunden. Der Paragraf „Tore“ zeigt die Tore insgesamt, pro Spiel und pro Minute. Anknüpfend an die Torzahl des Turnieres wurde noch eine

passende Tabelle mit den jeweiligen Toren und in welchen Spielminuten sie gefallen sind erstellt. Dafür wurden auch die Daten von der UEFA genommen und mit Excel ein Histogramm erstellt. Dieses Histogramm wurde zum Gesamtbild der Website blau abgesetzt und dann anschließend als .png implementiert. Um das Bild der Tabelle richtig zu zentrieren, wurde im Stylesheet, die „left & right margin“ auf „auto“ gesetzt und die „width“ auf 50% eingestellt. Mit diesem Element ist der erste Teil der Website abgeschlossen, da reine Auflistungen und Tabellen mit Statistiken nicht sehr ansprechend sind, wurde sich dazu entschlossen noch eine kleine Karte hinzuzufügen. Diese soll optisch ansprechender nochmals die Rangfolge der Mannschaften im Turnier darstellen. Die Karte wird zwar im html code implementiert, aber erstellt wurde sie im javascript in der statistik.js Datei. Als erstes wurde dort von der Leafletbibliothek mit dem Befehl „var map = L.map('map')“ eine Karte erzeugt, welche die Daten den Openstreetmap als Grundlage verwendet. Andere Layer in die Karte einzubauen, erschien nicht sinnvoll. Der Standard Openstreetmaplayer in den seinem schlichten grau, passt bestens um anschließend Marker deutlich abzubilden. Da es sich um eine Europameisterschaft handelt, sind natürlich nur Länder innerhalb von Europa vertreten und somit wurde die Karte als Standardzentrierung auf das geographische Europa ausgerichtet. Zusätzlich wurde noch der Zoom von vier als Standard festgelegt. Um die Rangliste der Mannschaften in der Karte sinnvoll und nachvollziehbar darzustellen wurden sich für die Implementierung von Circlemarkern entschieden. Diese wurden dann jeweils auf die Koordinaten der Hauptstädte der Länder, der teilnehmenden Mannschaften gesetzt. Die Circlemarker wurden in verschiedene Radien gegliedert, korrelierend mit der Platzierung des Landes. Die Abstufung wurde folgend gewählt; Radius 50 = Sieger, 40 = Zweiter, 30 = Gruppensieger, 20 = Gruppenzweiter, 10 = Gruppendritter und 5 = Gruppenvierter (Abb. 3). Zudem wurde für den ersten Platz ein goldener und für den zweiten Platz ein silberner Circlemarker speziell hinzugefügt.

```
L.circleMarker([41.9028, 12.4964],{
  radius: 50,
  color: "gold"
}).addTo(map)
  .bindPopup("<h1>Italien</h1> <h1>1. Platz</h1>");

L.circleMarker([40.4168, -3.7038],{
  radius: 20,
  color: "blue"
}).addTo(map)
  .bindPopup("<h1>Spanien</h1> <h1>Gruppenzweite</h1>");
```

Abbildung 3: Erstellung der Circlemarker

Außerdem wurden die Marker mit einem Popup versehen, welches nochmals das jeweilige Land und dessen Platzierung angibt.


```

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

L.control.fullscreen().addTo(map);

let miniMap = new L.Control.MiniMap(
  L.tileLayer.provider("OpenStreetMap.Mapnik"), {
    toggleDisplay: true
  }
).addTo(map);

var north = L.control({position: "bottomright"});
north.onAdd = function(map) {
  var div = L.DomUtil.create("div", "info legend");
  div.innerHTML = '';
  return div;
}
north.addTo(map);

```

Abbildung 4: Auflistung der benutzten Plugins

Wie in Abbildung 4 ersichtlich, wurden der Karte auch noch verschiedene Leaflet Plugins hinzugefügt. Einmal den Maßstab in Kilometern eingestellt. Als zweites das Fullscreen Plugin, welches die Karte mittels eines Klicks, in Fullscreenmode anzeigen lässt. Als drittes die Leaflet MiniMap, welchen auf der Weltkarte mittels eines roten Rechtecks die genaue Position des reingezoomten Kartenausschnittes darstellt. Als letztes Plugin kommt noch die Implementation eines Nordpfeiles dazu. Der Nordpfeil wird über den entsprechenden Code und einen Verweis in den „icons“ Ordner zum passenden .png erstellt.

Unterhalb der Karte befindet sich noch eine kurze Linksammlung mit drei der wichtigsten Links zu den Statistiken der EURO 2020. Im footer der Website angekommen sind jeweils noch die anderen zwei Websites verlinkt. So kann man immer von allen Websites beliebig hin und her wechseln. Abschließend sollte nochmals die Wichtigkeit des Stylesheets für das Aussehen der gesamten Website erwähnt werden. Ohne die Implementation des Stylesheets, würde nichts geordnet sein. Es wurden sich zu allen Platzierungen des Textes und der Elemente auf der Website Gedanken gemacht und diese anschließend in der statistik.css umgesetzt (Abb. 5 & 6).

```

img.table {
  border: 1px solid gray;
  display: block;
  margin-left: auto;
  margin-right: auto;
  margin-top: 40px;
  width: 50%;
}

```

```

#one {
  text-align: center;
  color: goldenrod;
  font-size: 30px;
}

```

Abbildung 5: Stylesheet Tabelle

Abbildung 6: Stylesheet Italien

2.3 Mannschaftsübersicht

In diesem Teil geht es um die Erstellung der Seite, welche die unterschiedlichen Mannschaften der Fußball Europameisterschaft 2021 vorstellen soll. Die Webseite soll dabei die Länder, welche sich qualifiziert haben und an der Europameisterschaft teilgenommen haben auflisten und auf einer Karte anzeigen. Zum Kartieren der unterschiedlichen Länder wird Leaflet verwendet. Mit weiteren Plugins soll die Karte dann interaktiver gestaltet werden. Welche Plugins wozu genau benutzt werden folgt im nächsten Kapitel „Arbeitsschritte“.

Grundlage hierfür ist ein HTML-file, zwei JS-files und ein CSS-file.

Zur Umsetzung dieser Seite wurden zuerst die vier Files erstellt. Zur Erstellung der HTML-Seite wurde das Neuseeland Projekt als Vorlage genutzt und für eigene Zwecke angepasst und erweitert. Die HTML-Seite trägt den Namen Mannschaft.html, das dazugehörige Stylesheet den Namen Mannschaft.css und das Javascript Mannschaft.js. Zu diesen drei kommt noch ein weiteres Javascript in dem die Informationen aller Mannschaften eingetragen sind. Hierzu hätte auch eine GeoJson-Datei erstellt werden können, jedoch wurde dies hier mit Javascript durchgeführt.

```
<!DOCTYPE html>
<html lang="de">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Mannschaften der EURO 2021</title>

  <!-- Ball als favicon -->

  <link rel="shortcut icon" href="icons/logo.png" type="image/png">

  <!-- eigene Stile und Hauptskript sowie Länder -->
  <link rel="stylesheet" href="mannschaft.css">
  <script defer src="mannschaft.js"></script>
  <script src="länder.js"></script>
```

Auf der HTML-Seite wurde dann der Titel, welcher im Tab zu sehen ist, sowie ein Ball-icon eingefügt. Das Hauptskript, Stylesheet und das Skript mit den Ländern wurden dann im HTML verlinkt.

Darunter auf der HTML- Seite kommen dann die benötigten Links für die Plugins und für die Leaflet Karte. Auf der erstellten Karte wurden fünf Plugins verwendet, diese sind: Das Leaflet providers Plugin, welches dazu dient, verschiedene Karten Layer auszuwählen. Das zweite Plugin, welches benutzt wurde, ist das Fullscreen Plugin, welches die Funktion hat,

die Karte im Fullscreen Modus anzuzeigen. Das dritte Plugin für die Mannschaftskarte ist das Minimap Plugin, welches dazu dient, eine kleine Karte als Übersichtskarte auf der Hauptkarte darzustellen. Des Weiteren wurde noch ein Plugin verwendet damit die Karte über einen Maßstab verfügt. Letztlich soll dann noch das Smooth Marker Bouncing Plugin dafür sorgen, dass die Karte bei einem Klick auf ein Marker anschaulicher und interaktiver wirkt.

```
<body>
  <main class="shadow">
    <header>
      
    </header>
    <article>
      <h1>Euro 2021</h1>
      <h2><em>Mannschaften</em></h2>
      <p>
        Die folgenden Mannschaften haben sich über die Qualifikation für das Endrunden-Turnier qualifiziert:
      </p>
    </article>
  </main>
</body>
```

Vor dem Text auf der HTML Seite wurde das Bild von dem Fußballstadium als Bannerbild eingefügt. Dieses stammt von Pixabay und wurde unten in den Quellenangaben verlinkt. Unter dem Bild folgen dann zwei Überschriften welche mit <h1> und <h2> im Stylesheet gestylt wurden. Bevor die einzelnen Länder der Euro 2021 aufgelistet werden, kommt ein einleitender Satz.

```
<ul style="list-style: none;">
  <li> Belgien </li>
  <li> Italien</li>
  <li> Polen</li>
  <li> Russland</li>
  <li> Ukraine</li>
  <li> Spanien</li>
  <li> England</li>
  <li> Frankreich</li>
  <li> Tschechien</li>
  <li> Türkei</li>
  <li> Finnland</li>
  <li> Schweden</li>
  <li> Deutschland
</li>
  <li> Kroatien</li>
  <li> Niederlande
</li>
  <li> Österreich</li>
  <li> Portugal</li>
  <li> Dänemark</li>
  <li> Schweiz</li>
  <li> Wales</li>
  <li> Nordmazedonien
</li>
  <li> Ungarn</li>
  <li> Slowakei</li>
  <li> Schottland</li>
</ul>
```

Die Länder wurden in einer ungeordneten Liste aufgelistet, hierbei war das Ziel, dass die Liste keine Punkte vor jedem Element hat, sondern die Flagge der jeweiligen Ländern, so würde die Liste vom Aussehen her zur Karte passen. Mit dem Befehl: style="list-style: none;" wurden die Punkte entfernt und die Flaggen mussten dann nur noch vor den Namen verlinkt werden. Das Flaggen Icon wurde dann im Stylesheet auf die richtige Größe angepasst. Um die Liste außerdem in zwei Säulen darzustellen wurde im Stylesheet wie folgt angepasst:

```
ul {
  columns: 2;
  -webkit-columns: 2;
  -moz-columns: 2;
}
```

Unter der Auflistung folgt dann die Karte und anschließend noch die Quellen sowie zwei Navigationspfeile, mit denen man auf die Stadien Übersicht und Statistikseite gelangt.

```
JS länder.js > [0] LÄNDER
1  const LÄNDER = [{
2    "name": "Belgien",
3    "info1": "Trainer: Roberto Martinez ",
4    "info2": "EM-Teilnahmen: 6",
5    "image": "icons/länder/land1.png",
6    "lat": 50.843333,
7    "lng": 4.363056
8  }, {
```

Im Länder Javascript sind die Informationen aller Länder aufgelistet, in diesen Informationen sind enthalten: die Koordinaten der Hauptstädte dieser Länder um als Marker auf der Karte eingetragen zu werden, sowie ein paar Fakten über die Mannschaften (zum Beispiel wer der Trainer der Mannschaft ist und wie oft diese Länder schon bei einer EM teilgenommen haben). Zusätzlich wurde noch die Landesflagge verlinkt, welche nicht nur im Popup angezeigt werden soll, sondern auch als Marker der jeweiligen Länder auf der Karte dienen soll.

```
JS mannschaft.js > ...
1  let zoom = 4;
2
3  // Kordinate: Zentrum von Europa
4  let coords = [47.751569, 1.675063];
5
6  let startLayer = L.tileLayer.provider("OpenStreetMap.Mapnik");
7
8  let map = L.map('map', {
9    center: coords,
10   zoom: zoom,
11   layers: [
12     startLayer
13   ],
14 });
15
16 let layerControl = L.control.layers({
17   "OpenStreetMap": startLayer,
18   "Open Topo Map": L.tileLayer.provider("OpenTopoMap"),
19   "Esri Satellitenbild": L.tileLayer.provider("Esri.WorldImagery"),
20
21 }).addTo(map);
22
23
```

Im Hauptskript wurden zuerst Variablen für den Zoomfaktor, den Koordinaten vom Zentrum von Europa sowie einem Start-layer definiert. Danach wurde die Karte mit den vorherigen Variablen als Parameter implementiert. Zu dieser Karte wurden dann das Layer Control Plugin hinzugefügt, welches erlaubt zwischen verschiedenen Karten-layer zu wechseln. Mit `.addTo(map);` wurde dieses Plugin dann der Karte hinzugefügt.

```
// Länder anzeigen
for (let länder of LÄNDER) {
  let popup = `

  <h1>${länder.name}</h1>
  
  <hr>
  <h3>${länder.info2}</h3>
  <h3>${länder.info1}</h3>
  `;

  L.marker([länder.lat, länder.lng], {
    icon: L.icon({
      iconUrl: länder.image,
      iconSize: [30, 30],
      popupAnchor: [0, -10]
    })
  }).addTo(map).bindPopup(popup).on('click', function() {
    this.bounce(1); // bounce 1 mal
  });
}
```

Mit Hilfe einer for-Schleife wurden die einzelnen Länder aus dem Länder-Javascript auf der Karte angezeigt, diese wurden mit einem Marker und einem Popup versehen. Die Marker sind in diesem Fall die Flaggen und das Popup enthält die Informationen der einzelnen Länder.

Zur Animation der Marker wurde das Smooth Marker Bouncing Plugin verwendet, dieses erzeugt ein Hüpfen der Marker, wenn draufgeklickt wird.

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

L.control.fullscreen().addTo(map);

let miniMap = new L.Control.MiniMap(
  L.tileLayer.provider("OpenStreetMap.Mapnik"), {
    toggleDisplay: true,
    minimized: true
  }
).addTo(map);
```

Zum Schluss wurden dann noch das scale Plugin sowie die Minimap eingefügt. Die minimap ist standardmäßig eingeklappt und öffnet sich erst beim Draufklicken.

4. Verwendete Daten und Plugins:

Flag icons: <https://www.freeflagicons.com/> [Zugriff 28.06.2022].

Font awesome 2022: <https://fontawesome.com/icons> [Zugriff 28.06.2022].

Fussball-Wm.pro: <https://www.fussball-wm.pro/em-2021/mannschaften/> [Zugriff 28.06.2022].

Leaflet.fullscreen: <https://github.com/brunob/leaflet.fullscreen>
<https://github.com/brunob/leaflet.fullscreen> [Zugriff 28.06.2022].

Leaflet "map": <https://leafletjs.com/examples/quick-start/> [Zugriff 28.06.2022].

Leaflet „Scale“: <https://stackoverflow.com/questions/40201113/how-to-display-the-map-scale-in-leaflet-js> [Zugriff 28.06.2022].

Leaflet „Smooth Marker Bouncing“:
<https://github.com/hosuaby/Leaflet.SmoothMarkerBouncing>

Leaflet.WorldMiniMap: <https://github.com/maneoverland/leaflet.WorldMiniMap> [Zugriff 28.06.2022].

Pixabay 2022: <https://pixabay.com/de/photos/architektur-trib%C3%BCnen-stadion-leer-1853520/> [Zugriff 28.06.2022].

Pixabay 2022: <https://pixabay.com/de/photos/stadion-football-budapest-uefa-6352107/> [Zugriff 28.06.2022].

UEFA 2022: <https://www.uefa.com/uefaeuro/history/seasons/2020/statistics/> [Zugriff 28.06.2022].

Wikipedia: https://de.wikipedia.org/wiki/Fu%C3%9Fball-Europameisterschaft_2021#Teilnehmer [Zugriff 28.06.2022].