TschufTschuf Handbuch

Hofer Alex, Siller Lukas

11.02.2022

Inhalt

[Aufgabenstellung 2](#_Toc95495537)

[Pflichtkriterien 2](#_Toc95495538)

[Team-Management 2](#_Toc95495539)

[Einsatz/Zielgruppe 2](#_Toc95495540)

[Frontend 2](#_Toc95495541)

[Backend 3](#_Toc95495542)

# Aufgabenstellung

Die Aufgabenstellung ist es, eine Webseite in der Programmiersprache JavaScript zusammen mit der textbasierte Auszeichnungssprache HTML und der Stylesheet-Sprache CSS zu programmieren. Diese soll es ermöglichen jede Busfahrt und Zugfahrt mit bestimmten Start, Ziel, Datum und Zeit zu finden.

## Pflichtkriterien

* Freie Eingabe eines Startpunktes und Zielpunktes sowie Datum und Zeit
* Es soll mit einer anderen API z.B.: Wetter-API verknüpft werden

# Team-Management

|  |  |  |
| --- | --- | --- |
| Aufteilung | | |
| Hofer Alex | sthofale@bx.fallmerayer.it | Frontend |
| Siller Lukas | stsilluk@bx.fallmerayer.it | Backend |

# Einsatz/Zielgruppe

Das Busfahrplanprojekt ist für den Privaten Gebrauch aber auch für Reisende gedenkt. Das Programm sollte beispielsweise von Schülern, Benutzer des Öffentlichen Verkehres in Südtirol und Reisenden benutzt werden.

# Frontend

Die Webseite wurde mithilfe eines Mockups(Abbildung 1) geplant.  
Es gibt zwei HTML Dokumente, eines für die Hauptseite, index.html, und das andere für das Impressum, impressum.html.  
Für den Style gibt es die CSS-Datei, style.css.   
Für die Änderungen des HTMLs gibt es die JavaScript Datei, htmlChanges.js

Die Webseite wurde mithilfe eines Grids aufgebaut, sie besteht aus den verschiedenen Containern:  
StartContainer, ChangeContainer, StopContainer, WetterContainer, AusgabeContainer.

Es wird mit Bootstrap gearbeitet.

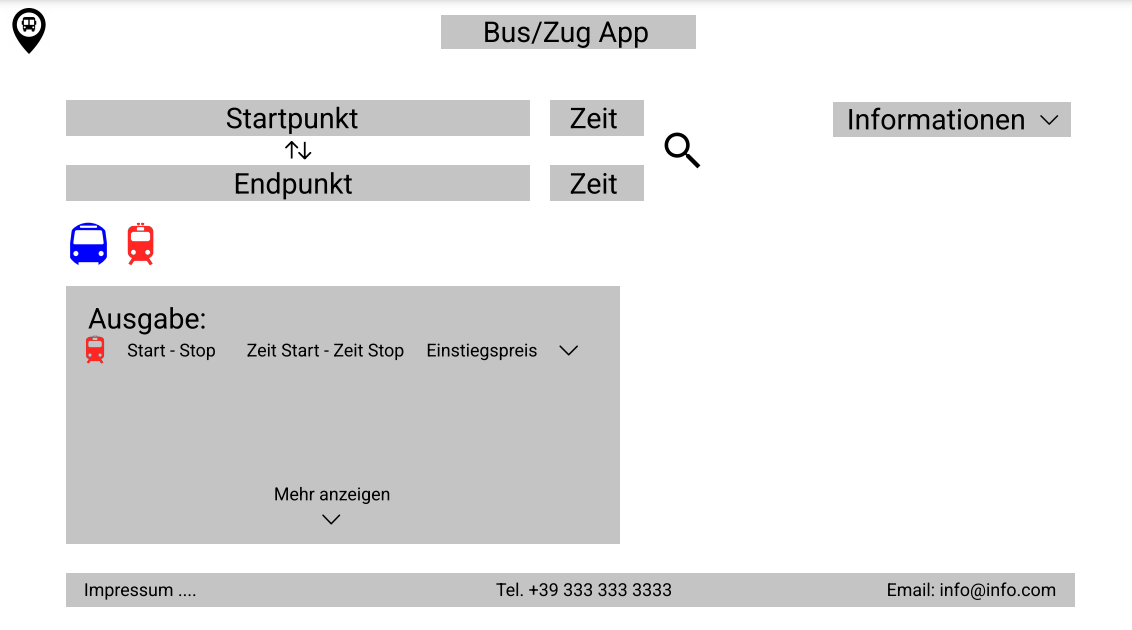
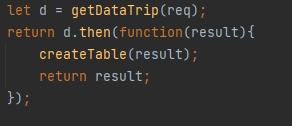


Abbildung : Mockup

# Backend

Im Hintergrund liegen 2 Dateien die sich mit den API abfragen beschäftigen. Die „verbindungReq.js“ beschäftigt sich mit der Abfrage für die Start und Endpunkte und alles was damit zusammenhängt. Dafür benutzt man die „fetch()“ Funktion die dafür verantwortlich ist, dass die Abfrage parallel lauft. Die erhaltenen Daten werden dann zurückgegeben und vom Frontend verarbeitet. Um die Daten von der aufgerufenen Funktion zu erhalten muss man mit .then auf die Rückgabewerte zugreifen (So wie im Bild 1) da man sonst bereits auf die daten zugreifen würde bevor man sie überhaupt bekommen hat.



*Bild 1*

Um den Richtigen Link für den Request zu erstellen bekommt die Datei auch Eingabe Daten und erstellt daraus dann den benötigten Trip link.

Bei der zweiten Datei „wetterReq.js“ wird dasselbe gemacht nur dass die daten für das Wetter ausgelesen werden.

Anfrage für Wetter daten:

