

Webprojektarbeit – HTML Spickzettel

1 Persönliche Angaben

Matr.-Nr.: 2240332

Git Repository: <https://github.com/Kgavaga/Web-Engineering>

2 Bearbeitete Aufgaben

- **Aufgabe 11:** Über ein HTML Canvas kann man mit dem Trackpad (Finger) oder Touchscreen unterschreiben (siehe 4.Hinweis zu Aufgabe 11)
- **Aufgabe 14:** Benutzung von VueJs
- **Aufgabe 17:** Installation der Komponenten läuft über Docker

3 Installationsbeschreibung

1. Github-Repository klonen

```
git clone https://github.com/Kgavaga/Web-Engineering
```

2. In den geklonten Ordner und dann in web-project navigieren

```
cd Web-Engineering/web-project
```

3. Docker **builden** und **starten**

```
docker compose up
```

4. Website aufrufen

<http://localhost:6001>

Der Docker Container heißt dann: web-project.

3.1 Falls Schritt 3 nicht funktionieren sollte

1. ...
2. ...
3. Docker **builden**

```
docker build -t web-project .
```

4. Docker **starten**

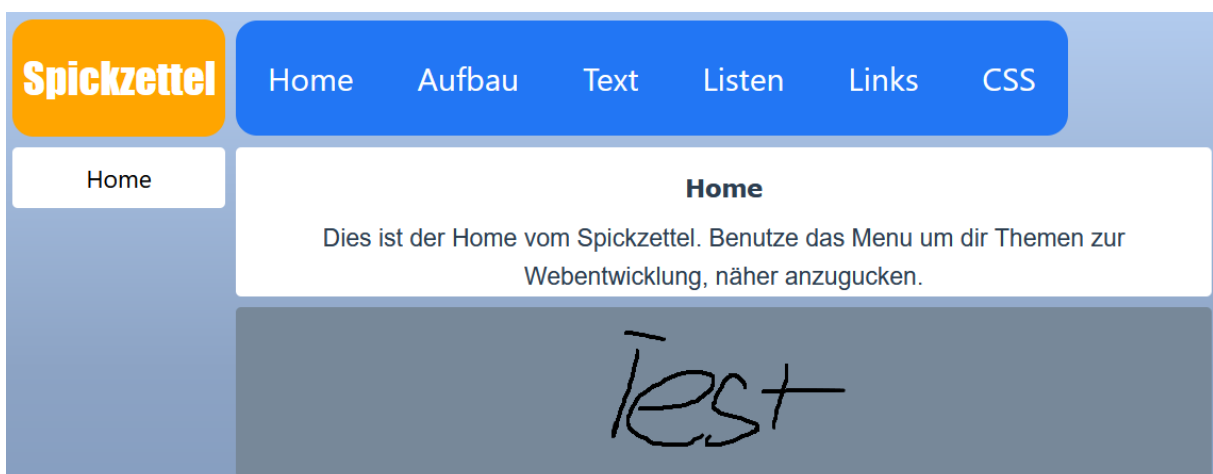
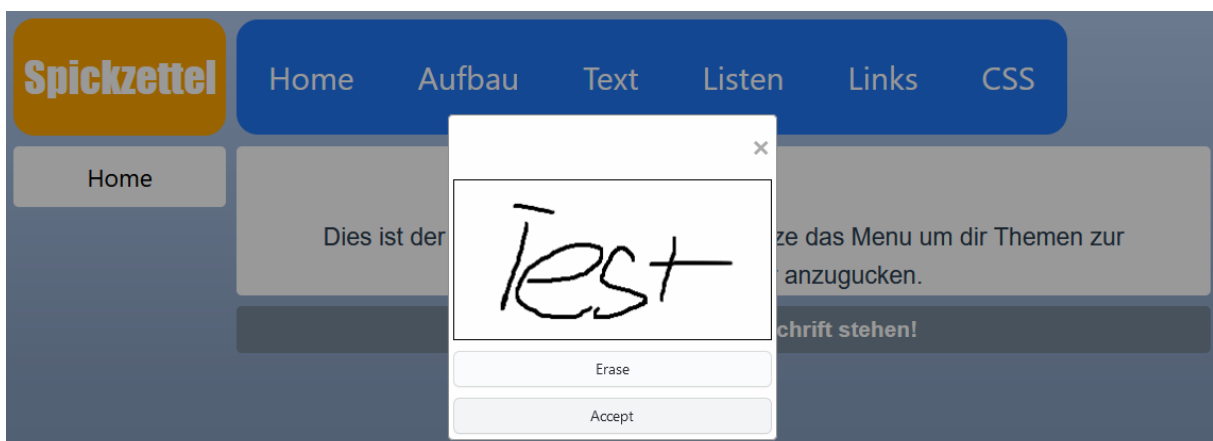
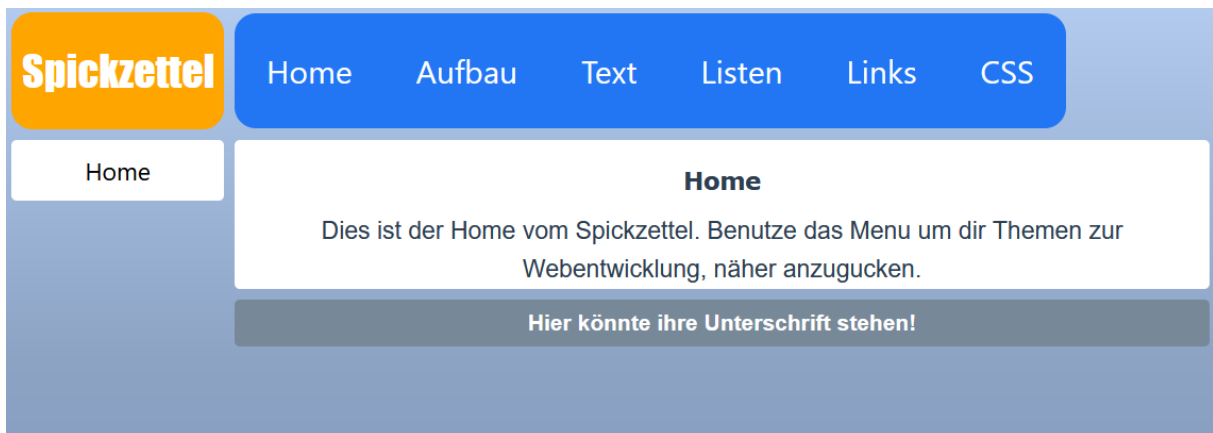
```
docker run -it -p 6001:8080 web-project
```

5. Website aufrufen

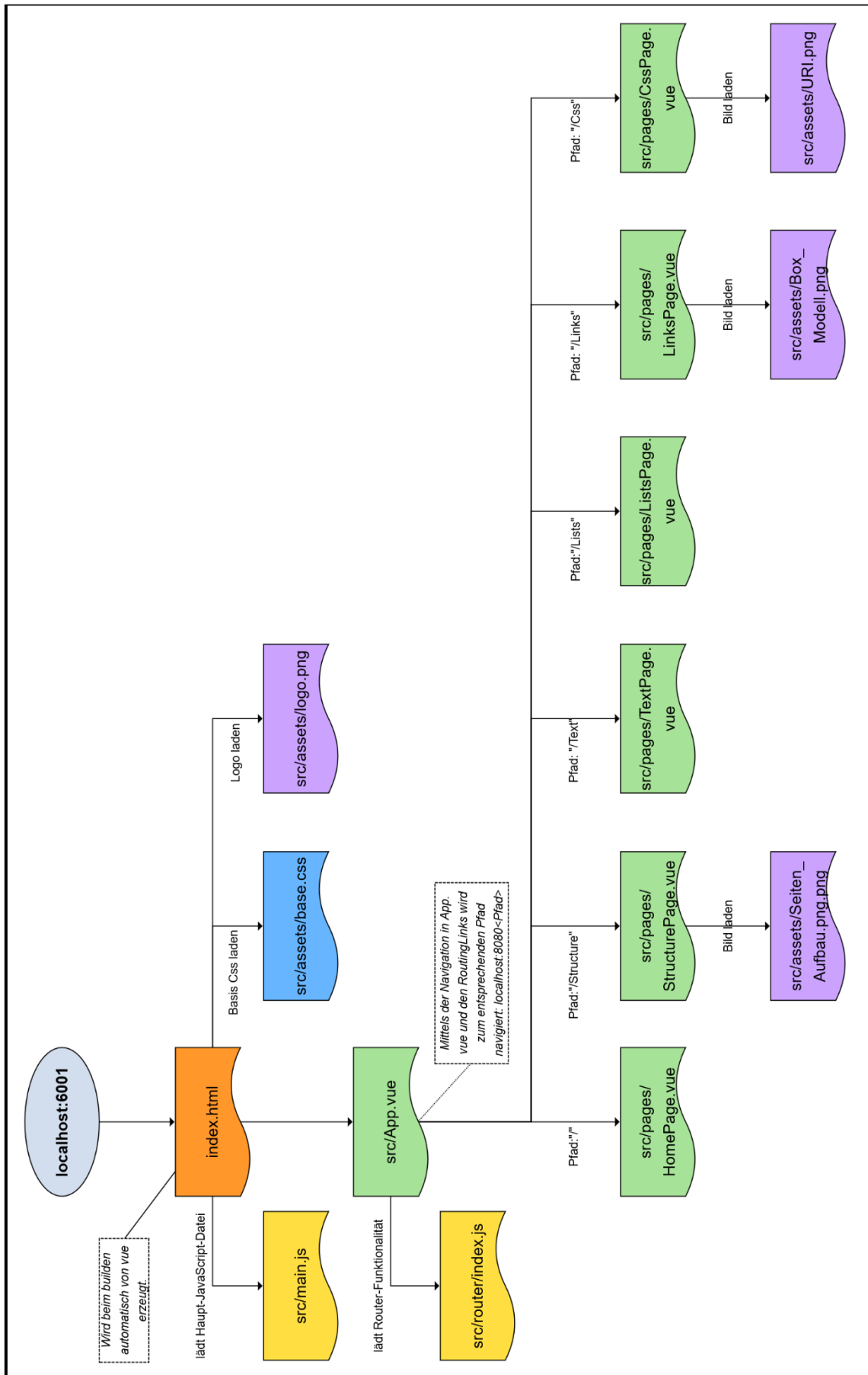
<http://localhost:6001>

4 Hinweis zu Aufgabe 11

Aufgabe 11 wurde so gelöst, dass man auf den Footer mit der Aufschrift „Hier könnte ihre Unterschrift stehen!“ klickt. Dabei öffnet sich ein Modal, wo man sofort loslegen kann und auf einer Fläche von 300x150 Pixel seine Unterschrift zeichnen kann. Dies geht sowohl für Desktop und Touch-Devices.



5 Datenflussdiagramm



6 Erfahrungsbericht

6.1 Was gut gelaufen ist

- Die Aufgabe 11 war relativ einfach zu lösen, da der Weg mit den Listenern sehr „straight-forward“ war und generell es JavaScript mit der DOM-Manipulation sehr einfach macht.
- Das Popup (Modal) programmieren, ging ziemlich gut, da aus Vorkenntnissen von der Arbeit im Betrieb zurückgegriffen werden konnte.
- Das Aufsetzen von Docker war etwas langwierig, hatte aber trotzdem gut funktioniert und brachte einiges an Spaß einher.
- Trotz Schwierigkeiten mit CSS konnte ich meine Vorstellungen zum Responsive Web-Design gut umsetzen und war zufrieden mit der Darstellung auf meinem Smartphone.

6.2 Was schlecht gelaufen ist

- Obwohl es gegen Ende hin gut lief, war CSS größtenteils ein sehr zeitwendiger und mühsamer Aufwand. Die vom vue-cli zur Verfügung gestellte base.css war dabei nicht sonderlich hilfreich, da sie zu Anfang eher verwirrend war.
- Erst sehr spät hatte ich gemerkt, dass ich Vite verwendete um mein Projekt zu builden und musste auf den Vue-cli erst noch umsteigen.
- Das Routing hatte Probleme bereitet, als ich versuchte ganz normal Anchors zu verwenden.

6.3 Fazit des Projektes

Zu Web-Engineering hatte ich schon seit jeher ein eher gespaltenes Verhältnis zu. Auf der einen Seite finde ich es überfordernd, wie viele verschiedene Möglichkeiten die Webprogrammierung einem gibt, Probleme zu lösen. Auf der anderen Seite ist es jedoch sehr interaktiv. So konnte ich super viel im Browser inspizieren und debuggen und Änderungen blitz-schnell anwenden und testen.

Durch dieses Projekt habe ich gelernt, den Debugger von Firefox sehr viel und gerne zu verwenden. Probleme mit CSS waren nicht mehr ein großes Fragezeichen, sondern konnten sehr schnell, interaktiv getestet werden.

Auch konnte ich meine Kenntnisse von AngularJs mit VueJs vergleichen und habe ein etwas besseres Verständnis bekommen, wie sie sich unterscheiden und wie VueJs genauer funktioniert.

Besonders habe ich endlich einmal Docker näher kennengelernt und gesehen was für nützliche Vorteile es mit sich bringt.