* Frontend

Unser Frontend ist prinzipiell in drei Teile aufgeteilt. Die Hauptseite für die Kunden, das Baukastensystem und das Dashboard für das Resteraunt. Dabei teilen sich erstere zwei die gleichen Technologien, während für das Dashboard teilweise unterschiedliche Technologien verwendet werden. In folgendem Abschnitt sind die verwendeten Technologien sowie deren Einsatzbereiche zu sehen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bereich | Materialize | Materialize Stepper | Halfmoon | JQuery |
| Hauptseite & Baukastensystem | ✓ | ✓ |  | ✓ |
| Dashboard |  |  | ✓ | ✓ |

* + Materialize

Materialize ist ein Frontend-Framework basierend auf der von Google entwickelten Designsprache Material Design. Es ist ein Framework, dass sich sehr gut für responsives Webdesign eignet. Wir haben uns für Materialize entschieden, da man relativ unkompliziert ansprechende User Interfaces nach dem Material Design Prinzip erstellen kann. Zudem haben uns die Komponenten für die mobile Darstellung überzeugt. Materialize wurde in unserem Projekt verwendet, um das Frontend der Hauptseite sowie des Baukastensystems zu realisieren. Eine sehr ähnliche Alternative wäre zum Beispiel das von Google entworfene Frontend-Framework Material Design.

* + Materialize Stepper

Der Materialize Stepper ist ein Plugin für das von uns verwendete Framework Materialize. Das Plugin erweitert das Framework um einen Stepper. Stepper sind Komponenten die einen verbundenen Prozess in kleine Schritte beziehungsweise in kleine User Interfaces aufteilen. Dies macht einen längeren Prozess (zb. Die Resteraunt-Registrierung) übersichtlicher. Jener Stepper wurde unter anderem bei der Registrierung der Kunden sowie der Resteraunt-Partner verwendet.

* + Halfmoon

Was ist Halfmoon?

Halfmoon ist ein Frontend-Framework das sich an Bootstrap orientiert. Jedoch hat Halfmoon Features die sich exzellent für unser Projekt eignen. So hat Halfmoon unter anderem einen eingebauten Dark Mode der sich sehr leicht anpassen lässt und voll anpassbare CSS-Variablen. Halfmoon eignet sich also sehr für das Erstellen von Dashboards. Der Aufbau der Komponenten in Halfmoon ist ebenfalls sehr für die Entwicklung von Dashboards ausgerichtet. Jedoch hat Halfmoon nicht so viele Komponenten wie beispielweise das verwandte Bootstrap. Dennoch haben wir uns aus den oben genannten Gründen für Halfmoon entschieden. Mithilfe von Halfmoon wurde das Frontend des Dashboards realisiert.

* + JQuery

JQuery ist eine beliebte JavaScript-Bibliothek, die unter anderem Funktionen zur DOM-Manipulation bereitstellt. Durch JQuery lässt sich unter anderem eine Menge Code sparen. Zudem besitzt JQuery ein breitgefächertes Repertoire an standardkonformen Funktionen zur Manipulation des „Document Object Models“.

Da wir hin und wieder für zum Beispiel die Validierung der Öffnungszeiten oder der asynchronen Änderung von Daten den DOM bearbeiten müssen kam uns JQuery sehr gelegen. Zudem haben wir alle samt bereits Erfahrungen mit JQuery gesammelt. Deshalb viel die Wahl für uns auf JQuery. Eine eher unpraktikable Alternative wäre es, jene Funktionalitäten in Vanilla-JavaScript zu lösen.

* Werkzeuge

Im Zuge der Planung sowie der Entwicklung unseres Projektes kommen selbstverständlich auch Werkzeuge für zum Beispiel die Erstellung von Grafiken und Modell oder der Versionierung von Quellcode zum Einsatz. In folgendem Abschnitt sind die von uns verwendeten Werkzeuge aufgelistet.

* + Visual Paradigm

Bei Visual Paradigm handelt es sich um ein sehr weit verbreitetes und bekanntes Werkzeug zum Erstellen von UML Grafiken. Visual Paradigm bietet eine frei verwendbare Version (Community Edition) an. Da wir öfters UML-Grafiken für zum Beispiel das ER-Modell, Use-Cases oder Klassendiagramme anfertigen müssen, sind wir auf so ein Tool angewiesen. Visual Paradigm lässt sich intuitiv steuern und bietet zudem sehr viele verschiedene Vorlagen an. Deshalb viel für uns die Wahl eindeutig auf Visual Paradigm. Alternativen wären hier unter anderem Lucidchard, Draw.io oder Dia. Jedoch bieten die genannten Tools keine so intuitive Handhabung an. Zudem fehlt es einigen Alternativen an Features beziehungsweise Vorlagen.

* + IntelliJ
  + Git