

## Anforderungen

Im Rahmen dieser Prüfungsleistung soll die Oberfläche eines Onlineshops auf Basis von React erstellt werden. Eine Datenbankbindung ist nicht Teil dieses Projekts. Die Website soll aus drei Bereichen bestehen: Eine Seite mit Angeboten, einem Warenkorb und einem Checkout-Formular. Shopseite und Warenkorb sollen dabei funktional sein, das Checkout-Fenster ist jedoch durch die fehlende Datenbankbindung rein dekorativ.

Dabei sollen folgende in der Vorlesung gelernte Methoden und Techniken sinnvoll angewandt werden:

- Flexbox oder Grid um das Aussehen der Seite dynamisch zu gestalten
- React Router um eine performante Navigation zwischen den Bereichen zu ermöglichen
- Conditional- und List-Rendering, um die Inhalte darzustellen
- Class- und functional-based Components, um die Teile der Website sinnvoll zu kapseln
- Props, um Daten zwischen den Komponenten auszutauschen
- State (oder useState/setState), um den Warenkorb performant umzusetzen

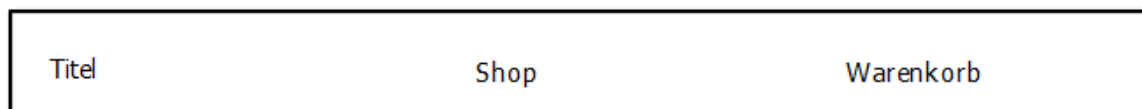
Des Weiteren sind Struktur, Namensgebung und Codequalität bewertungsrelevant. Diese beziehen sich auf eine sinnvolle Kapselung der Bestandteile, einheitliche Einrückung und Verwendung von Leerzeilen, sowie auf eine sprechende Benennung von Variablen nach einheitlichem Schema.

Die Dokumentation des Projektes soll neben den oben genannten Anforderungen einen Paper Prototype, ein Klassendiagramm, ein Konzept und ein Fazit enthalten.

## Paper Prototype und Konzept

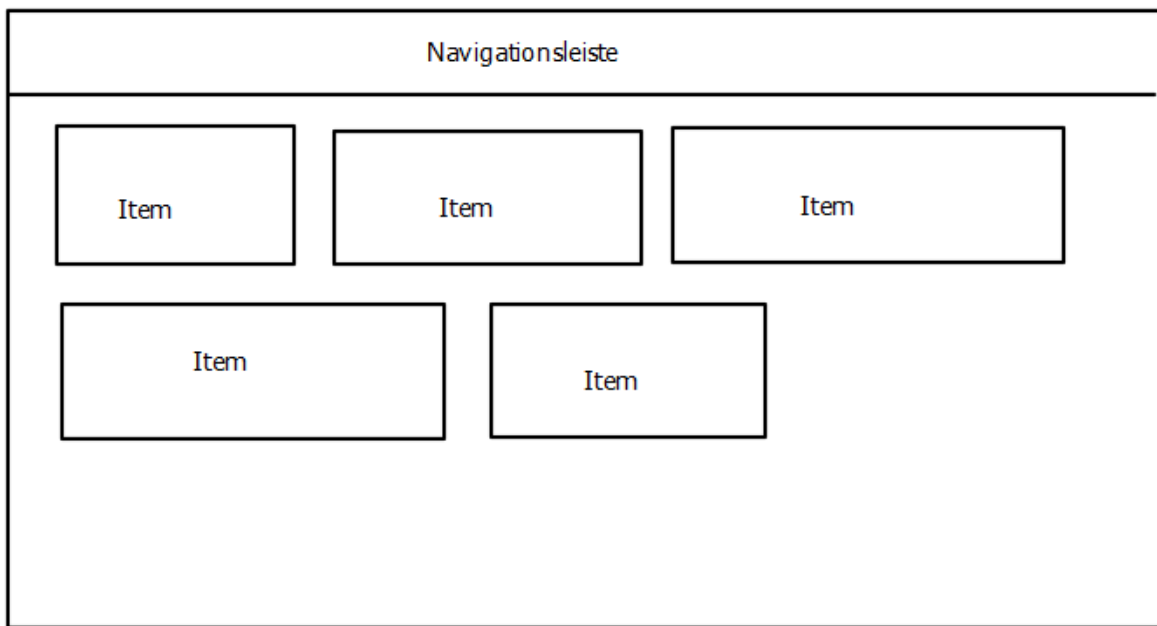
Hier wird das Design der Website schematisch dargestellt und Überlegungen zum Design werden erläutert.

### Navigationsleiste:



Die Navigationsleiste soll auf allen Seiten unverändert bleiben und immer die gleichen Optionen anzeigen. Aufgrund der wenigen Bereiche der Website ist das problemlos möglich. Der Warenkorb zeigt die Anzahl der darin befindlichen Gegenstände an.

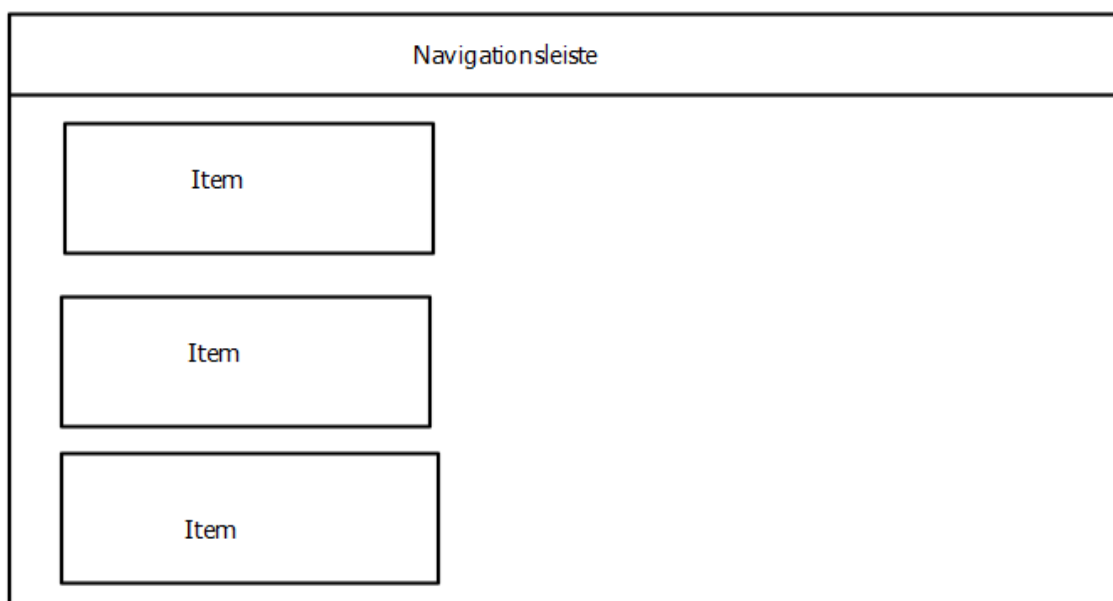
### Shopseite:



Hier werden alle verfügbaren Produkte angezeigt. Die Komponenten in einer Zeile sind immer gleich hoch. Wenn eine Komponente durch einen langen Titel in einem kleinen Browserfenster höher wird, wachsen auch alle anderen Komponenten der gleichen Zeile.

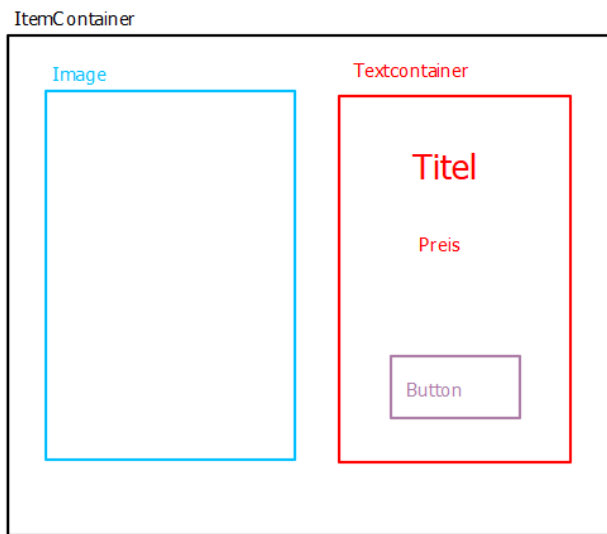
Die Maximalgröße einer Komponente beträgt 40vh, sodass immer mindestens zwei Komponenten nebeneinander passen und extrem lange Titel nicht zu riesigen Komponenten führen. Dadurch sieht die Website auf der halben Breite eines 1080p-Bildschirms noch gut aus.

### Warenkorb:



Im Warenkorb werden die Item-Komponenten untereinander angeordnet, wie es auch bei anderen Onlineshops üblich ist. Hier ist die Breite aller Komponenten gleich.

## Item-Komponente:



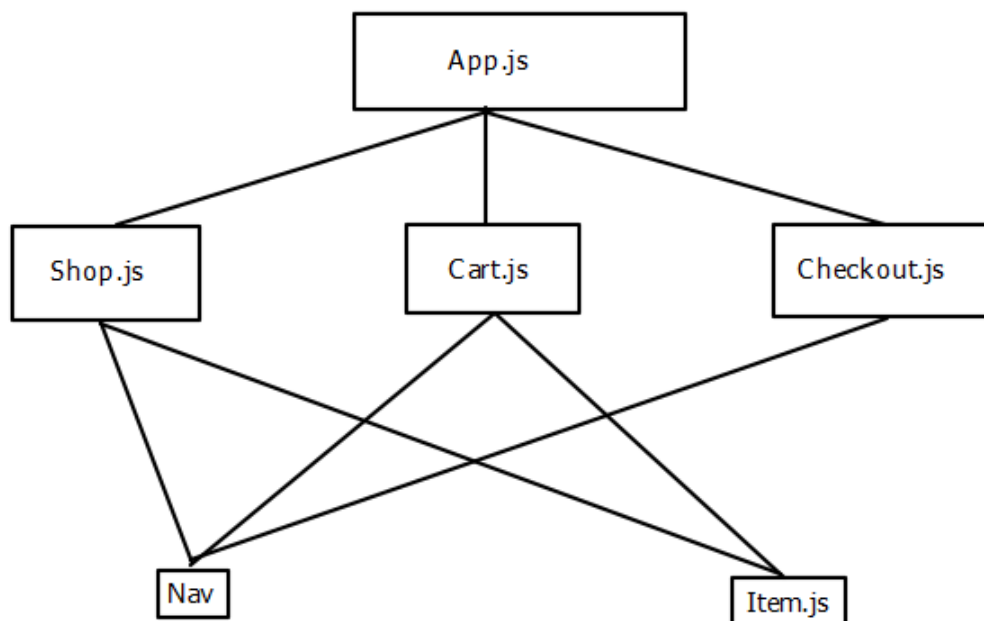
Diese Komponente wird sowohl für die Shopseite, als auch für den Warenkorb verwendet. Der gezeigte Button soll im Warenkorb durch die Anzahl und kompakte Buttons für Inkrement, Dekrement und entfernen verwendet werden.

ItemContainer ist ein div, das den Rest einschließt. Im Inneren werden die Elemente nebeneinander angeordnet.

Das Bild hat eine Höhe von 150px und eine dynamische Breite.

TextContainer enthält den Namen und Preis des Items sowie den Button, um das Item dem Warenkorb hinzuzufügen (oder für den Warenkorb Inkrement, Dekrement, Löschen und Anzahl)

## Klassendiagramm



## Fazit

Während ich zu HTML und CSS noch aus der Schule einige Vorkenntnisse hatte, waren mir JavaScript und React völlig neu. Die Einarbeitung in diese beiden Welten war nicht einfach und durch die Form dieses Projektes musste ich die Konzepte von React wirklich verstehen und es war nicht möglich, einfach Codeschnipsel zu kopieren.

Meines Erachtens ist das Übergeben von Funktionen als Parameter anderer Funktionen wenig übersichtlich und ich bin eher nach Tutorials und Beispielen vorgegangen, als die Logik dafür vollständig zu verstehen.

Props im allgemeinen und conditional-/List-Rendering haben mir kaum Probleme bereitet, weil es in vielen Programmiersprachen sehr ähnliche Konzepte gibt.

FlexBox und Router waren mir zwar neu, allerdings ist ihre Funktionsweise sehr anschaulich, wodurch ich beide fast auf Anhieb erfolgreich verwenden konnte.

Die größten Probleme hatte ich mit dem State. Wie die verschiedenen Funktionen durch Props zwischen den Elementen übergeben werden, war schwer zu verstehen und es sind viele Fehlermeldungen entstanden, die keinen Hinweis auf die fehlerhafte Stelle gaben.

Letztendlich habe ich es mit etwas Unterstützung doch hinbekommen.

Aufgrund von Zeitmangel habe ich das visuelle Design nicht ganz so hinbekommen, wie ich gerne hätte.

Grundsätzlich hatte ich mir von dem Projekt etwas allgemeineres Wissen erhofft, welches ich direkt anwenden könnte. In der Praxis werde ich mit SAP Interactive Forms by Adobe arbeiten, wo reines Javascript genutzt wird. Das war hier eher weniger vertreten.