DHBW
Duale Hochschule
Baden-Württemberg
Mannheim

Duale Hochschule Baden-Württemberg Mannheim

Projektarbeit im Fach Web-Development

Onlineshop mithilfe von React

Studiengang Wirtschaftsinformatik

Studienrichtung Data Science

Verfasser:	Canberk Alkan
Matrikelnummer:	3275561
Firma:	Linde Material Handling
Kurs:	WWI-19-DSB
Studiengangsleiter:	Bernhard Drabant
Lehrbeauftragter:	Jonas Heuer
Abgabefrist:	28.02.2021

Inhaltsverzeichnis

1 Einleitung

Im Folgenden wird die Vorgehensweise für das Erstellen eines Onlineshops im Fach Web-Development erläutert. Hierbei handelt es sich nur um ein Frontend, wodurch keine Datenbankanbindung (also Backend) erstellt werden muss. Ziel ist es einen Onlineshop mithilfe von „React“ zu erstellen, welcher beliebige Produkte darstellt. Dieser soll aus mindestens drei Seiten bestehen und bestimmte Anforderungen erfüllen. Es sollen mindestens 10 Produkte vorgestellt werden, welche in beliebig hoher Anzahl dem Warenkorb hinzugefügt werden können. Die Artikel sollen im Warenkorb angezeigt werden und entfernbar sein. Zudem soll es ein Checkout mit einem Kontaktformular geben, welcher den Namen, die Adresse, die E-Mail sowie einen Bestätigungs-Button enthalten soll. Wie bereits erwähnt, wird für das Erstellen der Anwendung die JavaScript Softwarebibliothek „React“ verwendet. Demzufolge gibt es bestimmte Anforderungen, welche im Quellcode implementiert werden sollen. Die folgende Liste soll die zu verwendenden React-Funktionen darstellen:

- FlexBox und Grid (in extra CSS-Dateien)
- BrowserRouter
- State, setState, useState
- Conditional Rendering / List-Rendering
- Class Based Components / Functional Based Components
- Hilfsmittel wie z.B. props

Wert gelegt wird zudem auf die Namensgebung, die Codequalität und Struktur sowie das Funktionieren der Anwendung im Allgemeinen.
Über GitHub soll diese Arbeit als Projekt angelegt werden. Für die Entwicklung der Anwendung sind mindestens 10 Commits vorgesehen.

2 Vorgehensweise und Umsetzung

2.1 Erste Schritte und allgemeine Struktur

Als ersten Schritt wird ein Projekt in GitHub eröffnet, sodass man Commits für das Projekt „pushen“ kann. Da vorher noch nicht mit React gearbeitet worden ist, wird die Umgebung installiert und eingerichtet. Hierfür wird „nodeJS“ und „npm“ benötigt. Als Entwicklungsumgebung wird Visual Studio Code (VSC) verwendet. Als nächstes wird mit „npx create-react-app webproject“ die React Umgebung erstellt und kann in VSC geöffnet werden. Die Datei `App.js` ist die Hauptkomponente der Anwendung, da hier alle erstellten Seiten eingebunden werden. `App.js` wird in `index.js` eingebunden, wo schließlich der `render` erfolgt. Um eine bessere Übersicht zu gewährleisten, wird für die einzelnen Seiten des Onlineshops der Ordner „Seiten“ in VSC erstellt. Dieser Ordner enthält sowohl die `.jsx`- als auch die `.css`-Dateien. Einzelne Komponenten und deren `.css`-Datei werden im Ordner „Komponenten“ gespeichert. Zudem werden in `App.js` auch props übergeben, sodass die Produkte bzw. die Artikel welche sich im Warenkorb befinden, auf mehreren Seiten verwendet werden können. Das Klassendiagramm in Abbildung 2.1 soll eine Übersicht über die verschiedenen „Website-Views“ und dessen „Parts“ geben.

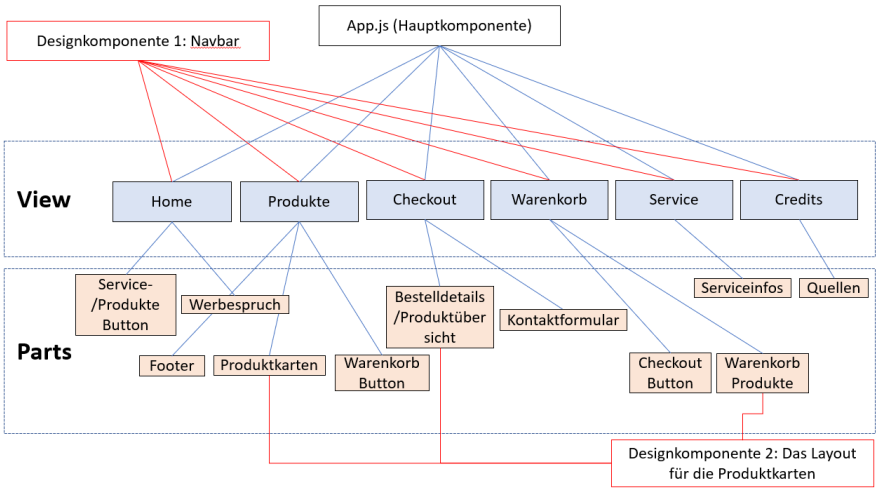

Abbildung 2.1: Klassendiagramm

Die „View“ sollen die Sichtweise des Nutzers aus den verschiedenen Shop-Seiten darstellen

und aus welchen „Parts“ bzw. Eigenschaften diese „View“ besteht. Die beiden Designkomponenten sind Komponenten, welche auf anderen Seiten wiederzufinden sind.

2.2 Komponente: Navbar

Die erste Komponente die erstellt wird, ist die Navbar. Die Navbar enthält das Logo mit dem Schriftzug „Kaffeetastisch“ sowie die Möglichkeit auf die entsprechenden Seiten „Home“, „Produkte“, „Services“ und „Warenkorb“ zu routen, was mithilfe von `BrowserRouter` in „App.js“ ermöglicht wird. Die Datei bzw. Seite „Home“ wird außerdem als Default-Seite ausgewählt. Der Warenkorb-Button leitet den Nutzer per Klick zum Warenkorb weiter. Der Button zeigt zudem die Anzahl der im Warenkorb befindlichen Artikel an. Für den Button ist eine extra `.jsx`-Datei erstellt worden, um eine bessere Struktur zu gewährleisten. Der Button wird dann in `Navbar.jsx` eingebunden. Zu erwähnen ist, dass die Navbar auf jeder Seite angezeigt und in der `.css`-Datei FlexBox verwendet wird.

2.3 Seite: Home

Als nächstes wird die Home-Seite im Ordner „Seiten“ erstellt. Die Gestaltung dieser Seite wird simpel gehalten. Im Zentrum befindet sich ein „Kaffeetastisch“-Namenszug sowie der Werbeslogan. Darunter befinden sich zwei Buttons für jeweils Produkte und Services. Als Hintergrund sind Kaffeebohnen zu sehen, um so einen Vorgeschmack für die angebotenen Produkte zu erzielen.

Abbildung 2.2: Paper Prototype: Home

2.4 Seite: Produkte

Auf der Produkt-Seite sind die Produktkarten zu sehen. Jedem Produkt wird eine Bild-URL, der Name sowie ein Preis zugewiesen. Diese Informationen werden in einer seperaten Datei `Liste.jsx` gespeichert. Damit nicht jede Karte einzeln gecodet werden muss, wird „List-Rendering“ verwendet bzw. eine `.map`-Funktion eingebaut. Mit „Props“ werden in `App.js` die Informationen aus `Liste.jsx` an die `Produkte.jsx` übergeben.
Die Karten enthalten außerdem ein Button, um die jeweiligen Produkte in den Warenkorb hinzuzufügen und diese Funktionalität ist in der const `addToCart` vorzufinden. Zudem wird eine Logik eingebaut, sodass ein Produkt, welches bereits im Warenkorb enthalten ist, nicht redundant auftritt. Mehrere gleichartige Artikel werden dann mit einem Counter innerhalb der jeweiligen Produktkarte im Warenkorb dargestellt. Die Karten werden in einer separaten CSS-Datei mithilfe von FlexBox gestyled. Auf der Produktseite ist ein Footer integriert, welcher einen Link zu „Credits.jsx“ enthält.

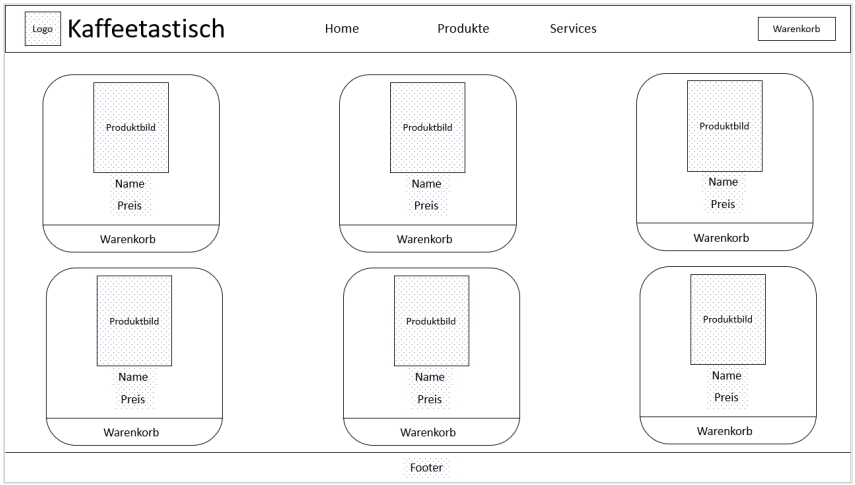

Abbildung 2.3: Paper Prototype: Produkte

2.5 Seite: Warenkorb

Der Warenkorb hat die Funktionalität Artikel, welche dem Warenkorb hinzugefügt werden zu speichern. Diese können entfernt, inkrementiert und dekrementiert werden. Auch hier wird die `.map`-Funktion verwendet, um dieselben Produktinformationen darzustellen. Allerdings ist das Layout der Karten abweichend. Es wird zum einen Grid für die gesamte Seite und allen Elementen innerhalb einer Karte verwendet. Für die einzelnen Elemente wie z.B.

dem Preis, Namen oder Bild wird FlexBox verwendet.
Falls sich keine Artikel im Warenkorb befinden wird mithilfe von Conditional-Rendering angezeigt, dass der Warenkorb leer ist.

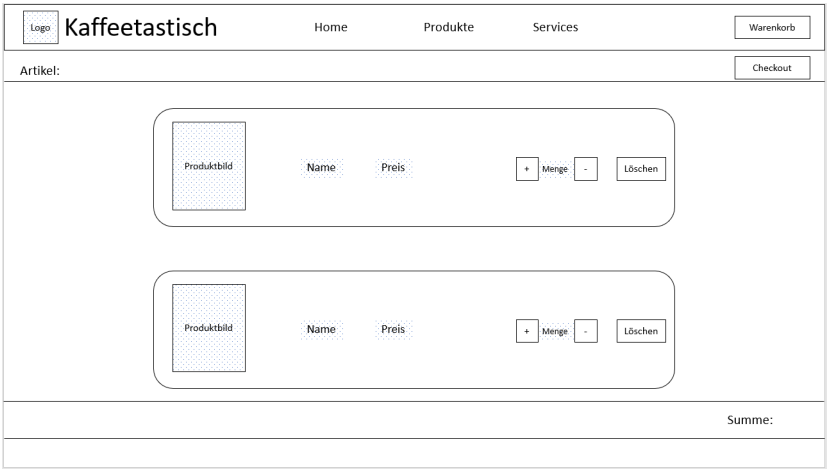

Abbildung 2.4: Paper Prototype: Warenkorb

2.6 Seite: Checkout

Zum Checkout gelangt man, indem sich Artikel im Warenkorb befinden und der Checkout-Button im Warenkorb betätigt wird. Da diese Seite in einer class geschrieben wird, werden sowohl `this.props` und `setState` verwendet. Die Liste mit den Warenkorbinformationen wird mit „props“ an Checkout weitergegeben. Es wird ein Button „Bestelldetails“ erstellt, welcher mithilfe von setState per onClick die sich im Warenkorb befindlichen Artikel in einer kleinen Übersicht nochmals darstellt. Darunter befindet sich das geforderte Kontaktformular. Jedes dieser Felder muss ausgefüllt sein, bevor die Daten bestätigt werden können. Ähnlich wie beim Warenkorb wird in der `.css`-Datei Grid und FlexBox verwendet.

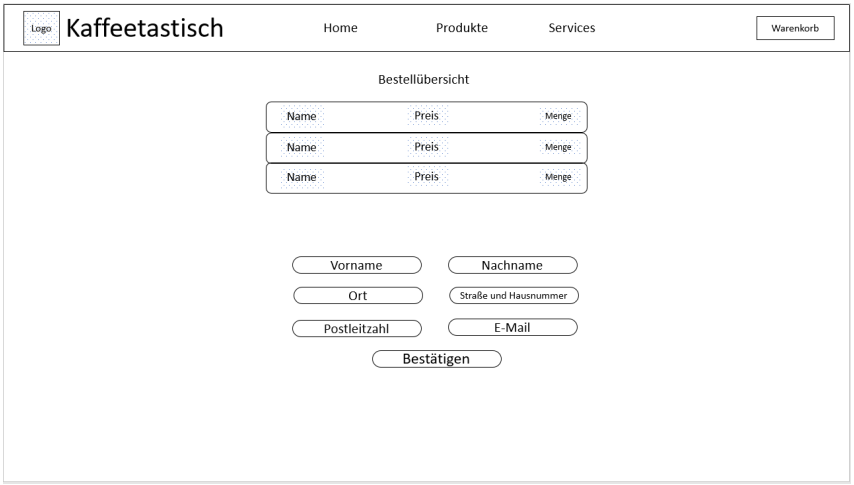

Abbildung 2.5: Paper Prototype: Checkout

2.7 Seite: Services

Die Service-Seite enthält lediglich Informationen darüber, wieso „Kaffeetastisch“ eine gute Wahl ist, woher die Produkte stammen, wie lang die Lieferzeit ist und ob Artikel retouniert werden können. Für das Layout wird Grid verwendet, da so die einzelnen Elemente leicht untereinander angeordnet werden können.

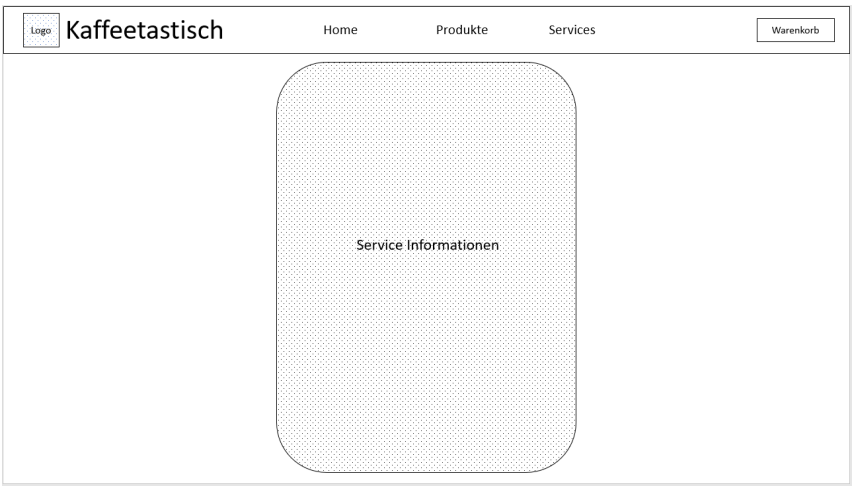

Abbildung 2.6: Paper Prototype: Services

2.8 Komponente: Footer und Seite: Credits

Der Footer weist eine ähnliche Struktur auf wie die Navbar, allerdings wird sich auf ein Element „Credits“ reduziert. Der Footer wird nur auf der Produkt-Seite repräsentiert, was ebenfalls mittels Conditional-Rendering ermöglicht wird.
Die Credits enthalten die verwendeten Quellen für die Bilder und Icons.

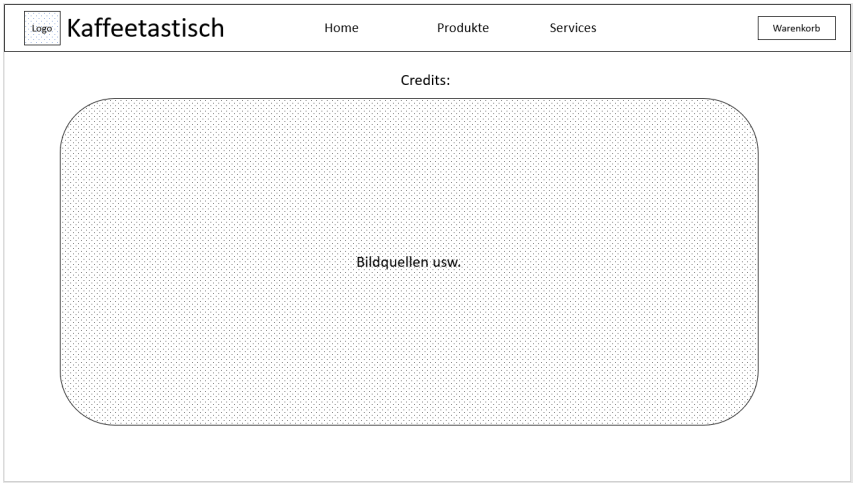

Abbildung 2.7: Paper Prototype: Credits

3 Fazit

Das Ziel dieser Projektarbeit war es, durch die Verwendung von React einen Onlineshop zu erstellen. Da ich zuvor keine persönlichen Erfahrungen im Bereich der Web Programmierung gesammelt habe, waren HTML, CSS sowie React neu für mich. Somit war eine gewisse Einarbeitung notwendig. Während die Grundlagen von HTML recht schnell erlernt worden sind, hat die Vielfalt und Funktionalität von CSS eine höhere Lernkurve dargestellt. Durch ständige Verwendung von FlexBox und Grid wurde das Prinzip mit der Zeit allerdings immer klarer und stellte somit kein Problem dar. Herausfordernd war die Übergabe von „props“ sowie die Verwendung von „useState“ und „setState“. Hier war die Lernkurve höher als bei HTML und CSS. Mithilfe von vielen Tutorials sowie der offiziellen Dokumentation von React, war ich in der Lage solch komplexere Funktionen zu verstehen und auf meinen Anwendungsfall zu adaptieren. Da ich viel Zeit in das Verstehen von „useState“ und „setState“ investiert habe, hat sich allerdings auch mein Verständnis für logische Zusammenhänge, wie z.B. die Übergabe von Artikeln in den Warenkorb, die Funktion Artikel zu inkrementieren oder zu dekrementieren, verbessert. Nach mehrmaligem Nutzen solcher Funktionen, war die Anwendung in anderen Komponenten leichter. Durch dieses Projekt habe ich die Web-Grundlagen wie HTML und CSS erlernt was im Bereich der Informatik sehr nützlich ist. Zudem bin ich nun in der Lage React-Basics zu verwenden und habe ein gutes Verständnis dafür bekommen, wie Websites aufgebaut sind.