Web Development: Projekt „Onlineshop“
Matrikelnummer: 3636997

MYFITNESSPLACE

Requirements:

Im Folgenden sind die jeweiligen Requirements der Aufgabe beschrieben.

Die Aufgabe ist es einen Onlineshop zu programmieren, welcher mindestens aus 3 Seiten besteht. Eine der jeweiligen Seite stellt die Produktseite dar, auf welcher mindestens 10 Produkte angeboten werden sollen, des Weiten sollen diese Produkte dem Warenkorb in beliebiger Anzahl hinzugefügt werden können. Der Warenkorb bildet somit die zweite Seite des Onlineshops. Hierbei sollen die gewählten Produkte angezeigt werden können, sowie auch wieder gelöscht werden können. Die letzte gefordete Seite, die Checkout Seite soll ein Kontaktformular beinhalten, mit Name, Adresse, Email und der Funktion Absenden.

Zu der Dokumentation gehört ein Paper Prototype, die Anfertigung eines Klassendiagramms, die Beschreibung der Requirements, das Konzept sowie ein Resümee.

Die Abgabe erfolgt über GitHub und Moodle und die Dokumentation wird in Form eines PDF abgegeben bis zum 28.02.2021

GitHub Link:

https://github.com/elisadandin/Web-Programmierung/commits/master

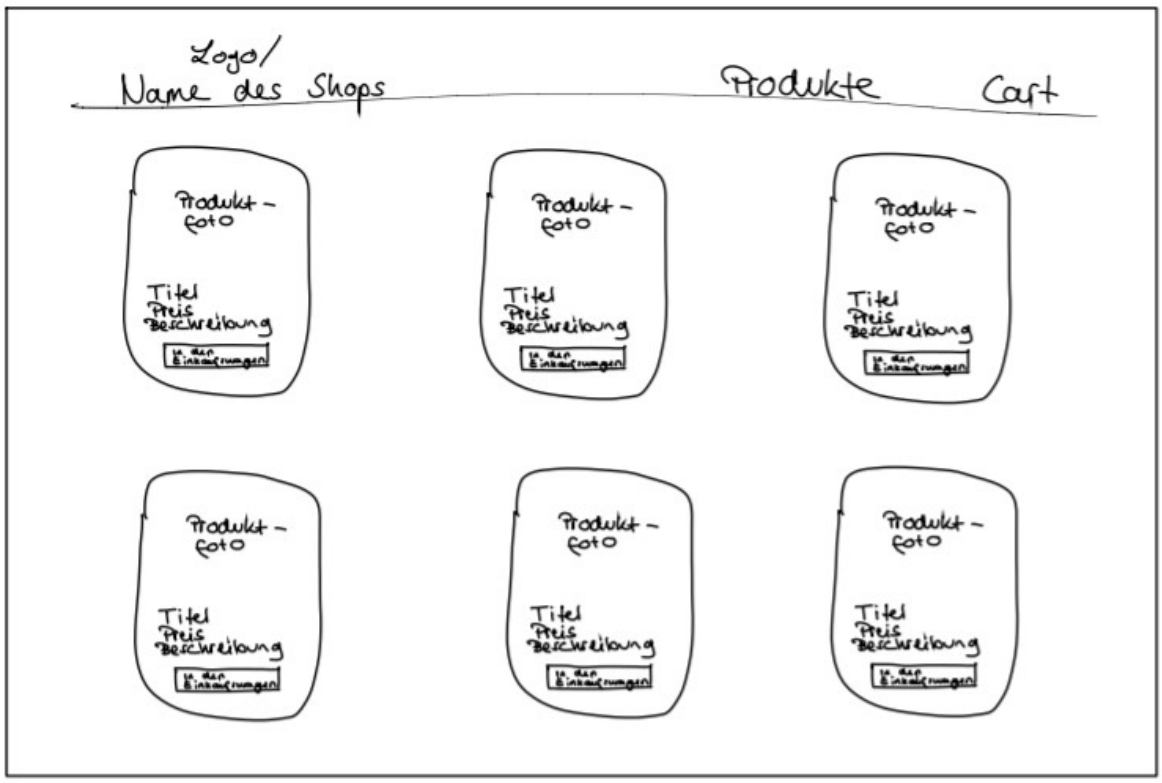
Web Development: Projekt „Onlineshop“
Matrikelnummer: 3636997

Paper Prototype:

Folgernd befinden sind die erstellten Modelle/ Prototypen des Onlineshops.

Produktseite:

Die Produktseite zeigt die jeweiligen Produkte des Webshops. Diese Produkte werden in einem Design von sogenannten Cards dargestellt. Innherhalb dieser Cards ist das jeweilige Produktbild, der Titel, eine kurze Produktbeschreibung sowie der Preis abggebildet. Des Weiteren befindet sich unter diesen Informationen ein Button, um die Produkte zum Warenkorb hinzufügen zu können.

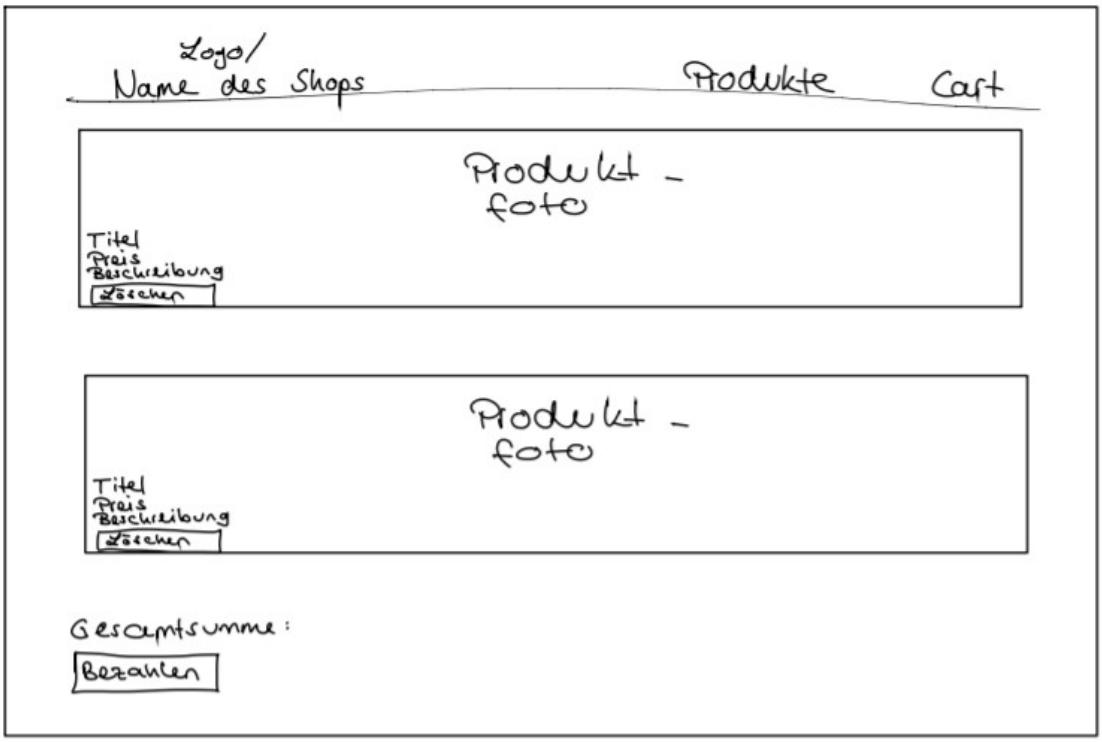


Web Development: Projekt „Onlineshop“
Matrikelnummer: 3636997

Warenkorbseite:

Die jeweiligen Produkte, die zum Warenkorb hinzugefügt werden, werden auch in Form von Cards angezeigt, und diese werden im Warenkorb untereinader gelistet angezeigt. Die jeweiligen Cards zeigen das Produktbild, den Titel, die kurze Produktbeschreibung sowie auch den Preis. Darunter befindet sich ein Button „Löschen“ um das Produkt aus dem Warenkorb zu entfernen.

Unter den Cards mit den jeweiligen Produkten, wird die Gesamtsumme abgebildet sowie ein weiterer Button mit der Aufschrift „Bezahlen“.



Web Development: Projekt „Onlineshop“
Matrikelnummer: 3636997

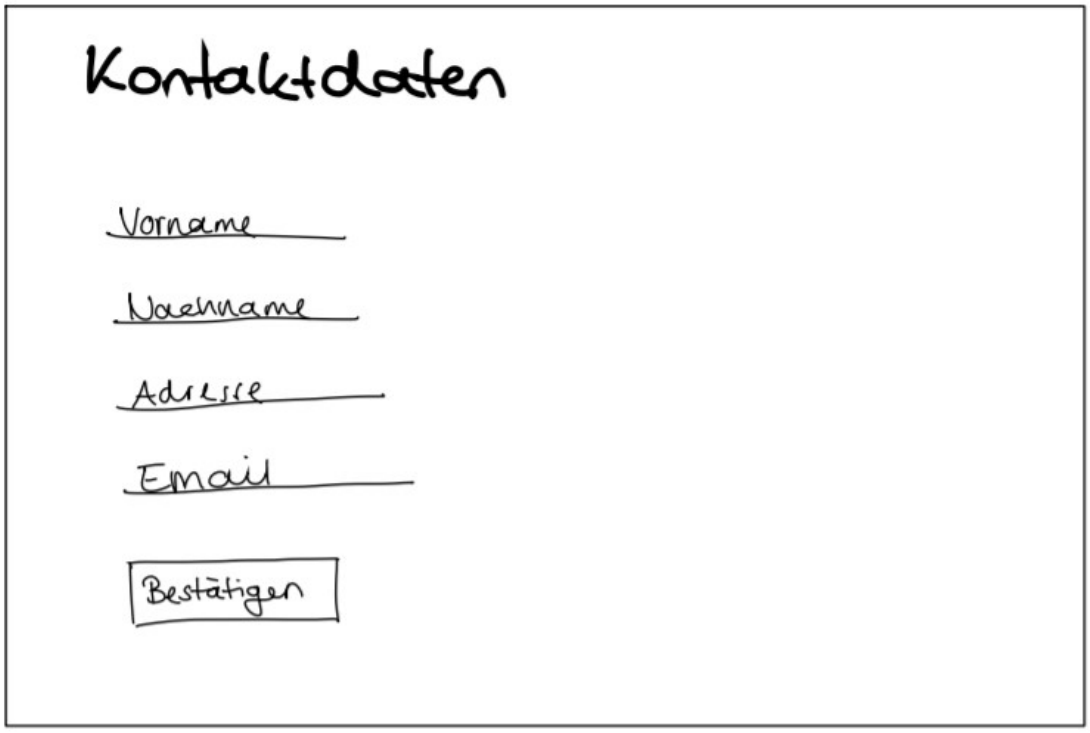
Checkoutseite:

Die jeweilige Checkoutseite besteht zum einen aus einem Konatktformular, in dem der jeweilige Benutzer seine Daten eingeben kann und einer weiteren Seite, der Success Seite, auf welcher der jeweilige Nutzer die Bestätigung sieht, dass seine Bestellung bestätigt wurde.

Kontaktformular:

Dieses Formular besteht aus verschiedenen Textboxen, in denen der jeweilige Benutzer seine persönlichen Daten angeben kann. Die Textboxen sind „Vorname“, „Name“, „Adresse“ sowie „E-Mail“ Adresse.

Unter dieser Eingabe befindet sich ein Button mit dem der Benutzer seine Eingaben bestätigt und somit die Bestellung der Produkte vollendet. Er wird automatisch zur Successseite weitergeleitet.



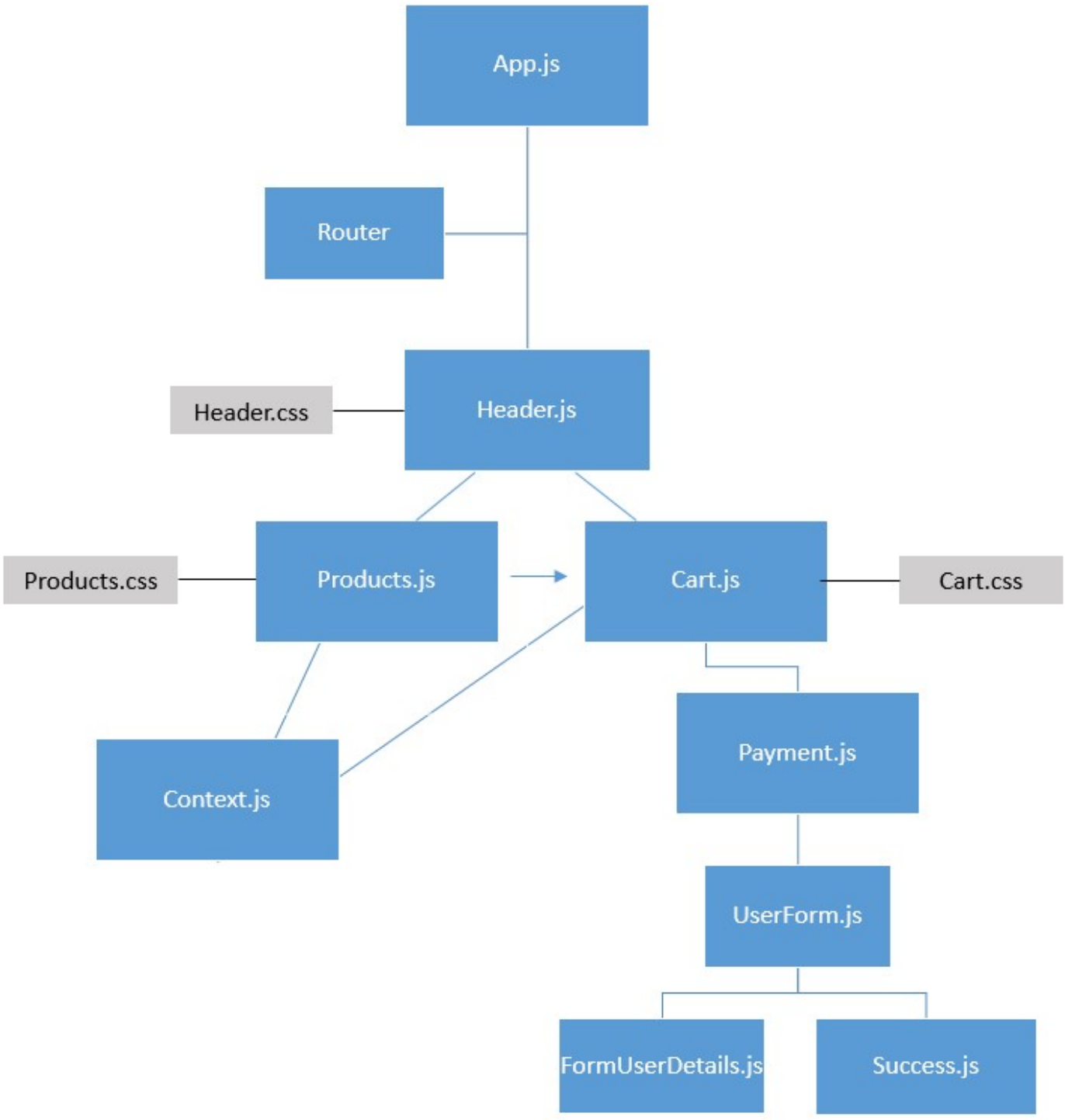
Successseite:

Hier werden die Information gezeigt, dass sich der Shop bei seinem Kunden für die Bestellung bedankt sowie, dass sich die jeweilige Rechnung in der Mail befindet.

Web Development: Projekt „Onlineshop“
Matrikelnummer: 3636997

Klassendiagramm:

Im folgenden ist der Aufbau der React App in Form eines Klassendiagramms dargestellt.



Web Development: Projekt „Onlineshop“
Matrikelnummer: 3636997

Konzept:

Router:
import {BrowserRouter as Router} from 'react-router-dom'

Einer der wichtigsten Funktionen der React App stellt der Router dar.
Jede Komponente innerhalb des <Router> </Router> - Elements kann nun auf den Router Context zugreifen und entsprechend darauf reagieren. Die verschiedenen Routen werden festgelegt, durch die Benutzung der Route-Komponente, welche eine path-Prop enthölt. In diesem Falle werden die path-Props in der Section.js File vergeben, sodass der Router auf die folgenden Seiten zugreifen kann

➔ Product.js; Cart.js; Payment.js

Die Header.js File, welche von der App.js File gerendert wird, rendert den Header des Webshops sowie die Navigationsleiste (<header> <nav> <nav/> <header/>). Der Header wird sowol bei der Produktseite, als auch bei der Warenkorbseite und der Paymentseite mitangezeigt.
Der Header zeigt links den Namen des Shops sowie das Logo (Herz), auf der rechten Seite ist zum einen „Produkte“ und ein Warenkorbicon zu sehen, welche mit den jeweilgen Links hinterlegt sind zur Produkt/Homepage und zur Warenkorbseite (Cart.js).

Context:
export const DataContext = React.createContext();

Diese Context Datei wird erstellt, um diesen Kontext auf einfachem Wege in anderen Komonenten refereinzieren zu können.
In diesem Fallen befinden sich darin die Produkte, die Funktionen um Artikel zum Warenkorb hinzufügen zu können / entfernen zu können und um die Gesamtsumme bestimmen zu können.

In der Context.js File werden die folgenden Funktionen verwendet:

- addCart : Produkte zum Warenkorb hinzufügen
- remove : Produkte aus Warenkorb entfernen
- getTotal : Gesamtsumme bestimmen

Die erzeugte DataContext Datei wird in die Product.js File importiert. In dieser File werden die folgenden Informationen der Contextdatei in Form von Shoppingcards dargestellt. Hierbei werden die jeweiligen Produkte durch die Nutzung der map-Funktion gerendert.
Der Button, welcher sich unten an den jeweiligen Produktcards befindet, „in den Einkaufswagen“ wird durch die Funktion onClick aktiviert, und in diesem Zuge die addCart Methode ausgeführt.

Cart: Die DataContext Datein wird auch hier zu Beginn in die Cart.js File importiert.

Mit der Hilfe von der componentDidMount() Funktion wird der Gesamtpreis bestimmt.
Zu dem Konzept meines Webshops gehört die Nutzung eines if else Statements. Überlegung war es im if-Statement zu formulieren, dass wenn der Warenkorb die Länge von Null hat, einen Text anzeigen zu lassen, dass der Warenkorb leer ist. Im else-Statement sollen dann die einzelnen Produkte, die zum Warenkorb hinzugefüt wurden wiedergegeben werden.

Payment.js / Checkoutformular:

Um dieses Formular zu Erstellen wurde die Hilfe von Material – UI genutzt, dies stellt React Komponenten dar, welche Google's Material Designs automatisch implementieren.
Um dies zu verwenden waren die folgende Importe relevant:

```
import MuiThemeProvider from 'material-ui/styles/MuiThemeProvider';
import AppBar from 'material-ui/AppBar';
import TextField from 'material-ui/TextField';
import RaisedButton from 'material-ui/RaisedButton'
```

Material UI wurde verwendet um die Bezahlung mit dem jeweiligen Kontaktformular zu erstellen. Dafür wurden verschiedene React Components erstellt, wobei die File „UserForm.js“ das Parent Component darstellt. Es enthält die States, das bedeutet die verscheidenen Steps und die verschiedenen Fields:

- Vorname
- Nachname
- Adresse
- E-Mail

Des Weiteren wurde für das Kontaktformular die folgenden Methoden angewand „nextStep()“ und „handleChange()“.

Web Development: Projekt „Onlineshop“
Matrikelnummer: 3636997

Resümee:

Zu Beginn des Projekts hatte ich häufig mehrere Probleme, die ich jedoch in den meisten Fällen schnell beheben konnte. Mit der Zeit des Projekts konnte ich mich besser in die Aufgabe der Entwicklung einarbeiten, sodass ich immer weniger Probleme bei der Entwicklung hatte und es für mich einfacher wurde den Shop zu programmieren.

Jedoch ist im großen und ganzen die gesamte Entwicklung eines Webshops ein sehr langer Prozess, da man alles im Detail planen und überlegen muss. Der Webshop besteht aus verscheidenen Phasen, die Konzeptentwicklung und die Entwicklung sowie Implementierung. Alle Phasen sind sehr relevant und bedeutend für den Webshop und bedürfen viel Zeit.

In meinem Konzept habe ich beschrieben, dass ich die Warenkorbseite mit Hilfe von if/else Statements anpassen wollte. Jedoch habe die dabei immer wieder aufkommende Schwierigkeiten, welche ich nicht beheben konnte. Mein Ziel war es mit Hilfe eines if else Befehls die Warenkorbseite anders anzeigen zu lassen, wenn sich keine Produkte im Warenkorb befinden. Ich habe es nicht geschafft und finde meinen Fehler nicht. Ich habe alle notwendigen Informationen importiert und viele verschiedene Möglichkeiten ausprobiert, aber ich konnte leider mein Problem nicht beseitigen.

Dennoch bin ich mit meinem Endergebnis des Webshops sehr zufrieden, und konnte während der Entwicklung sehr viel neues lernen.