Meine App

## FitCom

### Version 1.0.0, Datum: 29. Mai 2024

Hier kannst du einen Printscreen deiner App einfügen.

Abstract (Kurzbeschreibung)

Fitcom ist eine App die dazu da ist, dem Benutzer das Organisieren der eigenen Garderobe zu erleichtern. Man kann Kleidungsstücke erstellen, indem man einen Namen, eine Farbe und eine Kategorie gibt.

Es gibt die Kategorien Oberteil, Unterteil, Schuhe, Unterwäsche, Socken, Gürtel, Ringe, Ohrringe, Kopfbedeckung, Jacke, Taschen, Handschuhe, Armbänder, Halsketten.

Mit diesen Kleidungsstücken kann man dann seine eigenen Outfits zusammenstellen und diesen auch einen Namen geben und den Style benennen, den dieses Outfit repräsentiert.

So kann man Outfits planen und Outfits, die man gern hat abspeichern.

user stories

An dieser Stelle alle Anforderungen als User Stories auflisten:

* Als Benutzer möchte ich meine eigenen Kleidungsstücke hinzufügen, damit ich diese in der App benutzen kann.
* Als Benutzer möchte ich meine Kleidungsstücke bearbeiten können, damit diese immer auf neustem Stand bleiben und um Fehler zu korrigieren.
* Als Benutzer möchte ich Kleidungsstücke löschen können, damit ich keine Kleidungsstücke angezeigt bekomme, die ich nicht benutze.
* Als Benutzer möchte ich mit meinen Kleidungsstücken Outfits zusammenstellen, damit ich verschiedene Kombinationen ausprobieren kann.
* Als Benutzer möchte ich Outfits die mit gefallen abspeichern können, damit ich diese wiederverwenden kann.
* Als Benutzer möchte ich Outfits löschen können, damit ich keine Outfits abgespeichert habe, die ich nicht benutze.
* Als Benutzer möchte ich Outfits bearbeiten können, damit ich bestimmte Kleidungsstücke bei Outfits tauschen und Fehler korrigieren kann.
* Als Benutzer möchte ich angeben können, ob ein Kleidungsstück in meinem Besitz ist, damit ich zukünftige Outfits planen kann.
* Als Benutzer möchte ich Outfits verschieden Style-Kategorien zuteilen können, damit ich meine Outfits besser Organisieren kann.

Mockups

An dieser Stelle den Prototypen deiner App beschreiben. Füge einen Printscreen deiner Screens ein und beschreibe jeden Screen einzeln.

Technische Realisierung

Die Realisierung des Projekts lief eigentlich gut, doch da wir keine Hilfsmittel wie zum Beispiel Firebase benutzt und das ganze Frontend mit TypeScript anstatt JavaScript geschrieben haben, blieb uns am Ende nicht genug Zeit, um unser eigenes Backend in unsere App einzubinden.

Deshalb benutzt unsere App momentan die Test-/Beispieldaten aus dem «TestData» Skript in unserem Projekt, welches Beispiele für Kleidungsstücke, Outfits, und Benutzer beinhaltet.

Das Projekt hat drei Hauptordner: ‘app’, ‘components’ und ‘lib’.

Im app Ordner sind die Seiten, die in der App angezeigt werden. Das äusserste Skript im Ordner ist das index.tsx Skript, welches die Homepage mit Links zu anderen Seiten ist. Dann gibt es noch drei Ordner: ‘user’, ‘garments’ und ‘outfits’.

In den garments und outfits Ordnern ist jeweils ein index Skript (welches eine Seite ist, die alle Kleidungsstücke/Outfits auflistet), ein create Skript (welches eine Seite ist, bei der man ein neues Kleidungsstück/Outfit erstellen kann) und ein ‘[id]’ Ordner, der für dynamische Pfade für den Router da ist.

Im [id] Ordner ist bei garments und bei outfits jeweils ein index Skript, welches eine Seite mit allen Details über ein einzelenes Kleidungsstück/Outfit ist und ein edit Skript, welches eine Seite ist, bei der man das ausgewählte Kleidungsstück/Outfit bearbeiten kann.

Der dritte Ordner im app Ordner ist der ‘user’ Ordner. Dieser beinhaltet ein ‘login’ und ein ‘register’ Skript, welche, wie der Name schon verrät, eine Login Seite und eine Registrierungsseite sind.

Der ‘user’ Ordner hat auch einen ‘[id]’ Ordner, in dem ein ‘profile’ Skript ist, welches die Profilseite ist, bei der man Details über seinen Account sehen, den Account löschen oder sich ausloggen kann.

Die Seiten benutzen zum grössten Teil selbstgemachte Komponenten, welche im ‘components’ Ordner des Projects sind. Diese Komponenten sind zum Beispiel der CustomButton, der auf fast jeder Seite zu sehen ist. Ein paar Komponenten nehmen als Eigenschaft eine ‘variant’ Variable entgegen, welche die Farbe der Komponente verändert. Die Farben, die benutzt werden sind im CustomColors Skript im ‘lib’ Ordner des Projekts.

Der ‘lib’ Ordner beinhaltet Skripts, die mehr für die Funktionalität und Kommunikation mit der API zuständig sind. Alle Skripts die für die Kommunikation mit der API zuständig sind befinden sich im ‘api’ Ordner. Es gibt auch einen ‘hooks’ Ordner, in dem ein ‘session’ Skript ist, das den ‘useSession()’ hook zur Verfügung stellt, mit dem man sehen kann, ob man gerade eingeloggt ist oder sich ein- und ausloggen kann.

Ausserdem sind im ‘lib’ Ordner noch ein ‘CustomColors’, ‘CustomTypes’, ‘ECategory’, ‘help’ und ein ‘TestData’ Skript. Das CustomColors Skript beinhaltet die verschiedenen Farben, die in der App gebraucht werden. Das CustomTypes Skript beinhaltet selbstgemachte Datentypen, die oft im Projekt genutzt werden. Das ECategory Skript ist ein Enum mit den verschieden Kategorien, die ein Kleidungsstück haben kann. Das help Skript beinhaltet Methoden, die mehrmals im Projekt benutzt werden. Momentan ist im help Skript nur die «capitaliseFirstLetter()» Methode.

Im TestData Skript sind die Beispielsdaten die momentan von der App benutzt werden.

Testing

In diesem Kapitel definiert ihr die manuellen Tests, die durchgeführt werden. Es müssen mind. 4 manuelle Tests definiert werden.

Hier ein Beispiel eines Tests:

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | Testfallnummer (ST = Systemtest) |
| Anforderungen | Welche Anforderungen werden durch diesen Testfall abgedeckt. (User Stories) |
| Vorbedingungen | Was muss gegeben sein, damit dieser Test durchgeführt werden kann? |
| Ablauf | Welche Schritte werden bei der Durchführung des Tests durchlaufen? |
| Erwartetes Resultat | Was sollte nun passiert sein? |

Testprotokoll

In diesem Kapitel wird die Testausführung protokolliert. Welche Tests wurden wann von wem mit welchem Resultat durchgeführt.

Hier ein Beispiel eines Eintrages:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Person / Datum | Erfolgreich | Bemerkungen |
| ST-01 | D. Muster 18.06.2023 | Ja | Der Testfall war erfolgreich, der Testperson 1 ist jedoch aufgefallen, dass es in der angezeigten Fehlermeldung noch einen Rechtschreibfehler gibt. |
| … |  | … | … |
|  |  |  |  |

fazit

Hier kommt eure Reflexion zum Projekt.

* Was lief gut/schlecht?
* Wie seid ihr mit dem Endergebnis zufrieden?
* Was habt ihr gelernt?
* War alles vorhanden oder was fehlte noch?
* Usw.

Wir haben guten, organisierten Code geschrieben und zum grössten Teil die Funktionalität gut implementiert, doch wegen des Zeitmangels konnten wir nicht mehr das Backend einbinden und ein gutes Design machen. Wir haben uns ein Ziel gesetzt, das zu gross war für die verfügbare Zeit, weshalb wir nicht ganz fertig geworden sind.

Wie seiht ihr mit dem Endergebnis zufrieden?:

Wir sind nicht gerade zufrieden mit dem Endergebnis, da es nicht unseren Erwartungen entspricht. Es nervt ein wenig, dass wir mit 90% des Projekts fertig sind und deswegen die Funktionalität nicht ganz implementieren konnten.

Was habt ihr gelernt?:

Schmerz.

TypeScript ist nicht gut für Anfänger

GraphQL