

Instagram

Dokumentation Arbeitsblatt 3

Inhalt

Zusammenfassung und Anwendungszweck	1
Migration	1
Karussell	2
Partial-Views	2
Anpassungen im Zusammenhang mit «Devise» und «Bootstrap»	3
Lösung der Aufgaben.....	4
Aufgabe 1, Seite 3	4
Aufgabe 2, Seite 3	4
Aufgabe 3, Seite 6	4
Selbstreflexion	4
Fazit	5

Zusammenfassung und Anwendungszweck

Migration

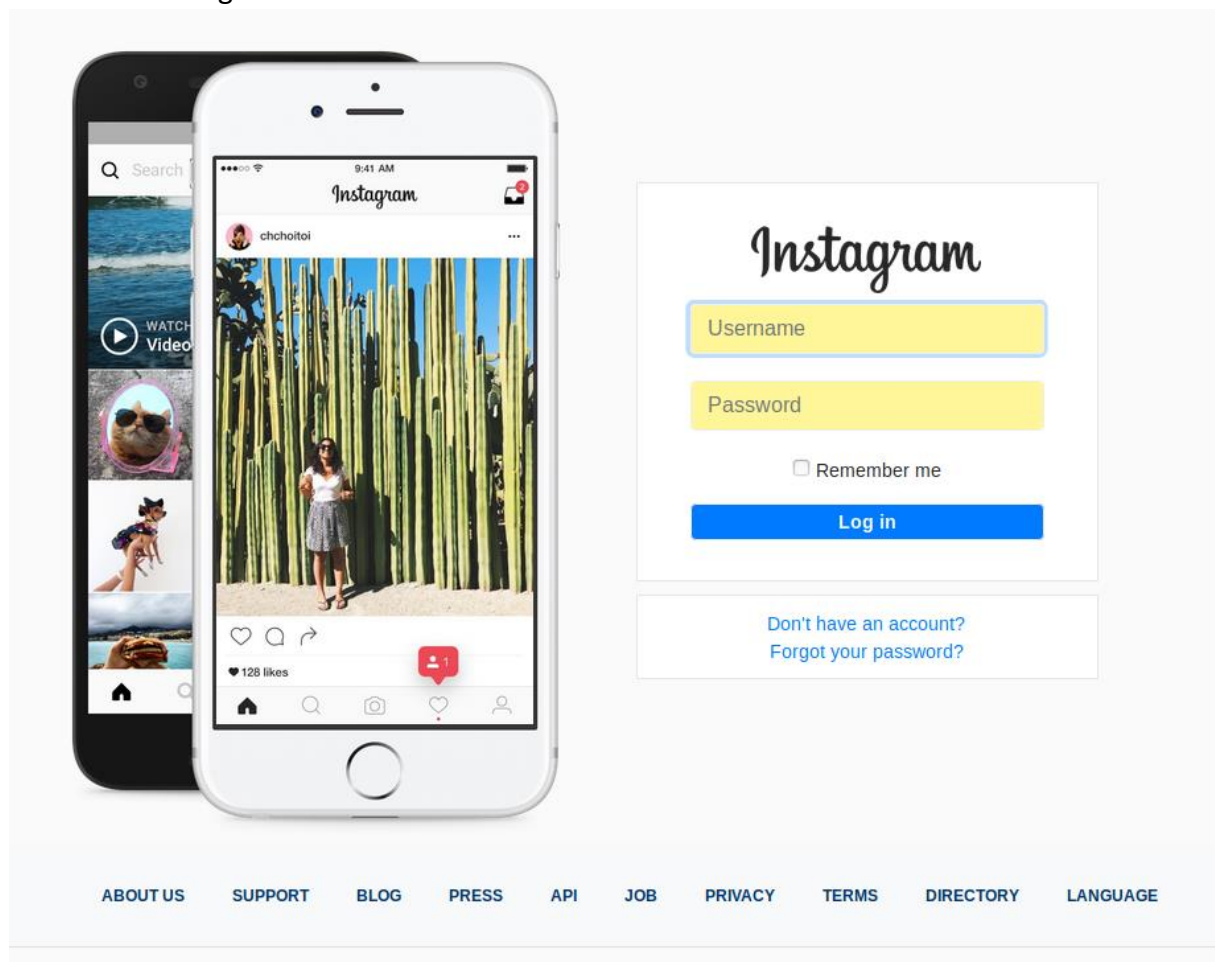
In der von Devise erstellten Datenbank hat ein Feld gefehlt, in welchem man die Benutzernamen der User speichern kann. Das kann man mit einer sogenannten “Migration” nachholen. Als erstes lässt man von Rails eine neue Migration generieren, indem man folgenden Befehl in die Konsole des Editors eingibt; “rails g migration AddNameToUser”. Rails generiert eine neue Datei unter db/migrate, in welche man die folgenden Befehle für die Änderung an der Datenbank eingeben kann:

```
class AddNameToUser < ActiveRecord::Migration[5.2]
  def change
    add_column :users, :name, :string
  end
end
```

Wenn die gewünschten Änderungen eingetragen wurden, kann man sie übernehmen, indem man in die Konsole “rails db:migrate” eingibt. Die neue Datei wird ausgelesen und die neue Spalte für den Usernamen wird eingefügt.

Karussell

Das Karussell im Dummy Phone wird von Bootstrap 4 zur Verfügung gestellt. Es besteht aus zwei Hauptcontainern. Der innere Container beinhaltet die einzelnen Elemente, welche angezeigt werden sollen. Die einzelnen Container, die das Bild enthalten, müssen mit der Klasse «carousel-item» gekennzeichnet werden und passen sich somit an das Karussell an. In diesen Containern können die Bilder angegeben werden. Für jedes Bild muss ein neuer Container erstellt werden, der in den Inneren Hauptcontainer eingebettet wird. Standardmässig wird alle 5 Sekunden das Bild gewechselt und lässt somit das Dummy Phone etwas spannender wirken. Das Karussell im Zusammenhang mit dem Dummy Phone sieht am Ende wie folgt aus:



Partial-Views

Partial-Views sind Teile von Ansichten, die nur einmal geladen werden müssen und trotzdem auf mehreren Ansichten angezeigt werden können. Man kann den Code in der einen Datei ändern und er wird für alle Ansichten, in denen die Partial-View vorhanden ist, übernommen. Jede Datei, die eine Partial-View darstellen soll, fängt mit einem «Underscore» an. Dann kann sie mit dem Befehl «render (Name der View)» in die gewünschte Ansicht gerendert werden. Beim Footer wird der Code in die Hauptdatei des Projekts gerendert, welche im Projekt unter app/views/layouts/application.html.erb zu

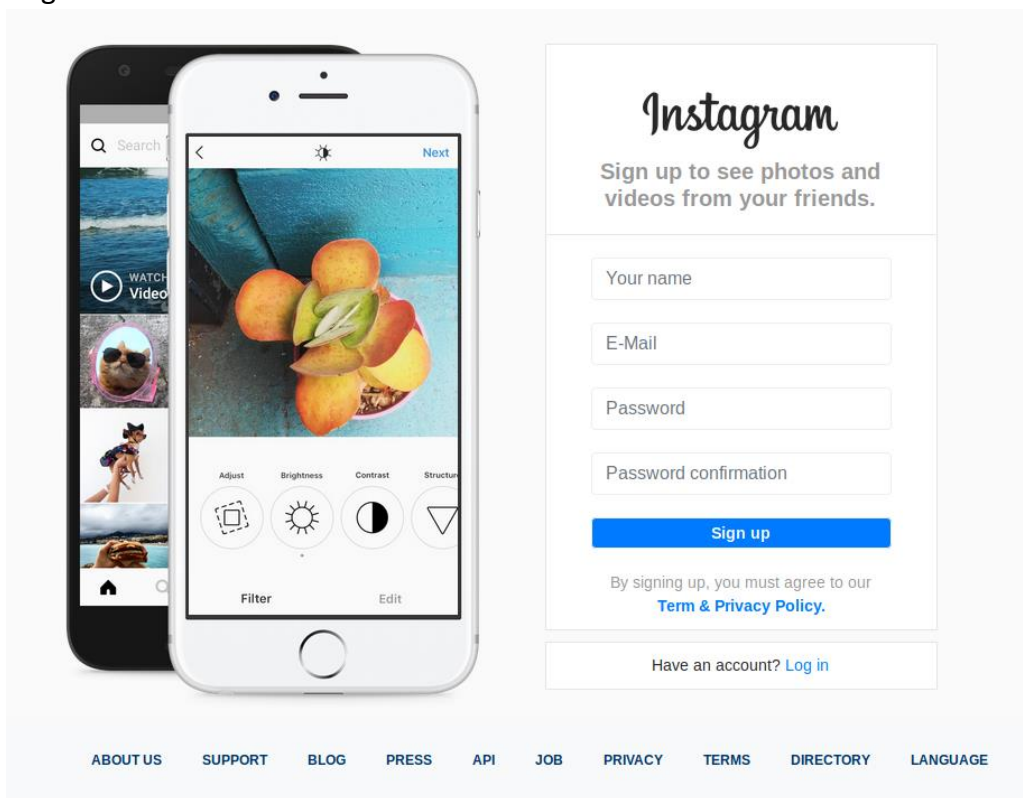
finden ist. Das Dummy Phone wird zwei Mal, bei der Login-Ansicht und bei der Registration, aufgerufen.

Anpassungen im Zusammenhang mit «Devise» und «Bootstrap»

«Devise» wurde für die Views Login und Registration benutzt. Die Vorlage von Devise wurde angepasst und erweitert. An den standardlinks, um zwischen den Views zu wechseln, wurden umbenannt. Die Links werden ganz einfach automatisch generiert, je nachdem, in welcher Ansicht man sich befindet. Beispielsweise wird beim Login der Link zu Registration angezeigt und bei der Registration der Link zum Login. Das kann ganz einfach mit dem folgenden HTML Tag erreicht werden, wobei der Tag in beiden Views genau derselbe ist:

```
<div class="redirect-links box">
  <%= render "devise/shared/links"%>
</div>
```

Devise erkennt, in welcher View man sich befindet und generiert die richtigen Links, wie im folgenden Foto:



Die

Anpassungen des Textes, der als Link angezeigt wird, kann in der folgenden Datei angepasst werden: `app/views/devise/shared/_links.html.erb`. Das ist eine bereits von Devise generierte Partial-View.

«Bootstrap» wurde dazu verwendet um den Footer und die Formulare in den beiden Ansichten Login und Registrieren ansprechend zu gestalten und um das Karussell zu erstellen. Mit der Klasse «form-control» werden die Input Felder auf den Standard von Bootstrap gestellt und sind somit etwas ansprechender bei der Bedienung. Der Submit-Button wird mit den beiden Klassen «btn» und «btn-primary» versehen und wird somit ebenfalls etwas ansprechender dargestellt.

Thema/Funktion	Vorteile	Nachteile
Migration	-Übersichtliche Darstellung der Änderungen an der Datenbank -Wenig Aufwand	-Änderungen sind erst aktiv, wenn sie migriert wurden
Karussell	-Kaum Aufwand -Übersichtlicher Code	
Partial-Views	-Ansichten werden schneller geladen -Code muss für alle Ansichten nur einmal verändert werden -Einfaches laden in die Ansichten	-Noch eine weitere Datei, die man beim Bearbeiten von Ansichten suchen muss

Lösung der Aufgaben

Aufgabe 1, Seite 3

Das Kommando um eine Migration generieren zu lassen lautet so: «rails g migration (Name of Migration)» und um die erstellte Migration auszuführen braucht man folgenden Befehl: «rails db:migrate».

Aufgabe 2, Seite 3

Um die Eingaben in das Formular auf Vorhandensein und maximale Länge zu validieren, braucht es folgenden Code: «validates :username, presence: true, length: {maximum: 50}»

Aufgabe 3, Seite 6

Die Verschachtelung der Container im HTML wird auch in der SCSS Datei übersichtlich dargestellt. Dadurch werden nur die Elemente angesprochen, welche sich innerhalb der Container befinden, die der höheren Klasse zugewiesen sind.

Selbstreflexion

Ich habe gelernt, was Partial-Views sind und dass sie sehr praktisch sein können. Ich habe auch gelernt mit Bootstrap ein Karussell zu erstellen und eine Validierung in Rails zu erstellen.

Ich habe bei Problemen erst selbst versucht sie zu lösen und habe mich in einem Fall an den Lehrer gewendet. Ich konnte dieses Problem schlussendlich jedoch selbst lösen. Ich habe mir die Zeit, die ich womöglich noch brauchen würde, eingeplant, jedoch habe ich den Auftrag teilweise etwas aufgeschoben.

Das erste Problem, das ich hatte, lag darin, dass der abgeschriebene Code für das Karussell nicht funktioniert hat. Das konnte ich lösen, indem ich den Code nochmals ganz genau

durchgegangen bin. Ich hatte ebenfalls Schwierigkeiten beim Erstellen der Validierung. Diese konnte ich mit Hilfe des Internets lösen.

Ich habe alles verstanden und konnte die Aufgaben problemlos lösen. Bei der Validierung allerdings bin ich mir nicht sicher, ob es noch eine bessere Version geben könnte.

Das nächste Mal sollte ich die Arbeit, wenn nötig, nicht aufschieben und sofort zu Hause erledigen.

Fazit

Die Instagram Applikation sieht nun schon etwas ansprechender aus und beinhaltet einige neue Bausteine. Ich konnte einiges mitnehmen und bin von den Partial-Views überzeugt.

Gelernt habe ich vor allen Dingen in der Mitte des Arbeitsblattes, da ich Bootstrap bereits recht gut kenne.