

Quicknote AB151-03

Attribut Benutzernamen erfassen

Da man sich bei Instagram einen Benutzernamen geben kann, musste ich im ersten Teil dieses Arbeitsblattes das Attribut 'name' zu der Tabelle User hinzufügen. Dies realisierte ich mit einer Rails Migration:

```
$ rails g migration AddNameToUser name:string
```

Dieser Befehl erstellte folgende Migration:

```
class AddNameToUser < ActiveRecord::Migration[5.2]
  def change
    add_column :table_name, :column_name, :type :string
  end
end
```

Mit rails db:migrate habe ich die Migration ausgeführt.

Um das Namensattribut zu validieren, habe ich folgende Zeile im User-Model hinzugefügt:

```
validates :name, presence: true, length: {maximum:50}
```

Somit wird sichergestellt, dass ein Benutzername vorhanden ist und dieser maximal 50 Zeichen lang ist. Damit man auf der kommenden Register-View einen Namen eingeben kann und versenden darf, habe ich im ApplicationController eine Methode geschrieben, welche das Attribut zulässt:

```
def configure_permitted_parameters
  devise_parameter_sanitizer.permit(:sign_up, keys: [:name])
  devise_parameter_sanitizer.permit(:account_update, keys: [:name])
end
```

Login-View anpassen

In diesem Abschnitt habe ich die Login-View an die originale Instagram-Seite angepasst. Dabei habe ich angefangen die navbar nur dann anzuzeigen, wenn auch ein User angemeldet ist. Dies habe ich durch folgende Abfrage im application.html.erb realisiert:

```
<%= if current_user
  render 'shared/navbar'
end %>
```

Danach habe ich den bereits existierenden Code auskommentiert. Um die Login-View so originalgetreu wie möglich nachzubauen, benutze ich Carousel-Klassen von Bootstrap, welche wie eine Diashow funktionieren. Dieses Carousel wird in einem Dummyphone auf der linken Seite und das Login-Formular auf der rechten Seite angezeigt. Die HTML-Struktur und der SCSS-Code wurden mir vorgegeben. Im SCSS-Code wurde nesting verwendet. Das ist eine sehr übersichtliche Verschachtelung, welche SCSS bietet.

Nachdem ich das Dummyphone integriert hatte, begann ich das Formular nachzubauen. Dazu war der Code wieder vorgegeben.

Footer einbauen

Für den Footer war wieder jeglicher Code vorgegeben. Diesen habe ich in einer neuen Partial-View namens `_footer.html.erb` unter `app/views/shared/` implementiert.

Unter `app/views/layouts/application.html.erb` habe ich die Partial-View dann eingebunden:

```
<%= yield %>  
<%= render 'shared/footer' %>  
</body>
```

Content
Footer

Registration-View

Da wir in der Login- und Registration-View exakt die dasselbe Dummyphone verwenden, habe ich dieses in eine Partial-View ausgelagert. Die Datei `_dummy_phone.html.erb` liegt nun unter `app/views/shared/`, beinhaltet das Dummyphone und wird in beiden Views eingebunden. Das Formular war anschliessend wieder vorgegeben und war gleich wie dasjenige der Login-View aufgebaut. Zum Abschluss habe ich auf der Registration View noch zwei Sätze eingebaut:

```
<h5 class="light-color"><strong>Sign up to see photos and <br>  
videos from your friends.</strong></h5>  
<p class="light-color">By signing up, you agree to our  
<br><strong>Term & Privacy Policy.</strong></p>
```

Code

```
<div class="container">  
  <div class="row">  
    <div class="col-lg-6 landing-left">  
      <%= render "shared/dummy_phone" %>  
    </div>  
    <div class="col-lg-6 landing-right text-center d-flex align-items-center">  
      <div>  
        <div class="form-login box">  
          <h2 class="core-sprite brand-name-img"></h2>  
          <h5 class="light-color"><strong>Sign up to see photos and <br>  
            videos from your friends.</strong></h5>  
          <%= form_for(resource, as: resource_name, url: registration_path(resource_name)) do |f| %>  
            <%= devise_error_messages! %>  
            <div class="form-group">  
              <%= f.text_field :name, autofocus: true, placeholder: "Your name", class: "form-control" %>  
            </div>  
            <div class="form-group">  
              <%= f.email_field :email, autofocus: true, placeholder: "Email", autocomplete: "email", class: "form-control" %>  
            </div>  
            <div class="form-group">  
              <%= f.password_field :password, autocomplete: "off", placeholder: "Password", class: "form-control" %>  
            </div>  
            <div class="form-group">  
              <%= f.password_field :password_confirmation, autocomplete: "off", placeholder: "Confirm Password", class: "form-control" %>  
            </div>  
            <div class="actions">  
              <%= f.submit "Sign up", class: "btn btn-primary" %>  
            </div>  
            <p class="light-color">By signing up, you agree to our  
              <br><strong>Term & Privacy Policy.</strong></p>  
          <%= end %>  
        </div>  
        <div class="redirect-links box">  
          <%= render "devise/shared/links" %>  
        </div>  
      </div>  
    </div>  
  </div>  
</div>
```

Linke Seite mit der Dummyphone Partial-View

Rechte Seite mit Formular

1 Registration View

```
<div class="container">
  <div class="row">
    <div class="col-lg-6 landing-left">
      <%= render "shared/dummy_phone" %>
    </div>
    <div class="col-lg-6 landing-right text-center d-flex align-items-center">
      <div>
        <div class="form-login box">
          <h3 class="core-sprite brand-name-img"></h3>
          <%= form_for(resource, as: resource_name, url: session_path(resource_name)) do |f| %>
            <div class="form-group">
              <%= f.email_field :email, autofocus: true, placeholder: "Email", class: "form-control" %>
            </div>
            <div class="form-group">
              <%= f.password_field :password, autocomplete: "off", placeholder: "Password", class: "form-control" %>
            </div>
            <%= if devise_mapping.rememberable? %>
              <div class="form-check">
                <%= f.check_box :remember_me, class: "form-check-input" %>
                <%= f.label :remember_me, class: "form-check-label" %>
              </div>
            <end %>
            <div class="actions">
              <%= f.submit "Log in", class: "btn btn-primary" %>
            </div>
          </div>
          <div class="redirect-links box">
            <%= render "devise/shared/links" %>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Linke Seite mit der Dummyphone
Partial-View

Rechte Seite mit Formular

2 Login View

```
<div class="dummy-phone">
  <div class="screen-shot">
    <div class="carousel carousel-fade" data-ride="carousel">
      <div class="carousel-inner">
        <div class="carousel-item active">
          <%= image_tag "screenshot1.jpg", height: '427', width: '240' %>
        </div>
        <div class="carousel-item">
          <%= image_tag "screenshot2.jpg", height: '427', width: '240' %>
        </div>
        <div class="carousel-item">
          <%= image_tag "screenshot3.jpg", height: '427', width: '240' %>
        </div>
        <div class="carousel-item">
          <%= image_tag "screenshot4.jpg", height: '427', width: '240' %>
        </div>
        <div class="carousel-item">
          <%= image_tag "screenshot5.jpg", height: '427', width: '240' %>
        </div>
      </div>
    </div>
  </div>
</div>
```

Karussell mit 5 Bildern.

3 Dummyphone Partial-View

```
<footer>
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <button class="navbar-toggler" type="button" datatoggle="
collapse" data-target="#navbar">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbar">
      <div class="navbar-nav mx-auto">
        <a href="#" class="nav-item nav-link" href="#">ABOUT US</a>
        <a href="#" class="nav-item nav-link" href="#">SUPPORT</a>
        <a href="#" class="nav-item nav-link" href="#">BLOG</a>
        <a href="#" class="nav-item nav-link" href="#">PRESS</a>
        <a href="#" class="nav-item nav-link" href="#">API</a>
        <a href="#" class="nav-item nav-link" href="#">JOB</a>
        <a href="#" class="nav-item nav-link" href="#">PRIVACY</a>
        <a href="#" class="nav-item nav-link" href="#">TERMS</a>
        <a href="#" class="nav-item nav-link" href="#">DIRECTORY</a>
        <a href="#" class="nav-item nav-link" href="#">LANGUAGE</a>
      </div>
    </div>
  </nav>
</footer>
```

4 Footer Partial-View

Anwendungszweck

Auch in diesem Arbeitsblatt haben wir Partial-Views verwendet. Diese sind sehr praktisch, da man damit seinen Code übersichtlich hält und Redundanzen ausschliessen kann. Partial-Views werden in fast allen grossen Webframeworks verwendet und werden einem Webentwickler sicherlich noch einige Male begegnen. Bei jedem Framework muss man sich im Vorfeld informieren und auch während dem Arbeiten Dokumentationen lesen. Genau das haben wir gemacht, um das Karussell einzubauen.

Vor- und Nachteile

| PRO | KONTRA |
|--|---|
| Viele Komponenten sind bereits in Bootstrap vorgefertigt und man muss diese nur noch anpassen. | Man muss sich über Bootstrap Klassen informieren. Einfaches HTML-CSS-Wissen reicht nicht aus. |
| Übersichtliche Struktur dank Partial-Views. | |

Selbstreflexion

1. Was habe ich gelernt?

- Ich habe gelernt, wie man eine Bootstrap Komponente einbaut und anpasst. Ansonsten war vieles Wiederholung für mich. Das Arbeiten mit Partial-Views und SCSS-Nesting kannte ich bereits. Wie man in Rails ein Attribut zu einer Tabelle, mit Hilfe einer Migration, hinzufügt war mir auch bekannt.

2. Was hat mich behindert?

- Behindert wurde ich durch nichts. Ich konnte Schritt für Schritt befolgen und kam gut vorwärts. Einzig meine Absenz von einer Woche hat mich ein bisschen aus dem Rhythmus geworfen und ich musste mich erst wieder reinlesen.

3. Was habe ich nicht verstanden?

- Alle vorgestellten Technologien sind mir bekannt und ich verstehe deren Anwendungszweck, und Benutzung.

4. Was kann ich beim Studium besser machen?

- Im Moment arbeite ich das Arbeitsblatt durch und informiere mich erst im Anschluss über benutzte Technologien. Dies sollte ich während dem Arbeiten tun.

Fazit

Das dritte Arbeitsblatt war eine «Fleiss-Arbeit». Es gab viel zu schreiben und zu verstehen. An sich wurden nicht viele neue Technologien vorgestellt. Einzig das Arbeiten mit Bootstrap-Komponenten war neu. Das bedeutet, dass ich nicht den grössten Lernfortschritt gemacht habe. (Im Vergleich zu AB2) Ich versuche mich immer über vorhandene Code-Sequenzen zu informieren, damit ich weiss, welches Attribut oder Klasse was bewirkt und nicht nur planlos kopiere. Ich konnte ausserdem alle Fragen im AB beantworten.