

Quicknote

Gem 'dropzonejs-rails'

Das Gem dropzone-rails bietet die Möglichkeit Bilder per Drag and Drop hochzuladen und mit einer Vorschau darzustellen. Man kann allerdings auch einen Button hinzufügen. Dieser Button öffnet ein kleines Fenster, auf dem man ein Bild auswählen kann. Auch bei dieser Option sieht man eine Vorschau.

Anwendung

Dieses Gem kann mit seiner Drag and Drop Funktion bei allen Webapplikationen sehr Nutz voll sein und ist garantiert keine schlechte Idee.

Vor- und Nachteile

- +Dieses Gem ist nicht wie die anderen, weil es keine Abhängigkeiten zu anderen Bibliotheken hat.
- +Dieses Gem ist sehr einfach zu konfigurieren, wie z.B. die maximale File Size.
- Wie man im Namen lesen kann, basiert dieses Gem auf JavaScript. Das heisst man sollte wenigstens die Grundlagen von JavaScript kennen.

Icons von Font Awesome

Font Awesome ist eine Bibliothek an Icons mit einer sehr grossen Auswahl von Icons. Diese Icons lassen sich ganz einfach mit CSS Klassen in einer Webapplikation einbinden. Als zweite Option kann man sie auch als SVG herunterladen.

Anwendung

Icons sind sehr wichtig um Webseiten noch schöner zu gestalten. Font Awesome ist eine sehr praktische Lösung, welche in allen Webapplikationen angewendet werden kann.

Vor- und Nachteile

- +Die Icons von Font Awesome sind, wie schon erwähnt, sehr einfach in CSS Klassen zu implementieren.
- +Ausserdem hat Font Awesome nicht nur eine sehr grosse Auswahl an Icons, sondern man kann sie in der Grösse und der Farbe verändern.
- Einige Icons sind leider nur in der Pro-Version verfügbar und daher kostenpflichtig.

Design der Seite mit den Posts

Das Design der Seite mit den Posts wurde mit eigenen CSS Klassen und mit Bootstrap gestaltet. Es wurden auch einige Icons aus unserem Core-Sprite Sheet oder von Font Awesome verwendet.

Anwendung

Alle unsere verwendeten Technologien (CSS, Bootstrap, Core-Sprite Sheet, Font Awesome) können in jeder Webapplikation verwendet werden. Ein gutes styling ist immer sehr wichtig für eine Webapplikation.

Vor- und Nachteile

- +Durch das Bootstrap sieht die Webapplikation sehr schnell sehr gut aus. Das Carousel kann sehr schnell und einfach erledigt werden.
- +Mit den Icons von Font Awesome sieht die ganze Webapplikation noch schöner und besser aus.
- Durch all den verschiedenen Container mit verschiedenen Bootstrap Klassen, Font Awesome Klassen und unseren Klassen kann die Webapplikation schnell unübersichtlich werden.

Partial-View für die Anzeige der Posts

Mit dem Code für den Slider wächst der Code für die Anzeige der Posts immer mehr an. Deshalb ist es nur zu empfehlen diesen Teil in eine Partial-View auszulagern und sie danach zu rendern.

Anwendung

Eine Partial-View ist immer dann zu verwenden, wenn man den gleichen Code mehrmals benötigt. Eine Partial-View ist sehr praktisch und immer sehr zu empfehlen.

Vor- und Nachteile

- + Eine Partial-View erspart unter Umständen sehr viel Code.
- + Eine Partial-View macht den ganzen Code viel übersichtlicher und lesbarer.
- Meiner Meinung nach gibt es keine Nachteile zu der Partial-View.

Selbstreflexion

Was habe ich gelernt?

In diesem Arbeitsblatt habe ich eine neues Gem kennengelernt: dropzonejs-rails. Zu diesem Gem habe ich ausserdem ein wenig JavaScript gelernt. Bootstrap kannte ich zwar schon, konnte aber mein Wissen erweitern. Allerdings kannte ich die Funktionen «previous» und «next» des Carousels noch nicht und durfte diese kennenlernen. Font Awesome kannte ich auch noch nicht. Ich fand es sehr einfach die Icons von Font Awesome einzubinden. Die Partial-View zu erstellen und zu rendern kannte ich schon von den vorherigen Arbeitsblättern und war somit nichts neues für mich.

Wie bin ich vorgegangen?

Ich ging so vor, wie bei den anderen Arbeitsblättern auch. Zuerst lass ich den ersten Auftrag und versuchte ihn direkt zu lösen, noch bevor ich den nächsten Auftrag las. Wenn eine Frage im Arbeitsblatt kam, antwortete ich gleich in der Quicknote. Erst am Ende des Arbeitsblattes verfasste ich die Quicknote.

Was hat mich behindert?

Behindert hat mich in diesem Arbeitsblatt nichts. Ich konnte sehr konzentriert und problemlos daran arbeiten, ohne dass ich in irgendeiner Form behindert wurde, was mich ehrlich gesagt positiv überrascht hat 😊.

Was habe ich nicht verstanden?

Verstanden habe ich alles sehr gut, nicht nur dank der Beschreibung, sondern auch dank den hilfreichen Links. Ausserdem sind meine Kollegen sehr hilfsbereit, falls ich etwas nicht verstanden hätte.

Was kann ich nächstes Mal besser machen?

Dieses Mal war ich sehr zufrieden. Ich löste das ganze Arbeitsblatt nicht am vorherigen Tag, sondern noch ein Tag davor. Die Quicknote hingegen löste ich ein wenig spät, was ich das nächste Mal ändern kann.

Lösungen der Aufgaben

Posts und Photos unterbinden

Meiner Meinung nach kann im Model in Rails mit «*validates :property, presence: true*» geprüft werden, ob das Property gesetzt ist. Leider hat dies bei mir nicht funktioniert. Allerdings wird durch die Implementation des Dropzone Gems schon sichergestellt, dass kein Post ohne mind. ein Bild erstellt werden kann.

Bildspeichergrößen ändern

In der Datei `app/assets/javascript/upload_post_images.js` wird die dropzone konfiguriert. Hier gibt es die Konfiguration «maxFileSize», die man auf 4 (für 4MB) ändern muss.

Abschliessende Reflexion

Alles in allem habe ich viele neue Technologien kennengelernt und auch repetiert, welche ich in der Zukunft für eine Rails-Applikation verwenden kann. Hauptsächlich habe ich gelernt, wie man einfacher und auch schöner einen Post erstellen kann. Ausserdem habe ich gelernt wie man ganz einfach Icons einbinden kann. Mit jedem Arbeitsblatt, das ich löse, wird mein Rails-Wissen erweitert.

Screenshots



Abbildung 1: Post mit einem Bild

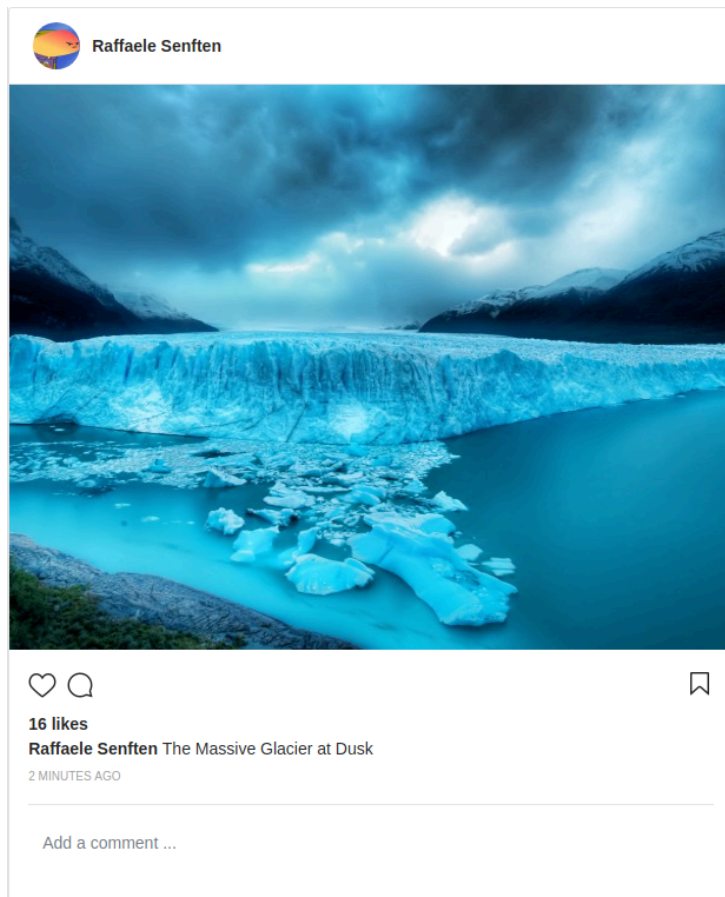


Abbildung 2: Post mit Kommentar

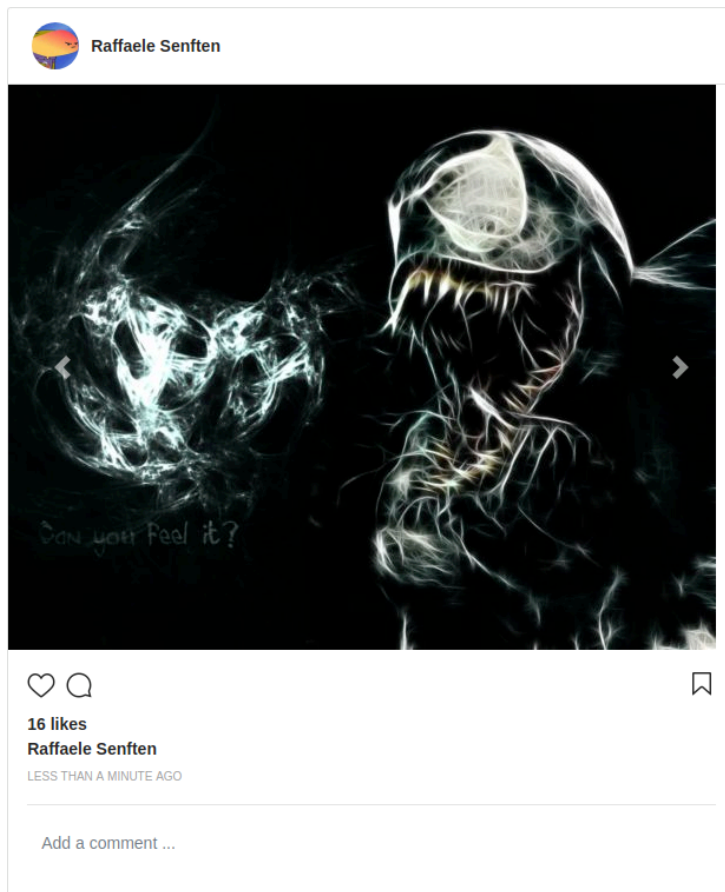


Abbildung 3: Post mit mehreren Bildern