

Readme

Γκάρτζιος Γκάρτζιος 2666

Μπάρζας Γιάννης 2765

1^η Εργασία Τεχνολογίας Διαδικτύου

Στην εργασία αυτή απαιτούνταν να τροποποιήσουμε τον κώδικα της εφαρμογής **Treegram** ώστε ο χρήστης να μπορεί:

- a) Να προσθέσει έναν Τίτλο στην κάθε φωτογραφία που ανεβάζει.
- b) Να μπορεί να ακολουθήσει άλλους χρήστες.
- c) Να φαίνονται οι φωτογραφίες του εκάστοτε χρήστη και αυτών που ακολουθεί σε αντίστροφη χρονολογική σειρά.
- d) Να προσθέσει σχόλια στις φωτογραφίες.
- e) Να διαγράψει τις δικές του φωτογραφίες και ταυτόχρονα να διαγράφονται τα σχόλια και τα tag αυτής.

Για να πραγματοποιηθούν αυτά τα ερωτήματα έπρεπε να τροποποιήσουμε τη βάση και συγκεκριμένα δημιουργήσαμε τα απαραίτητα migrations και κάναμε τις απαιτούμενες αλλαγές στο schema.rb.

Schema

Στον πίνακα “photos” προσθέσαμε το πεδίο:

- t.string “title” ώστε να αποθηκεύεται ο τίτλος της κάθε φωτογραφίας.

Επιπλέον δημιουργήσαμε τον πίνακα “follows” με πεδία:

- t.integer “following_id” δηλαδή το id του ακόλουθου .
- t.integer “followed_id” δηλαδή το id του ακολουθούμενου χρήστη.

Τέλος φτιάξαμε τον πίνακα “comments” με πεδία:

- t.integer “photo_id” δηλαδή το id της φωτογραφίας που σχολιάζει ο χρήστης.
- t.integer “user_id” δηλαδή το id του χρήστη που σχολιάζει.
- t.string “email” δηλαδή το email του χρήστη που σχολιάζει.
- t.text “comment” το σχόλιο.

Models

Φτιάξαμε τα μοντέλα:

1. follow.rb όπου δηλώσαμε ότι ανήκει στον user με την εντολή belongs_to:user.
2. comment.rb όπου δηλώσαμε ότι ανήκει στον user με την εντολή belongs_to:user και στην φωτογραφία belongs_to:photo.

Επιπλέον τροποποιήσαμε τα ήδη υπάρχοντα μοντέλα:

1. user.rb δηλώσαμε ότι έχει πολλά comments has_many: comments και πολλά follows has_many: follows

2. photo.rb δηλώσαμε ότι έχει πολλά comments has_many: comments, dependent: destroy και πολλά follows has_many: tags , dependent: destroy και δηλώσαμε πως τα tags, comments εξαρτούνται από την ύπαρξη της photo.
3. tag.rb δηλώσαμε ότι ανήκει στον user και την photo belongs_to :user belongs_to :photo

Controllers

Δημιουργήσαμε τους controllers:

1. Comments που προσθέσαμε τις μεθόδους:
 - Create για την δημιουργία του Comment
 - Destroy για την διαγραφή του Comment
 - Comment_params που ελέγχει και δίνει τα ορίσματα του comment
2. Follows που προσθέσαμε τις μεθόδους
 - Create για την δημιουργία του Follow
 - Follow_params που ελέγχει και δίνει τα ορίσματα του follow

Τροποποιήσαμε τους controller:

1. Tags που προσθέσαμε την μέθοδο:
 - Destroy για την διαγραφή του Tag
2. Photos που προσθέσαμε την μέθοδο:
 - Destroy για την διαγραφή της photo

Και τροποποιήσαμε την:

- Photo_params όπου προσθέσαμε την τιμή title.
3. Users που προσθέσαμε την μέθοδο:
 - Index που ορίσαμε σε ένα αντικείμενο users όλους τους User.

Views

Για την εμφάνιση του τίτλου προσθέσαμε τις αντίστοιχες εντολές ώστε να παίρνουμε τον τίτλο από την **new.html.haml** και να εμφανίζεται στην **show.html.haml**.

Στον user τροποποιήσαμε την:

1. **Show.html.haml** όπου προσθέσαμε τις αντίστοιχες εντολές ώστε:
 - Να εμφανίζεται το κουμπί users ώστε να μεταφερθούμε στην σελίδα εμφάνισης όλων των χρηστών.
 - Να εμφανίζονται οι φωτογραφίες (μαζί με τους τίτλους τους , τα tags και τα σχόλια τους) του χρήστη

και αυτών που ακολουθεί σε αντίστροφη χρονολογική σειρά.

- Να εμφανίζεται ο τίτλος της κάθε φωτογραφίας πάνω από αυτήν.
- Να εμφανίζονται τα σχόλια και τα tag κάτω από την φωτογραφία.
- Να εμφανίζεται η φόρμα υποβολής σχολίου δεξιά της φωτογραφίας.
- Να εμφανίζεται το κουμπί **Delete** (δεξιά της φωτογραφίας και κάτω από την φόρμα comment) που διαγράφει την αντίστοιχη φωτογραφία του συνδεδεμένου χρήστη και τα comments και tags που σχετίζονται με αυτήν.

2. *Index.html.haml* όπου προσθέσαμε τα αντίστοιχες εντολές ώστε:

- Να εμφανίζονται όλοι οι εγγεγραμμένοι χρήστες εκτός του συνδεδεμένου.
- Να δίνεται η δυνατότητα στον συνδεδεμένο χρήστη να ακολουθεί άλλους χρήστες μέσω ενός κουμπιού **Follow** που εμφανίζεται δεξιά του email του χρήστη
- Ένα κουμπί **Back** που σε επιστρέφει στην αρχική σελίδα