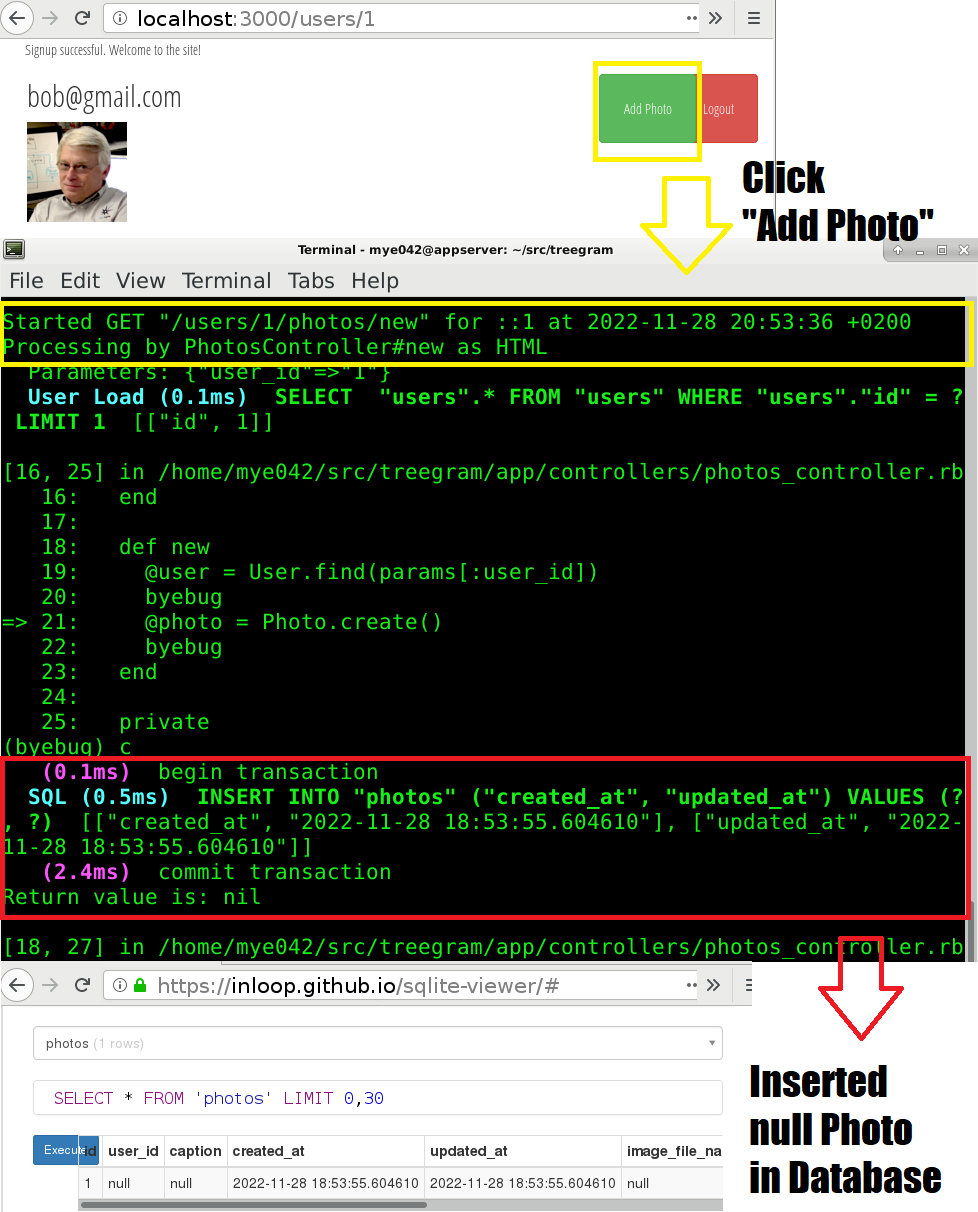
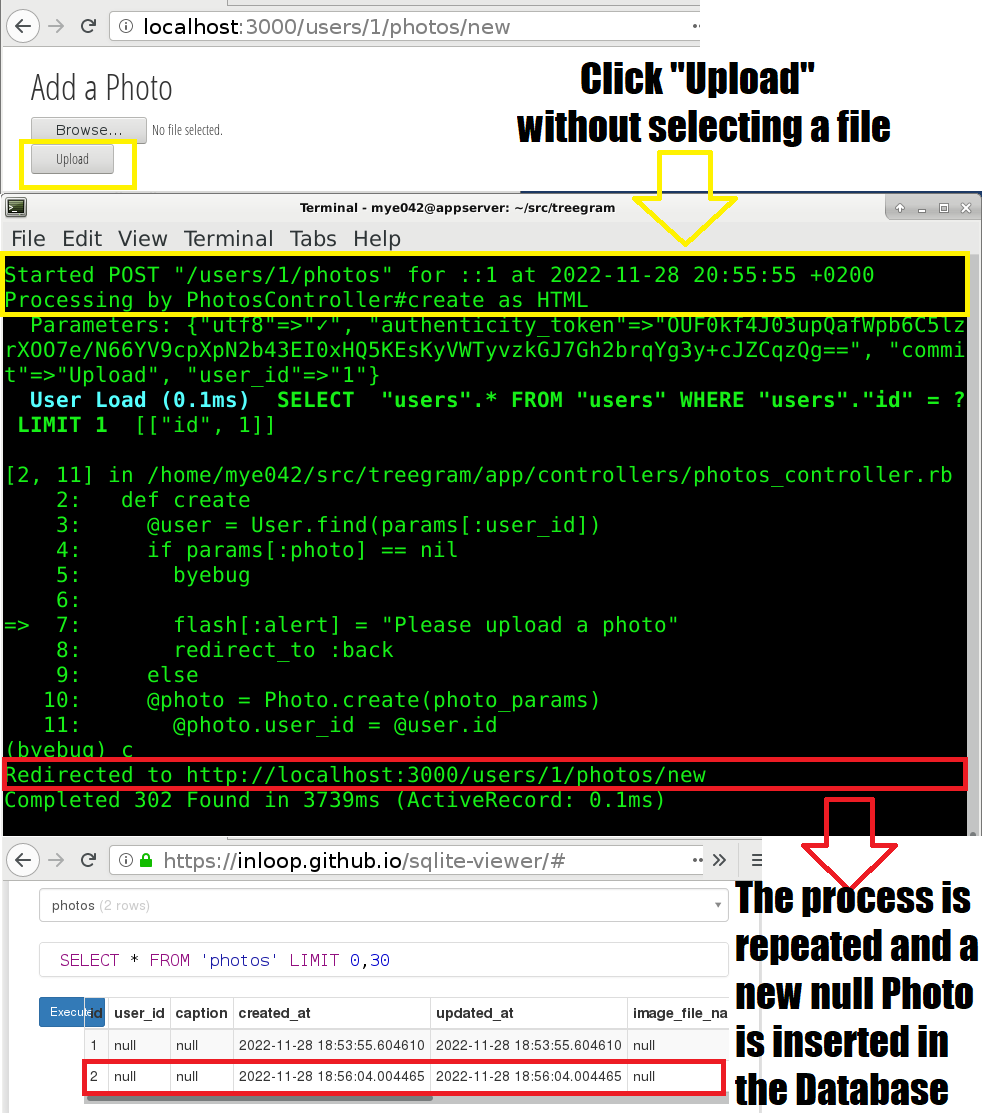
1. Προυπάρχοντα erors  
Όταν πατούμε το κουμπί Add Photo, δημιουργείται μια null Photo πριν καν φορτωθεί η προβολή new.html.haml. Αυτό γίνεται, διότι η συνάρτηση Ruby που υλοποιεί την ενέργεια new στον PhotoController καλεί την Photo.create().

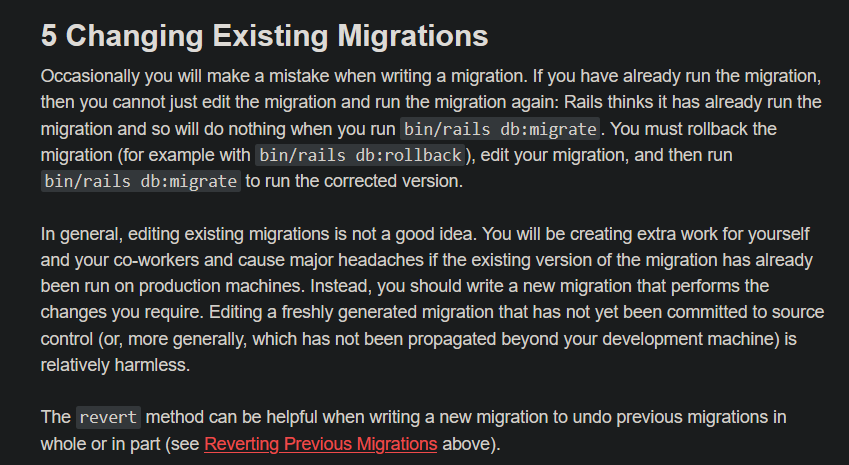
Επιβεβαιώσαμε τα παραπάνω, χρησιμοποιώντας τον διαδραστικό αποσφαλματωτή byebug και τον online [SQLite Viewer](https://inloop.github.io/sqlite-viewer).

Αφού ολοκληρωθεί η κλήση της new, φορτώνεται η προβολή στον διακομιστή μας.

Σε αυτό το σημείο, μπορούμε να ξαναδημιουργήσουμε μια null Photo, αν πατήσουμε στην φόρμα προσθήκης Photo το Upload χωρίς να έχουμε προσθέσει φωτογραφία.  
  
Αυτό γίνεται καθώς η συνάρτηση Ruby που υλοποιεί την ενέργεια create στον PhotoController, κάνει redirect στην new, αν δεν έχει επιλεχθεί φωτογραφία, και επαναλαμβάνονται όσα αναφέραμε παραπάνω.

# 2. Προσθήκη τίτλου σε φωτογραφία

Από <https://guides.rubyonrails.org/active_record_migrations.html>



Άρα για αυτόν τον λόγο θα φτιάξουμε ένα καινούργιο migration έτσι ώστε να προσθέσουμε τον τίτλο στις φωτογραφίες.

Για την προσθήκη του column title στον πίνακα photos της βάσης μας εκτελούμε τις εξής εντολές στo root directory της εφαρμογής:  
$ rails generate migration add\_title\_to\_photos title:string  
$ rake db:migrate  
  
Graphical user interface, text, application

Description automatically generated

A screenshot of a computer screen

Description automatically generated

Στο αρχείο “app/views/photos/new.html.haml” προσθέτουμε την παρακάτω γραμμή:  
$ %h2 Add a title: #{form.text\_field :title, placeholder:"Look at my dog!"}  
Συνεπώς, το prompt για το upload φωτογραφίας θα φαίνεται έτσι:  
Graphical user interface, application

Description automatically generated

Στο αρχείο “app/controllers/photos\_controller.rb” προσθέτουμε τους εξής ελέγχους για να αναγκάσουμε το χρήστη να έχει επιλέξει φωτογραφία προς ανέβασμα και να έχει προσθέσει τίτλο πριν του επιτραπεί να προχωρήσει:  
$ if params[:photo][:image] == nil  
$ flash[:alert] = "Please upload a photo."  
$ redirect\_to :back  
$ elsif params[:photo][:title] == ""  
$ flash[:alert] = "You forgot to add a title!"  
$ redirect\_to :back

Επίσης βάζουμε το argument :title στη γραμμή $ params.require(:photo).permit(:image, :title)  
Text

Description automatically generated

Στο αρχείο “app/views/users/show.html.haml” προσθέτουμε την εξής γραμμή για να υποδείξουμε στον κώδικά μας να εμφανίζει τον τίτλο κάθε φωτογραφίας:  
$ %h2= photo.title

Επίσης προσθέτουμε την εξής γραμμή για να εμφανίζεται το όνομα και το catchphrase της εφαρμογής όσο τη χρησιμοποιούμε:  
$ %h1.title Treegram - Show off to the world!  
Timeline

Description automatically generated

Graphical user interface, application

Description automatically generated

# 3. Προσθήκη λειτουργίας follow

Ξεκινάμε με την προσθήκη του μοντέλου follow:  
$ rails generate model follow  
Text

Description automatically generated

Έπειτα στο αρχείο “app/db/migrate/TIMESTAMP\_create\_follows.rb” ορίζουμε τη δομή του πίνακα με τα αναγνωριστικά του follower (αυτός που ακολουθεί τα post κάποιου) και του followee (αυτός του οποίου τα post ακολουθούν άλλοι):  
Text

Description automatically generated

Εκτελούμε $ rake db:migrate για να δημιουργήσουμε τον πίνακα των follow:  
Text

Description automatically generated

Ο οποίος σε αυτό το σημείο είναι άδειος:  
Graphical user interface, application

Description automatically generated

Σε αυτό το σημείο πρέπει να δημιουργήσουμε τη σχέση μεταξύ των πινάκων follow και user. Αυτό θα το κάνουμε με έναν follows controller.

Δημιουργούμε τον follows controller με την εξής εντολή:  
$ rails generate controller follows  
Text

Description automatically generated

Τροποποιούμε το αρχείο “app/controllers/follows\_controller.rb” προσθέτοντας λειτουργία follow/unfollow:  
Text

Description automatically generated

Τροποποιούμε το αρχείο “app/models/follow.rb” προσθέτοντας τη γραμμή  
$ resources :follows  
A screenshot of a computer

Description automatically generated with medium confidence

Τροποποιούμε το αρχείο “app/models/follow.rb” προσθέτοντας τη γραμμή  
$ validates :follower\_id, uniqueness: {scope: :followee\_id}  
Graphical user interface, text

Description automatically generated

Δημιουργούμε το αρχίο “/app/views/follows/index.html.haml” για προβολή όλων των χρηστών που μπορεί να ακολουθήσει ένας χρήστης:  
Text

Description automatically generated

Τροποποιούμε το αρχείο “/app/views/users/show.html.haml” για να προσθέσουμε:  
- Κουμπί για εμφάνιση εγγεγραμμένων χρηστών  
- Εμφάνιση φωτογραφιών των ατόμων που ακολουθεί ο χρήστης (folowees)