



Bomshika and TestAccount like this

Rails Project -

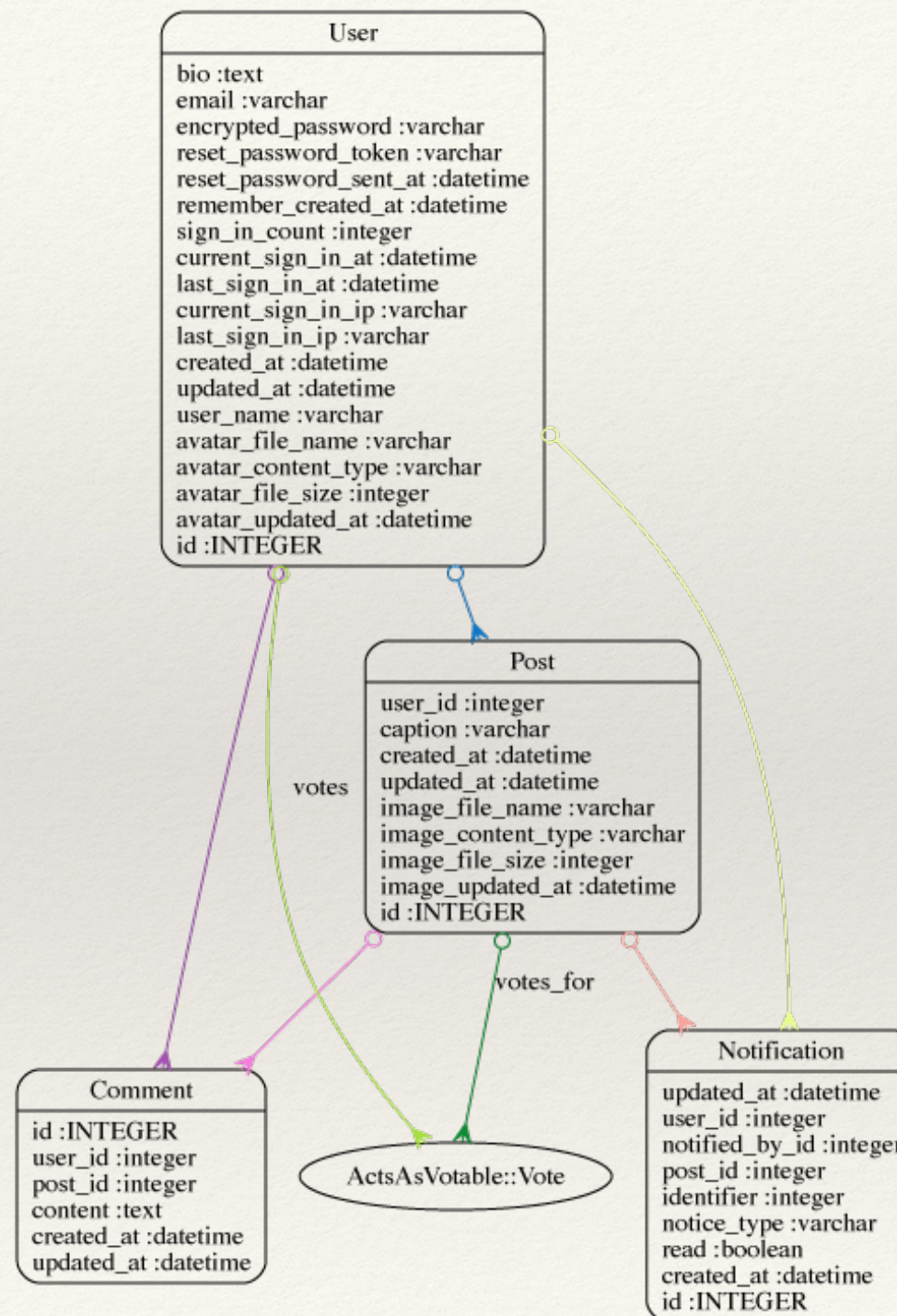
Pictogram

Michael Taylor
20064376
BSc (Hons) Multimedia
Application Development
Web Server Programming

Pictogram

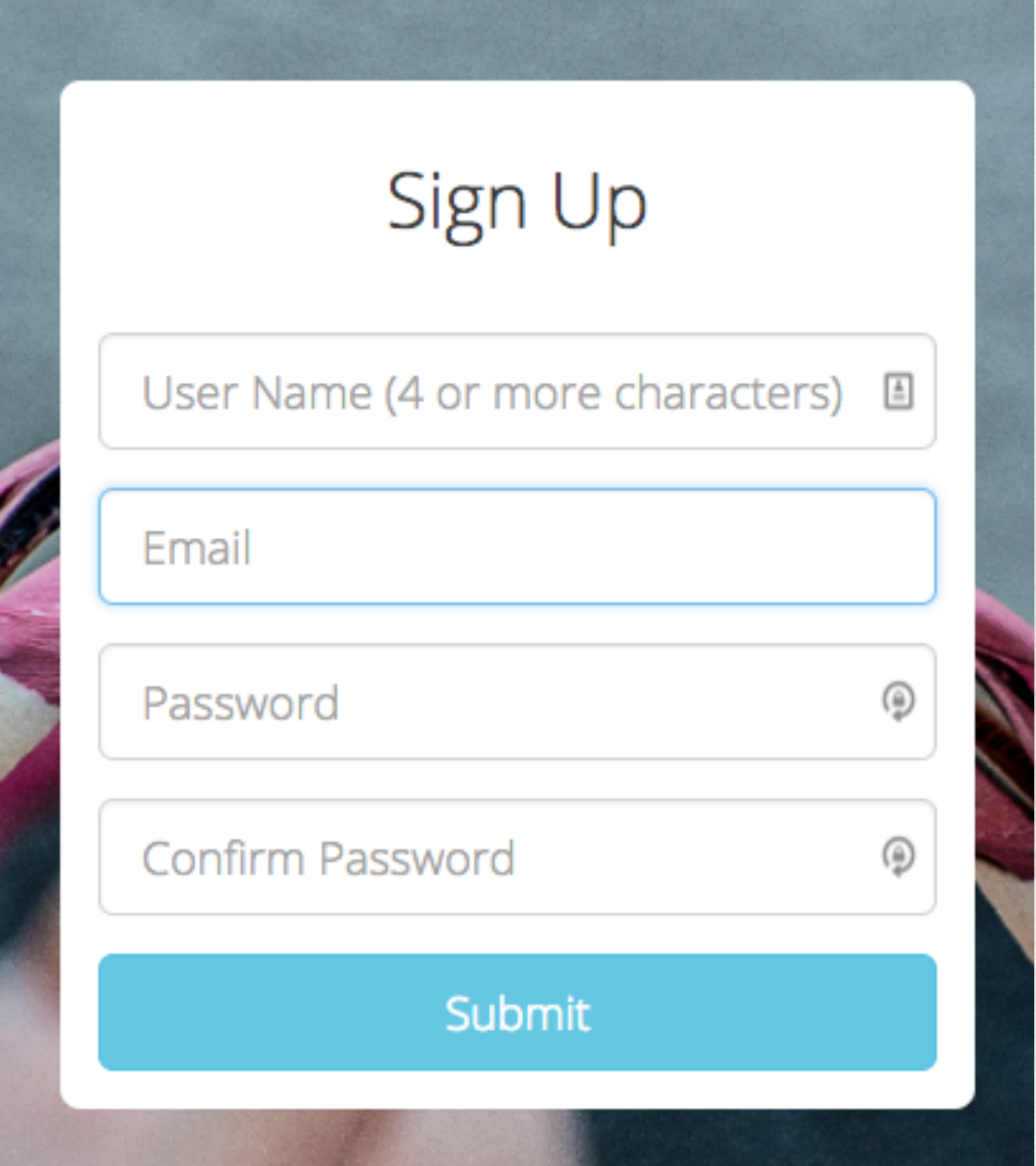
- ❖ Pictogram is an Instagram style application
- ❖ A user can create an account, upload pictures, comment and like their own posts as well as other pictogram users. Other users will be notified when one of their posts is commented on
- ❖ Before a user can view content on pictogram, they have to either login to their account or create an account
- ❖ A user can update posts, captions or if unsatisfied, delete the post
- ❖ Link to hosted application <http://178.62.123.29/>

Data Model




Sign up


- ❖ A user can choose a unique username, must be between 4 and 20 characters
- ❖ Email must be valid and contain @
- ❖ Their password must be at least 6 characters
- ❖ If a user fails to enter in the correct information, upon clicking submit, they will be prompted with error checks for each wrong field




Sign Up

User Name (4 or more characters) 

Email

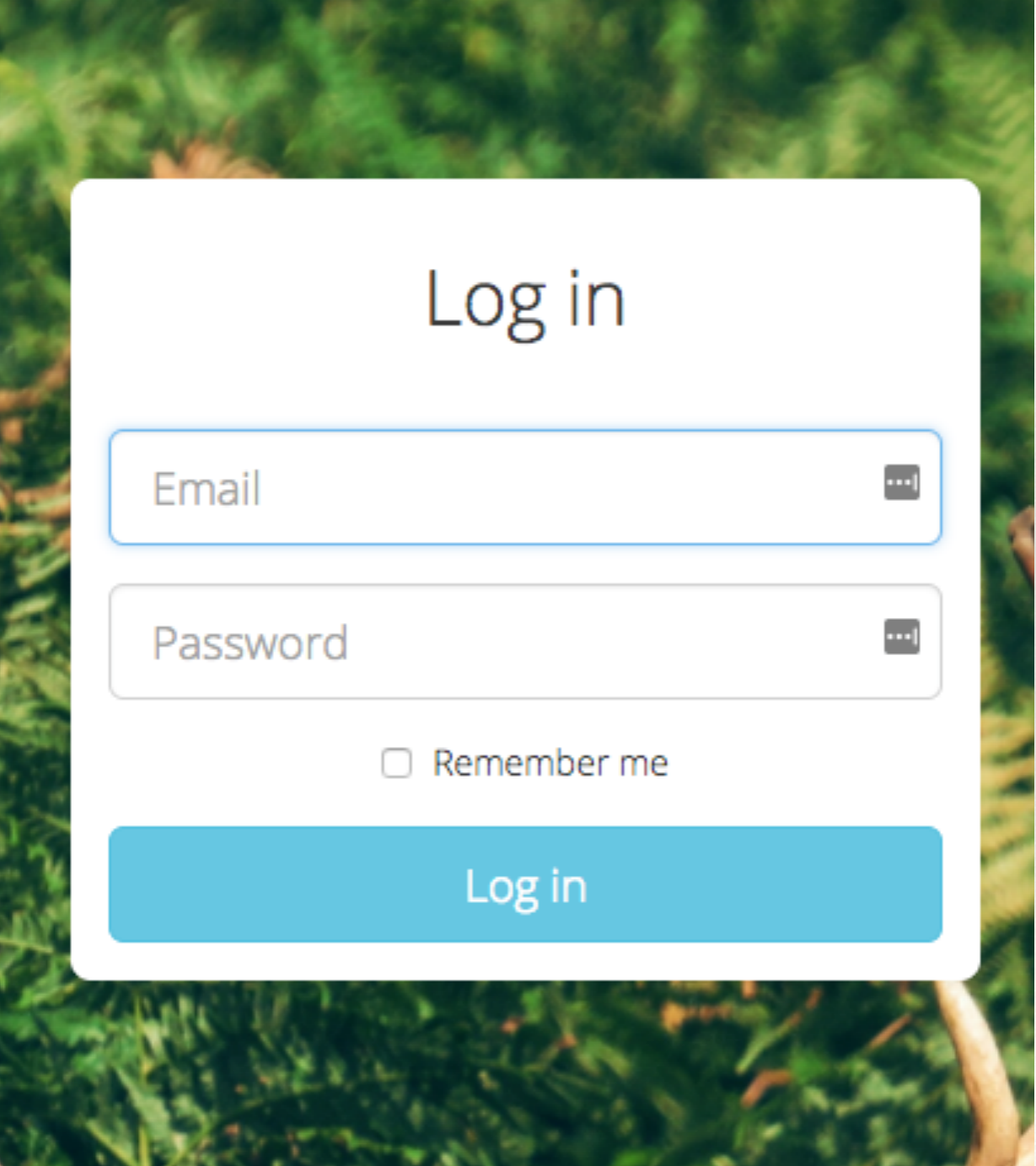
Password 

Confirm Password 

Submit

Login

- ❖ If a user has logged out, they can sign into their account
- ❖ If a user enters in the wrong information, a flash message will be displayed informing them of a wrong email / password combination



Log in

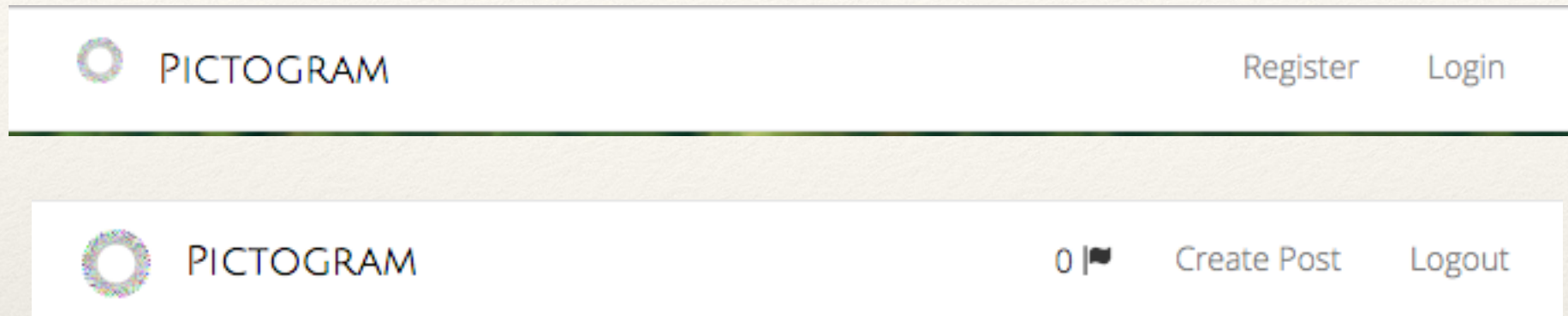
Email

Password

☐ Remember me

Log in

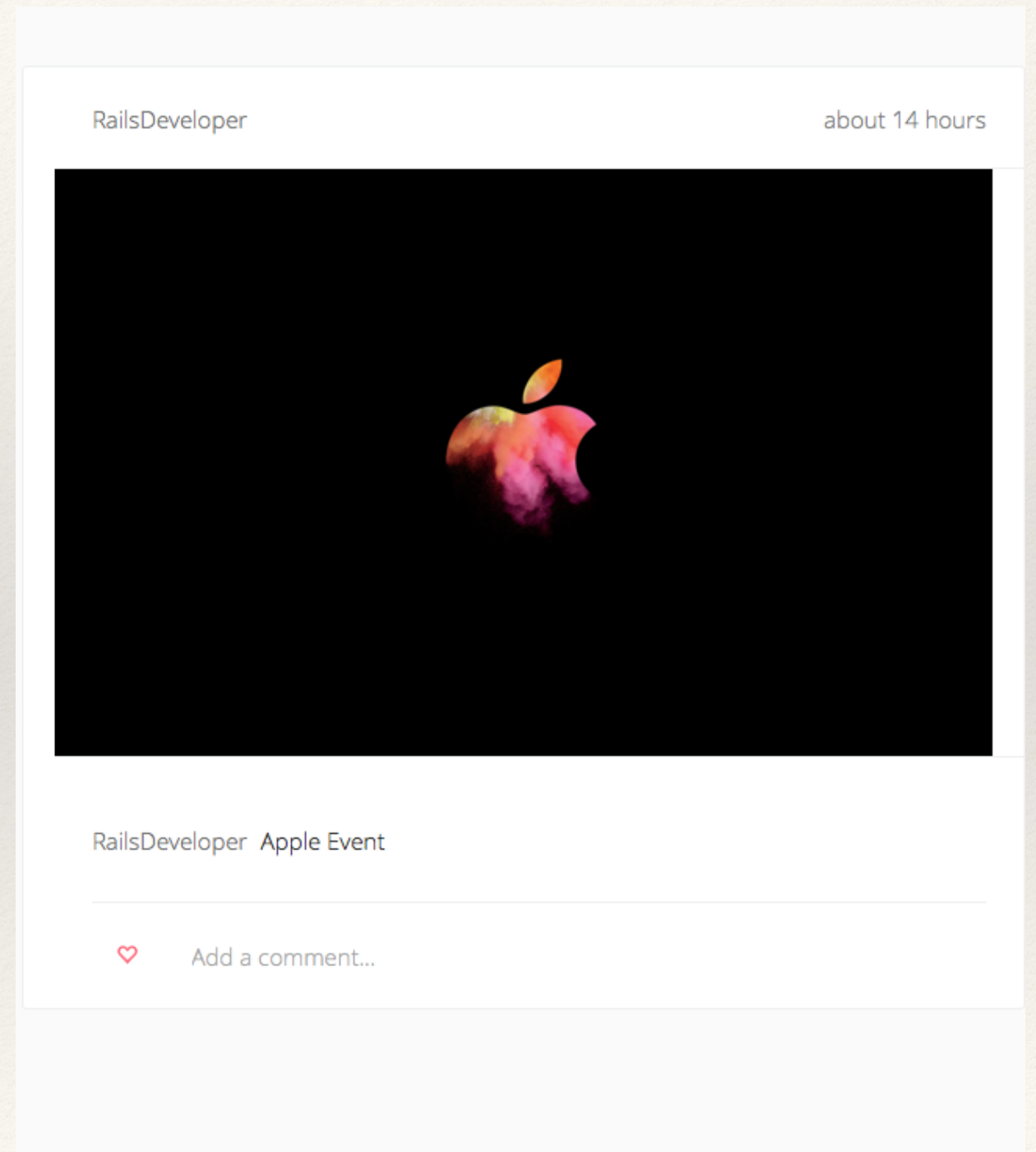
Navigation



- ❖ Pictogram has a clean and simple navigation
- ❖ The navigation is also dynamic, the first picture illustrates the navigation when not logged in
- ❖ When a user is logged in, the navigation will change to suit their needs better

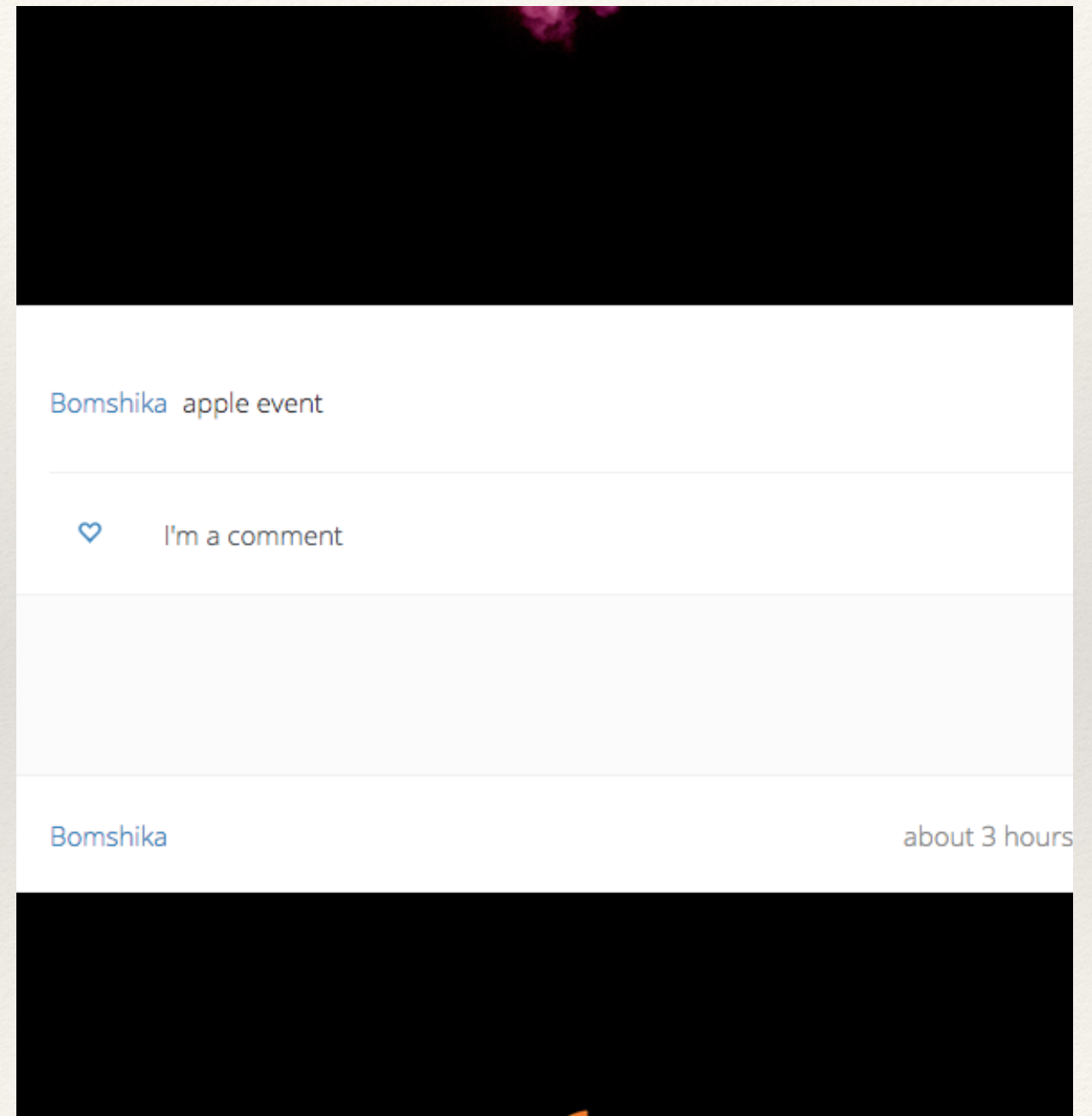
Feed

- ❖ When a user is signed in, they are directed to the main page called their feed
- ❖ The feed is made up of other pictogram users pictures
- ❖ the feed will show who created a post, roughly when they did, the image, and caption associated with the post



Comments

- ❖ A user can comment on pictures in their feed as demonstrated
- ❖ If a user spots a spelling error or decided their comment is not witty enough, they can delete their comment
- ❖ Using pagination, only the top four comments are shown, clicking on “more comments” will display all the comments on a post



Likes

- ❖ A user can also like their own post or another users
- ❖ The heart is animated, when a post is not liked, the heart is empty and when a post is liked, the heart is filled



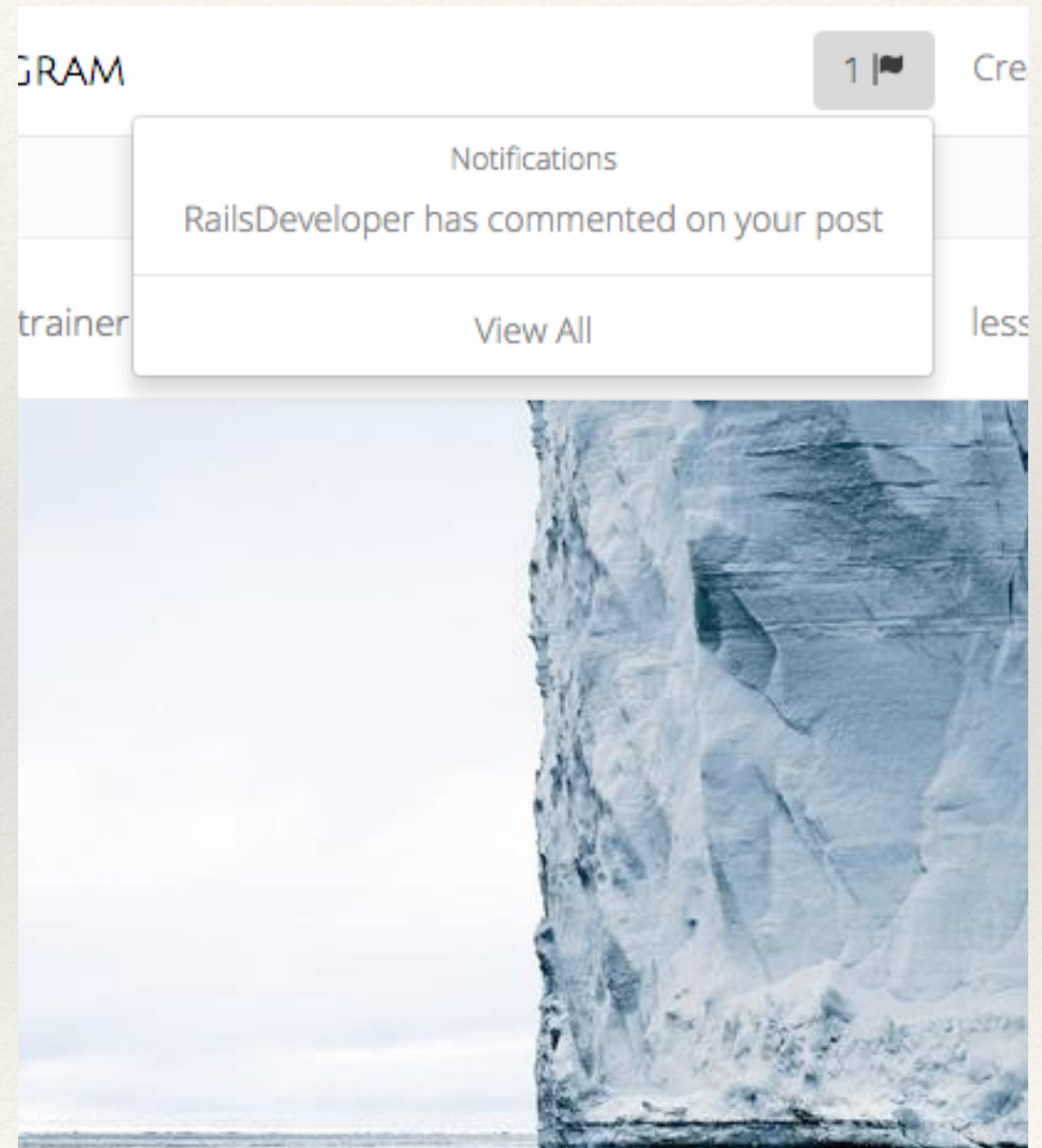
RailsDeveloper likes this
pokemontrainer iceberg
RailsDeveloper Such a great contrast



Add a comment...

Notifications

- ❖ If another comments on your post, the next time you log in you will receive a notification
- ❖ The notifications are clickable and will take you to the post that was commented on



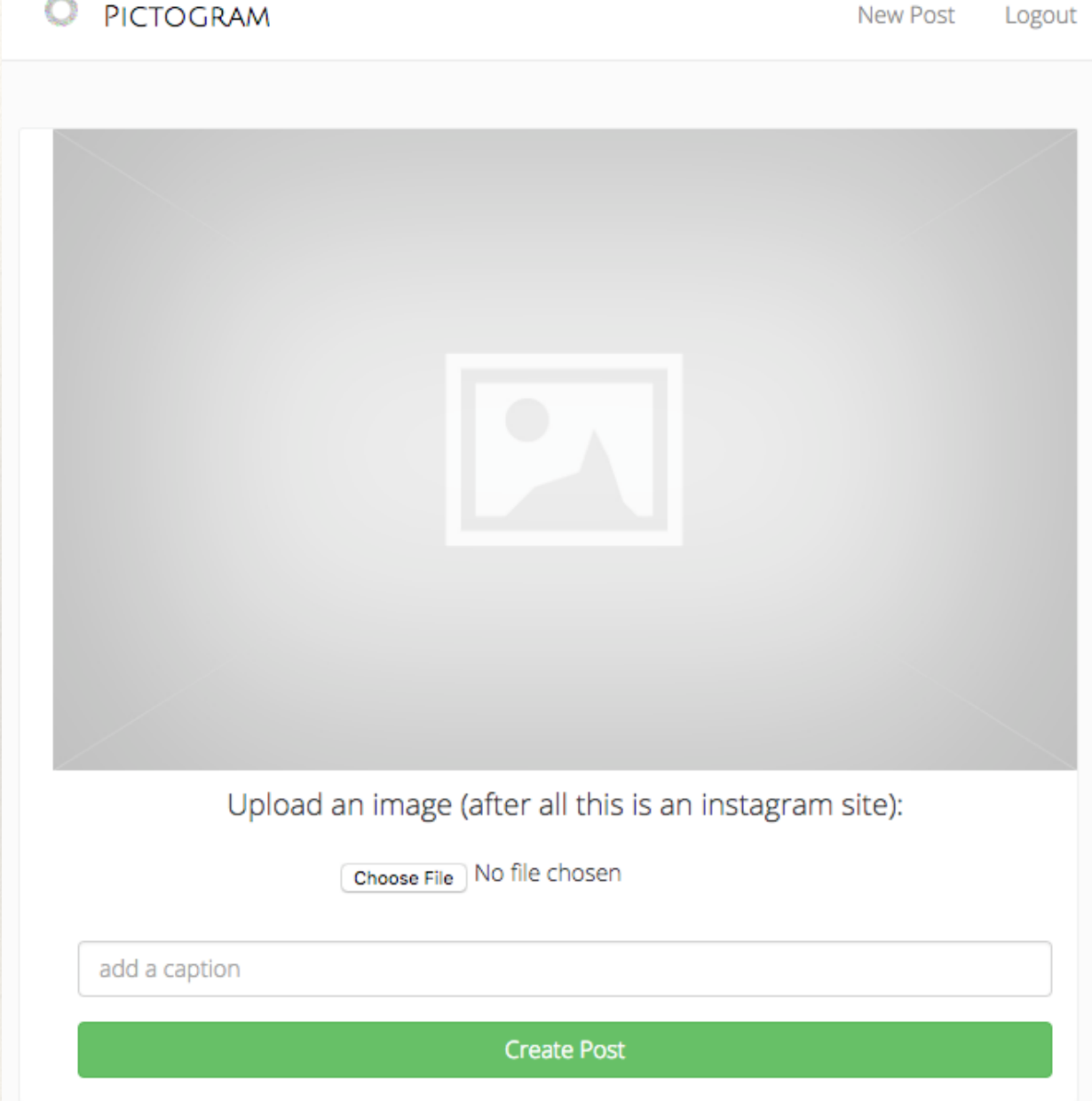
Pagination

- ❖ Using pagination, the number of posts per page is limited
- ❖ To see more posts the user has to click the “load more” button



Upload image

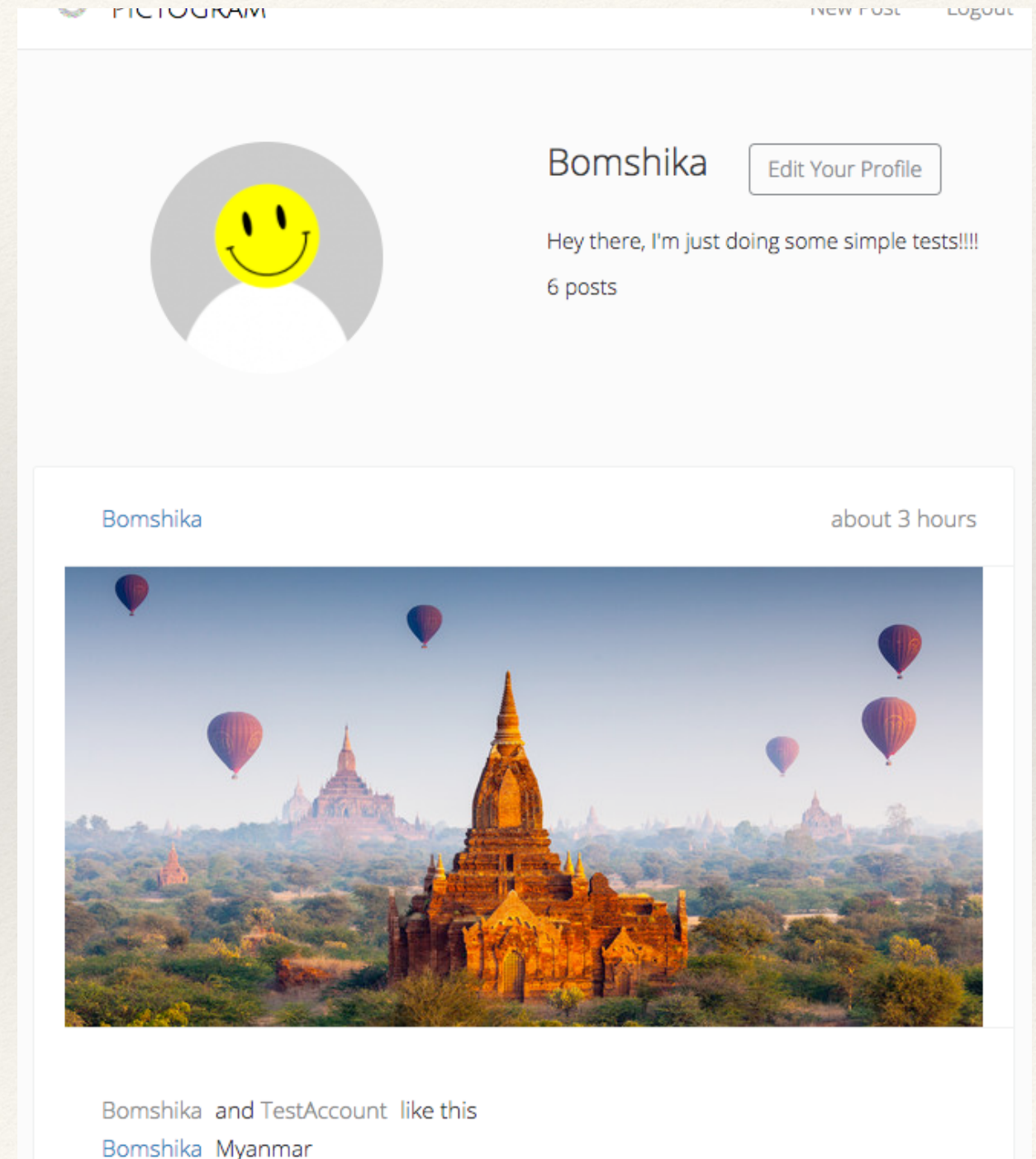
- ❖ As pictogram is an Instagram style application, one of the main focuses is images
- ❖ A user can upload an image from their computer, and add a caption
- ❖ On the upload page, the user is shown the image illustrated. The page has a placeholder image, and shows them where to add a caption



The screenshot shows the 'PICTOGRAM' web application interface for uploading a post. At the top, there is a navigation bar with the 'PICTOGRAM' logo on the left, and 'New Post' and 'Logout' links on the right. The main content area features a large, light gray square placeholder for an image, with a white icon of a picture and a mountain peak in the center. Below the placeholder, the text 'Upload an image (after all this is an instagram site):' is displayed. Underneath this text is a file selection interface consisting of a button labeled 'Choose File' and the text 'No file chosen'. Below the file selection is a text input field with the placeholder text 'add a caption'. At the bottom of the form is a prominent green button labeled 'Create Post'.

Profile

- ❖ Each user has their own profile page
- ❖ This page will contain all of their uploaded posts
- ❖ A user can edit their profile with a bio section and change their user image



Deployment

- ❖ For the hosting of their application, the student is using DigitalOcean.
- ❖ The student had to install imagemagik on the DigitalOcean server so that a user could upload an image.
- ❖ Nginx default settings wouldn't allow images over a certain size to be uploaded, the student changed this to allow images up to 10mb to be uploaded.

Independent learning

- ❖ For image uploading, the student used *gem paperclip*. This was an easy method to attach files to the ActiveRecord. The files aren't saved to their final location on disk but rather saved to the filesystem and can then be referenced by the browser.
- ❖ Instead of creating forms by hand, the student used *gem simple_form*, this allowed the student more flexibility with their forms.
- ❖ The student used *gem devise* for the authentication, devise allowed them to hash the users password before storing it in the database. Devise allowed for authentication to occur using POST requests.
- ❖ The student used *gem acts_as_votable*, to create the "this user has liked this post". This gem allows for any model to be voted on, or liked in this case. Using this gem, they student could show the first few likes, and list the remainder as "8 more people liked this".
- ❖ For pagination, the student used *gem kaminari*. This gem both for the posts and comments sections. The student changed the default configuration of this gem from displaying 25 items to 8 items.
- ❖ As the student is using DigitalOcean, they had to convert the application from local hosting and sql database. To do this the student used *gem unicorn* as the HTTP server and and *gem pg* for the postgresSQL interface for the database.

References

- ❖ The student followed this tutorial loosely for their application: <https://www.devwalks.com/lets-build-instagram-in-rails-part-1/>
- ❖ Rails Application DigitalOcean deployment: <https://www.digitalocean.com/community/tutorials/how-to-use-the-ruby-on-rails-one-click-application-on-digitalocean>
- ❖ Github Repo: <https://github.com/MikeZTaylor/pictogram>