

Qin Ying Chen, Erica Chou, Moira Duya March 31, 2020 Databases

GitHub Link: <https://github.com/Faitsu/Databases-Project>

Short Description: This project will have three group members working on five required features and six optional features for an app called “Finstagram” using Python/Flask. This app will require us to allow users to share photos with either all of their followers or with a specific group.

These are the required features for our project:

1. View visible photos: Finstagram shows the user the photoID of each photo that is visible to her, arranged in reverse chronological order.
2. View further photo info: Display the following for visible photos (You may include this with the results of “view visible photos” or you may supply a different way for users to see this additional info, such as clicking on additional links). a) Display the photo or include a link to display it. (If you have trouble actually displaying the photo, just display the photo’s pID for close to total credit for this part). b) The firstName and lastName of the photoPoster c) The postingDate, d) the usernames, first names and last names of people who have been tagged in the photo (taggees), provided that they have accepted the tags (Tag.acceptedTag == true) e) The usernames of people who have ReactedTo the photo and the emoji and/or comment they gave it
3. Post a photo: Finstagram prompts the User to enter

a) the location of a photo on the client computer,

b) a designation of whether the photo is visible to all followers (allFollowers == true) or only to members of designated FriendGroups (allFollowers == false). In response to the data submitted by the user, Finstagram

c) inserts data about the photo (including current time*, and the current user as photoPoster) and the value of allFollowers into the Photo table. *Finstagram can find the current time with an SQL function or a function in the host language.

d) either copies the photo to a dedicated folder with file name that includes the pID and stores this location as the filePath or it stores the photo as a BLOB datatype. (More details will be provided on how to do this).

e) gives the user a way to designate FriendGroups that the user belongs to with which the Photo is shared.

4. Manage Follows: a) User enters the username of someone they want to follow. Finstagram adds an appropriate tuple to Follow, with `acceptedFollow == False`. b) Finstagram displays list of requests others have made to follow the current user and the user has the opportunity to accept, by setting `acceptFollow` to `True` or to decline by deleting the request from the Follow table.

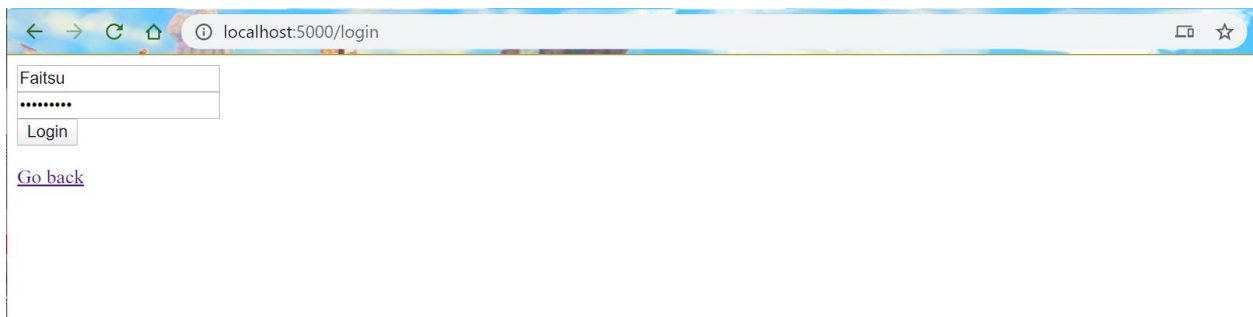
5. Add friendGroup:

- a) User provides a name for the friendGroup.
- b) Finstagram creates the group with current user as the groupOwner, provided that they don't already own a group with this name. Finstagram gives a meaningful error message if the current user already has a group with this name.
- c) If the group is created successfully, Finstagram adds the current user as a member of the group.

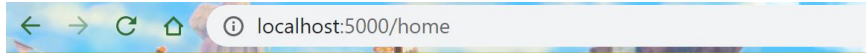
*For Part 3, the parts that were done are indicated by the highlight.

Screenshots of our work so far:

Login Page:



Home Page:



Welcome Faitsu

POST SOMETHING! :D

Choose a profile picture (JPEG/JPG/PNG): No file chosen

Pick who you want to be able to view your photo:

- ☐ All followers
- ☐ Members of designated friend groups

Select which friend group you would like to share your photo with:

Choose friend group(s):

Pikachu

Hold down the Ctrl (windows) or Command (Mac) button to select multiple options.

CREATE A GROUP!

Rest of the page:

localhost:5000/home
Choose friend group(s):

Pikachu

Hold down the Ctrl (windows) or Command (Mac) button to select multiple options.

Post

CREATE A GROUP!

Group Name:

Enter group name

Group Description:

Enter description

Make Group

YOUR POSTS:

pID	ImagePath	Caption	Posting Date	Who Can View
4	/uploads/Tanjiro.jpg	<3	2020-04-07 00:00:26	Visible to members of designated friend groups
3	/uploads/Dimitri.jpg	Dimitri is Cute	2020-04-06 23:59:12	Visible to all followers

YOUR GROUPS:

Group Name	Description of group
Pikachu	Pikachus are cute

GROUPS YOU ARE IN:

Group Name	Creator of Group
Pikachu	Faitsu

[Logout](#)

Because our group has three people, we have chosen these six features to add in to our app for Project 4:

Qin Ying Chen:

Manage tags:

1. Current user, whom we'll call x , selects a photo that is visible to her and proposes to tag it with username y (1) If the user is self-tagging ($y = x$), Finstagram adds a row to the Tag table: (x , photoID, true) (2) else if the photo is visible to y , Finstagram adds a row to the Tag table: (y , photoID, false) (3) else if photo is not visible to y , Finstagram doesn't change the tag table and prints some message saying that she cannot propose this tag.

2. Finstagram shows the user relevant data about photos that have proposed tags of this user (i.e. user's username is Tag.username and acceptedTag is false.) User can choose to accept a tag (change acceptedTag to true), decline a tag (remove the tag from Tag table), or not make a decision (leave the proposed tag in the table with acceptedTag == false.)

Erica Chou:

- 1)React to a photo: Users can add comments and/or emojis to photos that are visible to them.
- 2)Unfollow: Think about what should be done when someone is unfollowed, including some reasonable approach to tags that they posted by virtue of being a follower. Write a short summary of how you're handling this situation. Implement it and test it.

Moiria Duya:

- 1)Search by tag:Search for photos that are visible to the user and that tag some particular person (the user or someone else)
- 2)Search by poster:Search for photos that are visible to the user and that were posted by some particular person (the user or someone else)