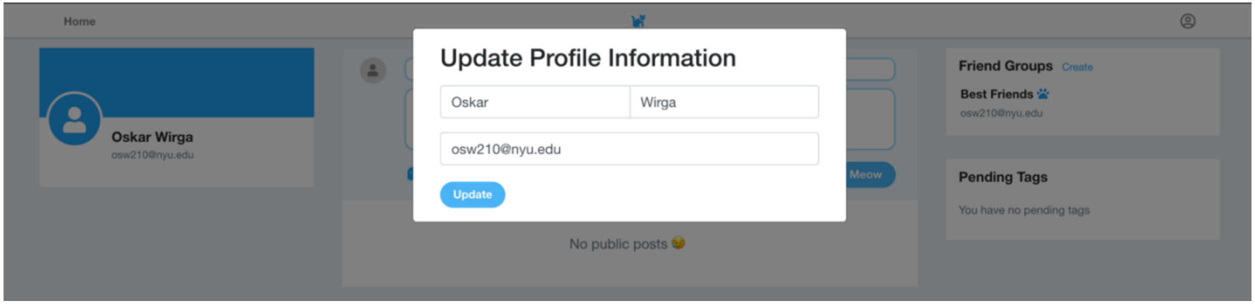
a. Oskar Wirga
b. Updating User Profile Information: Users open a modal in order to update their name or email information. A user can update one or all fields and so long as the email is not taken, they are able to change it.
c. Databases should be able to respond to change, a static primary key just is not realistic in content sharing platforms, people get new emails and so updating an email in the database should cascade through the entire database.
d. A clear explanation of any changes to the database schema that are needed (including additional tables, additional attributes, or additional constraints) Throughout the entire database, where ever there was a foreign key reference to the Person table's email primary key, I had to add an ON UPDATE CASCADE in order to reflect the change throughout the entire database if someone changes their email.
e. The queries (and any other SQL statements) used in your implementation of the feature
The first query ran is: SELECT * FROM Person WHERE email = new_email. New_email in this case is the new email the user wants to change to, but if this query returns a row, a user already exists with that email so we error out. The next query ran is UPDATE Person SET ' + parameters + ' WHERE email=%s. parameters is a string of the changed parameters that are joined by a comma in python. This allows me to dynamically update the Person's attributes without having to run individual UPDATE queries for each attribute.
f. db.py – Lines 343-363 What this code does is checks if there are any parameters that need to be updated. Then makes a query to check if there are any users with the email that the user is trying to change to. Then, the parameters string is made by joining the list of fields that are to be updated. This is done by using inline if statements alongside filtering None and finally joining with a comma. Then, I filter the actual variables by none in order to get the corresponding values. Then it is executed and the values update.
app.py – Lines 319-347 To edit the profile, we get the values from the form which are sent in the post request. Then I get the user information using the session['email'] key. Then I set the new_<attribute> variables by comparing them with the old ones. We set it to the new one if they are different, otherwise they are set to None. We then update the user with those parameters and based on the status code, set the session attributes to the updated version.
g. Before

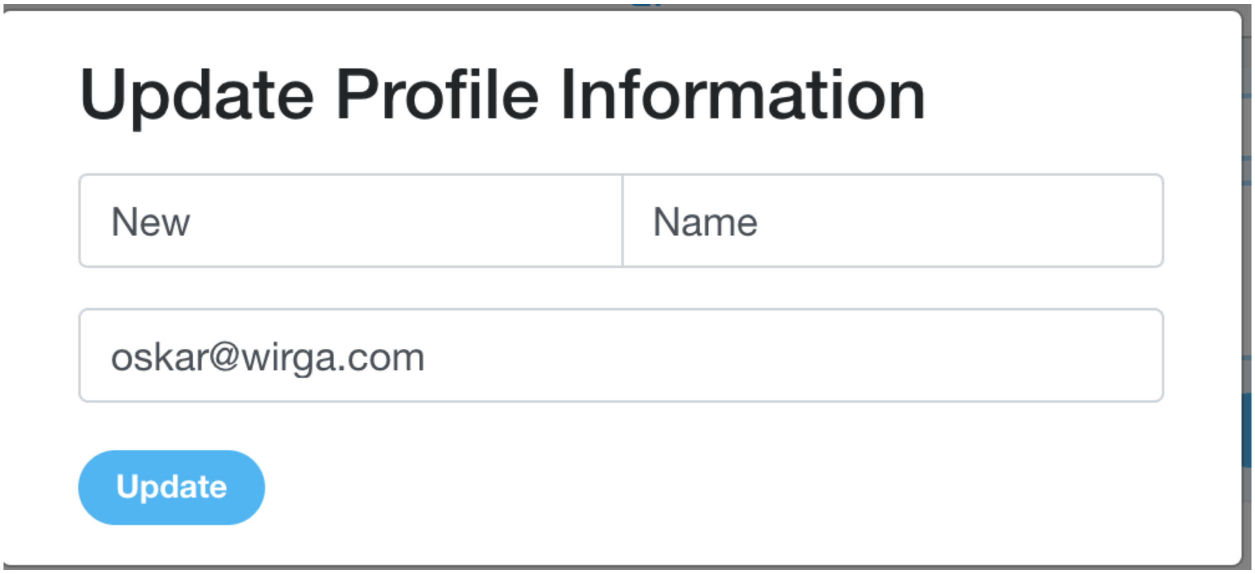


Update Profile Information

Oskar | Wirga

osw210@nyu.edu

New Info

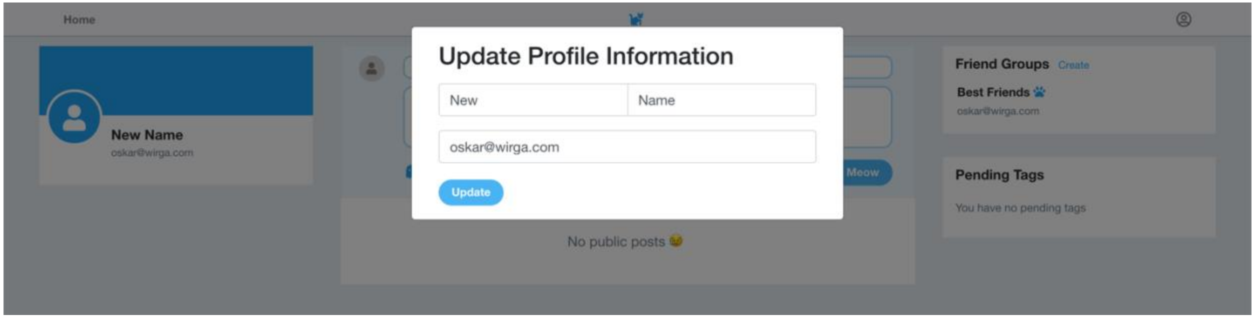


Update Profile Information

New | Name

oskar@wirga.com

Updated information



New Name
oskar@wirga.com

Update Profile Information

New | Name

oskar@wirga.com

No public posts 😆

Friend Groups

Best Friends 🐾
oskar@wirga.com

Pending Tags

You have no pending tags