

Names: Ikra Rashid, Keaton Smith, Antoine Assaf, Jahlil Walker

Additional Features:

Feature: Remove user from group

Name: Jahlil

Why it is a good feature: A user can remove another user from a group. This will allow users to be in control of their friend group space.

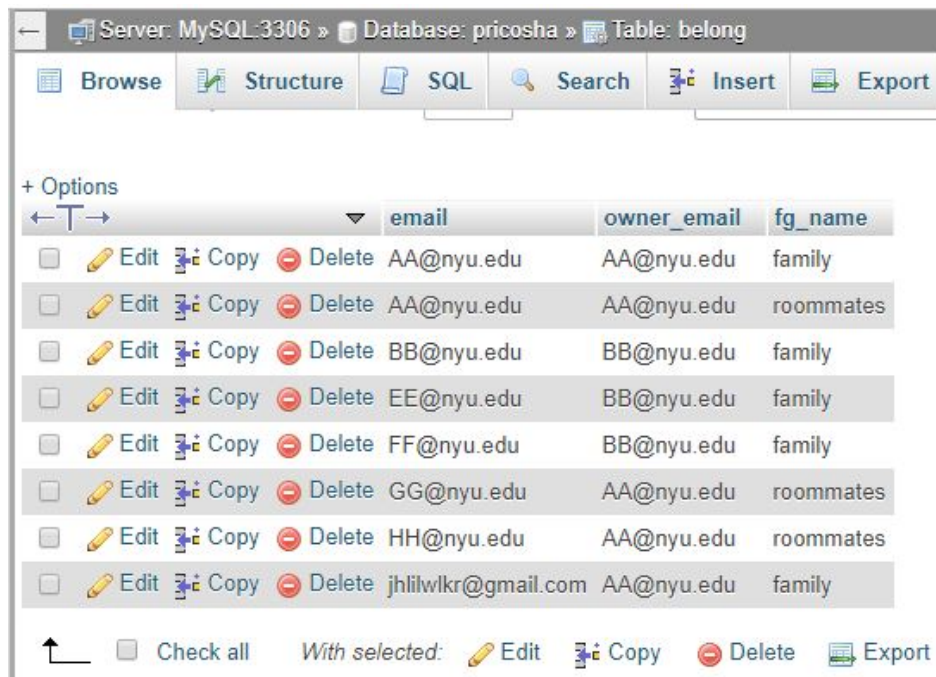
Changes to database: No changes to the database schema.

Queries:

```
170
171 #removes a user from a group
172 @app.route('/old_user', methods=['GET', 'POST'])
173 def delete_person():
174
175     cursor = conn.cursor();
176     groupName = request.form['fg_name']
177     ownerEmail = request.form['owner_email']
178     userEmail = request.form['user_email']
179     query = 'DELETE FROM belong WHERE email = %s AND owner_email = %s AND fg_name = %s'
180     cursor.execute(query, (userEmail,ownerEmail,groupName))
181     conn.commit()
182     cursor.close()
183     return redirect(url_for('home'))
```

Source code destination: Located on line 172 in file main.py and the HTML code is in file old_user.html

Screenshots:



	email	owner_email	fg_name
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	AA@nyu.edu	AA@nyu.edu	family
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	AA@nyu.edu	AA@nyu.edu	roommates
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	BB@nyu.edu	BB@nyu.edu	family
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	EE@nyu.edu	BB@nyu.edu	family
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	FF@nyu.edu	BB@nyu.edu	family
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	GG@nyu.edu	AA@nyu.edu	roommates
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	HH@nyu.edu	AA@nyu.edu	roommates
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	jhlilwlkr@gmail.com	AA@nyu.edu	family

← → ↻ ⓘ 127.0.0.1:5000/home

Welcome AA@nyu.edu

[Post](#)

Posts:

ID	Time	Post	Path	Poster
13	2018-12-07 05:09:39	TestPost	Test/Path	jhlilwlkr@gmail.com
10	2018-12-07 00:33:38	test	ign	WW@nyu.edu

[My Posts](#)

[My FriendGroups](#)

[Create Friendgroup](#)

[Manage Tags](#)

[Logout](#)

← → ↻ ⓘ 127.0.0.1:5000/my_groups

Here are the groups the you own:

Owner	Groupname
AA@nyu.edu	family
AA@nyu.edu	roommates

[Add user to group](#)

[Leave a group](#)

[Remove user from group](#)

[Go Back](#)

[Logout](#)

← → ↻ ⓘ 127.0.0.1:5000/old_friend

family
AA@nyu.edu
EE@nyu.edu
<input type="button" value="Submit"/>

[Go back](#)

Server: MySQL:3306 » Database: pricosha » Table: belong

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Exp](#)

+ Options

		email	owner_email	fg_name
<input type="checkbox"/>	Edit	Copy	Delete	AA@nyu.edu AA@nyu.edu family
<input type="checkbox"/>	Edit	Copy	Delete	AA@nyu.edu AA@nyu.edu roommates
<input type="checkbox"/>	Edit	Copy	Delete	BB@nyu.edu BB@nyu.edu family
<input type="checkbox"/>	Edit	Copy	Delete	EE@nyu.edu AA@nyu.edu family
<input type="checkbox"/>	Edit	Copy	Delete	EE@nyu.edu BB@nyu.edu family
<input type="checkbox"/>	Edit	Copy	Delete	FF@nyu.edu BB@nyu.edu family
<input type="checkbox"/>	Edit	Copy	Delete	GG@nyu.edu AA@nyu.edu roommates
<input type="checkbox"/>	Edit	Copy	Delete	HH@nyu.edu AA@nyu.edu roommates
<input type="checkbox"/>	Edit	Copy	Delete	jhlilwkr@gmail.com AA@nyu.edu family

Feature: View Own Posts

Name: Keaton

Why it is a good feature: This feature allows for users to view all of their own posts, whether or not public or private. This allows for the user to access their own data whenever they wish, as compared to how they were initially only able to see all public posts within the last 24 hours.

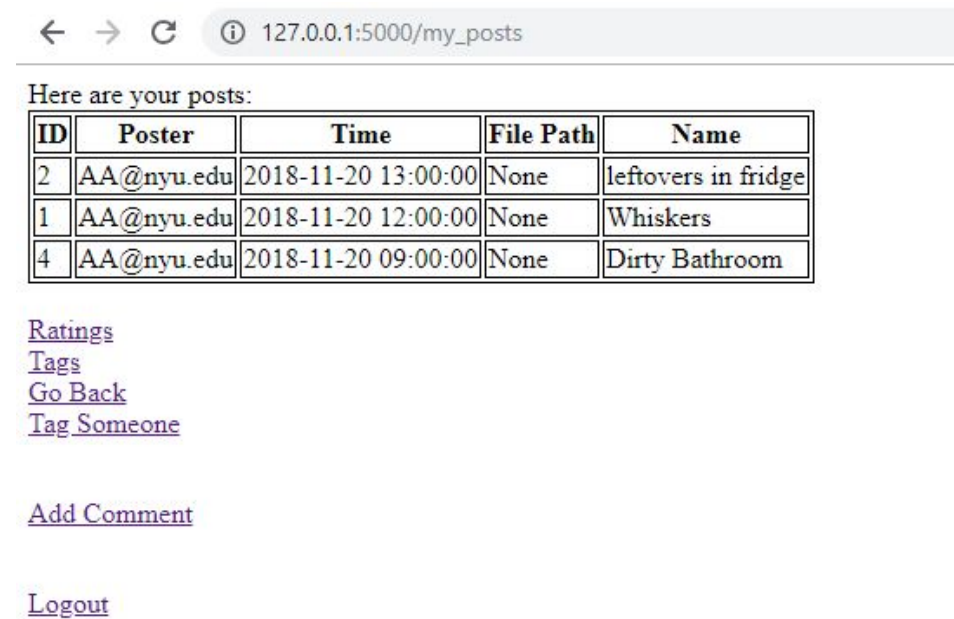
Changes to database: No changes to the database schema.

Queries:

query = 'SELECT * FROM contentitem NATURAL LEFT JOIN share WHERE email_post = %s
OR item_id IN (SELECT item_id FROM share NATURAL JOIN belong WHERE email = %s)
ORDER BY post_time DESC'

Source code destination: The query and the code associated with the query above can be found in the main.py file on line 196.

Screenshots:



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/my_posts". Below the address bar, the text "Here are your posts:" is followed by a table with five columns: ID, Poster, Time, File Path, and Name. The table contains three rows of data. Below the table, there are four links: Ratings, Tags, Go Back, and Tag Someone. Further down, there are two more links: Add Comment and Logout.

ID	Poster	Time	File Path	Name
2	AA@nyu.edu	2018-11-20 13:00:00	None	leftovers in fridge
1	AA@nyu.edu	2018-11-20 12:00:00	None	Whiskers
4	AA@nyu.edu	2018-11-20 09:00:00	None	Dirty Bathroom

[Ratings](#)
[Tags](#)
[Go Back](#)
[Tag Someone](#)

[Add Comment](#)

[Logout](#)

Feature: Create Group

Name: Jahlil

Why it is a good feature: This allows for a user to make a group of their own so that they aren't only being added to set groups. This will allow for the application to be more flexible and connect people together.

Changes to database: No changes to the database schema

Queries: query = 'INSERT INTO friendgroup (owner_email,fg_name) VALUES(%s,%s)'

Source code destination: The query and the code associated with the query above can be found in the main.py file on line 186. HTML code is in file new_group.html

Screenshots:

Server: MySQL:3306 » Database: pricosha » Table: friendgroup










Browse Structure SQL Search Insert Edit

✓ Showing rows 0 - 4 (5 total, Query took 0.0008 seconds.)

```
SELECT * FROM `friendgroup`
```

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

				owner_email	fg_name	description
<input type="checkbox"/>	 Edit	 Copy	 Delete	AA@nyu.edu	family	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	BB@nyu.edu	family	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	AA@nyu.edu	roommates	NULL

127.0.0.1:5000/home

Welcome AA@nyu.edu

[Post](#)

Posts:

ID	Time	Post	Path	Poster
13	2018-12-07 05:09:39	TestPost	Test/Path	jhlilwlkr@gmail.com
10	2018-12-07 00:33:38	test	ign	WW@nyu.edu

[My Posts](#)

[My FriendGroups](#)

[Create Friendgroup](#)

[Manage Tags](#)

[Logout](#)

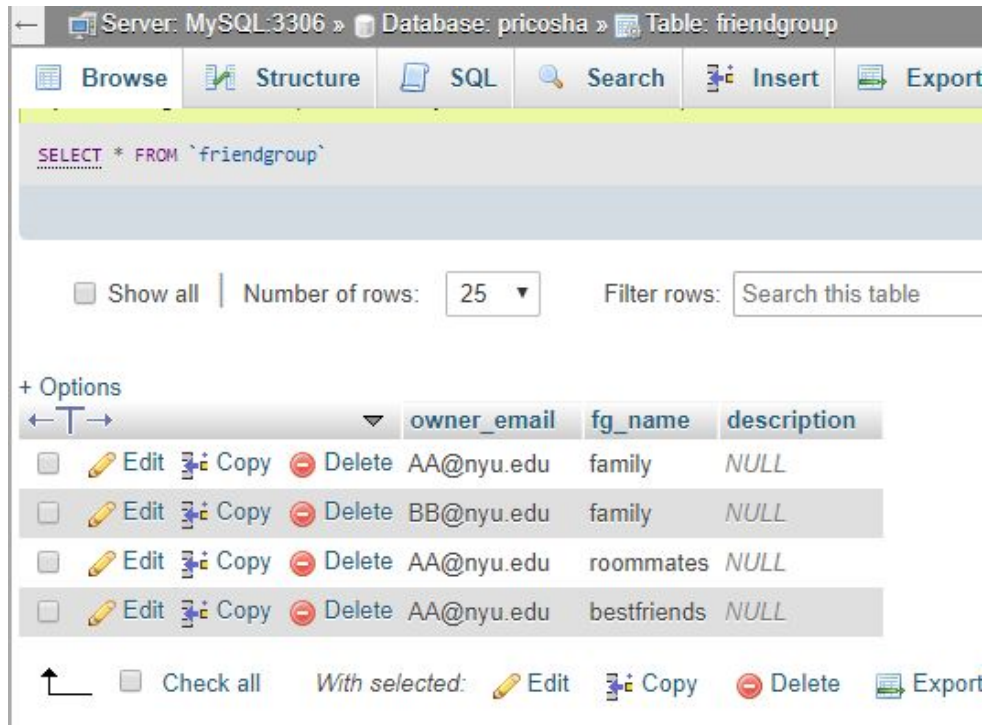
127.0.0.1:5000/create_group

What is the name of your new group?

bestfriends

Submit

[Go back](#)



Feature: Leave Group

Name: Ikra

Why it is a good feature: This feature allows for a user to have the ability to leave a group. This way, they are not required to stay in a group they no longer want to be in, and don't have to rely on the owner to drop them.

Changes to the database: No changes to the database schema

Queries:

```

236 @app.route('/leave_group', methods = ["GET", "POST"])
237 def leave_group():
238     email = session['email']
239     cursor = conn.cursor();
240     friendgroup_name = request.args.get('friendgroup_name')
241     query = 'DELETE FROM belong WHERE email = %s AND fg_name = %s'
242     cursor.execute(query, (email, friendgroup_name))
243     data = cursor.fetchall()
244     cursor.close()
245     return render_template('leave_group.html', posts = data)
246

```

Source code destination: The query and the code associated with the query can be found on line 236 in file main.py and the HTML code for leaving the group can be found in leave_group.html.

Screenshots:



Welcome HH@nyu.edu

[Post](#)

Posts:

ID	Time	Post	Path	Poster
13	2018-12-07 05:09:39	TestPost	Test/Path	jhlilwlkr@gmail.com
10	2018-12-07 00:33:38	test	ign	WW@nyu.edu

[My Posts](#)

[My FriendGroups](#)

[Create Friendgroup](#)

[Manage Tags](#)

[Logout](#)

Server: MySQL:3306 » Database: pricosha » Table: belong

Browse Structure SQL Search Insert Export

+ Options

		email	owner_email	fg_name
<input type="checkbox"/>	Edit Copy Delete	AA@nyu.edu	AA@nyu.edu	family
<input type="checkbox"/>	Edit Copy Delete	AA@nyu.edu	AA@nyu.edu	roommates
<input type="checkbox"/>	Edit Copy Delete	BB@nyu.edu	BB@nyu.edu	family
<input type="checkbox"/>	Edit Copy Delete	EE@nyu.edu	BB@nyu.edu	family
<input type="checkbox"/>	Edit Copy Delete	FF@nyu.edu	BB@nyu.edu	family
<input type="checkbox"/>	Edit Copy Delete	GG@nyu.edu	AA@nyu.edu	roommates
<input type="checkbox"/>	Edit Copy Delete	HH@nyu.edu	AA@nyu.edu	roommates
<input type="checkbox"/>	Edit Copy Delete	jhlilwkr@gmail.com	AA@nyu.edu	family

← → ↻ ⓘ 127.0.0.1:5000/leave_group

What group do you want to leave?

roommates

Submit

[Go back](#)

Server: MySQL:3306 » Database: pricosha » Table: belong

Browse Structure SQL Search Insert Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

		email	owner_email	fg_name
<input type="checkbox"/>	Edit Copy Delete	AA@nyu.edu	AA@nyu.edu	family
<input type="checkbox"/>	Edit Copy Delete	AA@nyu.edu	AA@nyu.edu	roommates
<input type="checkbox"/>	Edit Copy Delete	BB@nyu.edu	BB@nyu.edu	family
<input type="checkbox"/>	Edit Copy Delete	EE@nyu.edu	BB@nyu.edu	family
<input type="checkbox"/>	Edit Copy Delete	FF@nyu.edu	BB@nyu.edu	family
<input type="checkbox"/>	Edit Copy Delete	GG@nyu.edu	AA@nyu.edu	roommates
<input type="checkbox"/>	Edit Copy Delete	jhlilwkr@gmail.com	AA@nyu.edu	family

⬆️ ☐ Check all | With selected: Edit Copy Delete Export

Feature: Add comments

Name: Antoine

Why it is a good feature: Users can respond and be able to express their thoughts on the contents posted by themselves and other users. It is a good way to promote social interactions via the platform.

Changes to the database:

```
214
215 DROP TABLE IF EXISTS `comments`;
216 CREATE TABLE IF NOT EXISTS `comments` (
217     `commentor` varchar(20) NOT NULL,
218     `comment` varchar(25) NOT NULL,
219     `item_id` int(11) NOT NULL,
220     `commenttime` timestamp NULL DEFAULT NULL,
221     PRIMARY KEY (`commentor`,`item_id`)
222     KEY `commentor` (`commentor`),
223     KEY `item_id` (`item_id`)
224 ) ENGINE=MyISAM DEFAULT CHARSET=latin1;
225 COMMIT;
```

Queries:

query = 'SELECT item_id FROM contentitem WHERE item_id IN (SELECT item_id FROM contentitem NATURAL LEFT JOIN share WHERE email_post = %s OR item_id IN (SELECT item_id FROM share NATURAL JOIN belong WHERE email = %s))'

Source code destination: The query and the code associated with the query above can be found in the main.py file on line 215. HTML code is in file create_comments.html

Screenshots:

← → ↻ ⓘ 127.0.0.1:5000/my_posts

Here are your posts:

ID	Poster	Time	File Path	Name
2	AA@nyu.edu	2018-11-20 13:00:00	None	leftovers in fridge
1	AA@nyu.edu	2018-11-20 12:00:00	None	Whiskers
4	AA@nyu.edu	2018-11-20 09:00:00	None	Dirty Bathroom

[Ratings](#)
[Tags](#)
[Go Back](#)
[Tag Someone](#)

[Add Comment](#)

[Logout](#)

← Server: MySQL:3306 » Database: pricosha » Table: comments

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#)

✓ Showing rows 0 - 0 (1 total, Query took 0.0017 seconds.)

```
SELECT * FROM `comments`
```

☐ Show all | Number of rows: 25 ▼ Filter rows: Search this table

+ Options

← T → commentor comment item_id commenttime

↑ ☐ Check all With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

← → ↻ ⓘ 127.0.0.1:5000/add_comment

Add Comment!

2

Excited to eat leftovers!

Submit

[Go back](#)

← Server: MySQL:3306 » Database: pricosha » Table: comments

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#)

✓ Showing rows 0 - 0 (1 total, Query took 0.0008 seconds.)

```
SELECT * FROM `comments`
```

☐ Show all

Number of rows:

25 ▼

Filter rows:

Search this table

+ Options

← T →

▼

commentor

comment

item_id

commenttime

☐

Edit

Copy

Delete

AA@nyu.edu

Excited to eat leftovers!

2

2018-12-07 23:52:56

↑

☐ Check all

With selected:

Edit

Copy

Delete

Export

Feature: View comments

Name: Antoine

Why it is a good feature: This feature allows for a view all comments on any posts they can regularly view. It displays the Commenter's email, the post with which it is associated, the content of the comment, and the time of the comment.

Changes to the database: No additional features from the Add Comments feature.

Queries:

```
SELECT * FROM comments NATURAL LEFT JOIN share WHERE commentor = %s OR  
item_id IN (SELECT item_id FROM share NATURAL JOIN belong WHERE email = %s)  
ORDER BY commenttime DESC
```

Source code destination: The query and the code associated with the query above can be found in the main.py file on line 206. HTML code is in file comments.html

Screenshots:

☐ Show all

Number of rows: 25 ▼

Filter rows:

Sort by

+ Options

				commentor	comment	item_id	commenttime			
<input type="checkbox"/>		Edit		Copy		Delete	AA@nyu.edu	hello	2	NULL
<input type="checkbox"/>		Edit		Copy		Delete	AA@nyu.edu	test	4	2018-12-08 03:09:17



☐ Check all

With selected:



Edit



Copy



Delete



Export

Here are comments on your post:

commentor	ID	Comment	Time
AA@nyu.edu	4	test	2018-12-08 03:09:17
AA@nyu.edu	2	hello	None

[Go Back](#)

[Logout](#)