Link to GitHub repository:

https://github.com/amk960/Finstagram

Date of last commit to repository:

12/10/2019

Number of Team members:

3

Names and netIDs of Team members (one per line):

Asma Khursheed amk960

Shakthi Ganta Gameshwaran sgg329

Crystal Li yl4923

===

1.

Name of the feature

Like Feature

The full name of the team member who is primarily in charge of implementing this feature

Crystal Li

The queries (and any other SQL statements) used in your implementation of the feature. (If there are standard queries used in most or all of your features, you don't have to include them here; just include the SQL statements that do the main work for this feature.)

```
    query = "SELECT username,photoID FROM likes WHERE username=%s AND photoID=%s"
```

To check if the user had already liked the image. If no, insert into the like table their like.

```
   query = "INSERT INTO likes (username, photoID, liketime, rating) VALUES (%s, %s,
%s, %s)"
```

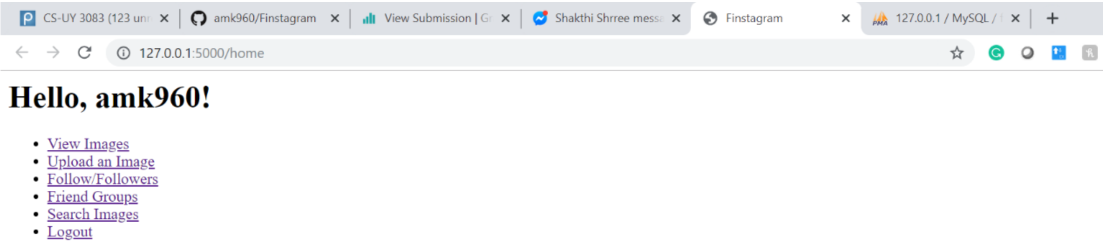
Otherwise delete previous like and insert new like rating.

```
query = "DELETE FROM likes WHERE username=%s AND photoID=%s"
query = "INSERT INTO likes (username, photoID, liketime, rating) VALUES (%s, %s, %s,
%s)"
```

A clear indication of where to find the application source code for the feature within your GitHub repository (filename and where to look in the file).

app.py has the source code under the function name like(). The html template images.html has the implementation of the like button and ratings box interface.

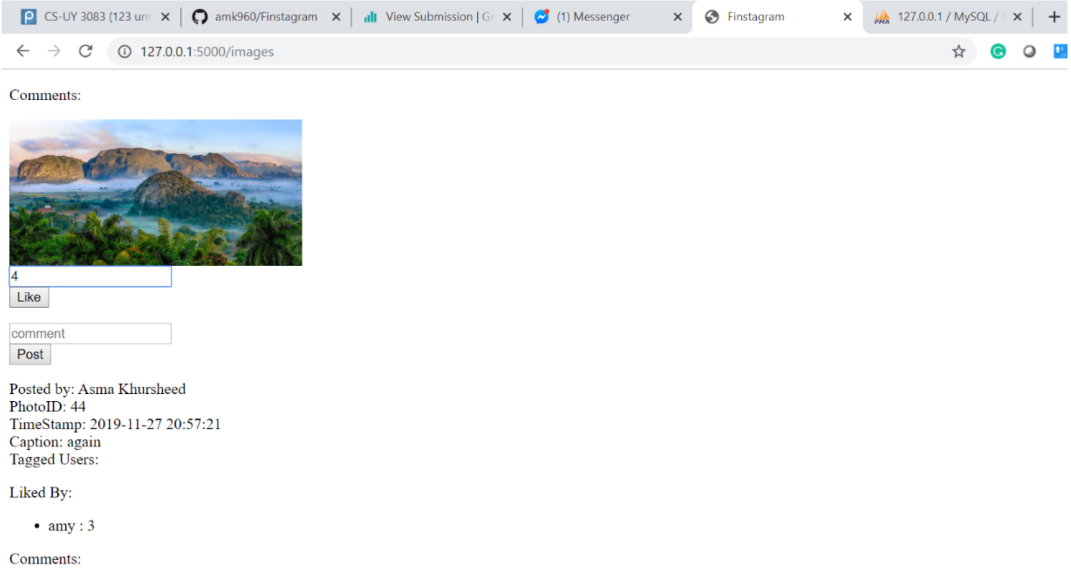
One or more screenshots or a short video demonstrating the feature, showing how it appears in the browser; Also show the relevant data that's in the database when you execute this demonstration (before and after if the feature changes the data), either as screenshots or as text. You may either include this in your GitHub and provide the link here or add screenshots here or on a separate page. Make it clear where the graders should look for this.



Hello, amk960!

- View Images
- Upload an Image
- Follow/Followers
- Friend Groups
- Search Images
- Logout

Screenshot 1: Home page (View Images has the link to all images you can view and has the like button there. Searching for an image also gives you the image and the like button for it!)



Comments:

4

Like

comment

Post

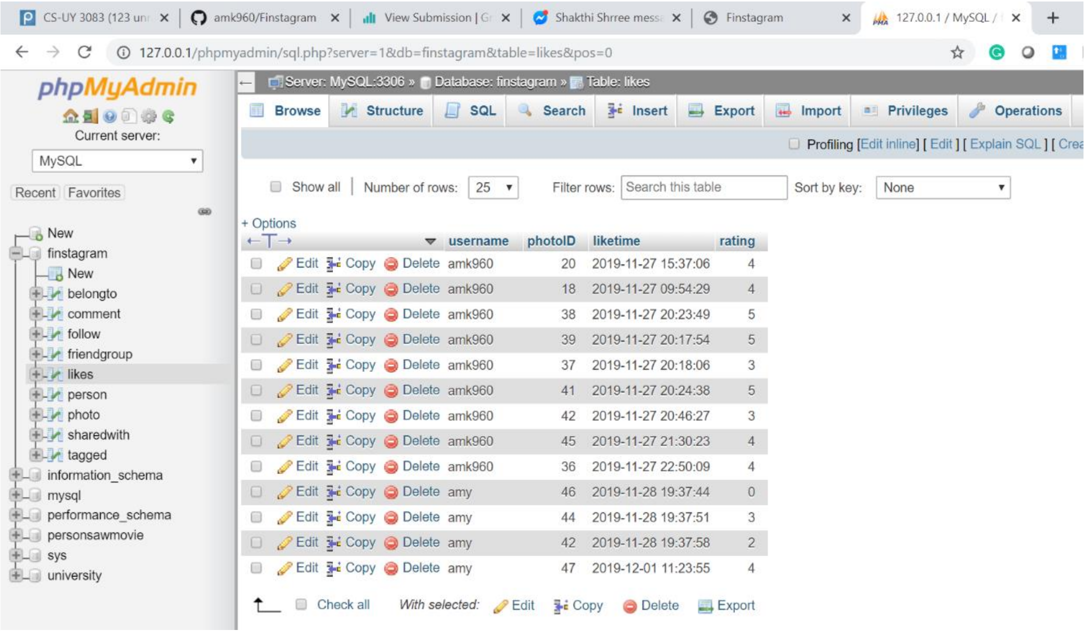
Posted by: Asma Khursheed
PhotoID: 44
TimeStamp: 2019-11-27 20:57:21
Caption: again
Tagged Users:

Liked By:

- amy : 3

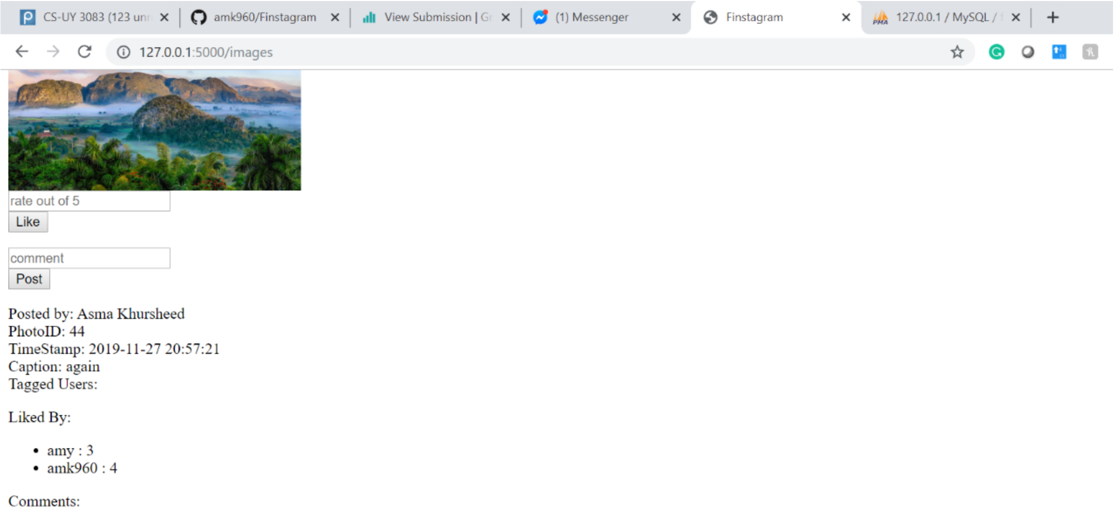
Comments:

Screenshot 2: View images link gives the like button for the image and now user amk960 is entering a rating of 4 for photoID 44.



	username	photoID	liketime	rating
□ Edit Copy Delete	amk960	20	2019-11-27 15:37:06	4
□ Edit Copy Delete	amk960	18	2019-11-27 09:54:29	4
□ Edit Copy Delete	amk960	38	2019-11-27 20:23:49	5
□ Edit Copy Delete	amk960	39	2019-11-27 20:17:54	5
□ Edit Copy Delete	amk960	37	2019-11-27 20:18:06	3
□ Edit Copy Delete	amk960	41	2019-11-27 20:24:38	5
□ Edit Copy Delete	amk960	42	2019-11-27 20:46:27	3
□ Edit Copy Delete	amk960	45	2019-11-27 21:30:23	4
□ Edit Copy Delete	amk960	36	2019-11-27 22:50:09	4
□ Edit Copy Delete	amy	46	2019-11-28 19:37:44	0
□ Edit Copy Delete	amy	44	2019-11-28 19:37:51	3
□ Edit Copy Delete	amy	42	2019-11-28 19:37:58	2
□ Edit Copy Delete	amy	47	2019-12-01 11:23:55	4

Screenshot 3: Before executing (hitting like button), photoID 44 only had 1 liker, amy as shown in screenshot 2 as well.



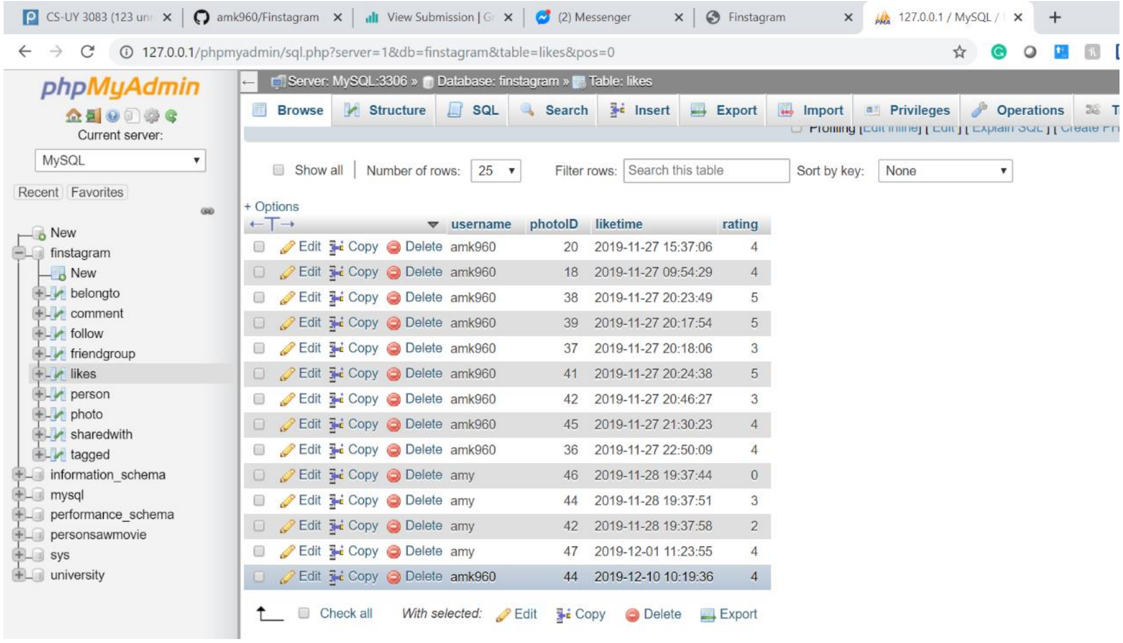
Posted by: Asma Khursheed
PhotoID: 44
TimeStamp: 2019-11-27 20:57:21
Caption: again
Tagged Users:

Liked By:

- amy : 3
- amk960 : 4

Comments:

Screenshot 4: After hitting the like button, the liked by now includes amk960 with rating 4



username	photoID	liketime	rating
amk960	20	2019-11-27 15:37:06	4
amk960	18	2019-11-27 09:54:29	4
amk960	38	2019-11-27 20:23:49	5
amk960	39	2019-11-27 20:17:54	5
amk960	37	2019-11-27 20:18:06	3
amk960	41	2019-11-27 20:24:38	5
amk960	42	2019-11-27 20:46:27	3
amk960	45	2019-11-27 21:30:23	4
amk960	36	2019-11-27 22:50:09	4
amy	46	2019-11-28 19:37:44	0
amy	44	2019-11-28 19:37:51	3
amy	42	2019-11-28 19:37:58	2
amy	47	2019-12-01 11:23:55	4
amk960	44	2019-12-10 10:19:36	4

Screenshot 5: After executing, user amk960 has an entry at the bottom for photoID 44.

2.

Name of the feature

Comment Feature

The full name of the team member who is primarily in charge of implementing this feature

Crystal Li

The queries (and any other SQL statements) used in your implementation of the feature. (If there are standard queries used in most or all of your features, you don't have to include them here; just include the SQL statements that do the main work for this feature.)

```
query = "INSERT INTO comment (username, photoID, commenttime, words) VALUES
(%s, %s, %s, %s)"
```

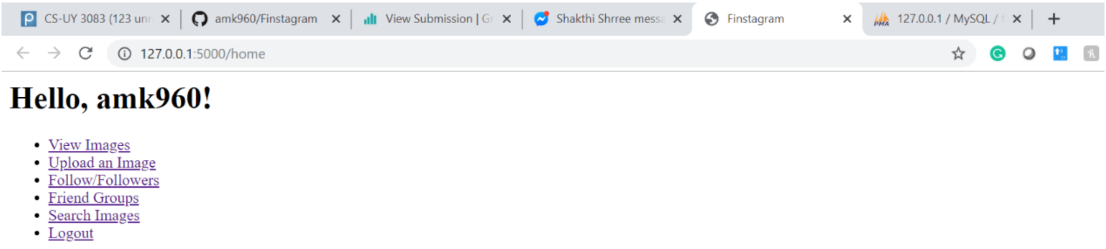
A comment table was created with

```
CREATE TABLE comment(
    username VARCHAR(20),
    photoID int,
    commenttime DATETIME,
    words VARCHAR(1024),
    PRIMARY KEY (username, photoID, commenttime),
    FOREIGN KEY(username) REFERENCES person(username),
    FOREIGN KEY(photoID) REFERENCES photo(photoID)
);
```

A clear indication of where to find the application source code for the feature within your GitHub repository (filename and where to look in the file).

app.py has the source code under the function name comment(). The html template images.html has the implementation of the comment button and data input box interface.

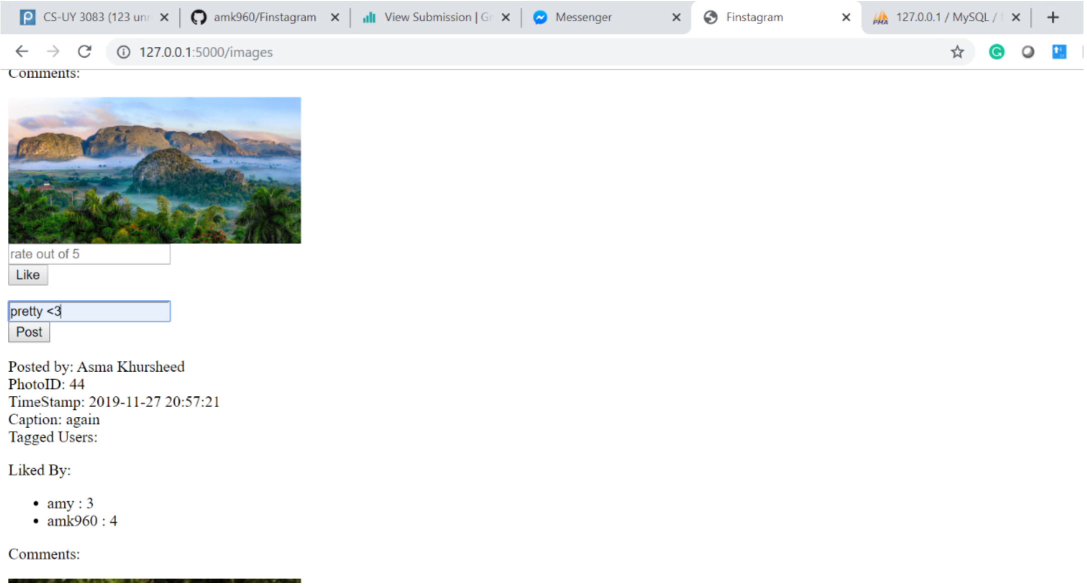
One or more screenshots or a short video demonstrating the feature, showing how it appears in the browser; Also show the relevant data that's in the database when you execute this demonstration (before and after if the feature changes the data), either as screenshots or as text. You may either include this in your GitHub and provide the link here or add screenshots here or on a separate page. Make it clear where the graders should look for this.



Hello, amk960!

- View Images
- Upload an Image
- Follow/Followers
- Friend Groups
- Search Images
- Logout

Screenshot 1: Home page (View Images has the link to all images you can view and has the comment button there. Searching for an image also gives you the image and the comment button for it!)



Comments:

rate out of 5

pretty <3

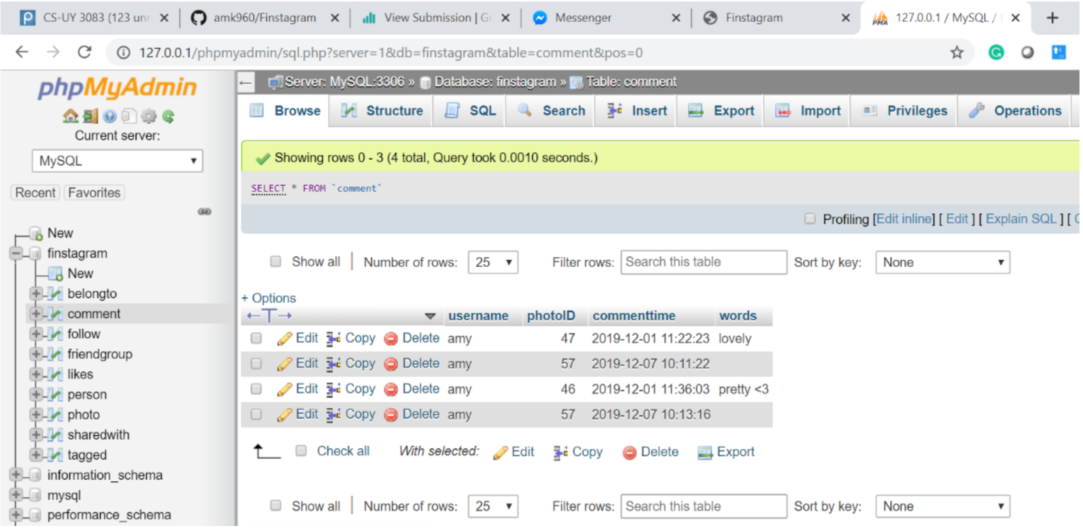
Posted by: Asma Khursheed
PhotoID: 44
TimeStamp: 2019-11-27 20:57:21
Caption: again
Tagged Users:

Liked By:

- amy : 3
- amk960 : 4

Comments:

Screenshot 2: View images link gives the like button for the image and now user amk960 is entering a comment for photoID 44.

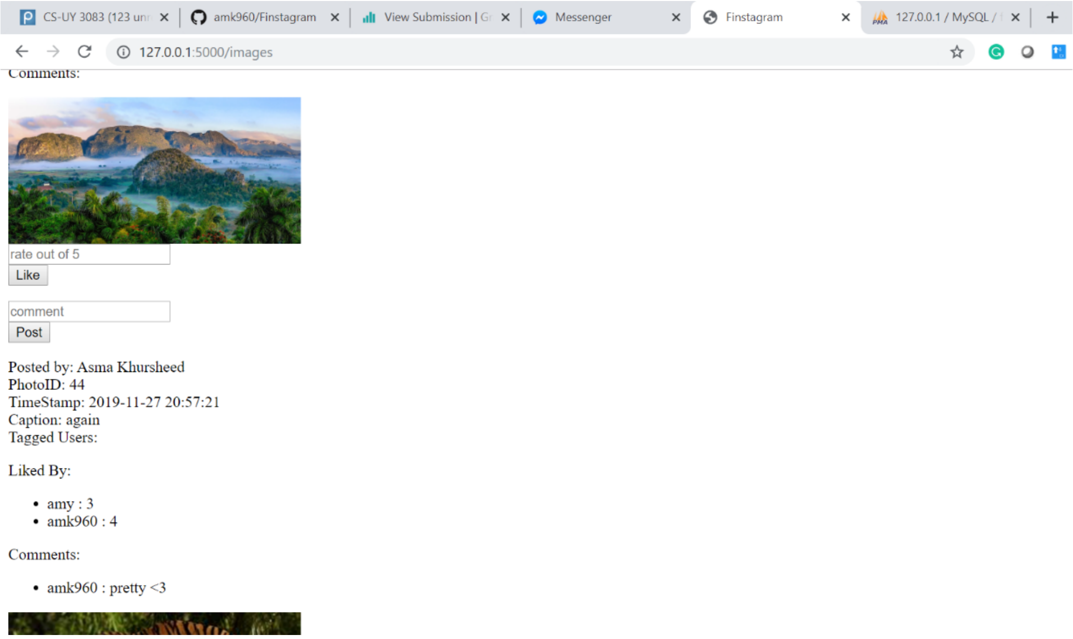


Showing rows 0 - 3 (4 total, Query took 0.0010 seconds.)

```
SELECT * FROM `comment`
```

	username	photoID	commenttime	words
□ Edit Copy Delete	amy	47	2019-12-01 11:22:23	lovely
□ Edit Copy Delete	amy	57	2019-12-07 10:11:22	
□ Edit Copy Delete	amy	46	2019-12-01 11:36:03	pretty <3
□ Edit Copy Delete	amy	57	2019-12-07 10:13:16	

Screenshot 3: Before executing (hitting post button), photoID 44 had no comment



Comments:

rate out of 5

Like

comment

Post

Posted by: Asma Khursheed
PhotoID: 44
TimeStamp: 2019-11-27 20:57:21
Caption: again
Tagged Users:

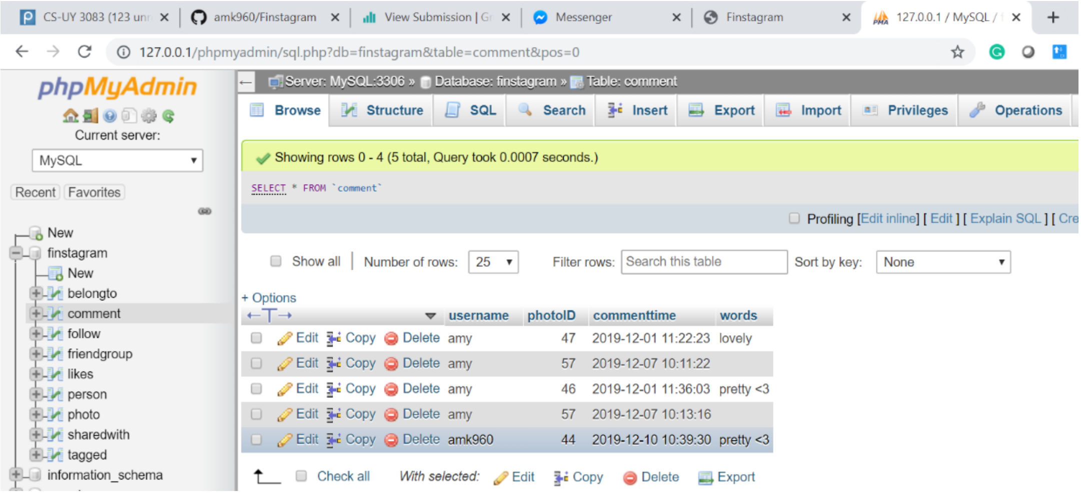
Liked By:

- amy : 3
- amk960 : 4

Comments:

- amk960 : pretty <3

Screenshot 4: After hitting the post button, the comments now includes amk960 with the data they input



✓ Showing rows 0 - 4 (5 total, Query took 0.0007 seconds.)

```
SELECT * FROM `comment`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Cre

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	username	photoID	commenttime	words
□ Edit Copy Delete	amy	47	2019-12-01 11:22:23	lovely
□ Edit Copy Delete	amy	57	2019-12-07 10:11:22	
□ Edit Copy Delete	amy	46	2019-12-01 11:36:03	pretty <3
□ Edit Copy Delete	amy	57	2019-12-07 10:13:16	
□ Edit Copy Delete	amk960	44	2019-12-10 10:39:30	pretty <3

□ Check all *With selected:* Edit Copy Delete Export

Screenshot 5: After executing, user amk960 has an entry at the bottom for photoID 44.

3.

Name of the feature

Add Friendgroup Feature

The full name of the team member who is primarily in charge of implementing this feature

Asma Khursheed

The queries (and any other SQL statements) used in your implementation of the feature. (If there are standard queries used in most or all of your features, you don't have to include them here; just include the SQL statements that do the main work for this feature.)

```
query = "SELECT * FROM friendgroup WHERE groupOwner=%s"
```

Get all the groups that the user owns

```
query = "SELECT * FROM friendgroup WHERE groupOwner = %s AND groupName = %s"
```

To check if the user already owns a group of that name

```
query = "INSERT INTO friendgroup (groupOwner, groupName, description) VALUES (%s, %s,
%s)"
```

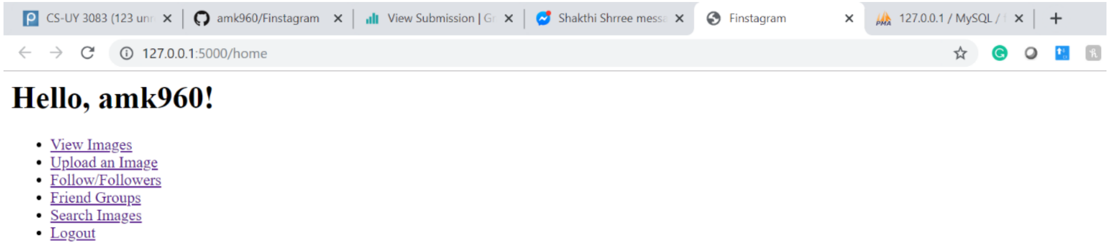
Upon creation, insert into the friendgroup table the new group info and also the user as a member of that group so they could view images shared to members of that group.

```
query = "INSERT INTO belongto (member_username, groupOwner, groupName) VALUES (%s, %s,
%s)"
```

A clear indication of where to find the application source code for the feature within your GitHub repository (filename and where to look in the file).

app.py has the source code under the function name groups() which connects to the raw template when clicking 'Friend Groups' on home screen. friend_groups() is the function with all the input data and queries when user enters data.. The html template groups.html has the implementation of the user interface.

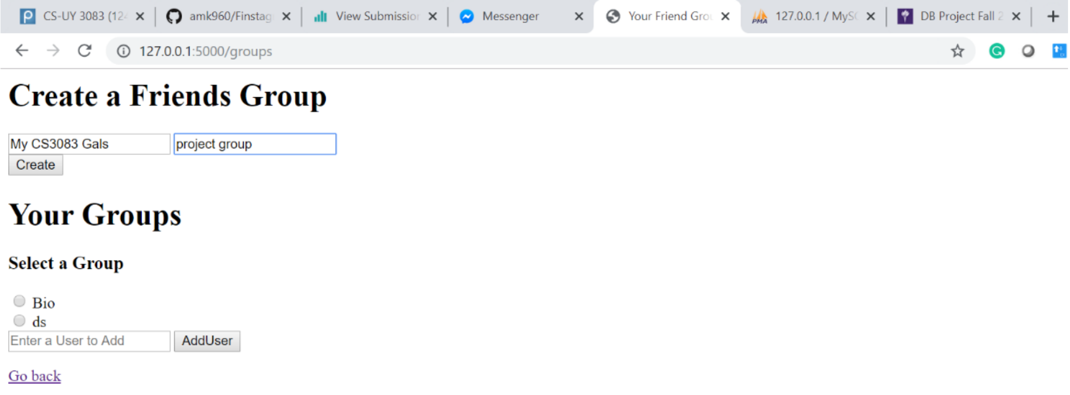
One or more screenshots or a short video demonstrating the feature, showing how it appears in the browser; Also show the relevant data that's in the database when you execute this demonstration (before and after if the feature changes the data), either as screenshots or as text. You may either include this in your GitHub and provide the link here or add screenshots here or on a separate page. Make it clear where the graders should look for this.



Hello, amk960!

- View Images
- Upload an Image
- Follow/Followers
- Friend Groups
- Search Images
- Logout

Screenshot 1: Home page (Friend Groups has the implementation for adding a user to a group and also creating a group)



127.0.0.1:5000/groups

Create a Friends Group

My CS3083 Gals | project group

Create

Your Groups

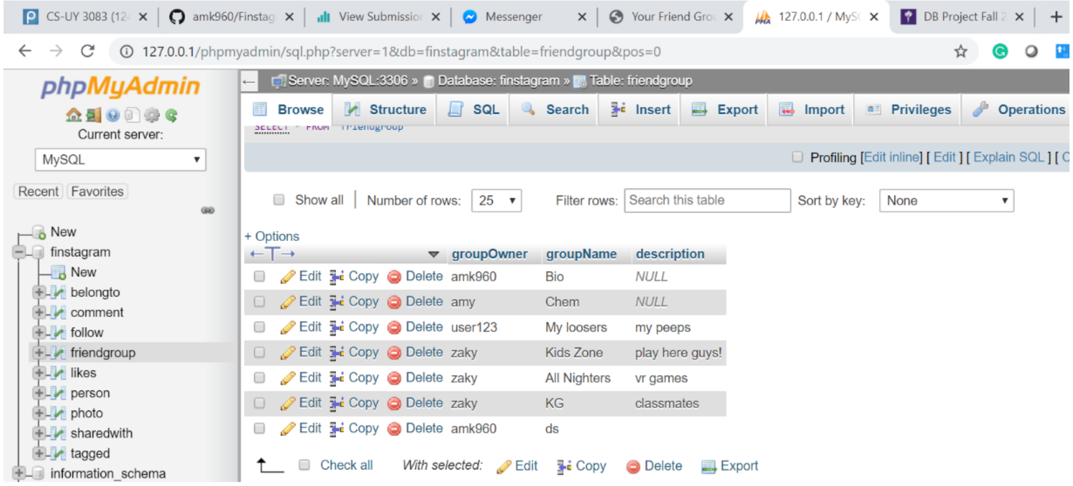
Select a Group

○ Bio
○ ds

Enter a User to Add | AddUser

Go back

Screenshot 2: Friend Groups link leads the user to this page for entering data.

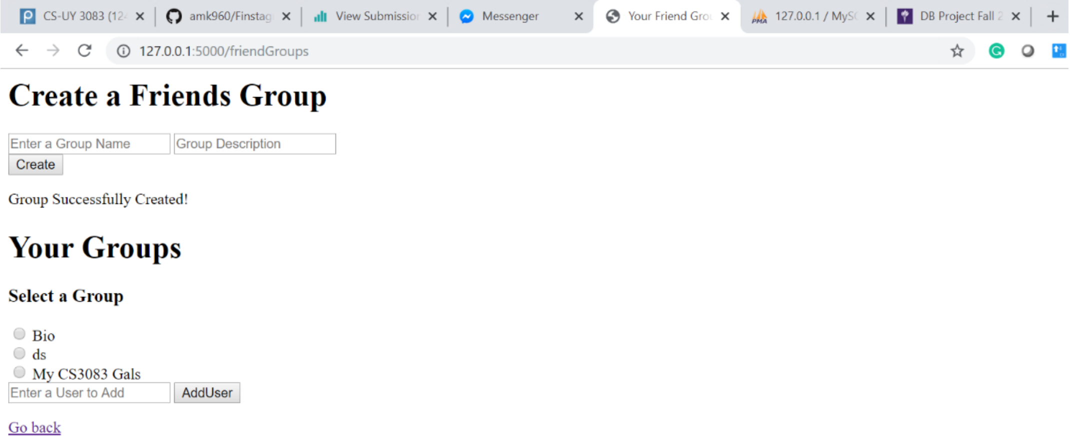


127.0.0.1/phpmyadmin/sql.php?server=1&db=finstagram&table=friendgroup&pos=0

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

	groupOwner	groupName	description
□ Edit Copy Delete	amk960	Bio	NULL
□ Edit Copy Delete	amy	Chem	NULL
□ Edit Copy Delete	user123	My loosers	my peeps
□ Edit Copy Delete	zaky	Kids Zone	play here guys!
□ Edit Copy Delete	zaky	All Nighters	vr games
□ Edit Copy Delete	zaky	KG	classmates
□ Edit Copy Delete	amk960	ds	

Screenshot 3: Before executing (hitting create), amk960 had 2 groups they owned.



127.0.0.1:5000/friendGroups

Create a Friends Group

Enter a Group Name | Group Description

Create

Group Successfully Created!

Your Groups

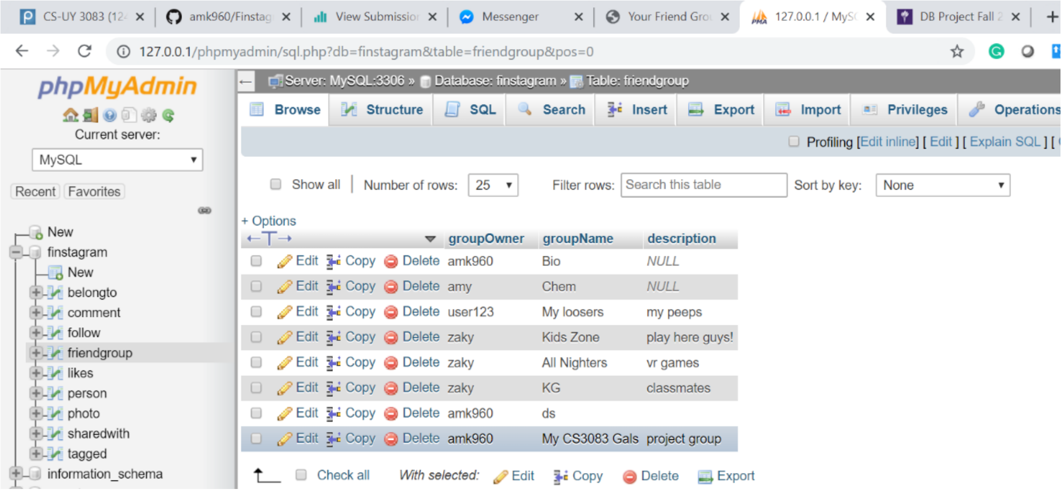
Select a Group

○ Bio
○ ds
○ My CS3083 Gals

Enter a User to Add | AddUser

Go back

Screenshot 4: After hitting the create button, the groups owned now includes new group



127.0.0.1/phpmyadmin/sql.php?db=finstagram&table=friendgroup&pos=0

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

		groupOwner	groupName	description
□	Edit Copy Delete	amk960	Bio	NULL
□	Edit Copy Delete	amy	Chem	NULL
□	Edit Copy Delete	user123	My loosers	my peeps
□	Edit Copy Delete	zaky	Kids Zone	play here guys!
□	Edit Copy Delete	zaky	All Nighters	vr games
□	Edit Copy Delete	zaky	KG	classmates
□	Edit Copy Delete	amk960	ds	
□	Edit Copy Delete	amk960	My CS3083 Gals	project group

Screenshot 5: After executing, user amk960 has an entry at the bottom for photoID 44.

4.

Name of the feature

Addfriend Feature

The full name of the team member who is primarily in charge of implementing this feature

Asma Khursheed

The queries (and any other SQL statements) used in your implementation of the feature. (If there are standard queries used in most or all of your features, you don't have to include them here; just include the SQL statements that do the main work for this feature.)

```
query = "SELECT * FROM person WHERE username = %s"
```

Check if the username entered by user exists.

```
query= "SELECT * FROM belongto WHERE groupOwner=%s AND groupName=%s AND
member_username=%s"
```

Check if the username entered already belongs to the group.

```
query = "INSERT INTO belongto (groupOwner, groupName, member_username) VALUES (%s, %s,
%s)"
```

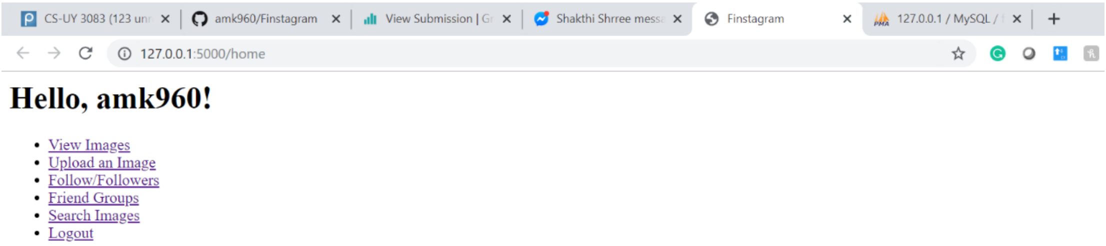
Return all the groups the user owns for outputting to user

```
query = "SELECT * FROM friendgroup WHERE groupOwner=%s"
```

A clear indication of where to find the application source code for the feature within your GitHub repository (filename and where to look in the file).

app.py has the source code under the function name groups() which connects to the raw template when clicking 'Friend Groups' on home screen. add_friend() is the function with all the input data and queries when user enters data.. The html template groups.html has the implementation of the user interface.

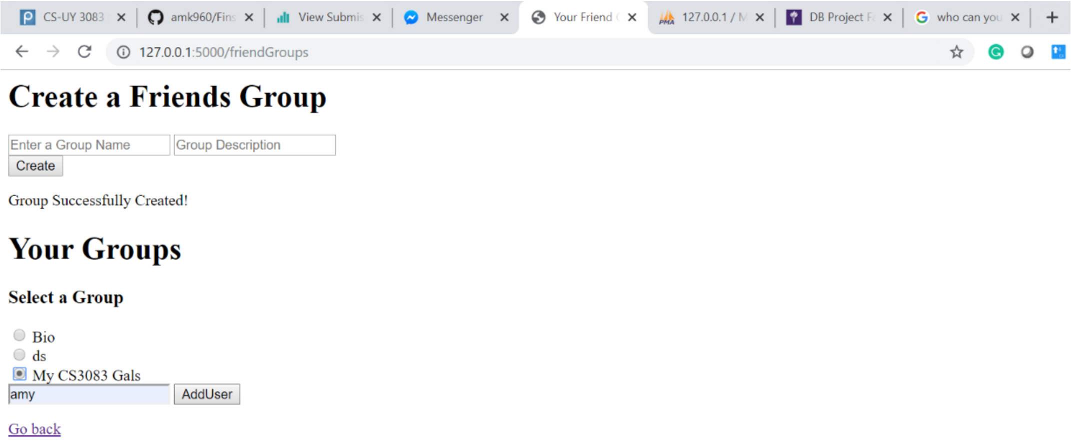
One or more screenshots or a short video demonstrating the feature, showing how it appears in the browser; Also show the relevant data that's in the database when you execute this demonstration (before and after if the feature changes the data), either as screenshots or as text. You may either include this in your GitHub and provide the link here or add screenshots here or on a separate page. Make it clear where the graders should look for this.



Hello, amk960!

- View Images
- Upload an Image
- Follow/Followers
- Friend Groups
- Search Images
- Logout

Screenshot 1: Home page (Friend Groups has the implementation for adding a user to a group and also creating a group)



Create a Friends Group

Enter a Group Name | Group Description

Create

Group Successfully Created!

Your Groups

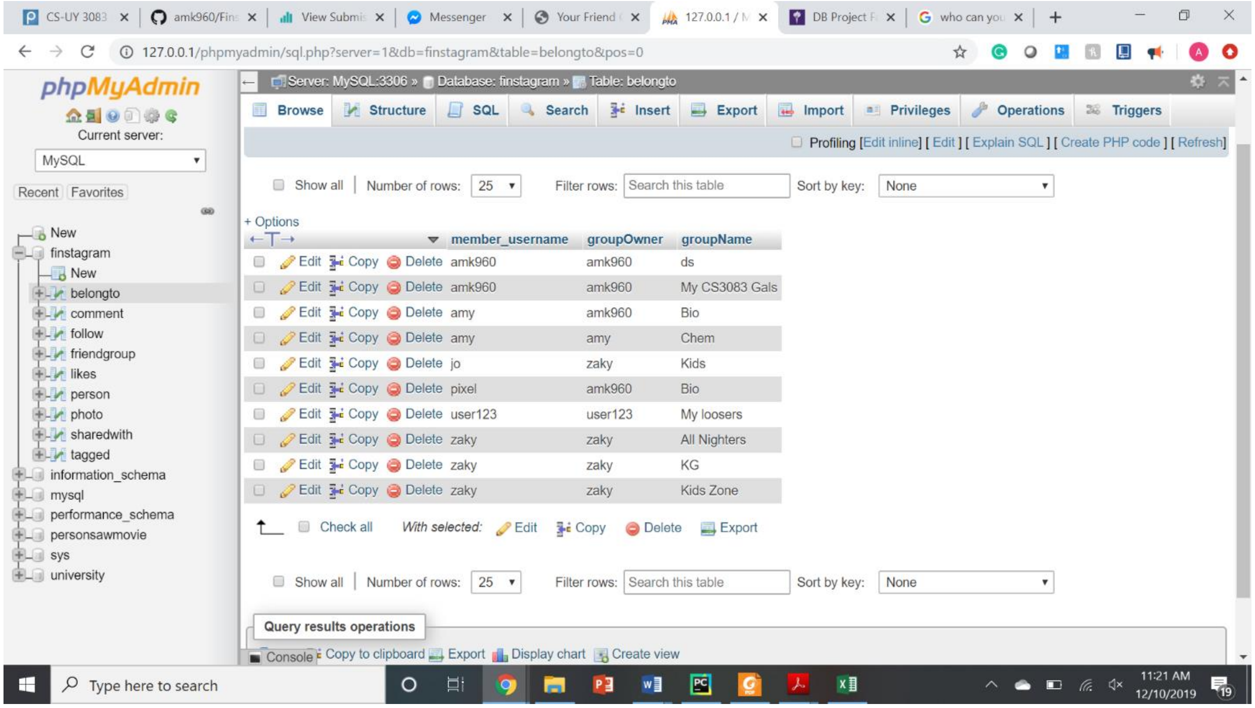
Select a Group

○ Bio
○ ds
● My CS3083 Gals

amy AddUser

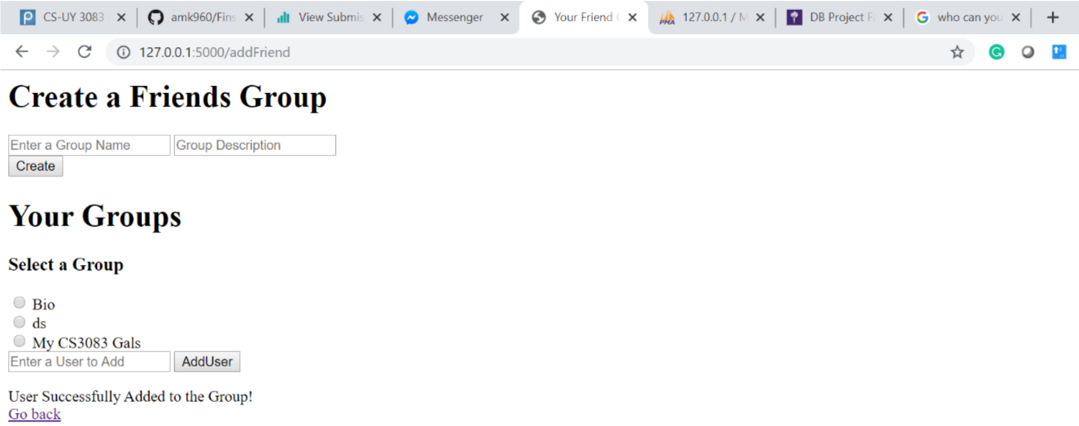
Go back

Screenshot 2: Friend Groups link leads the user to this page for entering data.



member_username	groupOwner	groupName
amk960	amk960	ds
amk960	amk960	My CS3083 Gals
amy	amk960	Bio
amy	amy	Chem
jo	zaky	Kids
pixel	amk960	Bio
user123	user123	My loosers
zaky	zaky	All Nighters
zaky	zaky	KG
zaky	zaky	Kids Zone

Screenshot 3: Before executing (hitting AddUser), amy belonged to 2 groups (Bio, Chem)



Create a Friends Group

Enter a Group Name | Group Description

Create

Your Groups

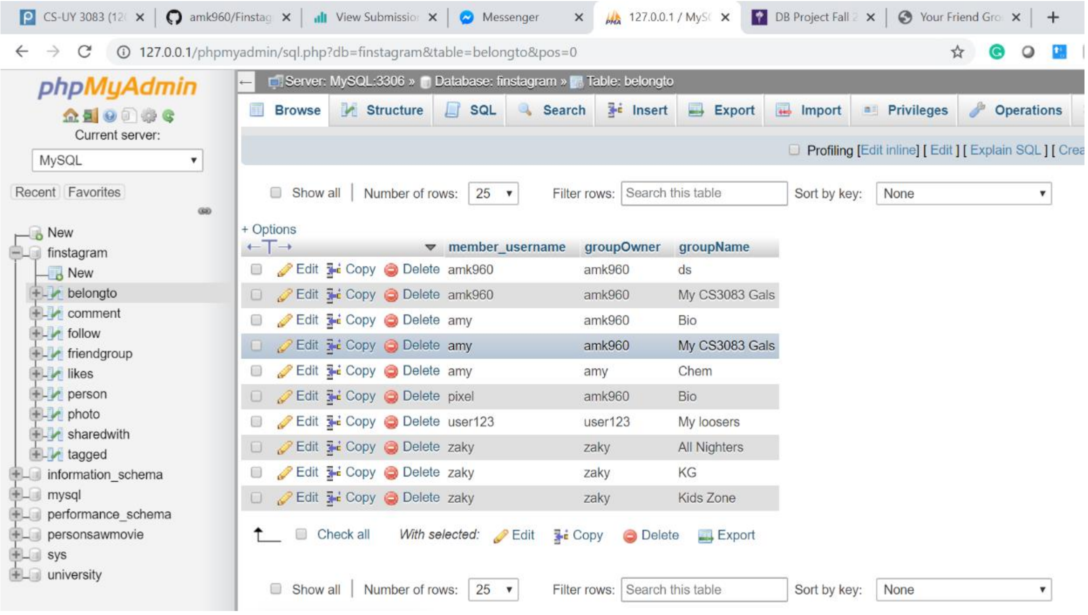
Select a Group

○ Bio
○ ds
○ My CS3083 Gals

Enter a User to Add | AddUser

User Successfully Added to the Group!
Go back

Screenshot 4: After hitting the AddUser button, message displays



	member_username	groupOwner	groupName
□ Edit Copy Delete	amk960	amk960	ds
□ Edit Copy Delete	amk960	amk960	My CS3083 Gals
□ Edit Copy Delete	amy	amk960	Bio
□ Edit Copy Delete	amy	amk960	My CS3083 Gals
□ Edit Copy Delete	amy	amy	Chem
□ Edit Copy Delete	pixel	amk960	Bio
□ Edit Copy Delete	user123	user123	My loosers
□ Edit Copy Delete	zaky	zaky	All Nighters
□ Edit Copy Delete	zaky	zaky	KG
□ Edit Copy Delete	zaky	zaky	Kids Zone

Screenshot 5: After executing, user amy has an entry in belongto table as member

5.

Name of the feature

unfollow Feature

The full name of the team member who is primarily in charge of implementing this feature

Shakthi Ganta Gameshwaramn

The queries (and any other SQL statements) used in your implementation of the feature. (If there are standard queries used in most or all of your features, you don’t have to include them here; just include the SQL statements that do the main work for this feature.)

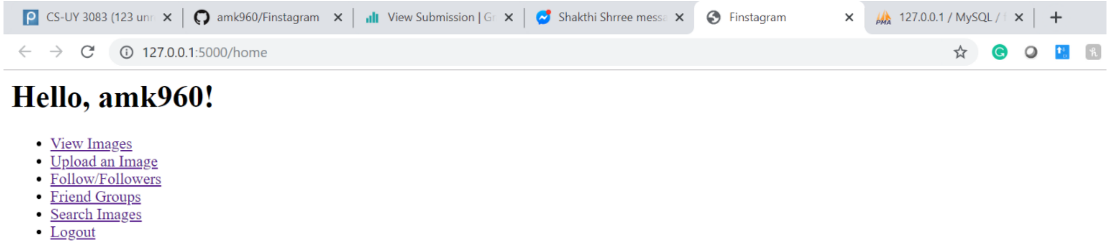
```
query = "DELETE FROM follow WHERE username_followed = %s AND username_follower = %s"
```

Delete from the table when the user selects cancel follow request, or when the user unfollows a following user.

A clear indication of where to find the application source code for the feature within your GitHub repository (filename and where to look in the file).

app.py has the source code under the function name followers() which connects to the raw template when clicking ‘Follow/Followers’ on home screen. add_follower() is the function with all the input data and queries when user enters data.. The html template followers.html has the implementation of the user interface.

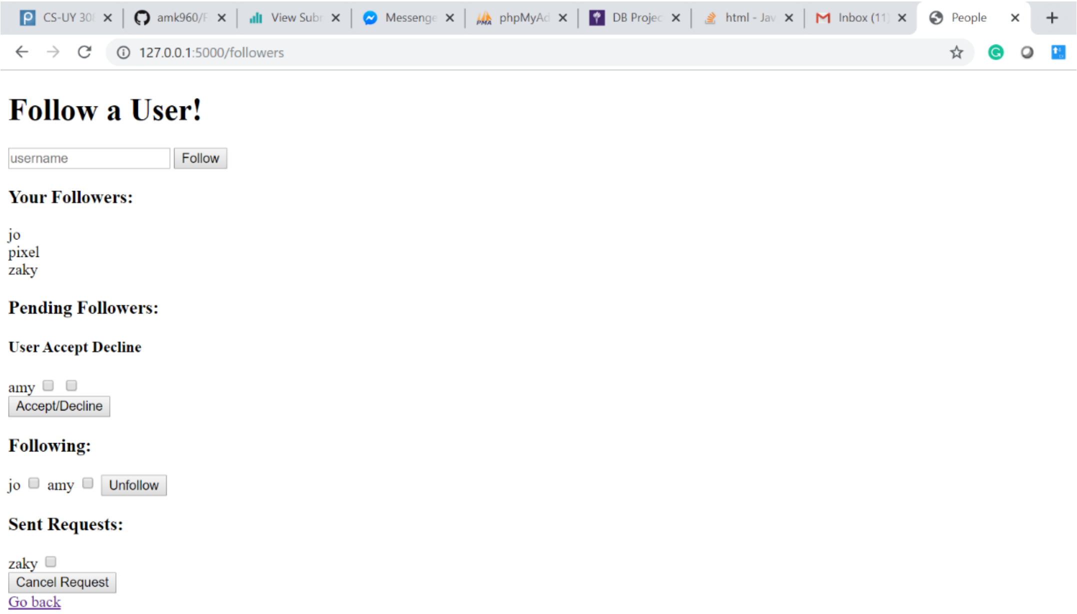
One or more screenshots or a short video demonstrating the feature, showing how it appears in the browser; Also show the relevant data that’s in the database when you execute this demonstration (before and after if the feature changes the data), either as screenshots or as text. You may either include this in your GitHub and provide the link here or add screenshots here or on a separate page. Make it clear where the graders should look for this.



Hello, amk960!

- View Images
- Upload an Image
- Follow/Followers
- Friend Groups
- Search Images
- Logout

Screenshot 1: Home page (Follow/Followers has the implementation for following/unfollowing/accepting followers etc.)



127.0.0.1:5000/followers

Follow a User!

username [Follow]

Your Followers:

jo
pixel
zaky

Pending Followers:

User Accept Decline

amy □ □
[Accept/Decline]

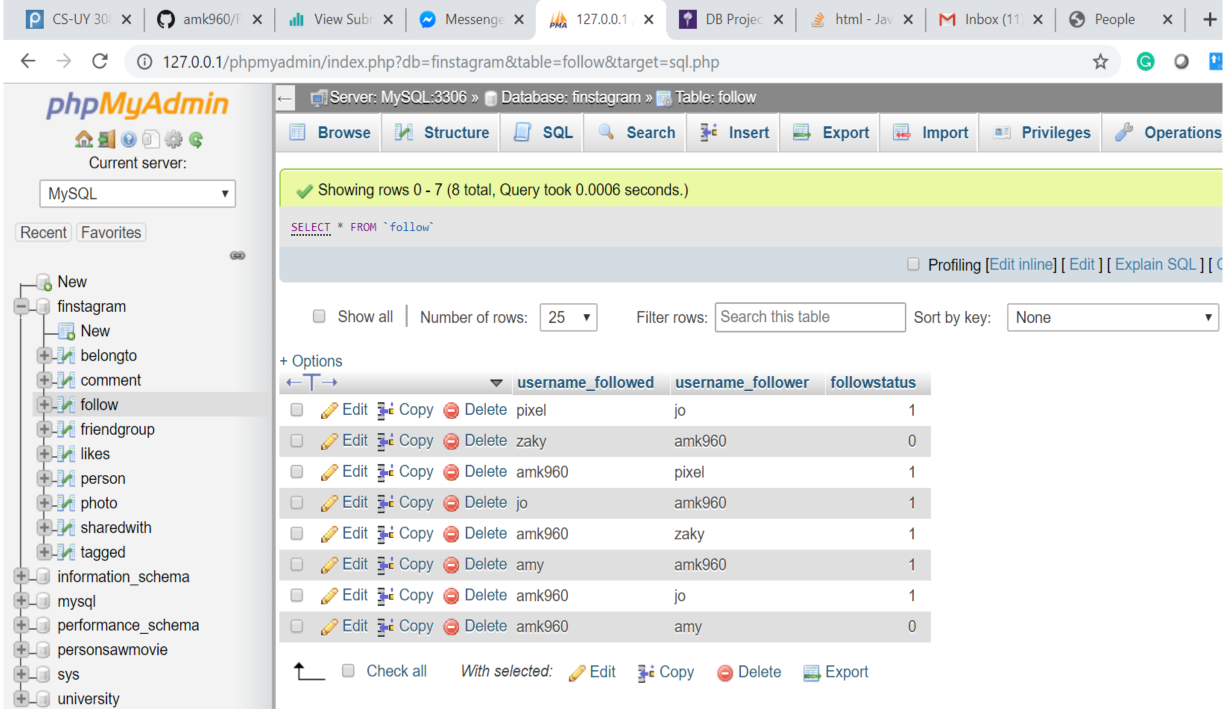
Following:

jo □ amy □ [Unfollow]

Sent Requests:

zaky □
[Cancel Request]
Go back

Screenshot 2: Follow/Followers link leads the user to this page for entering data.



127.0.0.1/phpmyadmin/index.php?db=finstagram&table=follow&target=sql.php

Server: MySQL:3306 » Database: finstagram » Table: follow

✓ Showing rows 0 - 7 (8 total, Query took 0.0006 seconds.)

```
SELECT * FROM `follow`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [C

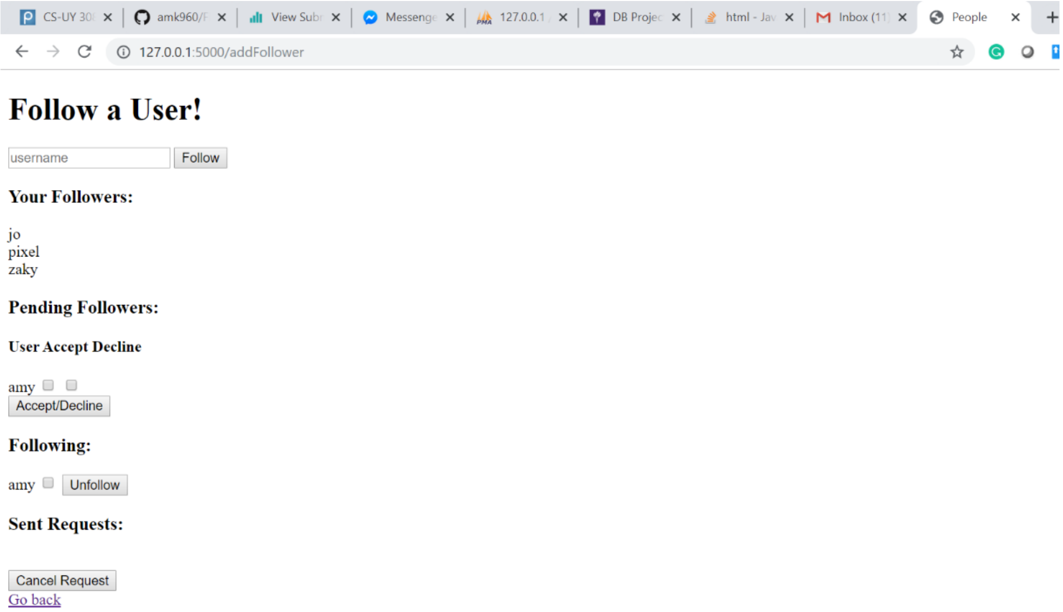
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	username_followed	username_follower	followstatus
□ Edit Copy Delete	pixel	jo	1
□ Edit Copy Delete	zaky	amk960	0
□ Edit Copy Delete	amk960	pixel	1
□ Edit Copy Delete	jo	amk960	1
□ Edit Copy Delete	amk960	zaky	1
□ Edit Copy Delete	amy	amk960	1
□ Edit Copy Delete	amk960	jo	1
□ Edit Copy Delete	amk960	amy	0

□ Check all *With selected:* Edit Copy Delete Export

Screenshot 3: Before executing (unfollow jo and cancel request to zaky) amk960 followed jo with status=1, and followed zaky with status=0.



Follow a User!

username Follow

Your Followers:

jo
pixel
zaky

Pending Followers:

User Accept Decline

amy □ □
Accept/Decline

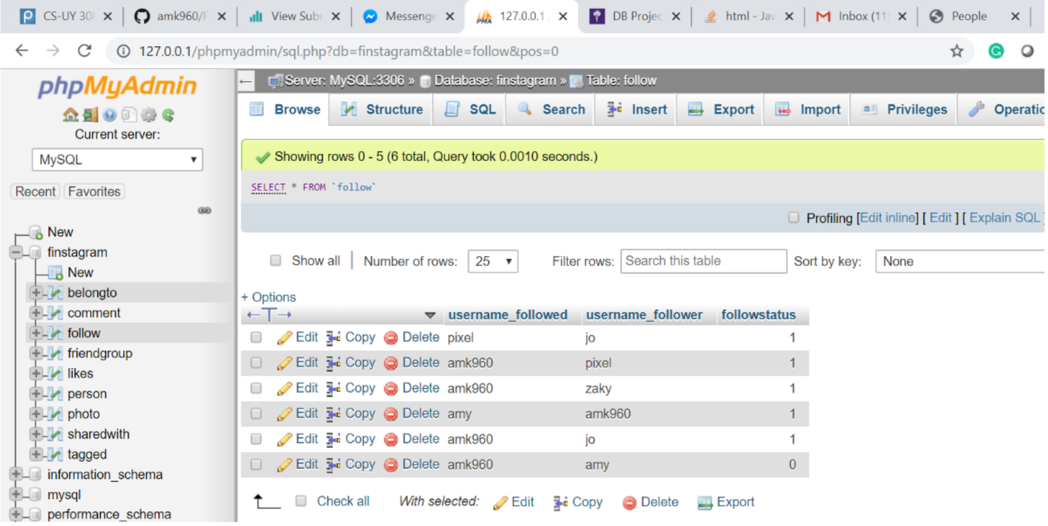
Following:

amy □ Unfollow

Sent Requests:

Cancel Request
Go back

Screenshot 4: After unfollowing jo and cancelling request to zaky



✓ Showing rows 0 - 5 (6 total, Query took 0.0010 seconds.)

```
SELECT * FROM `follow`
```

□ Profiling [Edit inline] [Edit] [Explain SQL]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

		username_followed	username_follower	followstatus
□	Edit Copy Delete	pixel	jo	1
□	Edit Copy Delete	amk960	pixel	1
□	Edit Copy Delete	amk960	zaky	1
□	Edit Copy Delete	amy	amk960	1
□	Edit Copy Delete	amk960	jo	1
□	Edit Copy Delete	amk960	amy	0

□ Check all *With selected:* Edit Copy Delete Export

Screenshot 5: After executing, amk960 no longer follows zaky and jo in the table

6.

Name of the feature

Search by poster Feature

The full name of the team member who is primarily in charge of implementing this feature

Shakthi Ganta Gameshwaran

The queries (and any other SQL statements) used in your implementation of the feature. (If there are standard queries used in most or all of your features, you don’t have to include them here; just include the SQL statements that do the main work for this feature.)

```
queryGroups = "CREATE VIEW mygroups AS SELECT DISTINCT photoID, timestamp, filepath,
allFollowers, caption, photoPoster FROM photo NATURAL JOIN sharedwith NATURAL JOIN
belongto WHERE member_username=%s AND photoPoster = %s"
```

Query of photos of the close friends groups you are in.

```
querySelf = "CREATE VIEW myphotos AS SELECT photoID, timestamp, filepath,
allFollowers, caption, photoPoster FROM Photo WHERE photoPoster=%s"
```

Query of the photos you posted

```
queryFollow = "CREATE VIEW myfollows AS SELECT DISTINCT photoID, timestamp, filepath,
allFollowers, caption, photoPoster FROM photo JOIN follow ON
(photo.photoPoster=username_followed) WHERE username_follower=%s AND allFollowers=%s
AND photoPoster = %s AND followstatus=%s"
```

Query of the photos of the friends you follow.

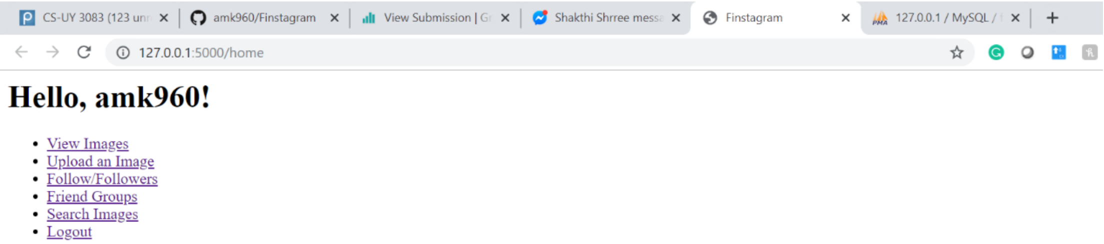
```
totalQuery = "SELECT * FROM mygroups UNION (SELECT * FROM myphotos) UNION (SELECT *
FROM myfollows) ORDER BY timestamp DESC"
```

Union of all the pictures visible to you with chosen poster.

A clear indication of where to find the application source code for the feature within your GitHub repository (filename and where to look in the file).

app.py has the source code under the function name search() which connects to the raw template when clicking ‘Search Images’ on home screen. search_image() is the function with all the input data and queries when user enters data.. The html template search.html has the implementation of the user interface.

One or more screenshots or a short video demonstrating the feature, showing how it appears in the browser; Also show the relevant data that's in the database when you execute this demonstration (before and after if the feature changes the data), either as screenshots or as text. You may either include this in your GitHub and provide the link here or add screenshots here or on a separate page. Make it clear where the graders should look for this.



Hello, amk960!

- View Images
- Upload an Image
- Follow/Followers
- Friend Groups
- Search Images
- Logout

Screenshot 1: Home page (Search Images for searching image by poster and then liking and commenting on that image)

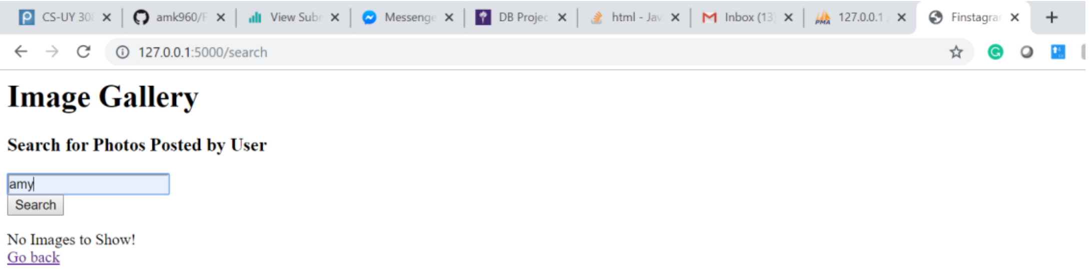


Image Gallery

Search for Photos Posted by User

amy

Search

No Images to Show!

Go back

Screenshot 2: Search Images link leads the user to this page for entering data (Now going to search for images posted by amy).

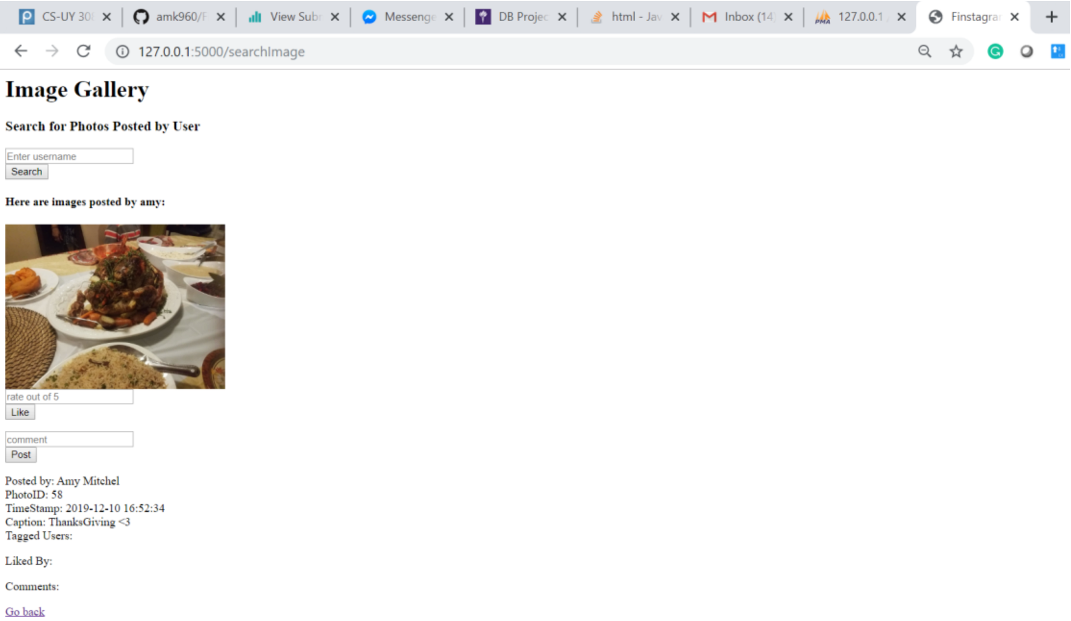


Image Gallery

Search for Photos Posted by User

Enter username

Search

Here are images posted by amy:

rate out of 5

Like

comment

Post

Posted by: Amy Mitchel
PhotoID: 58
TimeStamp: 2019-12-10 16:52:34
Caption: ThanksGiving <3
Tagged Users:

Liked By:

Comments:

Go back

Screenshot 3: After hitting the search button, images visible to user, that were posted by amy appear.

NOTE: This feature does not modify the DATABASE, unless user likes or comments which is explained in the implementation of those respective features.