

Link to GitHub repository: <https://github.com/maydanny/Database-Project>

Date of last commit to repository: **Dec 11, 2019**

Number of Team members: **1**

Names and netIDs of Team members (one per line): **Danny Kim (hmk353)**

=====

For each extra feature (i.e. the $2*n$ features beyond features 1 –5), include the following.

Please copy/paste the following template as many times as needed and then add your answers.

Add page breaks between descriptions of different features. You should have at least $2*n$ pages where n is the number of people in your group.

When this is finished save it in a pdf file named *ProjectSummary_<your-name>.pdf* with the name of the person who will hand it in for the group. Then hand it in on GradeScope. Remember to use the group handin option if you have more than one person on your team.

1. Manage Follows (originally planned to do like and comment, but I thought this was more necessary)

2. Danny Kim

3.

a) Python query

#To send a follow request to another user by going into his/her post page

```
@app.route('/followingAuth', methods=['GET', 'POST'])
```

```
@login_required
```

```
def followingAuth():
```

```
    username = session['username']
```

```
    user = request.form['photoOwner']
```

```
    cursor = conn.cursor();
```

```
    query0 = """SELECT filePath, ID, firstName, lastName, photoOwner, postingDate, caption
```

```
    FROM photo JOIN person ON (photoOwner = username)
```

```
    WHERE photoOwner = %s
```

```
    ORDER BY postingDate DESC"""
```

```
    cursor.execute(query0, (user))
```

```
    data = cursor.fetchall()
```

```
    query = 'SELECT followerUsername, followingUsername FROM follow where  
    followerUsername = %s and followingUsername = %s and followStatus = 1'
```

```

    cursor.execute(query, (username, user))
    followed = cursor.fetchone()
    query4 = 'SELECT * FROM follow where followerUsername = %s and followingUsername = %s
and followStatus = 0'
    cursor.execute(query4, (username, user))
    followed2 = cursor.fetchone()
    if (followed2):
        flash("You already sent a follow request and still pending")
        return render_template('show_posts.html', username=user, posts=data )
    if (followed): #duplicate
        flash("You are already following this user")
        return render_template('show_posts.html', username=user, posts=data )
    else:
        query2 = 'INSERT INTO follow VALUES (%s, %s, 0)'
        cursor.execute(query2, (username, user))
        flash("You have requested a Follow")
        cursor.close()
        return render_template('show_posts.html', username=user, posts=data)

```

←-----→

#Page for managing received tag and follow request

```

@app.route("/manage")
@login_required
def manage():
    username = session['username']
    cursor = conn.cursor()
    query = 'SELECT * FROM follow WHERE followingUsername= %s AND followStatus = 0'
    cursor.execute(query, (username))
    data1 =cursor.fetchall()
    query2 = 'SELECT * FROM tagged NATURAL JOIN photo WHERE username = %s AND
tagStatus=0'
    cursor.execute(query2, (username))
    data2 =cursor.fetchall()
    cursor.close()
    return render_template("manage.html",dataFollow=data1,dataTag=data2)

```

←-----→

#Accept or decline follow request

```

@app.route('/manageFollow', methods=['GET', 'POST'])
@login_required
def manageFollow():
    cursor = conn.cursor()

```

```

username = session['username']
choice = request.form['choice']
followerusername = request.form['followerUsername']
if (choice == '1'):
    query1 = 'UPDATE follow SET followStatus =1 WHERE followerUsername= %s AND
followingUsername = %s'
    cursor.execute (query1, (followerusername, username))
else:
    query2 = 'DELETE FROM follow WHERE followerUsername =%s AND followingUsername =
%s'
    cursor.execute (query2, (followerusername, username))
cursor.close()
return redirect (url_for('manage'))

```

3. (b) SQL query

#Another user's post page where there is a follow button to send a follow request

```

<html>
<head>
    <h1> Finstagram</h1>
</head>
<strong>
{% with messages = get_flashed_messages() %}
    {% if messages %}
        {% for message in messages %}
            {{ message }}
        {% endfor %}
    {% endif %}
{% endwith %}
</strong>
<br>
<body>

```

Here are posts by {{username}}:

```

<form action="/followingAuth" method="POST">
    <button type="submit" name = "photoOwner" value = {{username}}>Follow</button>
</form>

```

```

<form action="/show_posts" method="POST">
<style type="text/css">

```

```

        table, th, td{
            border: 1px solid black;
        }
</style>
<table>
    <th>Photo</th>
    <th>PhotoID</th>
    <th>Name</th>
    <th>Poster</th>
    <th>Posting Date</th>
    <th>caption</th>
    <th>Tagged Users</th>
    <th>Liked & Rating</th>

    {% for line in posts %}
        <tr>
            <td></td>
            <td>{{line.ID}}</td>
            <td>{{line.firstName}} {{line.lastName}}</td>
            <td>{{line.photoOwner}}</td>
            <td>{{line.postingDate}}</td>
            <td>{{line.caption}}</td>
            <td>{% for tag in tags %}
                {%if line.ID ==tag.ID %}
                    {{tag.username}}<br>
                {%endif%}
            {%endfor%}
            </td>
            <td>
                <table>
                    <th>username</th>
                    <th>rating</th>
                    {% for like in likepost %}
                        {%if line.ID ==like.ID %}
                            <tr>
                                <td> {{like.username}}</td>
                                <td>{{like.rating}}</td>
                            </tr>
                        {%endif%}
                    {%endfor%}
                </table>
            </td>
        </tr>
    {% endfor %}

```

```

</table>
</form>

<br>
<br>
<a href="/select_blogger">Go Back Public</a>
<br>
<a href="/home">Home</a>
<br>
<a href="/logout">Logout</a>

</body>
</html>

```

←-----→

#A manage page where all the received follow and tag requests are showing. The user can choose an user who sent the request and decide whether if the user wants to accept or decline the request.

```

<html>
<head>
<h1> Finstagram</h1>
</head>
<body>
<style>
table, th {
    border: 1px solid black;
    border-collapse: collapse;
}

</style>
<form action="/manageFollow" method = "POST">
    Follow Requests:
<table>
    <table>
        <th>Username</th>
        <tr>
            <td>
                {% for line in dataFollow %}
                <input type="radio" name="followerUsername" value={{line.followerUsername}}
required/> {{line.followerUsername}}</input>
                {%endfor%}
            </td>
        </tr>
    </table>

```

```

</table>

<table style="float: left">

    <th>Accept Decline</th>
    <tr>
        <td>
            <button type="submit" name = 'choice' value = "1" > Accept</button>
            <button type="submit" name = 'choice' value = "0" > Decline</button></td>
        </tr>
    </table>
    <br><br>
</form>
<table>

<br>

<form action="/manageTag" method = "POST">
    Tag Requests:
    <table>
        <th>PhotoID</th>
        <th>Photo</th>
        <th>Tagger</th>
        {% for line in dataTag %}
    <tr>
        <td>
            <input type="radio" name="ID" value={{line.ID}} required/>
{{line.ID}}</input></td>
        <td></td>
        <td>{{line.photoOwner}}</td>
    </tr>
    {% endfor %}
    <table>
        <th>Accept Decline</th>
    <tr>
        <td>
            <button type="submit" name = 'choice' value = "1" > Accept</button>
            <button type="submit" name = 'choice' value = "0" > Decline</button></td>
        </tr>
    </table>
</form>

<a href="/home"> Go Back </a>

```

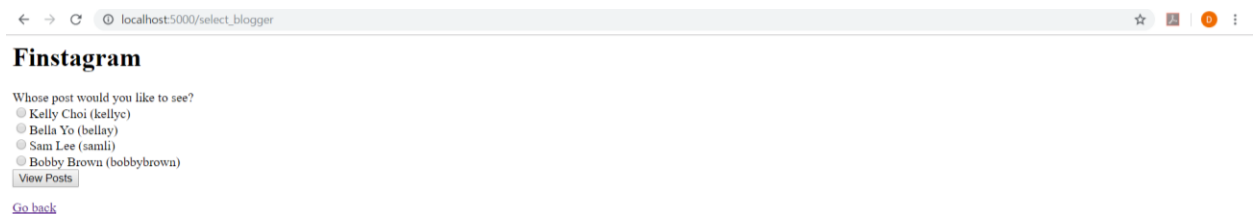
```
</body>  
</html>
```

4.

The above python code with all the queries of the optional feature is named as `finstagram_master.py` within the GitHub. The above queries are positioned in the bottom part of the file, and they are named as `def followingAuth()`, `def manage()`, `manageFollow()`.

The above html files are inside the template folder in GitHub. They are named `show_posts.html` and `manage.html`. To access `show_posts` page, `select_blogger` page needs to be opened first to choose which user's post page to look at.

5.




Within the `select_blogger` page, the logged-in user can choose a person from the people listed and see the user's personal postings. The listed people are the users that registered to Finstagram.

← → ↻ localhost:5000/show_posts?photoOwner=bobbybrown

Finstagram

Here are posts by bobbybrown:

[Follow](#)

Photo	PhotoID	Name	Poster	Posting Date	caption	Tagged Users	Liked & Rating				
	48	Bobby Brown	bobbybrown	2019-12-10 22:44:55	map		<table border="1"> <thead> <tr> <th>username</th> <th>rating</th> </tr> </thead> <tbody> <tr> <td>dannykim</td> <td>78</td> </tr> </tbody> </table>	username	rating	dannykim	78
username	rating										
dannykim	78										

[Go Back Public](#)
[Home](#)
[Logout](#)

If the user choose and visit another user's personal page, 'Follow' button shos on the top of the page. The user can press the button to send a friend request to the user. If the user is fist time sending the follow request, flash shows a message "You have requested a Follow". If the user is already following that user, a message appears as "You are already following this user". But if the user sent a request but the user who received the request did not yet accept it, the message appears as "You already sent a follow request and still pending".

← → ↻ localhost:phpmyadmin/sql.php?server=1&db=finstagram&table=follow&pos=0

Server: MySQL 3300 Database: finstagram Table: follow

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

1 row affected.

UPDATE `follow` SET `followStatus` = '0' WHERE `follow`.`followerUsername` = 'dannykim' AND `follow`.`followingUsername` = 'bobbybrown';

Showing rows 0 - 6 (7 total, Query took 0.0017 seconds.)

SELECT * FROM `follow`

Options

	followerUsername	followingUsername	followStatus
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	dannykim	bobbybrown	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	dannykim	bellay	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	dannykim	samli	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	samli	dannykim	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	bellay	samli	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	bobbybrown	kellyc	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	bobbybrown	dannykim	1

Check all With selected: Edit Copy Delete Export

Query results operations

Print Copy to clipboard Export Display chart Create view

When the user sent a follow request to a user named 'bobbybrown', the database gets new row of the follower username, followee username, and followstatus of 0. The followstatus gets updated when the received user accepts the request.

← → ↻ localhost:5000/manage

Finstagram

Follow Requests:

Username
<input type="radio"/> dannykim

Accept Decline

Accept Decline

Tag Requests:

PhotoID/Photo/Tagger
<input type="radio"/> Accept Decline

Accept Decline

[Go Back](#)

Now in the manage page, the user 'bobbybrown' can accept or decline follow requests received from other users. The user can choose which user's request to accept or decline by using radio format of html. It can be seen that the user received a follow request from another user named 'dannykim', and did not yet decide to accept it or decline.

← → ↻ localhost/phpmyadmin/sql.php?server=1&db=finstagram&table=follow&pos=0

phpMyAdmin

Current server: MySQL

Recent Favorites

- New
- finstagram
- New
- belongto
- follow
- friendgroups
- liked
- person
- photo
- sharewith
- tagged
- flaskdemo
- information_schema
- mysql
- performance_schema
- sys

Server: MySQL 3.306 Database: finstagram Table: follow

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 6 (7 total, Query took 0.0012 seconds)

SELECT * FROM "follow"

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

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

Options

	followerUsername	followingUsername	followStatus
<input type="checkbox"/> Edit Copy Delete	dannykim	bobbybrown	1
<input type="checkbox"/> Edit Copy Delete	dannykim	bellay	0
<input type="checkbox"/> Edit Copy Delete	dannykim	samli	1
<input type="checkbox"/> Edit Copy Delete	samli	dannykim	0
<input type="checkbox"/> Edit Copy Delete	bellay	samli	0
<input type="checkbox"/> Edit Copy Delete	bobbybrown	kellyc	0
<input type="checkbox"/> Edit Copy Delete	bobbybrown	dannykim	1

Check all With selected: Edit Copy Delete Export

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

Query results operations

Print Copy to clipboard Export Display chart Create view

Now when the user accepts the request from 'dannykim', the follow status changes to 1 and now dannykim can see bobby brown's images on his view page.

localhost/phpmyadmin/sql.php?db=instagram&table=follow&pos=0

Server: MySQL 3.306 Database: instagram Table: follow

Showing rows 0 - 5 (6 total. Query took 0.0005 seconds.)

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

Protting [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

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

	followerUsername	followingUsername	followStatus
<input type="checkbox"/>	Edit	Copy	Delete
	bobbybrown	dannykim	1
<input type="checkbox"/>	Edit	Copy	Delete
	dannykim	bellay	0
<input type="checkbox"/>	Edit	Copy	Delete
	dannykim	samli	1
<input type="checkbox"/>	Edit	Copy	Delete
	samli	dannykim	0
<input type="checkbox"/>	Edit	Copy	Delete
	bellay	samli	0
<input type="checkbox"/>	Edit	Copy	Delete
	bobbybrown	kellyc	0

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

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

Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

If bobby brown declines the request, the row with danny kim as follower and bobby brown as followee gets deleted.