

Database - Homework #7 (Group)

PRE-HOMEWORK TASK:

- Turn on your firewall (ufw).
 - Open ports 22 (SSH), 80 (HTTP), 443 (HTTPS)
 - <https://help.ubuntu.com/community/UFW>
 - **NOTE: When turning on firewall, make sure to not exit the terminal until you have allowed port 22 (SSH) and tested it in another terminal. Otherwise, you may be locked out of your own machine.**

Goals

- Introduce Relational Databases.
- Introduce bare-bones SQL.
- Perform SELECT, INSERT, UPDATE queries.
- Perform aggregate queries.

What you will need.

- Install PostgreSQL on your machine.
 - INSTALL: <https://www.liquidweb.com/kb/how-to-install-and-connect-to-postgresql-on-ubuntu-14-04/>
 - TUTORIAL: https://www.tutorialspoint.com/postgresql/postgresql_data_types.htm
 -
- Use PHP PDO to connect to PostgreSQL.
 - INSTALL: <http://php.net/manual/en/pgsql.installation.php> (Only use this link to install, don't follow these tutorials - See below).
 - USING PDO: <http://php.net/manual/en/book.pdo.php>
 - <http://php.net/manual/en/pdo.query.php>
 - <http://php.net/manual/en/pdo.exec.php>

Some helpful commands you will need to get started. NOTE: These commands are run on the postgresql command line.

```
1 -- Create a database called homework7.  
2 CREATE DATABASE homework7;
```

```
3
4 -- Create a user homework7 with password homework7.
5 CREATE USER homework7 WITH PASSWORD 'homework7';
6
7 -- Give the homework7 user full permission on database homework7.
8 ALTER DATABASE homework7 OWNER TO homework7;
9
10 -- For every new table you create, make sure you give user 'homework7' full
11 permission.
12 ALTER TABLE users OWNER TO homework7;
13
14 -- List databases.
15 \l
16
17 -- Connect to a database.
18 \c databasename
19
20 -- List table
21 \dt
22
23 -- More commands are here... https://www.postgresql.org/docs/current/static/app-psql.html
```

What you will NOT need.

- Any Frameworks.
- Any JavaScript.
- Any Database.

Tasks

Task 1

- Make copy of the homework #6 to start homework #7.

Task 2

- Create a table called users.
- Use the fields from Task 2 in homework #6
- Add a field 'id' that is a 'serial' type. This is an auto increment field. Make this field the primary key.
- Add a field 'username' that will store a user's username or netid. This field should be unique and should be enforced at the database level.
- Add a field 'registration' that will have the timestamp of when the user 'registered' (i.e. when the entry was created).

Task 3

- Rename your index.php to info.php
- Create a new index.php
- index.php will be a login form with the username field only.
- If user is logged in, they should not be prompted again and should be redirected to (home.php)
- When login form is submitted, one of the following should happen.
 - Use the username to check if that account exists. If it does, log them in and redirect them to the home.php page.
 - If username does not exist, create an entry. Log them in and redirect them to the home.php page.
- **SUGGESTION:** In order to know that a user has logged in, a value, preferably the id of the row in user table is stored in the PHP variable \$_SESSION. Use this variable to store this information.

Task 4

- Logout page should clear the login variable in 'Task 3'.
- Redirect the user to index.php
- Show a message: "You have been logged out."

Task 5

- Modify your 'page viewed' code to use the database.
- Create a table 'activity' that will track each page view. Columns are.
 - user_id: Foreign Key that references user table.
 - ip_address: User's IP address.
 - timestamp: Timestamp when user viewed this page.
- **NOTE:** For each page view, you should add an entry to this table.
- **SUGGESTION:** Define a function that performs this action in a separate file, include this file in all the pages you need, then call the function.

Task 6

- Create a page 'activity.php'.
- User must be logged in to view this page.
- Page will show the 50 most recent entries from the 'activity' table from 'Task 5' for that user, order by the most recent entry first.

Task 7

- Rename your profile.php to home.php
- Modify your home.php page to use the database.
- Use the info in \$_SESSION (See Task 3) to determine which information to retrieve.
-

Task 8

- Modify your profile-edit.php page to use the database.
- Use the info in \$_SESSION (See Task 3) to determine which information to retrieve.

Task 9

- Each page should have a page views counter.
- These counts should be generated from counting the number of rows in 'activity.php'.
- You must use the 'count' SQL aggregation function to do this.