

COMP 307 Assignment 4

Introduction

In this assignment, you will create a simple forum that allows users to create posts. Users can sign up, log, and create posts under their username. The forum must be implemented using PHP, with MySQL or MariaDB as the database. I suggest that you use XAMPP, which comes with Apache, PHP, and MariaDB.

Database Setup

Start by connecting to the database server via the command and running the following commands.

```
create database assign4;

use assign4;

create table user(id int primary key auto_increment, username varchar(100)
unique, password varchar(100));

create table post(id int primary key auto_increment, creator int not null,
title varchar(100), content varchar(10000), foreign key(creator)
references user(id));
```

This will create the database called assign4, as well as a user table and a post table for storing data. When grading your assignment, we will assume that your PHP code uses a database called assign4, which contains only these tables with exactly these columns. Note that the post table's creator column is a foreign key pointing to the id column of the user table, and "not null" is used to indicate that the foreign key cannot be NULL. Additionally, the unique keyword is used for the username column of the user table, to indicate that no two records can have the same username. I encourage you to try executing SQL statements that violate these restrictions and see what happens.

We will also assume that your PHP scripts connect to the database server using the default MySQL port on the localhost as the root user with no password.

PHP Specification

Using PHP, implement the following web pages.

[index.php](#)

This is the Home page of your forum. At first, your Home page looks like the following

[Home](#) [Sign up](#) [Log in](#) [Log out](#)

Posts

No posts available

You need to log in to make a post

The 4 links at the top are always shown no matter which page the user is currently on. A user needs to use the Sign up link at the top to create an account before the user can create posts.

[signup.php](#)

The signup page shows the following form, allowing the user to create a username and password.

[Home](#) [Sign up](#) [Log in](#) [Log out](#)

Sign up

Enter username

Create password

Upon submit, a record will be created in the user table, as long as the following criteria are met:

- The specified username and password are not empty strings
- The username does not already exist

If one of these conditions are not met, display an error message and ask the user to try again.

When creating a user record, ensure that the password is not stored in plain text, but rather hashed as we discussed in the lectured.

[login.php](#)

After the user has an account, the user must log in before posts can be created. The login page is accessed via one of the links at the top of each page. The log in page looks like the following

[Home](#) [Sign up](#) [Log in](#) [Log out](#)

Login

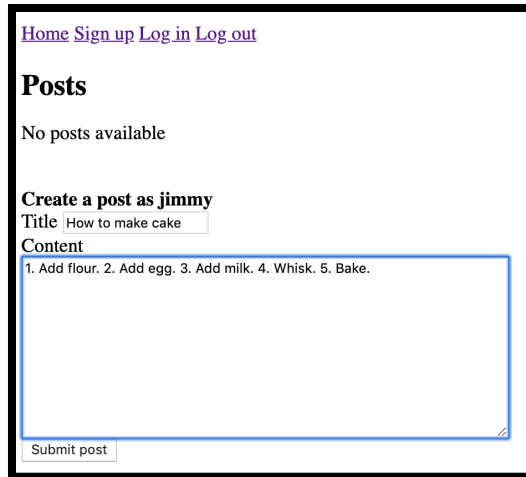
Enter username

Enter password

If the user enters a valid username and password, your website must use the PHP session to ensure that the user stays logged for subsequent page visits. Use the default session settings so that if the user closes the browser, the session is terminated.

[index.php \(after logging in\)](#)

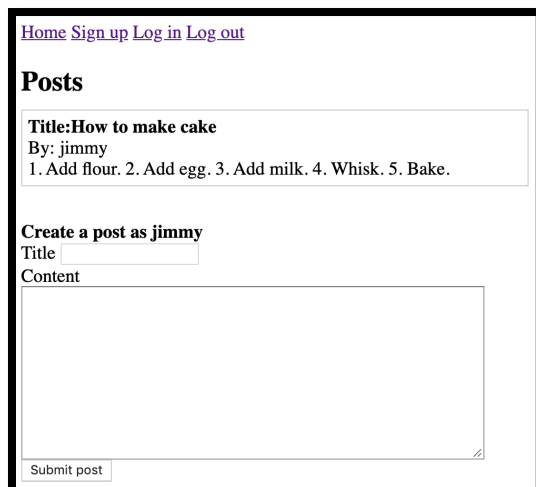
Once a user is logged in, the Home page looks like the following. If there are no records in the post table, it should say “No posts available”.



The screenshot shows a web page with a navigation bar at the top containing links: [Home](#), [Sign up](#), [Log in](#), and [Log out](#). Below the navigation bar is a section titled **Posts**. Under this title, it says "No posts available". Below that is a section titled "Create a post as jimmy". This section contains a "Title" input field with the text "How to make cake" and a "Content" text area containing the text "1. Add flour. 2. Add egg. 3. Add milk. 4. Whisk. 5. Bake.". At the bottom of this section is a "Submit post" button.

The user can now create posts as the logged in user. Each post consists of a title and some content. When the user clicks on submit post, the following this needs to happen:

1. Use JavaScript to check if either title or content is empty. If either is empty, display an error message and do not proceed to the following steps.
2. Perform an Ajax request using jQuery to save the post in the post table in the database. Ensure the id of the currently logged in user is saved in the creator column of the post table. You'll need to create a PHP script on the server side for this purpose.
3. Update the page using JavaScript (with the help of jQuery) to look like the following. Namely, the new post (including the title, creator username, and content) now shows up under Posts, and the title and content fields are cleared. All of this must be achieved using JavaScript, without refreshing or reloading the web page.



The screenshot shows the same web page as before, but now there is a post displayed under the **Posts** section. The post has a title "How to make cake", is created by "jimmy", and has the content "1. Add flour. 2. Add egg. 3. Add milk. 4. Whisk. 5. Bake.". Below the post is the "Create a post as jimmy" section, which now has empty "Title" and "Content" input fields and a "Submit post" button.

At this point, if the user decides to refresh the page for some reason, or if they navigate away from the Home page and navigate back to it, the Home page will remain looking like the above. In other words, all the posts in the post table will show up when the Home page loads.

If the user now clicks the Log out link at the top, and then comes back to the Home page, the following is shown. The posts that have been created are shown, but the user cannot create posts without being logged in.

[Home](#) [Sign up](#) [Log in](#) [Log out](#)

Posts

Title:How to make cake
By: jimmy
1. Add flour. 2. Add egg. 3. Add milk. 4. Whisk. 5. Bake.

You need to log in to make a post

Now suppose a different user comes to your site, creates an account with username andy, logs in, and creates a post. The Home page should now look like the following.

[Home](#) [Sign up](#) [Log in](#) [Log out](#)

Posts

Title:How to make cake
By: jimmy
1. Add flour. 2. Add egg. 3. Add milk. 4. Whisk. 5. Bake.

Title:How to grow plant
By: andy
Put seed in soil, add water, provide sunlight.

Create a post as andy
Title
Content

Submit post

Additional PHP Scripts

You will almost certainly need to create more PHP scripts (in addition to index.php, signup.php, and login.php) in order to accomplish all the listed features. Feel free to include additional PHP files in your site as you see fit.

Security Considerations

Your site must not be vulnerable to XSS or SQL injection attacks.

Path Considerations

When specifying paths in your HTML/JavaScript/CSS files, do **not** use paths that start with /. For the href attribute of links, and for Ajax calls, use a relative path that does not start with /. This will ensure that your site runs correctly regardless of where it is placed within the directory structure of the document root.

To load external JavaScript or CSS files, you must either use a relative path that does not start with /, or use a path starting with a protocol and domain name if you are loading them from a third party. For example, when loading jQuery, you could use one of the following

- `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>`
- `<script src="jquery.min.js"></script>`

External Libraries

You are required to use jQuery in this assignment to make Ajax calls and manipulate the DOM. You are not required to use any other front-end libraries or frameworks. However, you are highly encouraged to try using both Vue.js and Bootstrap.

What to submit

A single zip file containing all the code you wrote for your site. The grader should be able to extract your zip somewhere in the document root and start using your site.