

Goal

The goal of this project is to design an image sharing app / site using all the concepts, languages, and technologies learned throughout the class.

This task will be structured in a series of milestones, as follows:

- **Milestone 1: user authentication**
- **Milestone 2: session management**
- **Milestone 3: basic photo uploader**
- **Milestone 4: improved photo uploader**
- **Milestone 5: additional features**
- **Milestone 6: preparation of the final app**

Recommended Procedure:

1. Make sure you have your **lamp.cse.fau.edu** server space set up.
*Please note that you **must** have a server setup for your PHP code to work.*

Milestone 1: user authentication

2. **Create a table ('USERS') for storing user information** (see example 12.3 in Nixon's book). It should contain 2 fields per record, namely: *userid* and *password*. The password should be stored in encrypted form (after "salting" and hashing) (see example 12.4 in Nixon's book).
3. **Implement the functionality that allows a user to login** (by entering their *userid* and *password* and matching against the data stored in the USERS table in the database). See example 26.7 in Nixon's book for the basic functionality (ignore the visual aspects, please!). After successful login, display a page that shows a simple 'Success' message.
4. **Implement the functionality that allows a user to sign up** (by entering their intended *userid* and *password*): if the *userid* is no longer available, display an error message; otherwise, add the new *userid* and associated (encrypted) *password* to the 'USERS' table in the database and display a simple 'Success' message. (Optionally, you might want to enforce password rules, such as minimum length, etc.)
5. Improve the **visual aspects** of your login/signup page and make it **Bootstrap**-based. See examples at:
 - <https://getbootstrap.com/docs/3.3/examples/signin/>
 - <http://bootsnipp.com/tags/signup>
 - <https://gist.github.com/bMinaise/7329874>
6. **Create a page that looks like you want your future 'wall' to look**. Don't forget to include a 'Logout' button in the page.

Milestone 2: session management

7. Modify your authentication process to **implement the notion of a *session***. More explicitly, you should be able to *create a session upon successful login and destroy the session after the user logs out* (see examples 12.5 and 12.8 in Nixon's book).
8. **Test your implementation of a session** by conditioning the access to the wall page to successfully logged in users only. If a user is not logged in and tries to access the wall they should be redirected to the login page (see example 12.6 in Nixon's book).
9. **IMPORTANT: NAMING CONVENTIONS**
 - Name your main file (with login/signup screen) *index.php*
 - Name your wall file (which should only be accessible to registered users) *wall.php*
 - Make both files available at your *p7* folder.

Milestone 3: basic image uploader (starter code: 'BasicImageUploader')

10. **Create a table for storing photo information.** It should contain 5 fields per record, as follows: USER_USERNAME, STATUS_TEXT, STATUS_TITLE, IMAGE_NAME, TIME_STAMP

database.sql

```
1  -- Uncomment the following line if recreating the table
2  -- DROP TABLE WALL;
3  CREATE TABLE WALL(
4      USER_USERNAME VARCHAR(15) NOT NULL,
5      STATUS_TEXT VARCHAR(140) NOT NULL,
6      STATUS_TITLE VARCHAR(140) NOT NULL,
7      IMAGE_NAME VARCHAR(50) NOT NULL,
8      TIME_STAMP VARCHAR(50) NOT NULL,
9      PRIMARY KEY (TIME_STAMP)
10 );
```

11. Design a **form** for image uploading. It should contain an input for the uploader's username, the image title and caption, and a button (for file selection before upload).
12. Design the **script** that processes the form action, i.e., uploads the picture, saves it with a specific name, stores associated information (username, title, and text) in the WALL table, reads the contents of the WALL table, and displays them in reverse chronological order.
13. **Integrate the latest code with previous milestones, i.e., ensure that the form and the wall are only accessible to authenticated users during the course of a session.**

Milestone 4: improved image uploader

14. Modify the form to allow applying **image filters** (implemented using CSS) to an image before uploading.
15. **Change the WALL table** to include a field that will store information about applied/selected filter.
16. **Change the PHP scripts** accordingly.
17. **Integrate the latest code with previous milestone(s).**
18. **Test your app** after every significant change / addition.
19. **IMPORTANT: NAMING CONVENTIONS**
 - Name your starting page (with login/signup screen) *index.php*
 - Name the upload form (which should only be accessible to registered users) *form.php*
 - Name your wall file (which should only be accessible to registered users) *wall.php*
 - Make all files available at your *p7* folder.
 - In summary, my grader and I should be able to go to <http://lamp.cse.fau.edu/~username/p7/index.php> to signup/login, then to <http://lamp.cse.fau.edu/~username/p7/form.php> to test the upload functionality, and to <http://lamp.cse.fau.edu/~username/p7/wall.php> to see the results (and other people's photos, too).

Milestone 5: add unique features

20. **Implement another piece of functionality that allows a user to sign up and verifies whether a username is available or not (using AJAX).** See examples 26.5 and 26.6 in Nixon's book.
21. **Implement admin functionality to perform (at least one) restricted action (s)** (e.g., remove users, delete objectionable posts, etc.)
22. (OPTIONAL) Integrate your project with well-known sites by using their publicly available APIs.
23. It's time to **polish your project**, give it a unique flavor, and make it stand out.
You may want to focus on improved back-end functionality, additional options/features, and/or "branding".
24. Integrate the latest code with previous milestone(s).

Milestone 6: prepare the final app

25. **Integrate everything!**
26. **Test your app (extensively)** on the lamp.cse.fau.edu server.
27. Once you've reached a point where your app is complete and fully functional in the browser of your choice (Chrome, Firefox, Opera, or Safari), **prepare the final package** (single zip, all that is needed, and nothing else).
28. **Submit the final package** via Canvas.

Minimum requirements:

- Your app must be **your own work**. If you use a site, textbook example or any other source as “inspiration” along the way, please make a note of it in your report.
- You must host your project, **live**, on <http://lamp.cse.fau.edu>
- Your front-end should be based on a CSS framework (e.g., **Bootstrap**)
- Your solution must be **mobile-ready**
- Your solution must include **sign up and log in** functionality
- Your design should demonstrate **separation** between presentation (CSS), content (HTML5), and interactivity (JavaScript) on the client side
- You should use JavaScript for form validation on the client side **before** submitting the form for server-side processing
- Your **server-side scripts** should adhere to PHP best practices.
- Your app must be **fully functional**.

Deliverables

- A **single zip file** containing **all** files (.php, .sql, .css, .jpg, .js, etc.) associated with your project and your report (see below).
- A **detailed report** (5+ pages) describing the entire process, including: the learning curve (for PHP and MySQL, especially), the design process, coding troubles, functionality highlights, future improvements, etc.