

# SY306 Team Project 1

## Implement a message board

### Honor:

No collaboration between teams is allowed. All work for the project must be done by the team members. All sources outside of the course notes, and instructor need to be cited in the code, wherever that source is used.

### Team Assignments:

[Click here for team assignment information.](#)

### Functionality:

Public pages:

1. Sign-up – user can sign up – username, name, and password required. Store also the user role: regular or admin. Store all info in Users table (all new users will be by default regular users)
2. Login - users should be able to login using the username and password created at signup time

Member-only pages – accessible only if the member is logged in

3. Post a message – all messages, with timestamp, are stored in Messages table, where username is a foreign key
4. See all messages, from all users, with submission time and username, on the message board, sorted by the time of submission, most recent first
5. Delete your own messages if you are regular user
6. Admin can delete any message
7. Logout

### Other requirements:

- Server side functionality should be implemented using Python
- Client side functionality should be implemented with HTML5, CSS, and JavaScript

- All your HTML pages have to be valid HTML5, and all CSS needs to validate as CSS3. You should strive to generate valid HTML from Python and JavaScript.
- Server side data needs to be stored in the MySQL database of the team lead. You will have to turn in a SQL script with the statements used to create your tables.
- Before accessing the Member-only area section, you must first direct users to a login page where they provide their username and password, which is checked against the Users table. Only after successful login, should they be able to access the members area. You must use a cookie or some other mechanism so that:
  - they don't have to enter their username and password again
  - check that the user has successfully logged in and has the right role before accessing any information (e.g., admin-only section)
- The message board page with all the messages must be dynamically generated, based on the data in the Messages table
- Your site must be well-documented! You will be required to provide a brief overall write-up of how your site works. Each file should contain a description of that file, creation date, and author name (after the `#!` line for Python). If multiple team members modify the file, each should document the date and author of sub-sequent modifications. Each Python function should be preceded by a short description and you should have comments throughout your code.
- You should define functions and share code appropriately. Don't have the same code in multiple files. The DRY principle applies here!
- You must make use of CSS to make your pages attractive and consistent looking.
- You must use JavaScript to provide useful checking of forms before they are submitted to the server. The Python program should replicate those checks, to ensure users did not by-pass JavaScript.
- A final report and presentation will be required. See Deliverables section for details.
- The main entry page for your website should be named `index.html` or `index.py` and be placed in a top level directory folder called `'project01'` (without quotes).
- All Python, JavaScript, CSS, and HTML code must be constructed using Vim, Emacs, Atom, or a similar text-only editor. The use of programs such as Microsoft Word, Microsoft FrontPage, Dreamweaver, ColdFusion, Mozilla Composer, etc. will be considered an honor offense.

## Project Report:

You will submit a single-spaced written project report, which includes:

- A cover sheet with your team names and project members. If applicable, describe any requirements that you did not meet.
- A description of what each person did – who created each file, who tested?
- Username and password for an admin user, and for a regular user on your site. I may post messages to these accounts when I test and grade your site.

- A short technical report of how the overall system works from a coding standpoint. This should be a list of the scripts/files you have and a description of what they do. The report should also include a flow chart or logical diagram of the relationships between your code (HTML, CSS, JavaScript, Python, & MySQL ) together with information detailing authentication control and user-access methods. **HINT! This is one of the 1st things your team should do, develop a conceptual plan of how all your code will interrelate.**
- A SQL script with the create table statements that you used to create your tables. Executing this script should produce all the tables needed for the project.
- All of your files should be properly documented, as described in the requirements above.

## Project Grading

A portion of your individual grade will be a group grade based on the following:

1. Functionality and correctness – The site should operate as advertised without error.
2. Visual Appearance & Consistency – including good use of CSS.
3. Maintenance – Ensure the site is well organized, requires minimal maintenance, and can be expanded by someone else who doesn't know your code.
4. Documentation – Ensure all pages are documented. Each file should start with a description of that file, creation date, and author name. If multiple team members modify the file, each should document the date and author of sub-sequent modifications. Each Python function should be preceded by a short description. General documentation covering the system's layout and design is also required.

**NOTE:** Much of the above 4 items will be evaluated and graded during the team presentation.

The other portion of your individual grade will be administered by your teammates. Your assessment of your peers shall be based on effort, knowledge, team work, and professionalism.

## Deliverables:

### Electronic submission

One member of the team should create a zip file with all the files for your project and submit it to `submit.cs.usna.edu` under the "Project01" assignment by **23:59 on Sunday April 14th.**

### PUBLIC\_HTML:

In order to run/test the submission, all the Project1 files should also be on the `PUBLIC_HTML` of the team leader under a folder labeled 'project01'

(without quotes). Provide a link from the team leader's `default.html` to the `project01` folder.

## Presentation:

The members of your group should be prepared to demonstrate your system to the class. Each member in the team should present some part of the project. Each presentation will last approx. 5 minutes. Presentations will take place in class on **Monday, April 15th**.

## Report:

You will submit a single-spaced written project report, as described in the "Project Report" section.

## Feedback and Peer Grading:

You will be provided with an online form for giving feedback and grading the members of your team. You will submit this individually – peer grades are confidential. Check your email for more info.

## Thinking ahead...

Project 2 will involve securing your message board. As you work and design Project 1 together as a team, start thinking about how best to implement the following security measures...

- Input size limitations
- Secure password storage
- Duplicate all/any JS checks in Python
- Escaping user inputs to HTML safe character sequences
- Security measures to prevent SQL injections
- Security measures to prevent Cross-site Request Forgery Attacks
- Secure session management

