

Project 1: Build Web Application using PHP

For starters...

Goal: You have been asked to create a web application for an event management system.

Please pay attention to directory and file names. Also, make sure you include a link to your project along with any username and passwords in the dropbox.

Requirements

- Download the SQL file to create the tables from MyCourses and create the tables on Serenity. (There is also an EER diagram of the tables).
- You can add tables if you want (e.g. Permissions), but you cannot modify or delete the tables created by the script.
- You will have to populate the tables with a minimum of: 2 Venues, 2 events, 1 session for each event, 2 attendees for each event and at least 1 attendee of each type (admin (super admin), event manager - needs to be assigned to an event, and attendee - needs to be assigned to an event and session. Passwords need to be hashed using sha256.
- The application will consist a minimum of 3 pages: Events (listing of events with session schedule and venue location), Registrations (Manage registration for events and sessions) and Admin (add users, venues, events and sessions).
- Users need to be logged in in order to use the application (this can be done as a separate page or other method). If the user isn't logged in, you need to require them to login. Also, provide a logout option.
- You will use sessions to control access to the pages based on role:
 - Admin role:
 - Need to have one master admin that can't be deleted or edited.
 - Admin page:
 - Add/Edit/Delete/View any user
 - Add/Edit/Delete/View venues
 - Add/View/Edit/Delete events
 - Add/View/Edit/Delete sessions
 - Add/View/Edit/Delete attendees
 - Plus all functionality of all other roles
 - Event Manager role:
 - Admin page:
 - Add/View/Edit/Delete their own events

- Add/View/Edit/Delete sessions in their own events
 - Add/View/Edit/Delete attendees in their own events/sessions.
 - Plus Attendee role
- Attendee roles:
 - Events Page:
 - View all events.
 - Registrations Page:
 - Select, add, update, delete and view their registrations.
- CSS (in external style sheets) should be used.
- The site will pass HTML5 validation.
- All pages must share common visual and navigation elements.
- You can use any front-end technology you wish.
- Just in case: create backups of your db tables when you have completed the site and as you go.
- **The common elements for each page (navigation, etc) will NOT use include/required statements to include those elements. You need to achieve this via templates or functions.**
- **Your code should be structured to be re-usable and easily maintained. This means extensive use of functions and classes.**
- **All input must be validated and sanitized on the server-side.**
- **ALL DATABASE QUERIES MUST BE PARAMETERIZED QUERIES USING PREPARED STATEMENTS.**
- **DUE DATE: See dropbox on myCourses @ 11:59pm**
 - **Zip up entire project (file structure intact!) in a folder with your last name**
 - **Include links and all usernames and passwords in the comments of the dropbox**

Grading

- Meeting the above requirements will result in a 'B'. See the rubric for options required to get an 'A'