BCS 350: Web Database Development

JLi Spring 2024

Capstone Project Instructions

Objectives:

- Create a MySQL database and tables
- Query database using PHP (list, add, search for, delete records in a database)
- Implement best practices to secure user password
- Implement form-based web authentication in the user log in process
- Use prepared statements and user-defined sanitizing functions to prevent certain injection attacks
- Provide client-side validation using JavaScript and server-side validation with PHP
- Implement session management on chosen web pages
- Design a dynamic web database application.

Tasks:

This project is to acquaint you with fundamental and essential functions when building dynamic web pages with PHP and MySQL. You will make a web application that allows a user, upon successful login, to view, add, search, and delete records in a MySQL database. You will choose and design your own database and tables for your project. Your database can store information about products, inventories, orders, etc. You cannot use the same student table that we created in the Assignment 4. Be creative. The web application can register a new user, logs in a returning user, or logs out a user. You will practice on web authentication, input validation, data sanitization, session management, sign-up and sign-in process, connecting to web database, querying database, processing data, and displaying results on web pages. You will make several web pages showing your knowledge and skills of building a database-driven web application. The completion of this project will greatly help you in your senior project course.

Your project should consist of the following function modules:

- **Registration of a new user**. A new user should at least provide **email, username, password**, and **confirm password** during the sign-up process. Data validation must be provided (see below for requirements). If any data is invalid, display an error message. If all data are valid, register the user. Check if the username is available. Display an error message if the username is not available and provide a link back to the registration page so user can choose another username to complete the registration.

- **Login of an existing user** with correct username and password**.** If the user enters incorrect username or password, display an error message and provide a link back to the login page so user can reenter the username and password.

- **A main menu** is provided to a user **upon successful login**. The main menu contains links for the following tasks:

- **Listing records** of database table(s). You will design your own database and tables and decide what information will be listed on the web page. The data must be displayed in a table format.
- **Adding records** into the database. Provide text fields or other HTML forms to allow user to add a record into the database.
- **Searching for records** in the database. Use a drop-down list for user to choose a field to search and a text field for user to enter the information of that field to search for, for example, in our book example, if a user chooses "author" in the drop-down list, then the text field allows the user to enter the author information to look up; if a user chooses "title" in the drop-down list, then the text field allows the user to enter the title information to look up.
- **Deleting records** from the database. You have two ways to do this. The first approach is you can modify book examples to list all records and provide "Delete Record" button for each record. Another approach is you can allow user to search for a record and delete it if the record exists and the user confirms to delete it.
- **Log out** of user.
- Once a task is finished, a link should be provided for the user to return to the main menu.

Your project should also meet the following requirements:

- You must create a database and a user with password that are specified below and grant this user the full privilege to the database:

  Database name: **`bcs350sp24`**
  Username: **`usersp24`**
  Password: **`pwdsp24`**

  Hints: You need to run `mysql` as the root user to create a new database, a new user, and grant the new user the full access to the new database.

- Your database should have two types of tables, application table(s) and a *users* table. You are free to decide the application table(s) of your database (You can't use the same `classics` and `customers` tables as in the book examples). If your database has one application table, it must have at least 5 fields. Your application table must have a primary key and may have indexes for the fields that will be searched by the users.
- You must have a *users* table to store the registered user's information including username, email, and secured password. The username is the primary key. The password must be salted and hashed.
- Write a php file `setupDB.php` to create your database tables (application table(s) and the users table) and populate the application table(s) with initial values. The users table doesn't need initial values and will be filled out when users sign up (although you can add a user with salted and hashed password).

The `setupDB.php` must only contain php code for creating tables and populating application tables with initial values (not contain code for other tasks of the project).

- The form-based web authentication, not basic HTTP authentication, must be implemented in the user login process.
- Session management must be provided after successful login of a user. A user who has not successfully logged in should not be granted access to any functional module in the main menu (i.e., not be able to list, add, search, or delete records). Prompt or redirect the user to the login page in the case that a user accessed the main menu page without login.
- All input data must be sanitized to prevent injection attacks. Prepared statements with placeholders must be used in "adding records" module to sanitize the user input. Other modules' user input can be sanitized with either prepared statements or user-defined sanitizing functions (Please refer to Module 6 PowerPoints 2: mysqli vs PDO, pages11, 13, and 14. You can use `htmlentities` (or `htmlspecialchars`) and `real_escape_string` functions to sanitize input if using `mysqli`).

  All output data must also be sanitized to prevent injection attacks. You can use `htmlspecialchars` to sanitize the output.
- Input data for new user registration must be validated using both JavaScript and PHP with criteria similar to the book example including the format of email must be valid; no input field is empty; usernames must be at least 6 characters long; passwords must contain at least 8 characters; and passwords must contain at least one of lowercase letters, uppercase letters, and numeric digits. The password must match confirm_password.
- The navigation and usability of your project should be reasonable and applicable. Some suggestions: add a link to the user login on the user registration page and add a link to the user registration on the user login page so user can switch between sign up and sign in; a main menu page is displayed after user logs in; add a link back to the main menu after each function module is finished.
- Complete the capstone project report that includes the Self-Assessment of Capstone Project. List your files and briefly describe the purpose of each file.
  (Feel free to include screenshots to show your work, such as, your web pages before or after an action, the structures of database tables, and data in the tables.)
- Include the integrity statement **I certify that this submission is my own original work** with **your name** as comments in each of your source files.
- Make a short presentation to present how your project works. **The presentation must contain both video and audio.** Don't present your project without audio and expect me to guess your operations. **Missing project presentation will cause a major points deduction**.

  You will present your project in the following order:

1) **Present your name and show your Farmingdale student ID card**. **Missing this step will cause a moderate point deduction.**
2) Run `setupdb.php` to create the application table(s) and the users table, and populate the application table(s) with the initial values.
3) Go to the sign-up page. Sign up using invalid values:
   email: *alice*
   username: *alice*
   password: *mypass*
   confirm_password: *mypass2*
   This is to test the client-side validation.
4) Sign up using valid values:
   email: *alice@abc.com*
   username: *alice123*
   password: *Mypass123*
   confirm_password: *Mypass123*
5) Show all the contents of the *users* table.
6) Go to the login page and log in with username *alice123* and password *Mypass123*. The user should be directed to the main menu after login.
7) Demonstrate "Listing records" module.
8) Demonstrate "Adding records" module. Add a new record to the database.
9) Display the contents of your application table or list records.
10) Demonstrate "Search records" module. Search for the newly added record or other records.
11) Demonstrate "Delete records" module. Delete the newly added record.
12) Display the contents of your application table again or list records again.
13) Log out.
14) After log out, go to the "listing records" page by directly using its URL. This is to test if session control works on that page.
15) Sign up using username *alice123* again. This is to test if duplicated username can be registered.
16) Feel free to demo other features of your project.

Extra credit:
Demonstration of creative features or styles of your application.

Hints:
A project milestone system has been designed to guide you to complete a sequence of tasks that will lead to the successful completion of the capstone project. Please see the attached project milestones document. You should be able to successfully complete the project with what we learned in the course. Other web development technologies (Bootstrap, React, etc.) are not needed.

Submission:

You must submit a zip file that contains the following files by the deadline. No late submission will be accepted. **Missing any part of the following documents will result in a major deduction of total credits.**

1. All your source files
2. The backup file of your database and tables
3. Project presentation video
4. Project self-assessment report

Grading Criteria:

Your project will be graded based on the completeness and correctness of your work listed in the Self-Assessment of Capstone Project.