**BCS 350: Web Database Development**

CRN 24518 JLi Spring 2021

*----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*

**Capstone Project** **Report**

*----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------*

**1. Declaration of the Integrity Statement**

I certify that this submission is my own original work.

Your name and RAM ID: Joseph Nobile, R01730447

**2. Complete the Self-Assessment of Capstone Project**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Module** | **Points** | **Checklist** | **Completion of the task**  **Respond to the questions or answer**  **Yes - if your project completes the task,**  **No - if your project doesn’t complete the task,**  **Partial – describe what part doesn’t work.** |
| Which development stack do you use for your capstone project? AMPPS, XAMPP, WAMPServer, or other. | | | AMPPS |
| Database design (primary key, indexes) | 10 | What are the SQL commands that you used to create your project database and create/grant a user for your database? | Yes |
| Describe your database, how many tables your database has, and what data are stored in the database. | Yes |
| What is the primary key in each table? Any indexes? | Yes - \*Indexes not added at this point\* |
| Has a php file that creates the tables and insert initial data into the application table(s). | Yes |
| Listing records | 5 | List the records in your application table in a table format. | Yes |
| Searching for records | 10 | 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. | Yes |
| Adding records | 5 | Provide text fields or other HTML forms to allow user to add a record into the database. | Yes |
| Deleting records | 5 | Allow user to choose a record to delete or search for a record to delete. | Yes |
| User sign-up, password security | 15 | The form contains fields for username, email, password, and confirm\_password. | Yes |
| Username is checked against the database *users* table so duplicated username is not accepted. | Yes |
| If new user input meets the requirements, password matches confirm\_password, and username has not been used, new user is added to the *users* table. | Yes |
| Password is salted and hashed before stored in the *users* table. | Yes |
| User login,  Web authentication | 12 | A login page has a text field for username and a password field for password. A user successfully logs in if the submitted username and password match the stored record in the *users* table. | Yes |
| If the username or password doesn’t match the record, display an error message and provide a link to go back to the login page. | Yes |
| Log out | 3 | Log out a user. | Yes |
| Cookies and sessions | 10 | Session starts when a user logs in and session ends when a user logs out. Session management is provided to the main menu page and all function modules on the main menu page so that each function module can only work upon successful user login. | Yes |
| Data sanitization | 10 | All user submitted data must be sanitized before being used or stored into the database. | Yes |
| Prepared statement and place holders must be used in the “Adding Records” module. | Yes |
| JavaScript validation | 8 | JavaScript validation is provided to ensure all input values meet the criteria. | Yes |
| Check whether password matches confirm\_password using JavaScript. | Yes |
| PHP validation | 7 | PHP validation is provided to ensure all user input values meet criteria. | Yes |
| Check whether password matches confirm\_password using PHP. | Yes |
| Extra credits: Creative works | 10 | Describe any special design of your project. | Utilized basics of css file to provide some customization and add to presentation of the website. |

**3. List your source files and give a short description to each file**

add.php – Allows the user to add a new record to the music database table.

delete.php – Displays all records in the database, then prompts the user with a drop-down list to delete a specified entry in the database.

delete\_function.php – Contains the functionality necessary to delete the record selected in delete.php.

functions.php – Contains all additional functions necessary throughout the project. This includes string sanitization functions, add, randomizing a string for salting/hashing passwords, validating entries, etc.

home.php – Main home screen. Prompts the user to log in or register for any access to the website.

is\_logged\_in.php – Checks to see if the user is currently logged in, otherwise uses “not\_logged\_in.php” to redirect the user to login/register.

list.php – Displays a list of all the entries in the database.

login.php – Main login file necessary for initializing the database and all login variables.

login\_form.php – Prompts the user for login credentials, otherwise asks the user to register if they are not already a member.

logout.php – Logs the user out of the current session.

music.css – This file contains almost all of the custom visualization of the website. This includes background and text colors, table formatting, table heading colors, etc.

navigation\_bar.php – Contains links used as navigation control throughout the website. Displayed at the top of almost every page.

not\_logged\_in.php – File that is called if the chek in “is\_logged\_in.php” fails. This file displays an error that the user must be logged in to access whatever page they are attempting to access, then redirects them to the login screen.

register\_form.php – Form used to register a new user. Calls functionality on “register\_process.php”.

register\_process.php – This file contains all functionality necessary for registering a new user in the database. This file calls functions in the “functions.php” file to sanitize the user input, then create random strings of size 10, and salt/hash the user’s password for entry in the “users” table in the database.

search.php – Provides the user with a drop-down menu to select search criteria, then asks the user to search for entries within the database. Functionality is in “search\_results.php”.

search\_results.php – Takes the user input from “search.php”, sanitizes it, then queries the database for a search and displays the results.

setupDB.php – This file contains the code to be entered in the command window initially (displayed in comments), then calls the “login.php” file to verify the admin user/password specified, and creates the database tables via multiple queries.

user\_login.php – This file contains the process necessary for checking login credentials against existing login credentials in the database. If incorrect, will prompt the user to retry their login.

user\_validation.js – This is required for javascript validation. The file is called within the registration pages in order to validate new registered user credentials and ensure that each entry is unique.