**Requirements**

- Create a database table to persist user records for your application. At a minimum, include columns for a name, email, and encrypted password.

- Create a sign up form to capture the necessary information for a new user. Upon form submission, create a new record in your database table.

- Do not persist the password in plain text. Encrypt the password such that the password can be verified each time the user signs in. Use the `password_hash` function provided by PHP.

- Create a sign in form that asks for an email and password. Upon form submission, check to see if a user exists in the database table with the provided email, and also if the provided password can be verified against the encrypted password for that user. Use the `password_verify` function provided by PHP.

- If the credentials cannot be verified, inform the user and do nothing else. Otherwise, inform the user and use PHP sessions to establish a new user session to keep the user signed in until he or she closes the browser.

- When the user is not signed in, display links to sign in and sign up in your application header. Otherwise, when the user is signed in, display a personalized greeting that includes the name of the user, and also a link to sign out.

- Add logic to your application to restrict access to certain pages to only authenticated users. Implement at least one restricted page to demonstrate this functionality.

- Make your site look professional and presentable. Use valid HTML and CSS to structure and style your pages.

**Resources**

- Refer to the [PHP documentation](#) as needed when completing your assignment.

**Submission**

- Show your completed assignment to the instructor during class or office hours to receive credit.