

The purpose of this part of the second assignment is to use MySQL & PHP to make the pages created in the final part of assignment #1 functional.

Please use the files I've provided, which act as both my solution to the previous assignment, and a clean starting point for everyone. It also contains a few pages I didn't have you build before, but that you'll need now.

Create a proper header & footer in the includes folder (everything before the main in header, everything after main in footer), then modify all pages to remove the header/footer content, include the header and footer, and change the extension to PHP. This should be like what we did in the lab.

Import the necessary tables to your database using the SQL file I provided.

Once your database tables have been added, incorporate the PHP to make your application functional. Things you'll need to consider are:

- **Every page (header.php)**
  - The title should be different on every page (use a variable in the header)
  - The login/logout links should appear conditionally base on login status. The same with the appropriate links in the next section (i.e. Create account if not logged in, List stuff if logged in)
  - The random suggestion box should be randomly populated with a list item from in the database whose list has been made publically available. (it should change on every page load)
- **Create account (account.php)**
  - Should validate collected information
  - Properly salt and hash the password, then insert into the database.
  - Create the appropriate session for the logged in user, (after retrieving userid) then redirect to the main page.
- **The Login Page (login.php)**
  - Validates the user credentials against the database
  - Creates a session (including the user's database id).
  - This session is then checked on all other pages to make sure the user is logged in (otherwise send them to the login page).
  - Putting the user's id in session will allow this to be included when inserting list items.
  - It should also set a cookie where appropriate. (store the username)
  - The cookie if it exists on the login page should populate the username in the textbox
  - Redirect to main
- **Log Out (logout.php)**
  - Destroy session and cookie
  - Redirect to login page
- **Edit User (edituser.php)**
  - Populate the form fields with the database information for the logged in user.
  - On form submission, validate information and update database record accordingly.
  - You can decide how you want to handle password updates (separate it onto its own page, only update if something in box, etc)
    - Redirect back to main
- **Edit List (editlist.php)**
  - Populate the form fields with the database information for the logged in user's list.
  - On form submission, validate information and update database record accordingly.
  - Redirect back to main
- **Delete Account**
  - Add a menu item for this
  - Delete list and user information for the logged in user
  - Then destroy the cookie and session.
  -
- **The Main Page (index.php)**
  - Error message at top should be dynamic if list isn't marked available
  - Should dynamically display the user's list, with each list item having appropriate edit, delete links (the icons) with correct item id in the url.
  - List should be ordered by item number then title
- **Add Item (additem.php)**
  - Validate info and add to database
  - Redirect back to main
- **Edit Item**
  - Populate form with database information for item by the id in the URL.
  - On form submission, validate information and update database record accordingly.
  - Make sure image is renamed and moved if uploaded.
  - Redirect to item details
- **Delete Item (deleteitem.php)**
  - Delete item from database
  - Redirect back to main

- **Display List**
  - Error message at top should be dynamic if list isn't marked available
  - Dynamically populate list with user list data. Items marked completed should be linked to displaydetails.php, with appropriate id in URL.
- **Display Details Page** (displaydetails.php)
  - Display all the details for the current list item, using existing formatting. Item id should be retrieved from the URL.
- **Security** – make sure to show appropriate consideration for the security of your application. (password protection, sql\_injection, cross site scripting)

Your functional version must also validate and be tested cross browser and cross platform. If you are unable to fix cross browser problems, document them in a testing file and include it with your submission

## Submission

Compress (zip or rar) the completed directory structure (html/php files should be in root folder, images in an img folder, and css in a style folder, includes in the includes folder), along with a file containing a database\_dump of your current database, then submit the compressed file, along with credentials for an existing user account in your application.

The entire site should also be located on a publically accessible webserver (one provided or your own) and your submission should include a link to the index page.

Remember to submit a testing file which includes any cross browser/platform issues you were unable to resolve.