

Trent University
COIS 3420H
Culminating Project

The culminating project for this course is designed to give you experience implementing the concepts discussed all together in a full web application. The application itself is based on the following premise:

You have been hired to build a web application that allows users to track a physical and digital media collection (although for simplicity we're limiting it to movies). At a high level the application will allow the end-user to create/edit their account, add/edit/delete/view individual media items, and view/search their entire library.

The screenshots in the project folder contain some example pages of what your application could look like. You are welcome to replicate the design as is, or *Advanced Topic*: come up with your own (which to actually be considered advanced should be an improvement). Your **initial** design shouldn't include any JavaScript (we'll add some later), or the use of any external libraries (except the required Font Awesome)

Note: Attempting one or more of the listed **Advanced Topics**, will get you extra complexity marks (particularly if you're working in a big group where expectations are slightly higher). Since you're not actually being marked at Check-in, you can complete Advanced Topics at any time.

Check-In One

For the first check-in period you should have completed the HTML/CSS for all the front-facing pages, which include the following:

Create Account (account.html) – This page allows the user to create an account in the application. It should collect:

- Username
- Name
- Email
- password (twice).

The Main Page (index.html) will display all the movies in the collection in a grid format with previous and next links

- ◆ Eventually the movie info will come from the database...for now hard code enough in to get your layout worked out).
- ◆ You can use an actual table, flexbox or CSS Grid for the layout
 - *Advanced Topic*: Using flexbox or CSS Grid, make this page a little responsive (the number of movies in each row changes based on screen size).
- ◆ Underneath each movie should be image links to the edit, delete, and details pages for that movie. Use Font Awesome for your icons (suggestion: a pencil, a trashcan, and info)

The Add Page (addvid.html) – allows the user to enter details for a new movie. Page should collect:

- | | |
|-------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| ◆ title, | ◆ DVD release |
| ◆ star rating | ◆ actors |
| ◆ genre (user should be able to select more than one) | ◆ studio |
| ◆ MPAA rating | ◆ plot summary |
| ◆ Year | ◆ Video type (of which they can select more than one (DVD, Blu-ray, 4K Disk, Digital SD, Digital HD ,Digital 4K), |
| ◆ running time | ◆ allow the user to upload a cover image. |
| ◆ theatre release | |

The Display Details Page (displaydetails.html) display all the details for a particular movie, nicely formatted. (Eventually the movie info will come from the database, and be dynamic based on the movie selected...but for now hard code one to get your layout down)

The Search Page (search.html) this page contains a form allowing the user to specify what term to search for, and a grid of the search results of the search. (Eventually the movie info will come from the database...for now hard code some in to get the layout down)

The Login Page (login.html) Form to allow the user to login to an existing account. Should ask for username and password, and have a checkbox for “remember me” and a forget password link.

Note: You files can actually be php files at check-in if you want to simplify the layout changes by including a header and footer.

Check-in Two

For the second check-in period you should have completed the back-end functionality for all pages, during which you should consider the following:

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 navigation should include some sort of “you are here” indicator showing which page they’re currently on

Create account (account.php)

- Should validate collected information
- Properly 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.
-

Edit Account (editaccount.php)

- Note: front-facing page is a duplicate of the create account page
- 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

Delete Account (deleteaccount.php)

- Delete media and user information for the logged in user
- Destroy the cookie and session.

The Login Page (login.php)

- The cookie if it exists on the login page should populate the username in the textbox
- On submission: Validates the user credentials against the database
- On successful login
 - ◆ Creates a session (including the user’s database id). Note: Putting the user’s id in session will allow this to be included when inserting items.

- This session is then checked on all other relevant pages to make sure the user is logged in (otherwise send them to the login page).
- ◆ Set a cookie where appropriate (remember me). (store the username)
- ◆ Redirect to main on successful login
- If unsuccessful, display relevant errors

Log Out (logout.php)

- Destroy session and cookie
- Redirect to login page

The Main Page (index.php)

- Should dynamically display the user's media, with each item having appropriate info, edit, delete links (the icons) with correct item id in the url.
- *Advanced Topic:* Page should be limited to a reasonable number of items based on layout, and should have functioning previous and next links to show more content (Hint: you'll need to use LIMIT in your SQL, and pass the relevant limit values in the URL)

Add Item (addvid.php)

- Validate info and add to database
- Process image upload if necessary (move and upload)
- Redirect back to item details for the added media

Edit Item (editvid.php)

- 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 (deletevid.php)

- Delete item from database
- Redirect back to main

Display Details Page (displaydetails.php)

- Display all the details for the current media item, using existing formatting. Item id should be retrieved from the URL.

Search (search.php)

- Perform filtered query (search term should be run against title), and display matching search results (searched term should be sticky)
- *Advanced Topic:* Add additional search options (Category, plot, actor)

Security – make sure to show appropriate consideration for the security of your application. (password protection, sql_injection, cross site scripting)

Note: all front-facing forms should be self-processing to improve usability of error display

The finished project will also include a significant javascript component, which must be included by the final due date. Details on this part will be added shortly (I need to test some things first).