

Trent University
COIS 3420H
Assignment #1

The assignments in this class will build on top of one another to complete part of a small web application, and incorporate many of the ideas we talk about in class. The application itself is based on the following premise:

You have been "hired" to build a web application that allows users to track a video collection. The initial stage of the project involves writing the HTML and CSS for all the requisite pages. The application will involve the following:

- **Create Account** (register.html) – This page allows the user to create an account in the application.
- **Edit Account** (profile.html) – This page allows the user to edit their account credentials. (note: for Stage 1: just a duplicate of the contents of the create screen)
- **Delete Account** (leave.php) – this page will be empty in stage 1, since it has no html and css
- **The Main Page** (index.html) will display all the movies in the collection in a 3X4 table (12 movies at a time), with previous and next links to show the next/prev 12 movies (eventually the movie info will come from the database...for now hard code enough in to get the table structure worked out). Underneath each movie is an image link to the edit, delete, and details page for that movie
- **The Add Page** (addVid.html) – allows the user to enter details for a new movie
- **The Edit Page** (editVid.html) - allows the user to edit details of an existing movie (note: for stage one, this will contain the same html and CSS as the add page)
- **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)
- **The Delete Video Page** (deleteVid.php)- this page won't exist in stage one, since it has no front-end HTML & CSS
- **The Search Page** (search.html) this page contains a form allowing the user to specify what term to search for, and a four column table displaying the results of the search. (Eventually the movie info will come from the database...for now hard code some in)
- **The Login Page** (login.html) Contains a form which collects login credentials
- **The Logout Page** (logout.php) - this page will be empty in stage 1, since it has no front-end HTML & CSS

Stage One – XHTML/CSS

For index.html, addVid.html, displayDetails.html, search.html and login.html, use the images provided by the project management team (each page has a unstyled HTML view, as well as a styled view, all are located on blackboard), to generate the necessary HTML files and the CSS file.

For register.html, incorporate the same style used for the rest of the site, but design your own form layout. Information collected should include: username, first & last name, email, password, gender, age, and email preferences

For profile.html and editVid.html, duplicate the register.html and addVid.html pages respectively, rename and link appropriately.

For leave.php, deleteVid.php, and logout.php, just create and link the pages...for now they'll be empty.

Project Documentation:

In industry, proper project documentation is typically required. This helps deal with staff training/turnover, track issues and their solutions. It also, in theory, helps prevent similar issues from cropping up on subsequent projects. With your code submission, you will be required to include the following three documents:

Variable/label naming convention:

In the process of writing your code you will find that you require names for various elements. These will include, but are not limited to, the names of IDs and CLASSES as well as variables on forms as to/from the database. Your naming convention should make it possible to distinguish between these types of elements(i.e. if you looking at a CLASS or an ID or a variable name holding data from your database) . Proper naming conventions and adherence to those standards will help prevent issues with your code. For this portion of the assignment a simple description of the rules governing how names are created is sufficient.

e.g.

All "sessions" variables are in upper case.

data values extracted from the database are prefixed with "db_"

Data "Dictionary":

A complete list of your variable names and how they are used in your application. Your list should be divided into the appropriate groupings based on your naming convention.

e.g.

SESSION_USER - once login credentials have been validated, this field will contain the username used to log into the web application.

Issues/Solutions Document:

Every time you work on your code, document your work in the issues log. Create an entry when you make a significant decision about your project. This could include deciding on field sizes and database structures. Document when you encounter any problem. This would include spelling errors on variable names, forms data not getting to your processing scripts and database connection issues. Anything that at one point in time affected the functioning of your application should be logged. You should also log the solutions to each issue.

e.g. (maybe not quite this much info but as an example...)

14-Sep-2015 5:15pm: Unable to get the password field contents from the submitted data. I do however get the username OK.

14-sep-2015 5:35pm: D'oh! Typo in the field name on the receiving end. Didn't capitalize the name.

These documents should just be flat ASCII text files. No special formatting is required. Updated versions of these documents are required for each of the three assignment submissions.

Important Notes:

All HTML pages must validate to XHTML 1.0 Strict. (<http://validator.w3.org/>)

CSS must validate to CSS 3 standards. (<http://jigsaw.w3.org/css-validator/>)
or if your page is not web accessible: http://jigsaw.w3.org/css-validator/#validate_by_upload

ALL CSS must be in an external style sheet

Although they aren't visible in the screenshots, all form inputs must be contained within a form element. This is necessary for the forms to work properly later on

The menu should function correctly and link to the appropriate pages. The alternating color visible on the logout link (in index.php) is the hover effect which should be applied to all menu links on hover (the screen cap removes the mouse). Also note the style applied to the current page link as a "you are here indicator"

You'll have much better luck with the CSS if you start your CSS file by importing a reset file (<http://meyerweb.com/eric/tools/css/reset/>)

Your styled version must be tested cross browser and cross platform. Test with recent versions of IE, Firefox, Chrome, and Safari on your local machine. You can test on a Mac here in the Lab if you'd like. You can also use <http://browsershots.org/> or <http://browserstack.com> (if you got the free trial) to test other operating systems, and in particular older versions of IE. If you are unable to fix cross browser problems, document them in a testing file and include it with your submission

Marking Criteria

You will be marked both on your ability to create the correct pages (visually matching the screen captures provided), as well the quality of your actual HTML and CSS (taking into account standards, semantics, etc). There will also be marks for your code documentation and readability. The three project documentation text files are a mandatory part of your submission.

Submission

Compress (zip or rar) the completed directory structure (html files should be in root folder, images in an img folder, and css in a style folder), and submit the compressed file.

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.

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