

CE154 Assignment

This assignment is to create a simplified games catalogue, similar to [steam](#), [itch.io](#) or [good old games](#). The deadline for this assignment can is on faser.

Submission

Before the deadline, you should submit a single zip file (.zip) to faser. The submission should be a zip file: a RAR file is not a zip file, neither is a 7zip (.7z) file. The file should be named `ce154_username.zip`, where username is your Essex username.

In **addition** to the FASER submission, your website **must** be fully operational on the CSEE MyWeb service. The CSEE MyWeb service (W drive) will automatically become read-only after the deadline. You will not be able to upload after this time. Should the server not become read-only, then any files submitted after the deadline will not be counted.

WARNING

The work you submit must be your own. Any material you use, whether it is from textbooks, classmates, the web or any other source, must be acknowledged in your work. Also, you are assumed to have read the booklet at http://www.essex.ac.uk/plagiarism/docs/Plagiarism_and_how_to_avoid_it.pdf

Rules

- This assignment is to be submitted individually. **This is not a group assignment**
- Be sure to put your registration number as a comment at the top of all code and other files.
- Put your registration number near the top of all of your text (HTML, PHP and CSS) files in a comment - this should not make the page invalid.

NOTE

Please use the filenames as indicated in the task descriptions; otherwise, the marker may miss your work, and your work may not be marked.

NOTE

You are free to use any software you like for this assignment; however, we cannot provide support for packages other than Notepad+, Firefox, Web Development Tools (Firefox add-on) and a Windows PC.

WARNING

if you auto-generate code (such as using an IDE to design the layout), your marks could be adversely affected.

The brief

For this assignment, you will design a website using PHP, MySQL, HTML, CSS and (possibly) javascript. Pay careful attention to the requirements outlined below. I have provided examples of similar services in the introduction paragraph to this document.

For this assignment, you must write your server-side code in PHP, and use MySQL for your database. You can customise the SQL script to suit your needs, but you should not remove existing data. I have provided a sample database design with this document to get you started. **Your submission should contain the (working) reset script.**

The website will consist of four key parts:

- A list of games (the games catalogue)
- Game Reviews (submitted by users)
- Game Bookmarks (a simplified *wishlist*)
- User accounts

Functional Requirements (60%)

Your website should have the following functions:

Games Catalogue

The catalogue should be a page that allows users to view a list of games. This page should display (at least) the names and images of the games.

The list of games should be filterable by users:

- The users should be able to filter by different criteria (e.g., genre)
- The users should be able to search based on name (e.g., *Shadowrun* should return all of the Shadowrun games).

TIP

You can implement this however you like, but a sidebar with the options would be a sensible way to do this.

Clicking on one of the games should open a page describing the game, along with the reviews associated with the game. You should use a GET request to pass the game ID.

On the game details page, the game should display the details about the game from the database. The page should also display reviews left by logged in users.

Logged in users should be given the option to create a new review for the game if they have not reviewed it before. If the user has reviewed the game previously, the user should be able to modify their review.

Both of these pages should use the database to populate the pages.

Reviews and Bookmarks

A logged-in user should be able to post a review for the game. If the user

already has a review for the game, they should edit the review instead of creating a new one. A user

should **not** be able to post two reviews for the same game.

You can put the review form on its own page or put it at the bottom of the game details page.

A user should also be able to *bookmark* a game from the game detail page. The bookmark button should show up only if the user is logged in. Bookmarked games should be displayed on a page that contains their bookmarked games. From this page, they should be able to remove a bookmark.

- Users should not be able to see other user's bookmarks.
- A user should not be able to bookmark a game twice.

User management

A user should be able to log in and out using forms on your website. If logged in the name of the user should be shown on every page.

An admin account (an account with `is_admin` database field set to 1 in the database) should be able to create new games from the homepage. You do not need to support file uploads - asking the user for the URL of the image is sufficient.

Extra Functions (10\%)

You may choose to include extra features into your design. If you include extra features please add a document detailing what features you have added. The marker may award up to a bonus 10\% (mark capped at 100\%) for additional features.

If you are not sure if an additional feature is suitable, you can ask the lecturer (Joseph) in the labs or via email.

Non-functional requirements (40%)

There are some non-functional requirements for your assignment.

Code Correctness

The website should use valid HTML and CSS. Menu links should work correctly, and menus should be consistent on all pages (except or any links that should only work for admin/logged-in users).

The website should not crash, or display 500 errors during evaluation.

Website Style

The website should have a professional (clean) appearance. What a professional website looks like is somewhat open to interpretation, and the markers opinion will be final (based on their professional experience). You should aim to use a colour scheme, design and font choices that are suitable for the website.

The website theme should be consistent between pages, with all pages having a unified look and feel.

Website Correctness

Your website should use suitable conventions for website file names (lowercase, no spaces in filenames). The homepage should be named `index.php` and the folder structure should be suitable (e.g, assets folder for resources).

This category also includes submitting documents in the correct format.

Security

You should implement some basic security practices:

- Reviews cannot contain HTML - any HTML could be escaped
- Forms should protect against XSS, where appropriate
- Session management should use the PHP session management tools
- Admin-only pages should not be accessible if you are not logged in as an admin.
- User only features should only work if logged in.