

## CSG2431: Interactive Web Development

**Assignment 2:** Advanced Scripting/Database Website (Movie Discussion Site)  
**Assignment Marks:** Marked out of 25 (25% of unit)  
**Due Date:** 5 November 2018, 9:00AM (if working in a pair, inform tutor by 9AM Oct 8<sup>th</sup>)

### Contents

Assignment Overview .....	1
Access Level Changing.....	2
Moderator User Level .....	3
Site Statistics .....	5
Advanced Search Capabilities .....	6
Sample Content / Submission of Deliverables.....	7
Referencing, Plagiarism and Collusion.....	7
Marking Key .....	8

### Assignment Overview

In this assignment, you build upon the movie discussion site you developed in Assignment 1. You will be adding some more features and functionality, and a new access level – moderators.

A lot of the work you will do in this assignment will involve *modifying your files from Assignment 1*, rather than creating new files. The database will also need to be expanded in some places.

You **may work in pairs** to complete this assignment, or work alone. If working in a pair you must email the names/student numbers of your pair to your tutor by 8 October 2018, 9AM. If you work alone, be aware that there are *no extra marks* to compensate for the heavier workload. *Whether you work in a pair and who you work with may differ from Assignment 1 if desired/necessary.*

## Access Level Changing

When an admin user views a user profile, the page should now include a form to change the user's access level. Only admin users should have access to this functionality.

MovieTalk															
<p>You are logged in as asmith (<a href="#">Logout</a>)</p> <p><input type="text"/> <input type="button" value="Search"/></p> <p><a href="#">List All Movies</a></p> <p><a href="#">Add Movie</a></p> <p><a href="#">List all Users</a></p> <p><a href="#">Edit My Details</a></p>	<h3>Details of "jbloggs"</h3> <table border="1"><tr><td>Real Name:</td><td>Joe Bloggs</td></tr><tr><td>Email Address:</td><td>jbloggs@gmail.com</td></tr><tr><td>Year of Birth:</td><td>1982</td></tr><tr><td>Country:</td><td>Australia</td></tr><tr><td>Access Level:</td><td>Member</td></tr><tr><td>Banned Until:</td><td>Not currently banned</td></tr><tr><td>Ban Reason:</td><td>Not currently banned</td></tr></table> <p>Change access level to: <input type="text" value="Admin"/> <input type="button" value="Change"/></p>	Real Name:	Joe Bloggs	Email Address:	jbloggs@gmail.com	Year of Birth:	1982	Country:	Australia	Access Level:	Member	Banned Until:	Not currently banned	Ban Reason:	Not currently banned
Real Name:	Joe Bloggs														
Email Address:	jbloggs@gmail.com														
Year of Birth:	1982														
Country:	Australia														
Access Level:	Member														
Banned Until:	Not currently banned														
Ban Reason:	Not currently banned														

*Example of page showing user details including a form to change the user's access level  
(the example includes details of bans, which are detailed below)*

The form should allow the access level to be changed to "member", "moderator" (detailed below) or "admin".

Make sure that it is not possible to change *all* admins to members/moderators – there should be a *minimum of one admin user* at all times. Enforcing this will involve using a query to check how many admin users there are when trying to change an admin user to a different access level, and only proceeding if there are at least two admin users.

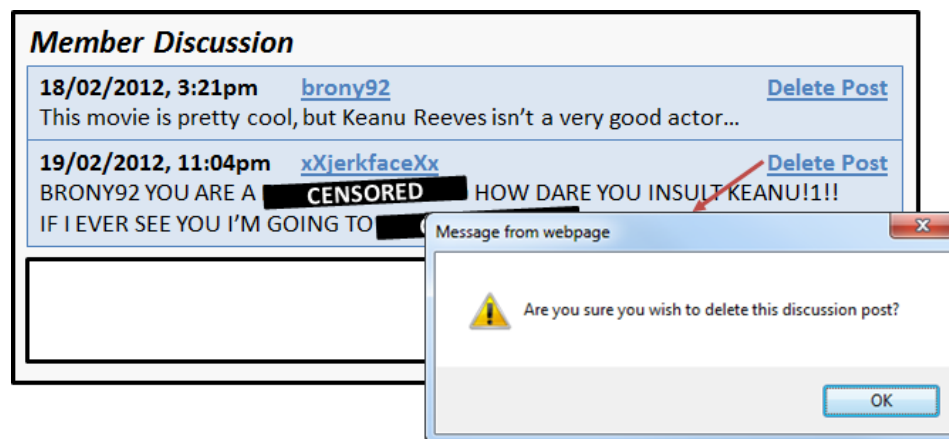
## Moderator User Level

The website needs a new type of user – moderators. Moderators have the same abilities as members – e.g. they can participate in discussion, rate movies, view profiles and update their user details. They also have the following two abilities (which are *not* available to members or admins):

- **Delete any discussion posts** which they feel are inappropriate
- **Ban/Unban members** who they feel have acted inappropriately

Moderators should also have access to the page listing all users, previously available only to admins.

When a moderator is logged in and viewing discussion about a movie, a “Delete Post” link should appear for each piece of discussion. Clicking the link should delete that piece of discussion. Remember to include a confirmation prompt before deleting the discussion.



*Example showing the deletion of a movie discussion post*

Moderators can delete posts by any user, regardless of if they are a member, moderator or admin.

Moderators can also *ban* members – but not moderators or admins. To ban or unban a member, a moderator must go to the view user profile page. Below the user details there should be a form allowing them to specify how many hours to ban the user for and provide a reason for the ban.

To store details of bans, add a “banned\_until” DATETIME column and a “ban\_reason” VARCHAR column to the “users” table of the database. Make sure they can both be NULL/empty. When the banning form is submitted, update those columns for the appropriate user. MySQL’s [DATE\\_ADD](#) function can be used to determine when the ban should last until. For example:

```
UPDATE users SET banned_until = DATE_ADD(NOW(), INTERVAL 24 HOUR),  
ban_reason = 'Swearing and threats' WHERE username='xXjerkfaceXx'
```

This query would set the ban\_reason column to “Swearing and threats” and the banned\_until column to the current date plus 24 hours for the user with a username of “xXjerkfaceXx”.

The details of bans should be included in the view user profile page – any logged in user should be able to see them. If the user has never been banned (i.e. their banned\_until and ban\_reason columns are NULL) or if the user was banned but the ban is over, show “Not currently banned”.

To determine if a ban is over, use the [strtotime\(\)](#) and [time\(\)](#) functions in PHP. For example:

```
if ( strtotime($user_details_row['banned_until']) < time() )
{
    echo 'Not currently banned';
}
```

Example of comparing a ban date to the current time in PHP. Note that this is only a small/simple example

There should also be a link or button on the view user profile page to unban a member. When this is clicked, the “banned\_until” and “ban\_reason” columns should be set to NULL for that user.

MovieTalk															
<p>You are logged in as bjones (<a href="#">Logout</a>)</p> <p><input type="text"/> <input type="button" value="Search"/></p> <p><a href="#">Advanced Search</a></p> <p><a href="#">List All Movies</a></p> <p><a href="#">List all Users</a></p> <p><a href="#">Edit My Details</a></p>	<h3>Details of “xXjerkfaceXx”</h3> <table border="1"> <tr> <td>Real Name:</td> <td>Jerkface McGee</td> </tr> <tr> <td>Email Address:</td> <td>jerkface@rudepeople.com</td> </tr> <tr> <td>Year of Birth:</td> <td>1998</td> </tr> <tr> <td>Country:</td> <td>Australia</td> </tr> <tr> <td>Access Level:</td> <td>Member</td> </tr> <tr> <td>Banned Until:</td> <td>Not currently banned</td> </tr> <tr> <td>Ban Reason:</td> <td>Not currently banned</td> </tr> </table> <p><b>Ban User</b>            Ban member for: <input type="text" value="24"/> hours.            Reason for ban: <input type="text" value="Swearing and threats"/> <input type="button" value="Ban"/></p> <p><b>Unban User</b>            Click <a href="#">here</a> to remove any existing ban for this user.</p>	Real Name:	Jerkface McGee	Email Address:	jerkface@rudepeople.com	Year of Birth:	1998	Country:	Australia	Access Level:	Member	Banned Until:	Not currently banned	Ban Reason:	Not currently banned
Real Name:	Jerkface McGee														
Email Address:	jerkface@rudepeople.com														
Year of Birth:	1998														
Country:	Australia														
Access Level:	Member														
Banned Until:	Not currently banned														
Ban Reason:	Not currently banned														

Example of the moderator version of the view user profile page – xXjerkfaceXx is about to be banned

To enforce a ban, you will need to modify the code used to process the login form. If the query that checks for a matching username and password returns a row, check if the banned\_until column is less than the current date – see the code example above for how to do this.

If the banned\_until column is *less* than the current date (if it is NULL, it will automatically count as being less), proceed with the login as normal – set the session variables and redirect the user. If the banned\_until column is *more* than the current date, show a message saying that the user has been banned. The message should include the date/time that they’ve been banned until and the reason.

## Site Statistics

Admin users need access to a new page which will show various statistics drawn from the content of the database. These statistics should be determined via SQL queries whenever the page is accessed – you do *not* need to change the database or store any additional data regarding the statistics.

The page should present the following statistics:

- How many members, moderators and admins there are, and the total number of users
- The average age of all users, calculated from their year of birth
- How many movies there are in the database
- The name of the movie with the most discussion, and how many discussion posts it has
- The name of the movie with the highest average rating, and what that average rating is

MovieTalk											
<div>You are logged in as asmith (<a href="#">Logout</a>)</div> <div><input type="text"/> <input type="button" value="Search"/></div> <div><a href="#">Advanced Search</a></div> <div><a href="#">List All Movies</a></div> <div><a href="#">Add Movie</a></div> <div><a href="#">List all Users</a></div> <div><a href="#">Statistics</a></div> <div><a href="#">Edit My Details</a></div>	<div>Site Statistics</div> <table><tr><td>Users:</td><td>Total: 21 (15 Members, 4 Moderators, 2 Admins)</td></tr><tr><td>Average User Age:</td><td>29</td></tr><tr><td>Number of Movies:</td><td>102</td></tr><tr><td>Most Discussed Movie:</td><td>Jaws (12 discussion posts)</td></tr><tr><td>Highest Rated Movie:</td><td>Avatar (9.2 average rating)</td></tr></table>	Users:	Total: 21 (15 Members, 4 Moderators, 2 Admins)	Average User Age:	29	Number of Movies:	102	Most Discussed Movie:	Jaws (12 discussion posts)	Highest Rated Movie:	Avatar (9.2 average rating)
Users:	Total: 21 (15 Members, 4 Moderators, 2 Admins)										
Average User Age:	29										
Number of Movies:	102										
Most Discussed Movie:	Jaws (12 discussion posts)										
Highest Rated Movie:	Avatar (9.2 average rating)										

*Example of the statistics page*

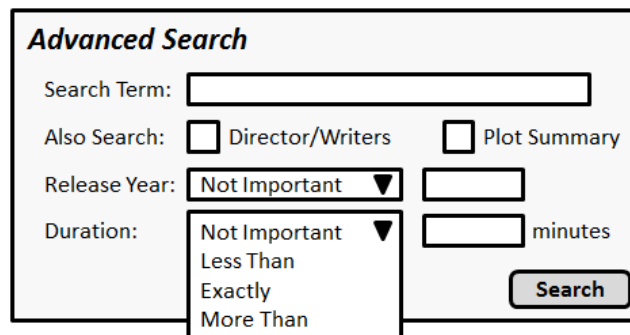
These statistics will involve writing SQL queries that include [aggregate functions and the GROUP BY clause](#). Aim to write efficient queries and minimise the number of queries needed overall.

You may need to search for examples or tutorials regarding this, or refer to the content of CSG1207.

## Advanced Search Capabilities

Previously, all users could search for movies in the database by movie name only. This feature should remain, however all users should also be presented with a link to an “Advanced Search” page. This page should allow for the following features:

- A text field to type a search term into (all searches will search the movie name column)
- A check box to search for the term in the director and writers columns as well
- A check box to search for the term in the plot summary column as well
- A drop down box and text field to specify the release years to search
- A drop down box and text field to specify the movie durations to search



The image shows a web form titled "Advanced Search". It contains the following fields and controls:

- Search Term:** A text input field.
- Also Search:** Two checkboxes labeled "Director/Writers" and "Plot Summary".
- Release Year:** A dropdown menu currently showing "Not Important" and an empty text input field.
- Duration:** A dropdown menu with options "Not Important", "Less Than", "Exactly", and "More Than". It is currently open, showing these options. Next to it is an empty text input field followed by the word "minutes".
- Search:** A button at the bottom right of the form.

*Example of the advanced search form*

The search term is the only required field. If an option other than Not Important is chosen for the release year or duration drop down lists, then the corresponding text field is required. The drop down lists for specifying a release year or duration should contain four options: Not Important (the default), Less Than, Exactly and More Than. If the form is submitted with Not Important selected, it does not become part of the search. If any of the other options are selected, the values in the drop down list and text field are used in the search, e.g. “release\_year = 2005”.

The form processing code for the form will involve using conditional statements to check which fields have been checked/selected/filled in, and generating an appropriate search query. If needed, research the WHERE clause of the SELECT statement, or refer to the content of CSG1207.

The example below shows an appropriate search query. The code in orange is what would be run if only a search term was entered. The code in green would be added as needed based on what the user has selected in the form. Note the position of the parentheses in the WHERE clause.

```
SELECT * FROM movies
WHERE (name LIKE '%into%'
      OR director LIKE '%into%' OR writers LIKE '%into%'
      OR summary LIKE '%into%')
AND release_year = 2005
AND duration < 150
```

*If Director/Writers checkbox ticked*  
*If Plot Summary checkbox ticked*  
*If Release Year fields completed*  
*If Duration fields completed*

*This example would search for movies with “into” in the movie name, director/writers or summary, that were released in 2005, and have a duration of less than 150 minutes.*

## Sample Content / Submission of Deliverables

Make sure that your database contains enough data for me to test the new features.

Once your assignment is complete, re-read the assignment brief to make sure you have implemented everything that is asked for. Then, submit the following in a zip/rar file:

- **The folder inside your htdocs folder which contains your assignment code files.** Make sure you have used relative paths for any links in your assignment – I should *not* need to deploy your assignment into the same folder structure that you developed it in when marking
- **An SQL file of your database, exported using phpMyAdmin** (see [this video](#))
- **A text file named readme.txt containing the login details for all users in your database, and references for any external code or resources you have used**

Include **your name and student number** in the file name of the zip/rar file. Upload the file to the Blackboard assignment submission area on or before the assignment due date and time.

**If working in a pair**, include the **names and student numbers of both students in the readme.txt file** and your **submission comment on Blackboard**. Both students should submit the same zip/rar file.

*NOTE: I will be marking your assignments with the same XAMPP setup used for this unit. I should be able to simply drop your code folder in my htdocs folder and import your database file and start marking. Assignments are marked **AS IS** – if there are strange configuration issues or code errors that cause some or all of the functionality to fail, marks will be subtracted accordingly. Always test what you are planning to submit, particularly if you have used a different environment than XAMPP.*

## Referencing, Plagiarism and Collusion

Web development often relies on new developers finding examples of code that performs a task that they need, so that they can understand how it works. Many websites provide freely available code snippets, functions and tutorials. Learning from and using these are perfectly legitimate, *as long as it is done within the university rules*. Blackboard contains links to guides on plagiarism, referencing and academic misconduct. Any code you use (even if you modify it) must be fully referenced. Any code or interface features found in your assignments that have not been referenced, or have been written by someone other than yourself will typically attract zero marks and a report of academic misconduct on your university record.

Also, while helping your friends and learning together is great, be careful about working too closely with others on your assignment. **NEVER SHARE YOUR CODE** – EVEN AFTER THE DUE DATE. If multiple people submit code which is identical or almost the same, it is collusion. Collusion will be penalised and result in a report of academic misconduct on your university record. It is important that **you** are the author of your code, and understand it. If you are uncertain about plagiarism, collusion or referencing, simply email your tutor, lecturer or unit coordinator and ask.

## Marking Key

Criteria	Marks Allocated
<b>Access Level Changing</b> See details in assignment brief.	4
<b>Moderator User Level</b> See details in assignment brief.	9
<b>Site Statistics</b> See details in assignment brief.	6
<b>Advanced Search Capabilities</b> See details in assignment brief.	6
<b>Total</b>	<b>25</b>