

Database Evidence

Working website: <https://php.mmc.school.nz/13/shilohwarner/Database/>

Trello board: <https://trello.com/b/lx3lIBsx/moviedatabase>

The project on Github: https://github.com/shiloh25/movie_database

Planning

Add information about the purpose of your website/ database, who the target audience will be and some of their requirements

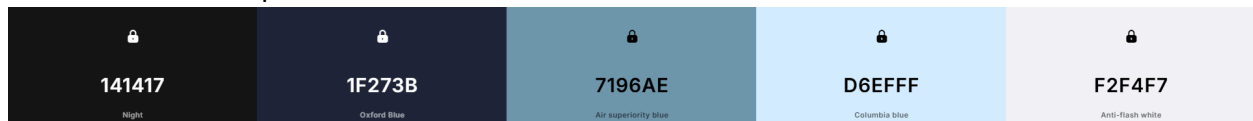
Colours:

Potential palette options:



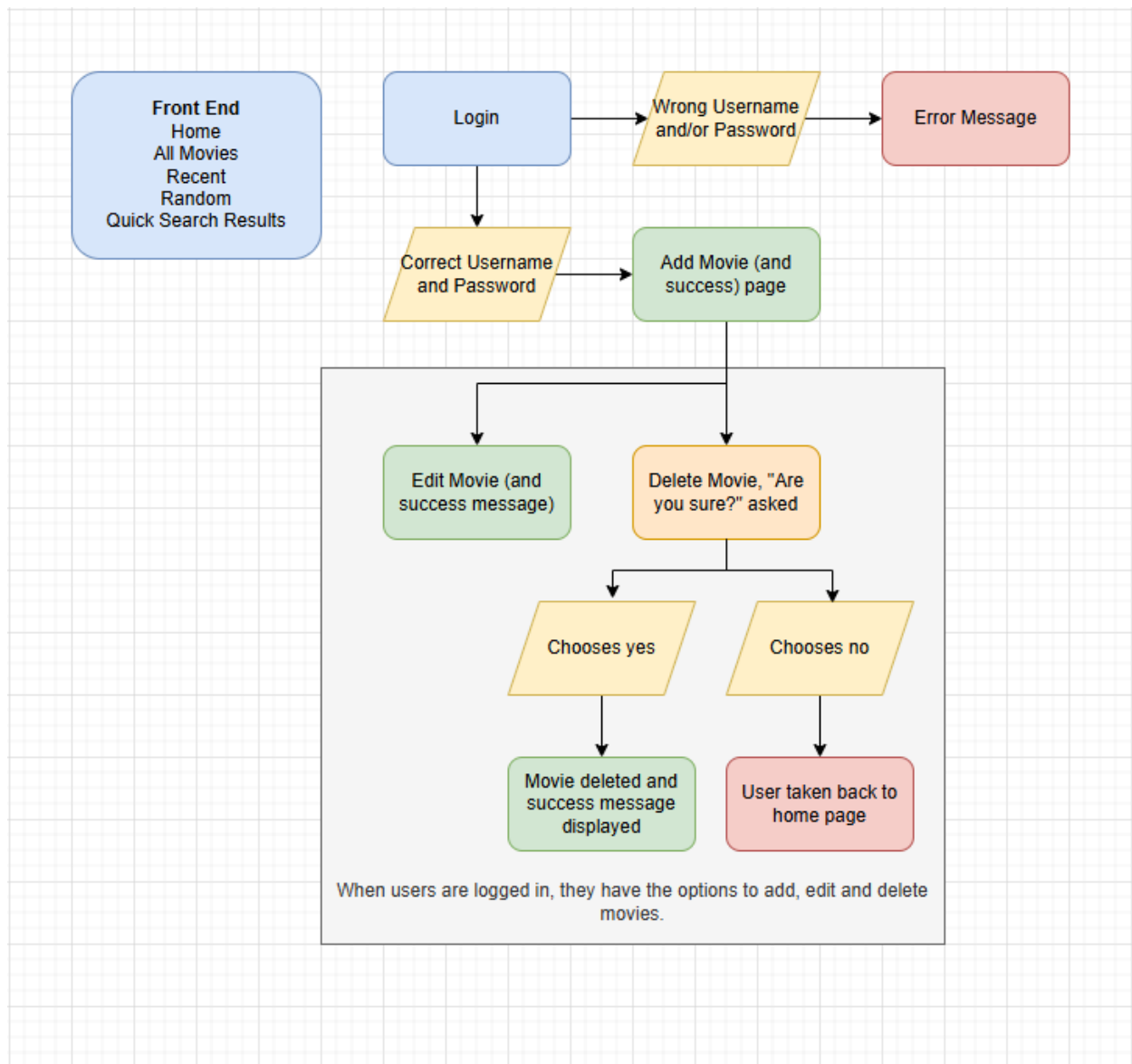
Chosen colour scheme:

I choose this as my final colour scheme because of a few different factors. Firstly, it fits my original criteria of wanting to use a colour scheme that is somewhat associated with popular movie streaming platforms as these colours will have built in associations in the user's mind so they will be relevant. Secondly, I wanted the colour scheme to be colourful but also not too overwhelming as there will already be a lot of information on the interface so having bright colours could reduce the aesthetics. This eliminated the first two palettes as options. I ended up decking on the final colour scheme by getting opinions from end users, particularly my friends. They agreed that the darker blue colour looked better than the grey colour in terms of fitting in with the rest of the palette and so I chose these colours.



Name: Shiloh Warner

Flowchart:



Wireframes:

Below are the wireframes for the layout of my database on the website interface that the users will see. This design is based on the colour scheme decided on earlier, and the general structure of the website from the practise task. The wireframe contains the necessary features for the database to function including navigation and search bars, login options, and a screen to add movies. Edit and delete buttons will be added once the user is logged in.

The image displays three wireframe screens for a website titled "Quick Movies".

Top Wireframe (Main Page): This screen features a dark blue header with the "Quick Movies" title. Below the header is a navigation bar with links for "All Movies", "Recent", and "Random". To the right of the navigation bar is a search bar with a "Select" dropdown menu, a magnifying glass icon, and a "Go" button. The main content area is titled "All Movies" and contains four placeholder boxes, each with the text "Lorem ipsum dolor sit amet, consectetur adipiscing elit."

Bottom Left Wireframe (Login Page): This screen has a blue background and is titled "Login". It includes a "Username:" label, a text input field with "Username..." as a placeholder, a "Password:" label, a text input field with "Password..." as a placeholder, and a "Log In" button.

Bottom Right Wireframe (Add Movie Page): This screen has a blue background and is titled "Add Movie". It includes a "Movie Title (Required)" label, a text input field, a "Director Name (First, Last)" label, a text input field, and an "Add Movie" button.

Database Design

Add design information about the tables that are used, the data type and the relationship between them.

Example tables:

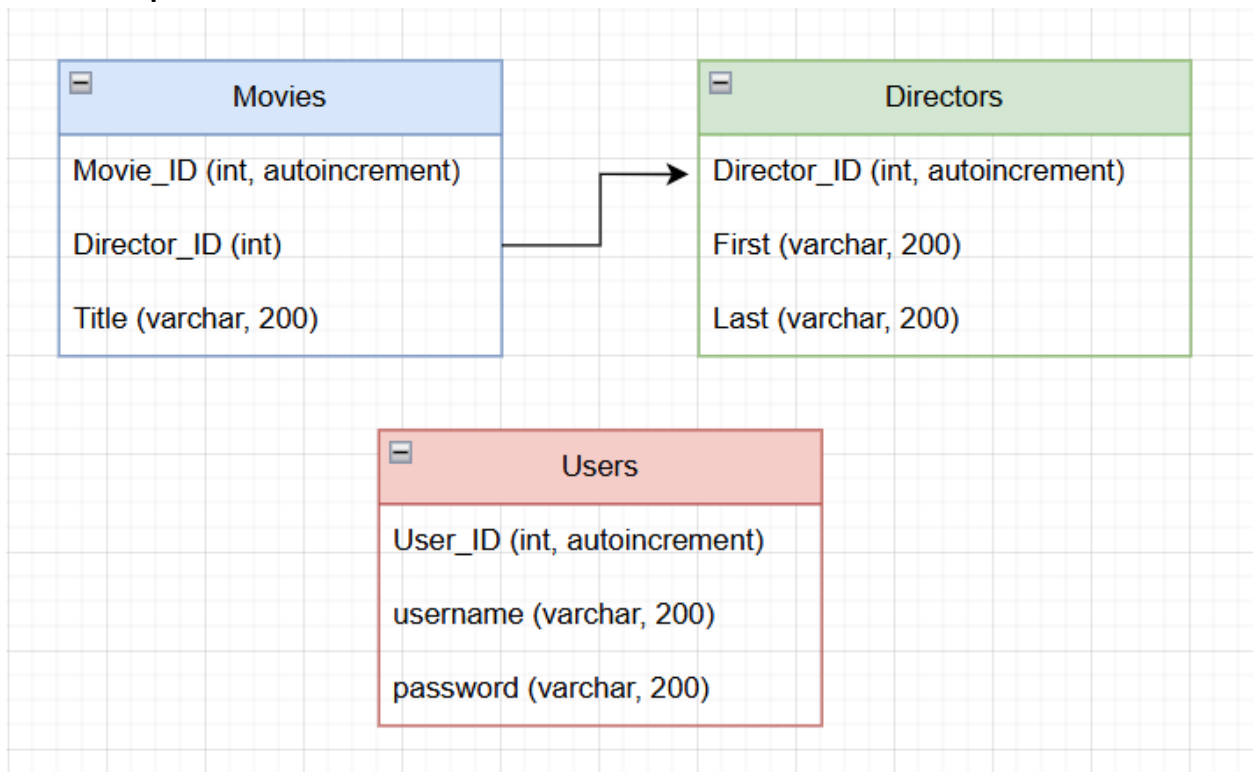
In my practise task, there was a row for first, middle and last name in the table. While I was gathering my data, it became clear to me that no directors were known by their middle names, it was all just first and last names. Due to this, I decided to leave out the middle names row from my data table in order to reduce this kind of redundant information from being in my database.

Name	Data Type
Director_ID	int
First	varchar
Last	varchar

Name	Data Type
Director_ID	int
Movie_ID	int
Title	varchar

Name	Data Type
User_ID	int
username	varchar
password	varchar

Relationship Tables:



Development

Database setup:

Place screenshots of all your tables used in your database

The left screenshot shows the 'movies' table structure in phpMyAdmin. The table has three columns: Director_ID (int), First (varchar, 200), and Last (varchar, 200). The right screenshot shows the 'movies' table data in phpMyAdmin. The table has three columns: Movie_ID (int), Director_ID (int), and Title (varchar, 200). The data is as follows:

Movie_ID	Director_ID	Title
1	1	West Side Story
2	2	Ready Player One
3	3	BFG
4	4	Jaws
5	5	Jurassic Park
6	1	The Lost World: Jurassic Park
7	2	Little Women
8	2	Barbie
9	2	Lady Bird
10	3	Avatar
11	3	Avatar: The Way of Water
12	3	Titanic
13	3	The Terminator
14	3	Terminator 2: Judgement Day
15	4	Clueless
16	4	Loser
17	5	The Hobbit: The Battle of the Five Armies
18	5	The Hobbit: The Desolation of Smaug
19	5	The Hobbit: An Unexpected Journey
20	5	The Lord of the Rings: The Return of the King
21	5	The Lord of the Rings: The Two Towers
22	5	The Lord of the Rings: The Fellowship of the Ring
23	6	Wonder Woman
24	6	Wonder Woman 1984
25	7	Harry Potter and the Order of the Phoenix

Name: Shiloh Warner

	User_ID	username	password
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	admin	\$2y\$08\$UZsTr4efri2K8.jGfMeVVOeqjRaCP1QWEeQNrBuBDk...
<input type="checkbox"/> Check all	With selected: <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete <input type="checkbox"/> Export		
<input type="checkbox"/> Show all	Number of rows: 25	Filter rows: Search this table	

Database SQL Testing & Trialing:

Add screenshots showing different SQL queries that you are trialing and testing

Director SQL Testing:

SQL Query	Outcome
SELECT * FROM `director` WHERE Director_ID = 1;	<p>Director ID search for 1 which is Steven Spielberg, this is the result that shows up so it is working.</p>  <p>The screenshot shows the SQL query results for Director_ID = 1. The query is: <code>SELECT * FROM `director` WHERE Director_ID = 1;</code>. The results show one row: Director_ID 1, First Name Steven, Last Name Spielberg. The interface includes a status bar at the top indicating 'Showing rows 0 - 0 (1 total, Query took 0.0001 seconds.)', a toolbar with 'Edit', 'Copy', and 'Delete' buttons, and a table view with columns 'Director_ID', 'First', and 'Last'.</p>
SELECT * FROM `director` WHERE Director_ID = 4;	<p>Director ID search for 4 which is Amy Heckerling, this is the result that shows up so it is working.</p>  <p>The screenshot shows the SQL query results for Director_ID = 4. The query is: <code>SELECT * FROM `director` WHERE Director_ID = 4;</code>. The results show one row: Director_ID 4, First Name Amy, Last Name Heckerling. The interface includes a status bar at the top indicating 'Showing rows 0 - 0 (1 total, Query took 0.0001 seconds.)', a toolbar with 'Edit', 'Copy', and 'Delete' buttons, and a table view with columns 'Director_ID', 'First', and 'Last'.</p>
SELECT * FROM `director` WHERE CONCAT(First, ' ', Last) LIKE '%greta%';	<p>Director search for a first or last name of 'greta'</p>

	<div>Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)</div> <div>SELECT * FROM `director` WHERE CONCAT(First, ' ', Last) LIKE '%greta%';</div> <div>Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]</div> <div>Show all Number of rows: 25 Filter rows: Search this table</div> <div>Extra options</div> <div><div>Director_IDFirstLast</div><div><input type="checkbox"/> Edit Copy Delete 2GretaGerwig</div></div> <div>Check all With selected: Edit Copy Delete Export</div> <div>Show all Number of rows: 25 Filter rows: Search this table</div>
--	---

From these various tests of different SQL Queries tested in the director table of the database, I know that the searches for both Director_ID and Director Name are working as they should be and therefore these queries are ready to be implemented into the code.

Movie SQL Testing:

SQL Query	Outcome
SELECT * FROM `movies` ORDER BY Movie_ID DESC LIMIT 10;	<div>Retrieving 10 recent movies</div> <div>Showing rows 0 - 9 (10 total, Query took 0.0001 seconds.) [Movie_ID: 97... - 88...]</div> <div>SELECT * FROM `movies` ORDER BY Movie_ID DESC LIMIT 10;</div> <div>Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]</div> <div>Extra options</div> <div><div>Movie_IDDirector_IDTitle</div><div><input type="checkbox"/> Edit Copy Delete 9722Independence Day: Resurgence</div><div><input type="checkbox"/> Edit Copy Delete 9622Independence Day</div><div><input type="checkbox"/> Edit Copy Delete 95222012</div><div><input type="checkbox"/> Edit Copy Delete 9422Moonfall</div><div><input type="checkbox"/> Edit Copy Delete 9322White House Down</div><div><input type="checkbox"/> Edit Copy Delete 9221Red Sparrow</div><div><input type="checkbox"/> Edit Copy Delete 9121Sunderland</div><div><input type="checkbox"/> Edit Copy Delete 9021The Hunger Games: The Ballad of Songbirds & Snakes</div><div><input type="checkbox"/> Edit Copy Delete 8921The Hunger Games: Mockingjay - Part 2</div><div><input type="checkbox"/> Edit Copy Delete 8821The Hunger Games: Mockingjay - Part 1</div></div> <div>Check all With selected: Edit Copy Delete Export</div> <div>Query results operations</div> <div>Print Copy to clipboard Export Display chart Create view</div>

SELECT * FROM `movies` WHERE
Movie_ID = 1;

Movie ID search for 1 which is West Side Story. This is the movie that shows up so it is working correctly.

Showing rows 0 - 0 (1 total, Query took 0.0001 seconds.)

```
SELECT * FROM `movies` WHERE Movie_ID = 1;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

	Movie_ID	Director_ID	Title
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	West Side Story

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

SELECT * FROM `movies` WHERE
Movie_ID = 81;

Movie ID search for 81 which is Percy Jackson and the Olympians: The Lightning Thief. This is the movie that shows up so it is working correctly

Showing rows 0 - 0 (1 total, Query took 0.0001 seconds.)

```
SELECT * FROM `movies` WHERE Movie_ID = 81;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

	Movie_ID	Director_ID	Title
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	81	20	Percy Jackson & the Olympians: The Lightning Thief

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

SELECT * FROM `movies` WHERE Title
LIKE '%little%';

Movie search for 'little'

Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

```
SELECT * FROM `movies` WHERE Title LIKE '%little%';
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

	Movie_ID	Director_ID	Title
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	2	Little Women

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

SELECT * FROM `movies` WHERE Title
LIKE '%and%';

Movie search for 'and'

✓ Showing rows 0 - 12 (13 total, Query took 0.0003 seconds.)

SELECT * FROM `movies` WHERE Title LIKE `&and`;

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all
 Number of rows: 25
 Filter rows: Search this table
 Sort by key: None

Extra options

	Movie_ID	Director_ID	Title
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	25	7	Harry Potter and the Order of the Phoenix
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	26	7	Harry Potter and the Half-Blood Prince
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	27	7	Harry Potter and the Deathly Hallows: Part 1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	28	7	Harry Potter and the Deathly Hallows: Part 2
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	29	7	Fantastic Beasts and Where to Find Them
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	40	10	Nomadland
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	47	13	Edward Scissorhands
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	51	13	Alice in Wonderland
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	52	13	Charlie and the Chocolate Factory
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	76	19	Aquaman and the Lost Kingdom
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	82	20	Harry Potter and the Philosopher's Stone
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	83	20	Harry Potter and the Chamber of Secrets
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	91	21	Sunderland

☐ Check all
 With selected:
 ☐ Edit
 ☐ Copy
 ☐ Delete
 ☐ Export

Media Used

Add screenshots showing different linking SQL queries that you are trialing and testing

Free to use under the Unsplash License

Quick Movies

[Home](#)
[All Movies](#)
[Recent](#)
[Random](#)

All

▼

🔍

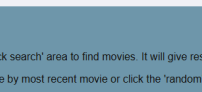
+

➡

Welcome

Please use the 'quick search' area to find movies. It will give results if your search term is in the movie or director name. It will also find movies related to a subject (try typing in something like 'love' or 'live').

You can also browse by most recent movie or click the 'random' link to get the system to generate ten movies for you.



CC shalohwarner 2025

All movie titles are copyright free and therefore I can use them freely without breaking any copyright laws. I did not use movie posters or any other visual or audio media associated with the films meaning the website is copyright free and will not face any copyright infringement or other legal consequences of using stolen media if it were to become a live website.

Iterative Process

Provide evidence you have used the iterative process to improve the database

For all test plans, evidence of results is shown in the final testing video.

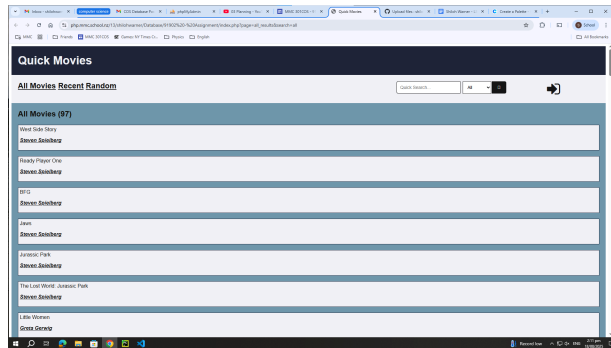
Test Plan for Searching:

Action/Input	Expected Result/Output	Actual Result/Output
User clicks 'Home'	Taken to the home page	Taken to the home page
User clicks 'All Movies'	All Movies show up in order based on the director ID with the number of movies next to it (eg. 97)	All Movies (97) title with all of the movies in the database listed below.
User clicks 'Recent'	The 10 most recently searched or selected movies show up.	The 10 most recently searched or selected movies show up.
User clicks 'Random'	10 random movies show up	10 random movies show up
User clicks 'Steven Spielberg'	All of Steven Spielberg's movies show up in a list	All of Steven Spielberg's movies show up in a list
User searches 'Harry Potter' in 'Movies'	All of the Harry Potter movies in the database show up in a list	All of the Harry Potter movies in the database show up in a list
User searches 'Gerwig' in 'Directors'	All of the movies directed by Greta Gerwig show up	All of the movies directed by Greta Gerwig show up
User searches 'The' in 'All'	All of the movies containing 'the' in the title or director show up	All of the movies containing 'the' in the title or director show up
User searches 'Potato' in 'All'	Message to say there are no results for that search	Message to say there are no results for that search
Blank text input in search box	Error message to say that search box cannot be blank	Error message to say that search box cannot be blank

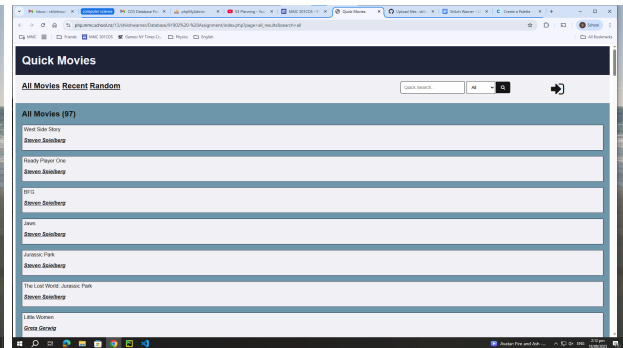
Other Iterative Changes:

As I was testing my website, I realised that the search icon on my search button was not working and a default square had been placed there. This was an aesthetic and usability issue as it was very unclear that this was the search button. For all an external user knew, the button could have done anything. To fix this issue, I checked how my css files were linked as it seemed that the main one was not linking to font-awesome which was the file that contained the necessary information to show the search icon. I was able to fix this error so in the final screenshot, the search icon can be seen and therefore the aesthetics and usability have increased.

Before:

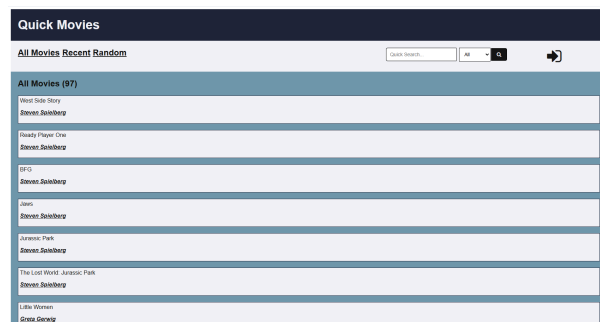


After:



During my testing stages, I realised that I had no link back to the homepage in my navigation bar. This is a functionality error as it means that once the users click away from the homepage at the beginning, they have no way of getting back to it. To fix this, I added a homepage link into my banner_navigation file that now allows users to go back to the homepage whenever they want to.

Before:



After:

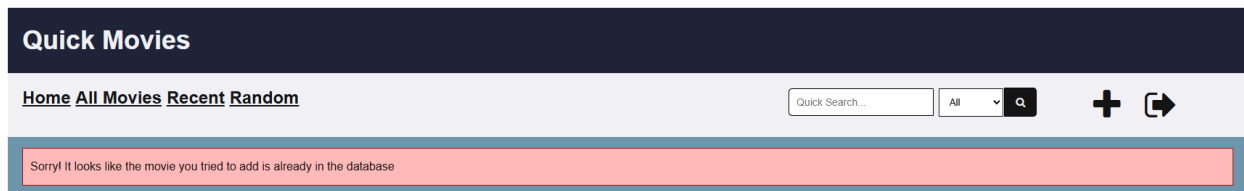


As I was testing my adding movies section, I discovered that the database currently allows you to add the same movie by the same director twice even though this would make it a duplicate. This is a functionality error as it now means there is redundant data in the database. To fix this, I wrote some code that checks if the title already exists in the database. If it doesn't then it allows the user to add the new movie. If it does, then an error message is displayed to notify the user that the movie they tried to add is already in the database.

Before:

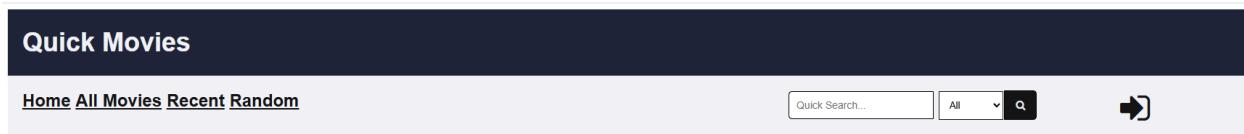


After:

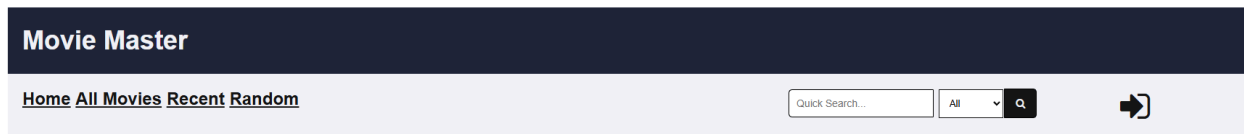


As a measure to increase the aesthetics and meet end user requirements, I changed the title of my website from 'Quick Movies' to 'Movie Master'. Changing the title of the website to 'Movie Master' makes it more informative and enticing to the users which therefore gives them a better first impression of the database and makes the title more engaging.

Before:



After:



User Log in

Provide evidence that users can only log in to the admin area if they use the correct username / password combination

Test Plan for Login:

Action/Input	Expected Result/Output	Actual Result/Output
User enters correct login information	The user is successfully logged in and taken to the	The user is successfully logged in and taken to the

Name: Shiloh Warner

	add movie page.	add movie page.
User enters incorrect login information	Error message to tell the user that they have entered the incorrect login information	Error message to tell the user that they have entered the incorrect login information
User enters blank login information	Error message to tell the user that they have entered the incorrect login information	Error message to tell the user that they have entered the incorrect login information

Evidence Screenshots:
Correct username and password

Quick Movies

[Home](#) [All Movies](#) [Recent](#) [Random](#)

Quick Search...

All

Q

➔

Login

Username:

admin

Password:

Log In

CC shilohwarner 2025

Page after correct login information has been entered

Quick Movies

[Home](#) [All Movies](#) [Recent](#) [Random](#)

Quick Search...

All

Q

+

➔

Add Movie

Movie (Required)

Director Name

(First Last)

Submit Movie

CC shilohwarner 2025

Incorrect message for if the wrong username and password have been entered

Quick Movies

[Home](#)
[All Movies](#)
[Recent](#)
[Random](#)

All

Q

➔

Login

Username:

Password:

Incorrect username / password

Log In

CC shilohwarner 2025

Add, edit and delete a record

Provide evidence that users can add, edit or delete a record

Test Plan for Add Movie:

Action/Input	Expected Result/Output	Actual Result/Output
User adds valid input into both boxes such as "test" and "test"	Movie is added successfully and a success message tells the user this	Movie is added successfully and a success message tells the user this
User adds valid input into the movie title box but leaves the director box blank	The movie is added successfully with the director dubbed as "anonymous". Success message shown	The movie is added successfully with the director dubbed as "anonymous". Success message shown
User leaves both boxes blank	Message shown to tell the user they need to fill out the movie title field	Message shown to tell the user they need to fill out the movie title field

Screenshot Evidence:

Here is the page to add a movie to the database

Quick Movies

[Home](#) [All Movies](#) [Recent](#) [Random](#)

Quick Search...

All

Q

+

→

Add Movie

test

test

Submit Movie

CC shilohwarner 2025

Success message to say that the movie has been added to the database

Quick Movies

[Home](#) [All Movies](#) [Recent](#) [Random](#)

Quick Search...

All

Q

+

→

Movie Success (1)

test

test

✎

🗑

CC shilohwarner 2025

Test Plan for Edit Movie:

Action/Input	Expected Result/Output	Actual Result/Output
User changes movie title and director	Movie information is changed successfully	Movie information is changed successfully
User changes movie title	Movie information is changed successfully	Movie information is changed successfully
User changes director	Movie information is changed successfully	Movie information is changed successfully
Neither movie title or director is changed	Movie information is saved successfully	Movie information is saved successfully
Movie title is changed to blank	Error message to communicate that the movie title cannot be blank	Error message to communicate that the movie title cannot be blank

Screenshot Evidence:

Edit movie screen

Quick Movies

[Home](#) [All Movies](#) [Recent](#) [Random](#)

All Q + →

Edit a Movie

If you edit an director, it will change the director name for the movie being edited.
It does not edit the director name on all movies attributed to that director.

CC shilohwarner 2025

Quick Movies

[Home](#) [All Movies](#) [Recent](#) [Random](#)

All Q + →

Edit a Movie

If you edit an director, it will change the director name for the movie being edited.
It does not edit the director name on all movies attributed to that director.

CC shilohwarner 2025

Success message to show that the movie has been edited

Quick Movies

[Home](#) [All Movies](#) [Recent](#) [Random](#)

All Q + →

Edit Movie Success

You have edited the movie. The entry is now...

test1

~~test~~

✎ 🗑

CC shilohwarner 2025

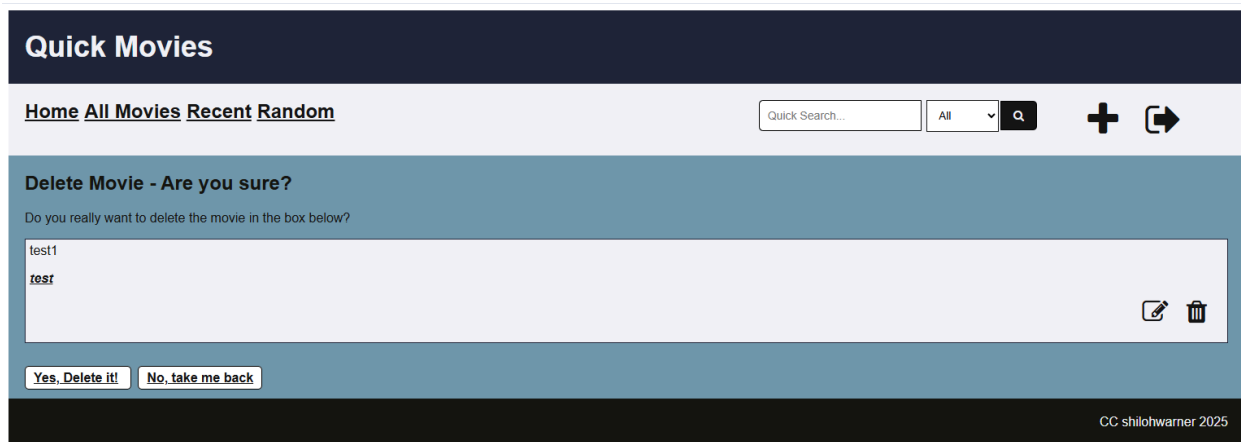
Name: Shiloh Warner

Test Plan for Delete Movie:

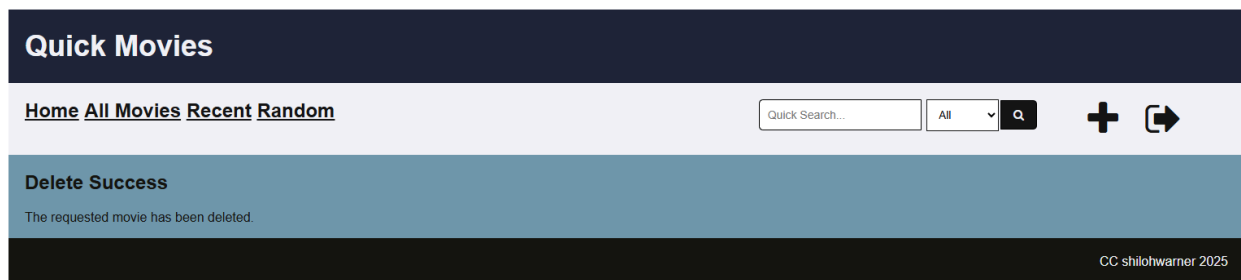
Action/Input	Expected Result/Output	Actual Result/Output
User clicks 'Yes, delete it'	Movie is deleted with a success message to say it has been deleted	Movie is deleted with a success message to say it has been deleted
User clicks 'No, take me back'	User is taken back to home page	User is taken back to home page

Screenshot Evidence:

Delete movie page



Success message to confirm that the movie has been deleted



End User Feedback

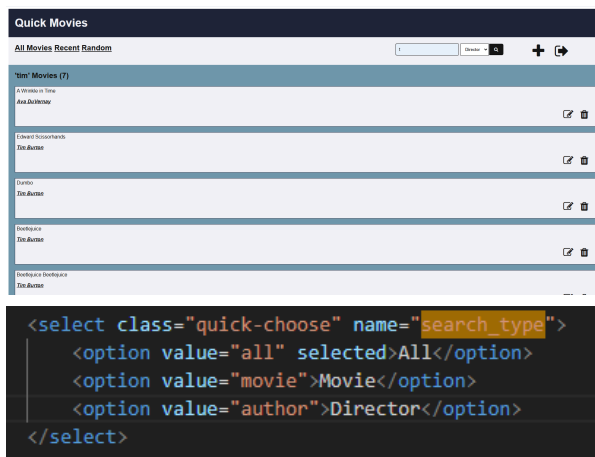
Add evidence of end user feedback that has helped improve your website throughout the project

During the early stages of my user testing, I tested with my computer science teacher Mr Tuahine. While he was looking around on the website, he found that there was an error in the director search. As seen in the screenshot below, when you search for "Tim" as a director, all of his movies show up, however, the movie "A Wrinkle in Time" also shows up which is not directed by anyone named Tim. This led me to believe that the search was not actually searching for a director but rather was just searching all for "Tim" because this would result in

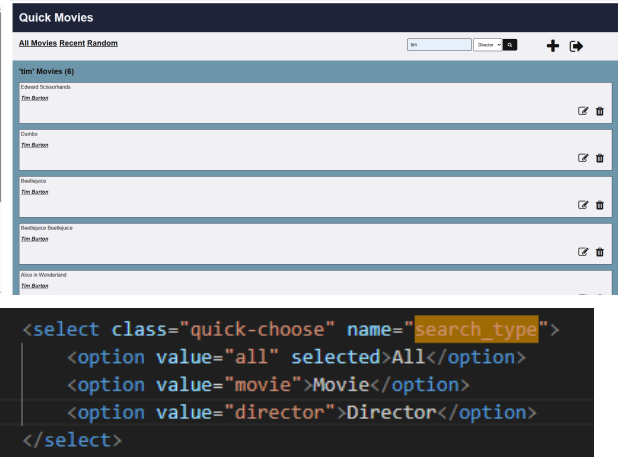
Name: Shiloh Warner

“A Wrinkle in Time” showing up due to the tim in time. When I looked into my code to find the error, I found that my banner navigation file had an error in it when naming the search type. The search type for director was tagged “author” instead of “director”, as seen in the before screenshot below, meaning that the search could not be linked to the other files as it had the wrong name. This resulted in the search defaulting to an all search rather than a specifically director search. To fix this error, I changed the name of the tag to be “director” instead of “author” as seen in the after screenshot below. This error found through user testing made my database director search significantly more functional as prior to this, it was not searching specifically for directors at all.

Before:

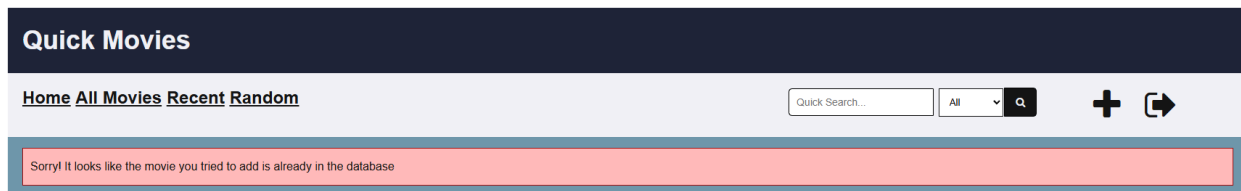


After:

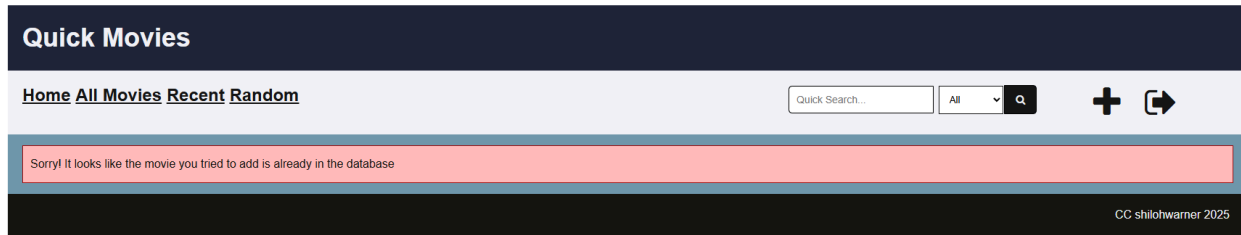


The person I tested with was my Mum. During her testing the database, she tried to enter a movie that was already in the database and was therefore met with an error message. However, she noticed something I hadn't noticed before which was that the footer was missing from the page with the error message. Looking at my code, I realised this was due to using exit() after my error message. The exit() command was causing the code to completely end and skip the footer. To fix this, I switched the exit() method to an if-else. This means that the code continues and the footer is seen again.

Before:



After:



The next person I tested with was my friend Sam. When she saw the add movie screen, she was instantly put off by the mismatched font, random word spacing and weirdly spaced layout. For this reason, I changed the fonts to be the same, easy to read font, I fixed the spacing, and changed the margin between the input boxes and submit button to make everything a bit closer together so the spacing didn't seem so random. These actions increased the aesthetics of my database interface and therefore made it more enjoyable and easy for end users to operate.

Before:

After:

The next person I tested with was Tye, a fellow computer science student and friend. When she was testing the edit movie component, she shared that she was initially confused as to what was input boxes and what was message boxes because all of the boxes looked the same. For this reason, I changed the message box to be a light blue colour to obviously differentiate it from the input boxes, making the edit movie component more usable for the end users.

Before:

After:

Final Outcome:

Final Testing:

Add links to videos showing the website interface functions as intended. E.g. show a logged in user editing, adding and deleting data.

Link to testing video following above test plans: <https://drive.google.com/file/d/1T7Ne9>

Name: Shiloh Warner

Relevant Implications

How have you addressed the relevant implications? Consider functionality, aesthetics, legal, accessibility, and usability.

Functionality:

Functionality means to ensure that the database meets its intended purpose and works as expected. It should work for expected, boundary and unexpected cases. For example, my database needs to respond to input in the add_movie component such as expected input like adding a movie with a title and director, boundary input like adding a title but no director meaning director will default to anonymous, and unexpected input such as trying to add a movie with no title or director. It should accept the expected and boundary input and continue to function and recover from unexpected input rather than crashing or freezing. In the cases of unexpected input, it was also required that appropriate error messages were presented in order to help the user recognise and recover from errors as quickly and easily as possible. It is vital that I account for every possible expected, boundary and unexpected case that the user could enter. Functionality is important because if my database was constantly crashing or giving the user the incorrect output, the user will likely become very frustrated to the point that they may even stop trying to use the database. Databases that do not work correctly and are not functional are of limited use and would cause the developer of the information to lose users due to the impatience and frustration surrounding the database not working as it should be. An example of how I have increased the functionality of my database is by doing SQL testing to make sure that my proposed SQL queries worked before implementing them into my code. This meant that I was confident that searches would return the correct results, and therefore the database would be functioning as it needed to be for the end users. If I did not do this SQL query testing, there would have been increased possibilities of errors in the queries getting through to the final code and causing the search function to not work correctly. By testing my SQL queries I increased the functionality of my database by ensuring the search worked as intended.

Aesthetics:

Aesthetics improves the overall appearance of an outcome to make it visually appealing for the end users. A database that is well spaced out and has nice, cohesive colouring is more aesthetically pleasing than a database that is randomly or unevenly spaced with inconsistent and random colouring. Aesthetics matter because how the database is presented will be one of the first things that the end user notices. If they find it cluttered and confusing, they are much less likely to continue trying to use that database. It is important to make sure that the database presents an interface with clear, easy to read spacing and colours that work well together as this makes the interface more appealing to the user which increases the likelihood of the user happily using the database based on their first impression, and having a nice and easy experience using the database based on the aesthetics that follow. This also relates to usability. If all of the text, buttons, headings, and labels are overlapping, poorly spaced, and difficult to read, the database would have reduced usability as the user would struggle to understand what was what. They would become very confused by the layout and stop trying to figure out the difficult to understand database. By ensuring that the aesthetics make the database easy to

understand, usability is also increased as it makes the database more straightforward for the end user. An example of an aesthetic consideration I took was people's embedded association with different colours and their meanings. In my database, the message that comes up if you enter incorrect login information, try to add multiple of the same movie, or enter a search option with no results is red. I made this decision because red is commonly associated with incorrectness or errors meaning that the users know a moment quicker whether they have entered valid input or not as they know by the colour before they read the words. My particular choices for colours also extended to the colours of my overall interface. This is discussed in the planning section of the documentation.

Accessibility:

Accessibility involves making the database and interface as accessible as possible to as many people as possible, despite any barriers that could make it more difficult for them to use the outcome. Key aspects of accessibility include making sure the database can still be accessed by those who are colourblind, have dyslexia, as well as various other disabilities. Accessibility is important because if some users cannot access the website due to disabilities, the database is not inclusive and the number of users becomes significantly more limited. One of the ways I addressed accessibility was by ensuring that all of the text on my database was in a simple font, spaced out well, and quite a large size to ensure that people who are dyslexic or have sight impairments can still read all of the text easily. To ensure that I was meeting these requirements, I tested the website with my Dad as he is both dyslexic and needs glasses. He told me that all of the text was easily readable to him and therefore I knew I had met these requirements. Another example of how I addressed accessibility is by ensuring that my chosen colours were colour blind friendly. I did this through using coolers.com to see what the palette would look like to someone who is colour blind. As can be seen in the screenshot below, the colourblind colours are still distinct enough that they would not cause confusion to a colourblind user and therefore, this requirement has been met.

