PHP Documentation

Autocomplete Search Functionality

This functionality is heavily reliant on JavaScript and PHP in order for it to work, due to the quantity of JavaScript, a separate file was created for this called search.js. This allows the user to type into the search bar and have a prediction appear based on what the user is typing out. The user is then able to use the up and down arrow keys to flow through the autocomplete predictions and press enter to fill the value into the search box. When enter is pressed, the user is taken to a new page containing a table of shows based on the search value, but if no results are found, then the user is told this. This is all retrieved from the database using PHP.

Watchlist

The basis of the watchlist functionality is to give some interaction between the standard-level user and the website. When the user selects the call to action (cta), more info, a modal appears presenting more information about the shows, while the user is reading this information they have the choice to either close the modal or add the show to a watchlist. The user can choose to access their watchlist at any given time, even when there are no shows added to the watchlist. This functionality uses a session array which targets the unique ID of each show that will be entered into the database in a prepared sql statement. The sql statement is then looped through and pulled onto the watchlist page, showing the show name, category and studio name, all delimited by vertical lines. For users who wish to reset their watchlist, they can click the reset cta and this will unset the watchlist session.

Next to the reset button on the watchlist page is a CTA (Call To Action) called download watchlist. The majority of the PHP in watchlist.php is similar to that which is in downloadwatchlist.php, which does the same loops and checks but instead of just echoing the watchlist shows, it writes the list to a text file and downloads it. After this is done, the watchlist session array is unset, so if they wish to do so, they can create another list for a friend.

Content Management System

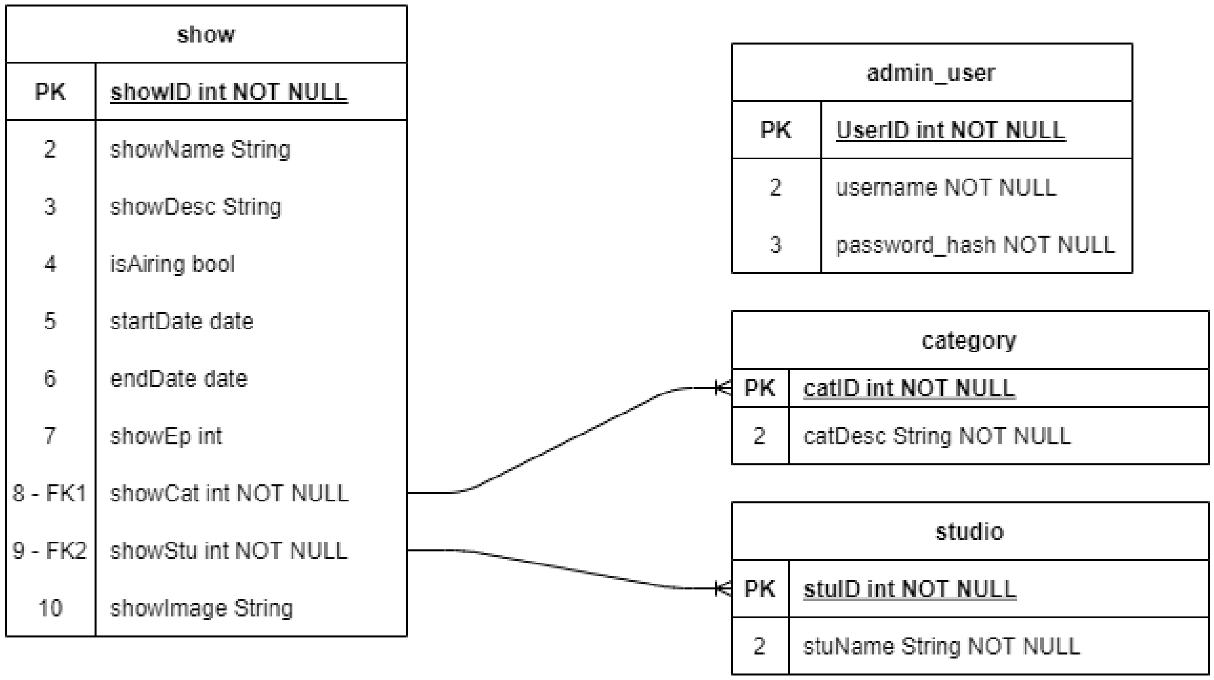
The content management system (CMS) can be used by admin-level users only. To prevent unauthorised users from accessing the pages by typing in the URL, session data is used to prevent this. Checking if the user is logged in, only giving them access if the logged-in session variable is set. If this is not the case, the user will be automatically redirected to the login form. Once the admin-level user has been granted access, they then have an option to either add a new show, edit a show which is currently already on the system, or delete a show which they no longer wish to have on the system. This CMS is what is used to allow the admin users to communicate with the database without needing to access the live database directly. Once all admin tasks are completed, the user is able to log out, which will redirect the user to the homepage, however if they wish to access the admin section again, they will need to log back in. If the user just wants to see how new content displays when added, they can click ‘Home’ redirects to the home page without destroying the session and logging the user out.

Homepage Content

On the homepage, a list of shows are displayed, and when the user clicks the more info button, the user is greeted with details about the shows. In order to display this in the way that was intended, AJAX was used so that the modal data is able to change dynamically without the need to reload the whole page. The AJAX was programmed using JQuery as it is much more lightweight than JavaScript as it is able to achieve the same outcome using less lines of code.

Database Documentation

ERD:



show	
PK	showID int NOT NULL
2	showName String
3	showDesc String
4	isAiring bool
5	startDate date
6	endDate date
7	showEp int
8 - FK1	showCat int NOT NULL
9 - FK2	showStu int NOT NULL
10	showImage String

admin_user	
PK	UserID int NOT NULL
2	username NOT NULL
3	password_hash NOT NULL

category	
PK	catID int NOT NULL
2	catDesc String NOT NULL

studio	
PK	stuID int NOT NULL
2	stuName String NOT NULL

Data Dictionary:

admin_user

Attribute	Datatype	Size	Comments
userID	INT	11	NOT NULL
username	STRING		NOT NULL
password_hash	STRING		NOT NULL

category

Attribute	Datatype	Size	Comments
catID	INT	11	NOT NULL
catDesc	STRING		NOT NULL

studio

Attribute	Datatype	Size	Comments
stuID	INT	11	NOT NULL
stuName	STRING		NOT NULL

shows

Attribute	Datatype	Size	Comments
showID	INT	11	NOT NULL
showDesc	STRING		NOT NULL
showEp	INT		
startDate	DATE		
endDate	DATE		
showCat	INT		NOT NULL
showStu	INT		NOT NULL
showImage	STRING		UPLOAD IMAGE TO WEBSITE
isAiring	BOOL	1	

Description:

For the nature of this project, the website is to be hosted locally using XAMPP localhost software, there is no username or password needed to access the database level, allowing the machine which is being used to act as a server. However, if the database and the website were to be hosted on a live URL, the data.The table ‘shows’ is the main data storage table which interacts with the website. The columns ‘showCat’ and ‘showStu’ are foreign keys linked to tables ‘category’ and ‘studio’ respectively. This is done to allow for the data to be pulled to the website dynamically, meaning when the site needs to be updated, it can be updated by an admin-level user rather than hard coding the content in. The table ‘admin_user’, has no relationship with the other tables due to its purpose. It’s only used to allow for admin-level users to access the CMS.