Index.php

This file is a simple web application for user authentication, including both login and signup functionalities. It starts by initiating a session with session\_start() and includes a database connection file using require('dbconnect.php'). The script initializes an empty string $error to store any error messages that may arise during the login or signup processes.

The first major block of code handles the login functionality. It checks if the login form has been submitted by verifying if $\_POST['login'] is set. If so, it retrieves the email and password from the POST request and constructs an SQL query to check if a user with the provided credentials exists in the database. If a matching user is found, their details are stored in the session, and the user is redirected to dashboard.php. If no match is found, an error message "Invalid login credentials!" is set.

The second major block of code handles the signup functionality. It checks if the signup form has been submitted by verifying if $\_POST['signup'] is set. It retrieves the name, email, and password from the POST request and constructs an SQL query to check if a user with the provided email already exists. If a user is found, an error message "User already exists!" is set. If no user is found, another SQL query is constructed to insert the new user's details into the database. Upon successful insertion, the new user's ID is retrieved, and their details are stored in the session. The user is then redirected to index.php. If the insertion fails, an error message indicating the failure is set.

The HTML part of the code includes a simple form for both login and signup, styled using Tailwind CSS and a custom font from Google Fonts. The form displayed depends on whether the action parameter in the URL is set to 'register'. If not, the login form is shown by default. If the action parameter is set to 'register', the signup form is displayed. Both forms include fields for email and password, with the signup form also including a field for the user's name. Error messages are displayed above the forms if any errors are set during the login or signup processes. The forms are styled to be user-friendly and visually appealing, with responsive design elements to ensure a good user experience.

Dashboard.php

Submit\_review.php

The provided PHP code in submit\_review.php handles the submission of movie reviews by authenticated users. It begins by starting a session with session\_start() and including the database connection file using require('dbconnect.php'). This ensures that the script has access to the session data and the database.

The script then checks if the user is logged in by verifying if the $\_SESSION['user'] variable is set. If the user is not logged in, they are redirected to the index.php page, and the script execution is terminated using exit(). This is a security measure to prevent unauthorized access to the review submission functionality.

Next, the script retrieves the logged-in user's information from the session and stores the user ID in the $userid variable. This user ID will be used to associate the review with the correct user in the database.

The script then checks if the request method is POST, indicating that the form has been submitted. It retrieves the movie ID and review text from the POST request and stores them in the $movieid and $review variables, respectively. It also gets the current date in the format 'Y-m-d' and stores it in the $date variable.

To insert the review into the database, the script prepares an SQL statement using $conn->prepare(). The SQL statement inserts the review date, review text, user ID, and movie ID into the review table. The bind\_param() method is used to bind the variables to the prepared statement, ensuring that the data is safely inserted into the database.

The script then executes the prepared statement using $stmt->execute(). If the execution is successful, the user is redirected to the dashboard.php page. If there is an error during execution, the script outputs the error message using echo. Finally, the prepared statement and the database connection are closed using $stmt->close() and $conn->close(), respectively.

Overall, this script ensures that only authenticated users can submit reviews, securely inserts the review data into the database, and handles any potential errors during the process.

**A) Potential New Features**

1. **User Profile Management**:
   * Allow users to update their profiles, including profile pictures and personal information.
   * Implemented in [index.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") and [dashboard.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
2. **Movie Recommendations**:
   * Provide personalized movie suggestions based on user ratings and watchlist.
   * Utilize data from [rating](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and [watchlist](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) tables.
3. **Admin Dashboard**:
   * Create an interface for administrators to manage movies, genres, and user accounts.
   * Could be added as a new file, e.g., admin\_dashboard.php.
4. **Social Features**:
   * Enable users to follow each other, share watchlists, and comment on reviews.
   * Extend functionality in [submit\_review.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") and [my\_watchlist.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
5. **Advanced Search Filters**:
   * Add filters for release year, rating range, and multiple genres.
   * Enhance search functionality in [dashboard.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
6. **Movie Trailers and Media**:
   * Incorporate trailers and additional media for each movie.
   * Integrate in [overview.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
7. **Pagination for Movie Listings**:
   * Implement pagination to handle large lists of movies efficiently.
   * Modify queries and UI in [dashboard.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").

**B) Improvements to Existing Features**

1. **Enhanced Security**:
   * **Password Hashing**:

<?php

// In [`index.php`](index.php)

$hashed\_password = password\_hash($password, PASSWORD\_DEFAULT);

* + **Prepared Statements**:
    - Ensure all SQL queries use prepared statements to prevent SQL injection.
    - Example in [add\_to\_watchlist.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").

1. **User Authentication Enhancements**:
   * Implement session timeout and secure session handling.
   * Improve in [index.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") and [logout.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
2. **Watchlist Functionality**:
   * **Status Updates**:
     + Allow users to mark movies as "Watching" or "Completed".
     + Update in [my\_watchlist.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
   * **Notes and Ratings**:
     + Enable users to add personal notes or ratings to their watchlist entries.
3. **Review System Improvements**:
   * **Edit and Delete Reviews**:
     + Allow users to modify or remove their reviews.
     + Update in [submit\_review.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") and [overview.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
   * **Rich Text Formatting**:
     + Support markdown or rich text in reviews for better expression.
4. **Performance Optimizations**:
   * **Database Indexing**:

-- In [`muvi.sql`](muvi.sql)

ALTER TABLE `movie` ADD INDEX idx\_title (`Title`);

ALTER TABLE `user` ADD INDEX idx\_email (`Mail`);

* + **Caching Mechanisms**:
    - Implement caching for frequently accessed data to reduce database load.

1. **User Interface Enhancements**:
   * **Responsive Design**:
     + Ensure all pages are mobile-friendly.
     + Improve CSS in files like [dashboard.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") and [my\_watchlist.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
   * **Feedback Messages**:
     + Provide more detailed feedback for user actions, such as successful updates or errors.
2. **Error Handling and Validation**:
   * **Input Validation**:
     + Validate all user inputs on both client and server sides.
     + Enhance in files like [add\_to\_watchlist.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") and [submit\_review.php](vscode-file://vscode-app/c:/Users/advfa/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
   * **Comprehensive Error Messages**:
     + Offer clear and actionable error messages to users.

Implementing these features and improvements will enhance the functionality, security, and user experience of your application.