

Developer's Perspective: Implementing The Site 'Eclipse' Online Discussion Forum

Source Code Files Breakdown:

about-us.php:

An informative page, detailing an Overview of the website and project, our website statement, and a brief description of our site's function and inspiration. It is primarily composed of html contained in a php file.

addComment.php:

This page takes on the features of postPage.php and adds a form that lets a user submit their comments regarding the post. This means that the post remains on screen along with the topics and other possible content as if the user is viewing the post. The site takes this comment and uploads it into the database using PHP and SQL. The form submission is handled through HTML.

adminpage.php:

To begin, this PHP file checks if the user logged in and is an admin. If not, it redirects them to the homepage. Its main function is to display statistics based on the selected filter (posts, comments, or users) and display the corresponding data in an HTML form. Admins can view these statistics by selecting a filter and then submitting the form. There is a link to this page on the home page header, which is only viewable if the user is logged in and has admin status.

change-password.php:

This page is shown when the user clicks on the "Change Password" button in their profile page. It offers a simple form asking the user for their current password, then asking the user to enter their new password and to confirm this new password. If the user enters any invalid values (wrong current password or mismatched new password), feedback messages are shown prompting the user to try again.

connectDB.php:

This php file is what we use to connect to our database. It contains The necessary host, user, password, and database info. It also contains the required info to connect to our server in the form of comments. We made this file to save time and increase usability, it will be referenced in most of the other files.

contact.php:

An informative page, showing a form to the user that takes their contact information along with their name and any comments, suggestions, complaints or other body of text they want to send to us!

create-post.php:

This is responsible for allowing users to create new posts on the website. It starts if a user accesses the page without being logged in, they are redirected to the login page, however, the home page is configured so this shouldn't ever be the case. It includes a form for users to create a new post, with fields for title, content, and topic. When submitted, the form data is sent to process-post.php for further processing. Additionally, it lists the user's previous posts if there are any, if not, it displays a message saying so.

edit-profile.php:

This PHP file allows logged in users to edit their profile information. If a user tries to access the page without being logged in, they are redirected to the login page. Upon form submission, it updates the user's profile information such as username, email, bio, and profile picture. If an uploaded profile picture is provided, it is stored in the "pfp" directory and its path is updated in the database. The Page displays the current profile picture if available. After making changes and clicking "Save Changes," the form data is processed and the user is redirected to their profile page (user-profile.php).

home.php:

The main landing page of our website. This page offers the user numerous things based on whether they are logged in or not. A user that is not logged in can view all post titles, use the search bar, click on our information pages and topics, and have an option for them to login or register. A logged in user can do all of these with the additional features of viewing their profile, and creating a post. A special feature is unlocked for an admin account where they can view basic site information as well. These features are restricted to each state so a logged in user will not see a login/register button and vice versa.

login.php:

This PHP script deals with the log-in function. Upon receiving a POST request, it retrieves the email and password provided by the user and queries the database to find a matching user record. If a matching record is found, the user's session variables (email, username, role, and userId) are set, and the user is redirected to the home page (home.php). If no matching record is found, an alert is displayed indicating invalid credentials.

logout.php:

When this page is accessed, it calls a function that destroys the session. Logging out the user.

postPage.php:

Since the home page only showcases post titles, this page shows the full content of the post along with comments regarding the post. A logged in user will also have the option to add their own comment while a user that isn't logged in will not see this option. The sidebar from the home page is also visible in this page as well showing our information pages and available topics.

process-post.php

This file handles the submission of a new post form. It checks if the request method is POST and retrieves the post title, content, topic, and user ID data, as well as the current date for the post date. Then, it prepares an SQL statement to insert these values into the Post table in the database.

processComment.php:

This first starts the session and checks if the request method is POST and if the required fields postId and comment are set. If the user is logged in, it retrieves the userId from the session. Otherwise, it terminates the script with an error message indicating that the user is not logged in. It then retrieves the postId and comment from the POST data and prepares an SQL statement to insert these values into the Comments

table. If the SQL statement executes successfully, it redirects the user to the post page to view the added comment.

register.php

This page is our main registration page where users are greeted with a simple form that takes basic account information that will be stored in the database using PHP and SQL. This information will be taken into account when the user views their profile and when they login. There are validators that ensure no form is left blank and that the email (valid email address syntax) and password (number, symbol, upper case) are checked before submission.

tos.php

An informative page, showing both logged in and free users the terms of services and rules for engaging and usage of our website. These cover basic guidelines in ensuring a safe community for everyone involved.

user-profile.php

This page provides a landing page for logged in users to see their basic info that was given to the database upon registration. They are able to view their profile picture, name, email, and bio. They also have two buttons that lead the user to a page where they can edit their profile information OR change their password. The user can also view their previous posts and comments and clicking on it will bring them to their respective post page!

index.php:

A helper page that directs the online web server to our home page.

Limitations:

We feel as though most of our bases are covered, however, there are some areas where people may argue there are limitations. A user can not post videos/pictures, however, this was intentional as we wanted the platform to be primarily text-based, to avoid overlap and stand out when juxtaposed with other popular social media platforms and forums. We feel as though much nuance is lost when only a photo or video is provided. Additionally, we have not implemented a direct message feature between users, however, we believe the lack of this feature will facilitate greater conversation in the direct comments of the post itself.