My Nature Forum

Software Implementations

Website link:

https://cosc360.ok.ubc.ca/kaff/project-Kaf-Mahir/kaf/login.php

Github Link:

https://github.com/COSC360/project-Kaf-Mahir

Mahir Rahman - 71811509

Abdallah Kaf - 47359534

Table Of Contents:

1.0 Navbar …….

This code is a navigation bar written in HTML and PHP for a web application. It has a fixed position on the top of the page and contains links and buttons for navigating the application. The PHP code is used to dynamically generate the contents of the navigation bar based on whether the user is logged in or not. The navigation bar consists of a container that contains the logo or profile picture of the user if they are logged in, a search bar, and links to various pages of the application. The navigation bar is collapsible, and clicking on the toggle button will reveal or hide the links. The PHP code checks if the user is logged in by checking the $_SESSION['username'] variable. If the user is logged in, the code will display the user's profile picture and links to the profile, settings, and logout pages. If the user is not logged in, the code will display default values instead of the profile picture and links to the login and registration pages.

1.1 Admin.php ……..

This opadmin dashboard for a server. It checks if the user is logged in as an admin and displays a table of all the users on the server, along with their email and whether they are enabled or disabled. There is also an option to toggle the status of each user.

It then checks if the user is logged in as an admin, and if not, redirects them to the home page. After that, it checks if the "toggle_status" POST variable is set, and if so, updates the status of the user in the database. The HTML portion of the script contains a navigation bar and two containers. The first container displays a welcome message for the admin, while the second container displays a table of all the users on the server, along with the option to toggle their status.

1.2 Login.php ……..

Our login page that allows users to enter their username or email and password. It contains a form that sends a POST request to a login.php file upon submission. If the user clicks the "login-btn" button and there are errors, the form will display them in an alert box. The page also has links for social media login options and a link to sign up for an account. If the user does not have an account, they can click the "Enter without login" link to access the home page. The code also checks for any messages passed through the URL and displays them in a warning alert box

if there are any. The HTML code uses Bootstrap for styling and includes a custom CSS file and a JavaScript file for form validation.

1.3 Signup.php ……

This page consists of form elements for collecting user data such as username, email, password, and password confirmation. The form is set to post to a PHP file called "signup.php". The code also includes error handling for the form input using PHP variables. If there are any errors, they are displayed on the page. Additionally, there are social media login buttons and an option to subscribe to a newsletter. There is also a link to a login page for users who already have an account. Finally, there is a script tag at the end of the code for a custom validation script for the form.

1.4 Home.php ……

This page consists of a PHP script that handles a request from an AJAX call. The request sends a POST parameter 'post_id' which is used to update the upvotes of a post in a MySQL database. The script first checks if the user is logged in, and then checks if they have already liked the post. If they have not, it adds the user to the upvotes table, otherwise, it removes them. It then returns the updated number of upvotes in a JSON object. If the user is not logged in, the script returns the string 'login'. The script also includes an HTML page, which is used to display a navbar on the webpage. The navbar includes links to the user's profile and a search bar, among other things. It is worth noting that the code appears to be missing some sections, indicated by comments such as "//what to show when user is not logged in instead of profile picture".

1.5 Profile.php ……...

The PHP code is responsible for generating an HTML page. The code first sets the error reporting level and includes two PHP files, "auth.php" and "load-profile.php". It then sets the $isMyProfile variable to false. The HTML document starts with the usual meta tags and includes Bootstrap's CSS and icons. It sets the document's title to "Profile". A navbar is then created using Bootstrap's classes. The navbar includes a logo, a search form, links to the home page, log out, and settings. The navbar also includes a button to create a new post that will show only if the

user is logged in. Finally, a modal form is created to create a new post. The form includes an input field for the post title, a textarea for the post content, and buttons to submit and close the modal.

1.6 Load-profile.php ……

This is a PHP script that handles user profile information and profile picture uploads. It first checks if a username is provided in the URL parameter "username" or if a username is stored in the session. If a username is found, it queries the database for profile information such as the full name and bio associated with the username. The script then sets the profile picture based on whether an image file with the user's username exists in the "img/" directory or not. If an image file exists, it is set as the profile picture. Otherwise, a default image is used. The script also handles the uploading of a new profile picture if the user submits a form with a file input named "inputPic". It checks if the uploaded file is a valid image file and if it is, it moves the file to the "img/" directory and updates the user's profile picture. It then redirects the user back to the profile page with a success message. Lastly, the script handles the updating of profile information such as the full name and bio. It checks if the "save-profile" button was clicked and if it was, it updates the corresponding fields in the database with the new values provided by the user. It then redirects the user back to the profile page with a success message. The script also includes some functions for validating uploaded image files and checking their file extensions.

1.7 Post.php …….

This script is responsible for upvoting posts and updating the corresponding upvote count. The code first sets the error reporting level to report all errors and display them on the screen. It then includes two PHP scripts: 'auth.php' and 'load-profile.php', which are not included in the question and may contain functions or classes that are used in the script. If the 'post_id' parameter is set in the AJAX request, the script checks if the user is logged in. If the user is logged in, it retrieves the 'post_id' and the current upvote count of the post from the 'posts' table in the database. It then checks if the user has already upvoted the post by querying the 'upvotes' table using the 'post_id' and the username of the logged-in user. If the user has already upvoted the post, the script removes the upvote by deleting the corresponding row from the 'upvotes' table and decreasing

the upvote count of the post in the 'posts' table. If the user has not upvoted the post, the script adds an upvote by inserting a new row into the 'upvotes' table and increasing the upvote count of the post in the 'posts' table. The script then retrieves the updated upvote count by querying the 'upvotes' table and returns it as a JSON object in the response.

1.8 Load-Post.php ……

This PHP code fetches a specified number of posts from a database and displays them in a formatted card style. The post information, including the title, body, author username, and date created, is displayed. Additionally, there are three buttons in a dropdown menu for liking, commenting, and sharing. The commenting button opens a popover with a list of example comments. The code is designed to be used with jQuery to retrieve the number of posts to display.

1.9 Results.php ……

This page is related to our search bar results. The search is performed on a MySQL database table named 'posts', which has columns such as 'Title', 'Body', 'AuthorUsername', etc.

The PHP code first checks if the form with the name 'submit-search' has been submitted or not. If it has been submitted, the search query entered by the user is extracted and stored in the $search variable using mysqli_real_escape_string() to prevent SQL injection attacks. A SQL query is then constructed to search the 'posts' table for rows where the concatenation of 'Title', 'Body', and 'AuthorUsername' contains the search query. The results of this query are stored in the $result variable. The code then checks the number of rows returned by the query using mysqli_num_rows(), and if it is greater than zero, it iterates through each row using a while loop and displays the results using HTML code. The HTML code creates a card for each post that matches the search query, displaying the post's title, body, author, and date of creation. The card is wrapped in an anchor tag that links to a PHP file named 'post.php', passing the post ID as a URL parameter. The code also checks if the user is logged in or not and displays different content based on this condition. For example, if the user is logged in, the profile picture of the user is displayed in the navigation bar, and links to the user's profile, settings, and logout pages are shown. If the user is not logged in, a different content is shown.

2.0 Setting.php ……

This is a PHP file that contains code for a website's setting page. It starts with HTML code that imports Bootstrap and Google Fonts. It also includes a navigation bar with a search bar and icons for Home, Logout, and Settings. The page's content is divided into two sections, each with its own tab. The first tab is for Profile settings, while the second tab is for Security settings.
The Profile settings section has a form for users to update their name, bio, and profile picture. The form's action is set to setting.php, indicating that the form data will be processed by the same PHP file. There is also a conditional statement that checks if the user uploaded an unsupported image format, in which case a warning message is displayed.