

# **Messi Is The Goat**

**Kussh Satija – 80384878**

**Yahia Kamel Hussein – 98890320**

**Ronak Jain – 70015854**

## Table of Contents

<b>GitHub Repository</b>	3
<b>Walkthrough</b>	
Homepage	3
Sign in / Login	4
Sign up / Register	5
Create Post	6
Create Thread	6
Search Bar	
User Profile	7
User Profile Settings	8
Admin Portal – Dashboard	8
Admin Portal – Users	9
Admin Portal – Threads	
<b>Implementation from a System and Developer’s Perspective</b>	
Summary of Implemented Features	10
Unregistered User	10
Registered User	11
Admin	10
Description of PHP files	10-14

GitHub Repository URL: <https://github.com/ronakJain21/360-project.git>

Website Link: <https://csc360.ok.ubc.ca/yhussein/360-project/PHP/>

Admin

Username : ron

Password:123

Non Admin User :

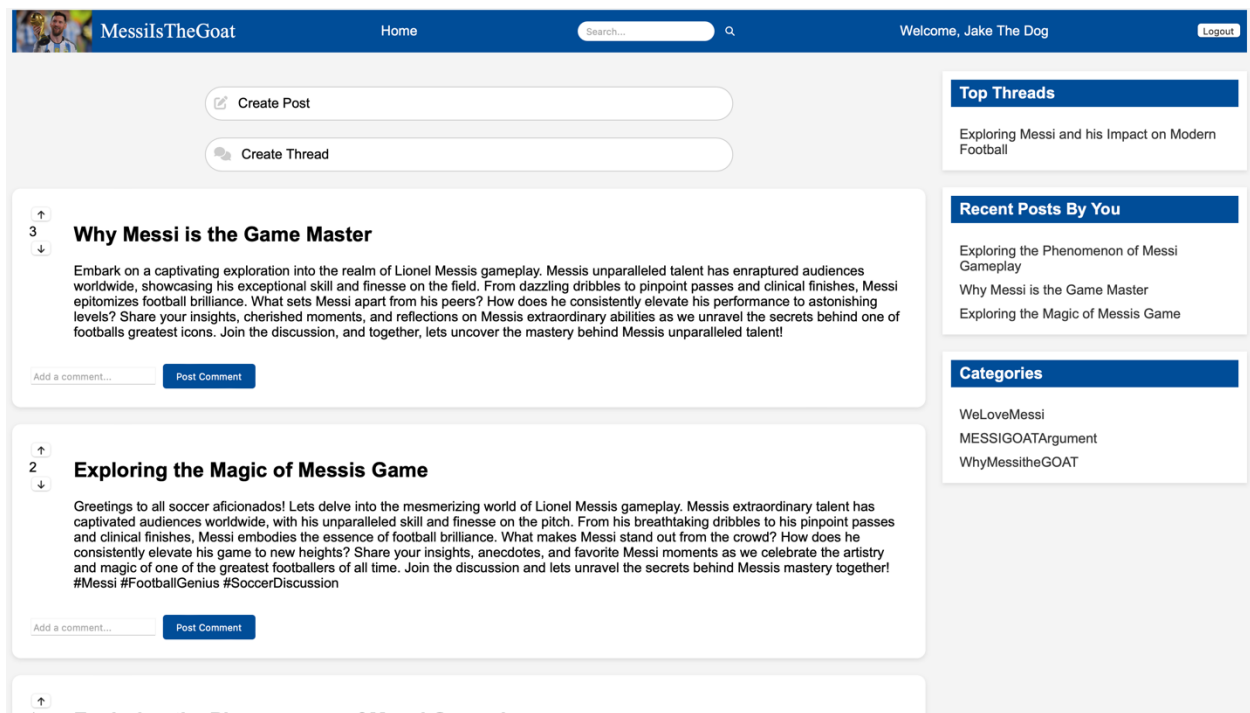
Username : ron1

Password: 123

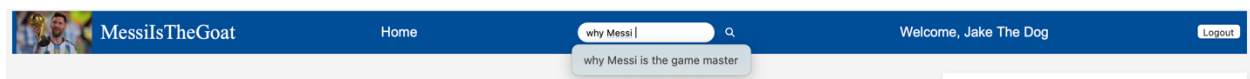
## Walkthrough:

### 1. Home page.

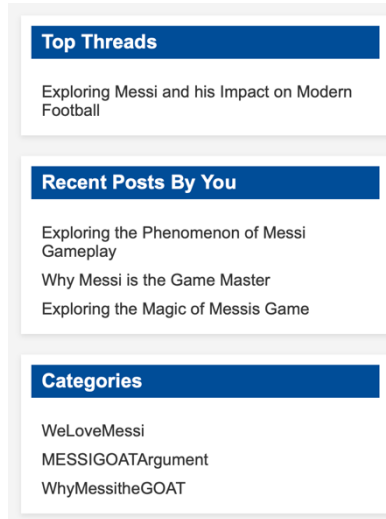
Upon arriving at the home page, without needing any login information, users are greeted with a vibrant and interactive interface designed to facilitate engaging conversations and community interaction. A limitation of an unregistered user is that they will not be able to interact with the community via creating posts, recommending new threads, voting and/or commenting.



Positioned at the top of the page are easily accessible menus, allowing users to navigate through different sections of the platform, including logging in/out, user profile, and search functionalities. An unregistered user is limited to use only the “Home” button to navigate back to the home page and search functionality to search for their desired post/thread.



Upvoting and downvoting posts on the forum allows users to express their agreement or disagreement with the content shared by fellow members. When users upvote a post, they signal their appreciation for its quality, relevance, or helpfulness, thereby highlighting it within the community. Conversely, downvoting indicates disagreement or dissatisfaction with a post, helping to downgrade content that may be less valuable or relevant to the discussion.



The sidebar on the homepage allows users to swiftly access posts categories, overview recent posts published by them, and navigate themselves to top threads on the website.

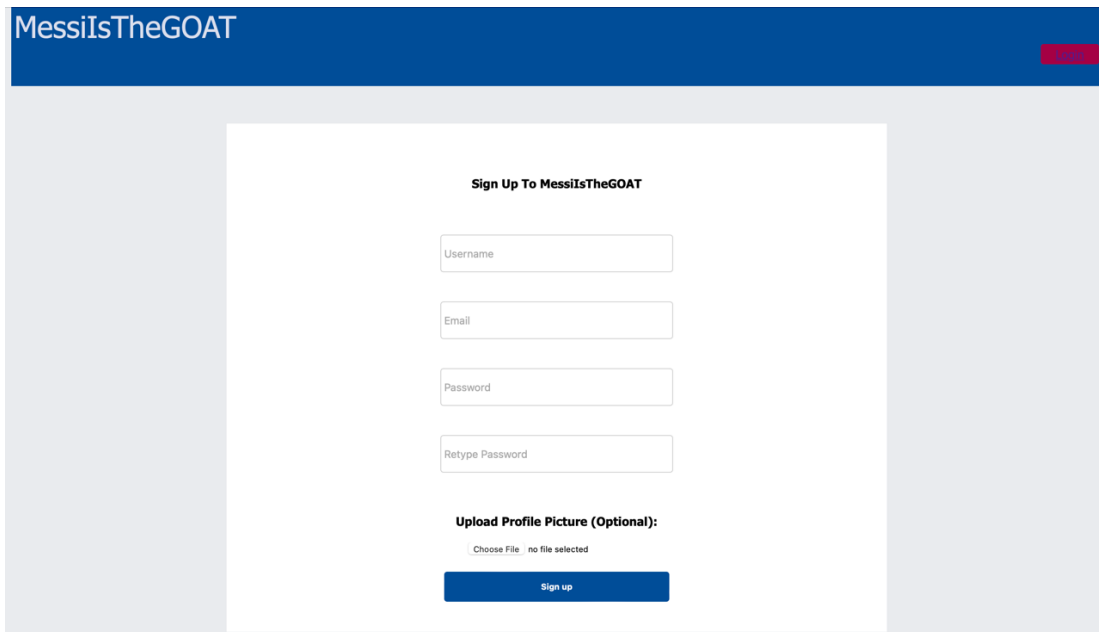
## 2. Sign in / Login.

Upon clicking on the “Sign in button” from the home page, the user is redirected to the Login page. If a user is registered with the website, they can input their credentials and click “Log in” to actively start participating with the community. Otherwise, an unregistered user must click the “Sign up” button to create an account.

The image shows a login page for a website called 'MessiIsTheGOAT'. At the top, there is a dark blue header with the website name 'MessiIsTheGOAT' in white. Below the header, the main content area has a light gray background. In the center, there is a white login form with a blue border. The form has a title 'Login To MessiIsTheGOAT' at the top. It contains two input fields: 'Email or Username' and 'Password'. Below the password field, there is a checkbox labeled 'Show Password'. At the bottom of the form, there is a blue button labeled 'Log in'.

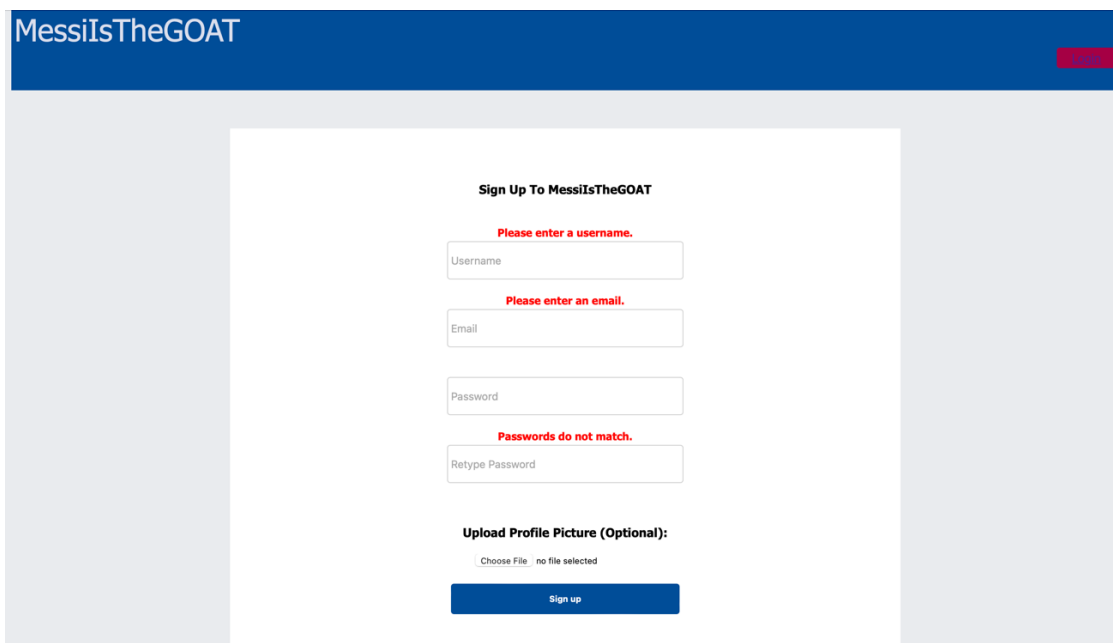
### 3. Sign up / Register.

Upon clicking the “Sign up” button, the user is redirected to the registration page where they will be asked to input their desired username, email, password, and an optional profile picture. After a successful registration, the user will be brought back to the login page that will allow them to use their newly created credentials to explore our diverse range of topics, connect with fellow members, and start engaging in the discussion.



The screenshot shows the registration page for 'MessiIsTheGOAT'. The page has a blue header with the site name and a 'Login' button. The main content area is white and contains a 'Sign Up To MessiIsTheGOAT' form. The form includes input fields for Username, Email, Password, and Retype Password. Below these fields is an 'Upload Profile Picture (Optional):' section with a 'Choose File' button and a 'no file selected' status. At the bottom of the form is a blue 'Sign up' button.

The user is prompted to fill out required input fields correctly for successful registration.



The screenshot shows the registration page for 'MessiIsTheGOAT' with validation errors. The form fields are: Username (with error 'Please enter a username.'), Email (with error 'Please enter an email.'), Password (with error 'Passwords do not match.'), and Retype Password. The 'Upload Profile Picture (Optional):' section is still present with the 'Choose File' button and 'no file selected' status. The blue 'Sign up' button is at the bottom.

### 4. Create post.

Upon clicking “Create post” button from the homepage, a registered member will be redirected to a user-friendly interface awaiting their input. To create a post, it's crucial for users to begin by crafting a compelling title that effectively conveys the essence of their intended discussion. Once a suitable title has been chosen, users can proceed to compose the body paragraph of their post. Within this section, users are encouraged to express their thoughts, share insights, pose questions, or initiate discussions on topics of interest to them. The users can then choose a specific category or thread for their post to encourage more responses from the community. After filling out the required input fields, the users can click the “Post” method to publish their thoughts on the website.

The screenshot shows a 'Create Post' form with the following elements:

- Title:** A text input field with the placeholder 'Add a unique title to your post!!!'. The example text is 'Messi scored 2 goals against PSG'.
- Body:** A text area with the placeholder 'Add body text to your post'. The example text describes Lionel Messi's performance.
- Select Thread (Optional):** A dropdown menu with the option 'No Thread' selected.
- Select Category (Optional):** A dropdown menu with the option 'WeLoveMESSI' selected.
- Post Button:** A blue button labeled 'Post'.

Numbered steps are indicated on the left side of the form:

1. Title input field
2. Body text area
3. Select Thread dropdown
4. Select Category dropdown

## 5. Recommend new thread.

Upon clicking the “Create thread” button from the homepage, registered users will find a user-friendly form prompting them to input essential details. Beginning with suggesting a descriptive and engaging title that accurately reflects the proposed topic of discussion, users can then provide a summary or introduction to the thread idea, outlining its relevance and potential for stimulating conversation. The users can then select a particular category that best fits their thread idea to attain more activity on their thread. After filling out the required input fields and approval from an admin, the users can view their threads on the home page.

## Recommend a New Thread !!!

*Initiate a new conversation on your topic of interest with our online community*

1 Add a unique title for your thread !!!

Messi's Dribbling Skills

2 Add a description for the thread

Are you mesmerized by Lionel Messi's extraordinary dribbling skills? Let's start a thread discussing his unparalleled talent on the ball. From his effortless changes of direction to his impeccable close control, Messi's dribbling ability is second to none. Share your favorite moments of Messi leaving defenders in his wake and discuss what makes his dribbling style so unique and effective. Join us in celebrating the artistry of one of football's greatest magicians!

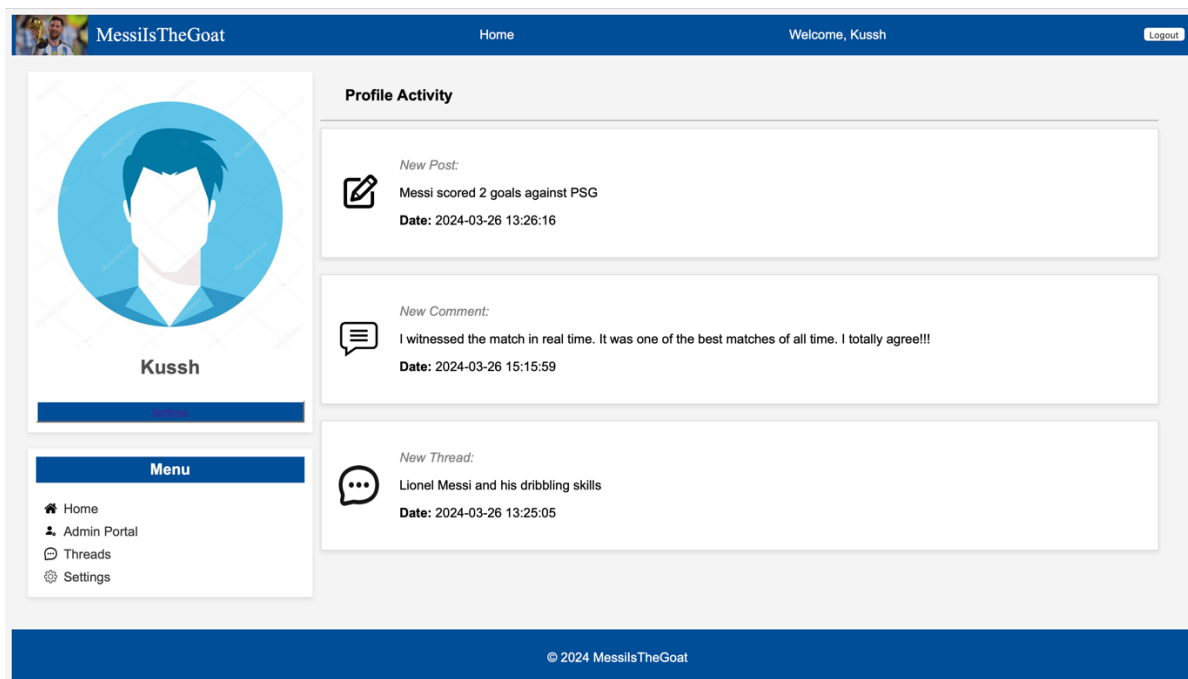
3 Select Category (Optional):

WhyMessitheGOAT

NEW THREAD

## 6. User profile.

Upon clicking “Welcome, User” button from the navigation bar, the users will be able to view their profile. Here, users encounter an overview display of their activity within the forum under “Profile Activity” that allows them to view their interactions with the community in the form of threads, posts, and/or comments with their corresponding time stamps. Moreover, using the sidebar “Menu”, the user can promptly navigate themselves back to the home page, account settings or the admin portal if the user is registered as an admin with the website.

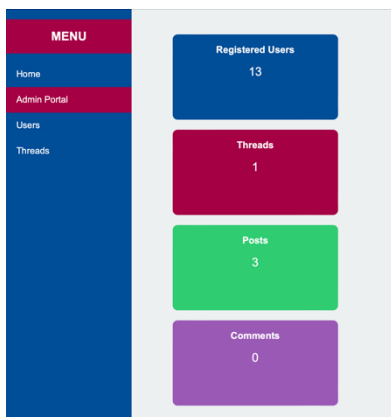


## 7. User profile – settings.

Upon clicking “Settings” button from the profile page, the registered members will be redirected to a user-friendly form prompting them to customize their account settings. The array of settings allows users to modify their profile picture, username, and/or password. After making desirable alterations, users can click “Update Settings” button to successfully change their settings.

The screenshot shows a user profile settings page for a user named 'Kussh'. The page has a dark blue header with the site name 'MessIsTheGoat', a 'Home' link, a 'Welcome, Kussh' message, and a 'Logout' button. The main content area is light gray. On the left, there's a profile picture placeholder with a blue circular background and a white silhouette of a person's head and shoulders. Below the picture is the name 'Kussh' and a 'Settings' button. To the right of the profile picture, there are three sections: 'Change Profile Picture:' with a 'Choose File' button and 'no file selected' text; 'Change Username:' with a text input field containing 'Kussh'; and 'Change Password:' with two text input fields labeled 'Old Password' and 'New Password', and an 'Update Settings' button. Below these sections, a message says 'Profile picture updated successfully.' At the bottom, there's a dark blue footer with the copyright notice '© 2024 MessIsTheGoat'.

## 8. Admin portal – dashboard.




The admin portal provides an insightful overview of key metrics for the discussion forum website, including the total number of registered users, threads, posts, and comments. These metrics offer valuable insights into the platform's growth, user engagement, and content distribution. By tracking these statistics, administrators can make informed decisions to optimize the forum's performance and enhance the overall user experience.

## 9. Admin portal – Users.



Upon clicking the “Users” button from the admin portal sidebar, admins can view registered users and their essential information such as user ID, username, email, joined date stamp and their status on the website. The display allows admins to promptly access user information and take actions such as blocking their account with respect to website policy guidelines.



MessisTheGoat


- Home
- Admin Portal
- Users**
- Threads

### All Users

ID	Username	Email	Joined	Status	Action
15	Monica Geller	monicacleans@gmail.com	2024-03-26 15:37:49	Active	<button>Block</button>
14	Jake The Dog	adventuretimewithjake@gmail.com	2024-03-26 15:37:20	Blocked	<button>Unblock</button>
13	Nick Fury	furnick@gmail.com	2024-03-26 15:36:51	Active	<button>Block</button>
12	Jessica Day	dayJessica@gmail.com	2024-03-26 15:36:23	Blocked	<button>Unblock</button>
11	Nick Miller	nickmiller09@gmail.com	2024-03-26 15:36:04	Active	<button>Block</button>
10	Winston Schmidt	schmidt@gmail.com	2024-03-26 15:35:36	Active	<button>Block</button>
9	John Watson	watson@yahoo.com	2024-03-26 15:34:55	Blocked	<button>Unblock</button>
8	Sherlock Holmes	holmes@gmail.com	2024-03-26 15:33:47	Active	<button>Block</button>
7	Robin Hood	robinhood@gmail.com	2024-03-26 15:31:17	Active	<button>Block</button>
6	Rachel Green	rachgreen@gmail.com	2024-03-26 15:30:44	Active	<button>Block</button>
5	Ted Mosby	mosbyted@yahoo.com	2024-03-26 15:30:15	Blocked	<button>Unblock</button>
4	Chandler Bing	chandlerbing@gmail.com	2024-03-26 15:29:56	Active	<button>Block</button>

## 10.Admin portal – threads.

Upon clicking the “threads” button from the admin portal sidebar, admins can view all threads and their essential information such as thread ID, name, URL, created date stamp, status, members active and toggle buttons to hide/delete threads with respect to website community guidelines.



MessisTheGoat

- Home
- Admin Portal
- Users
- Threads**

### All Threads

ID	Name	URL	Created	Owner	Status	Members	Action
1	Messi's Greatest Goals	/l/messi-goals/	Jan 15, 2024	MessiFan	Active	300,000	<button>Delete</button> <button>Hide</button>
2	Messi vs. Ronaldo	/l/messi-vs-ronaldo/	Feb 10, 2024	FootballDebate	Active	450,000	<button>Delete</button> <button>Hide</button>
3	Messi's Philanthropy	/l/messi-charity/	Mar 5, 2024	CharityWatch	Active	150,000	<button>Delete</button> <button>Hide</button>

## Implementation from a System and Developer’s Perspective

## Summary of Implemented Features

### Unregistered User:

- Can view posts on the home page.
- Can search for posts, threads, and comments by using the search functionality on the navigation bar.
- Can create an account by clicking on the “Sign up” button.
- Have access to the exact styling that a registered user would.

### Registered User:

- Can view posts on the home page.
- Can search for posts, threads, and comments by using the search functionality on the navigation bar.
- Can recommend new threads to the admin.
- Can create posts on any threads and/or category.
- Can vote on posts.
- Can comment on posts.
- Can view their profile activity.
- Can delete their posts.
- Can delete their comments.
- Can view their profile.
- Can update their profile picture, username, password, or email.
- Can log in swiftly using their email or username and password.
- Maintain logged in state while navigating between pages.

### Admin:

- Can view registered users and their essential information such as username, email, date joined and activity status.
- Can view crucial metrics such as registered users, number of posts, number of threads, and number of comments published.
- Can block or unblock users.
- Can hide, restore, or delete threads.
- Can delete any comment or post.

## Description of PHP Files

### 1. **add\_comment.php**

This PHP script facilitates the addition of comments to posts on the website. It begins by configuring error reporting and managing user sessions, including necessary files for user authentication and database connection. Upon ensuring the user is logged in, it receives POST parameters for the post ID and comment content. The script executes a SQL statement to add the comment, providing feedback to the user based on the success or failure of the operation. In cases of missing or invalid parameters, appropriate error messages are displayed.

### 2. **adminPage.php**

This PHP script serves as an admin dashboard for the website. It retrieves and displays key metrics such as the number of registered users, threads, posts, and comments from the database. Admins can access this dashboard after logging in, providing them with valuable insights into the forum's activity and performance.

3. **adminThreadPage.php**  
This PHP script generates an admin dashboard page for managing threads on the website. It retrieves thread data from the database, including thread ID, title, creation date, owner, and visibility status. The page displays this information in a table, allowing the admin to easily view and toggle the visibility of threads by hiding or unhiding them via AJAX requests triggered by button clicks.
4. **adminUser.php**  
This PHP script generates an admin dashboard page for managing users on the website. It retrieves user data from the database, excluding the admin user, and displays it in a table format. The admin can view user details such as ID, username, email, registration date, and status, and can also toggle the status of users between "active" and "blocked" using AJAX requests triggered by button clicks.
5. **category.php**  
This PHP script retrieves posts and threads from the database based on the category name provided in the URL parameter.
6. **config.php**  
This PHP script establishes connection with the database using hostname, username, password, and database name parameters.
7. **createPost.php**  
This PHP script serves as a form for users to create new posts on the website. It validates the user session, retrieves necessary data from the database such as thread titles and category names, and processes the submitted form data to insert a new post into the database. If there are any errors during the form submission, it displays appropriate error messages to the user.
8. **vote\_handler.php**  
This PHP script handles user voting functionality on posts, allowing users to upvote or downvote posts. It first checks if the user is logged in and has provided the necessary parameters for voting (post ID and vote type). Then, it manages the vote transactionally, updating the database to reflect the user's vote and adjusting the post's total vote count accordingly. The script returns a JSON response indicating the success or failure of the vote operation, along with updated vote counts.
9. **createThread.php**  
This PHP script serves as a form for users to recommend and create new threads on the website. It validates the user session, retrieves necessary data from the database such as category names, and processes the submitted form data to insert a new thread into the database. If there are any errors during the form submission, it displays appropriate error messages to the user.
10. **delete\_comment.php**  
This PHP script handles the deletion of comments from the database. It first checks if the user is logged in and has the necessary session data. Then, it verifies if the user is an admin

by querying the database. If the user is an admin, it proceeds to delete the comment associated with the provided comment ID.

11. **delete\_posts.php**

This PHP script manages the deletion of posts from the database. It first checks if the user is logged in and has an active session. Then, it verifies if the user is an admin by querying the database. If the user is an admin, it proceeds to delete the post associated with the provided post ID.

12. **fetch\_posts.php**

This PHP script retrieves posts from the database along with their total vote count and the associated comments. It first constructs a query to fetch posts and their vote counts, ordered by vote count and timestamp. Then, for each post, it constructs another query to fetch comments associated with that post.

13. **fetch\_recent\_posts.php**

This PHP script retrieves the three most recent posts from the database and stores them in an array.

14. **fetch\_top\_threads.php**

This PHP script retrieves the top four threads based on the number of posts associated with each thread from the database.

15. **hide.php**

This PHP script is responsible for hiding a thread and all its associated content (posts and comments).

16. **index.php**

It includes sections for displaying posts, creating new posts and threads (if the user is logged in), and sidebar sections for top threads and recent posts. The file utilizes PHP to fetch data from the database such as posts, top threads, and recent posts, and JavaScript to handle user interactions like voting, commenting, and deleting posts and comments via AJAX requests.

17. **login\_status.php**

This PHP script sets up error reporting, starts a session if not already started, and includes the database configuration file. It checks if a user is logged in by verifying the presence of a 'user\_id' in the session. If a user is logged in, it fetches their details from the database and determines if they are an admin by querying the 'admins' table.

18. **Login.php**

This PHP script handles user authentication for logging in. It verifies the submitted login credentials against the database, including checking if the user account is blocked. If the login is successful and the account is not blocked, it starts a session, sets session variables, and redirects the user to their profile page.

19. **logout.php**

This PHP script initiates the logout process by closing the current session, effectively logging out the user.

20. **search.php**  
This PHP script performs searches across posts, comments, and threads based on the user's input query. Additionally, it includes JavaScript code for AJAX interactions like adding comments, voting on posts, and deleting posts or comments.
21. **SignUp.php**  
It validates user input for username, email, password, and profile picture upload. If validation passes, it inserts the user's information into the database, including hashing the password and storing the profile picture as a blob in the database. If there are any errors during the registration process, appropriate error messages are displayed to the user on the sign-up form.
22. **thread.php**  
This PHP script is responsible for displaying a thread and its associated posts and comments. It fetches the thread details and associated posts from the database, along with their vote counts and usernames. It also handles adding comments to posts via AJAX requests and dynamically reordering posts based on their vote counts. Additionally, it includes JavaScript code to handle voting functionality using AJAX requests.
23. **unhide.php**  
This PHP script is handling a POST request to unhide a thread and its associated content. It first checks if the user is an admin and if a thread ID is provided in the request. Then, it verifies if the thread exists and if it's currently hidden. If the thread is hidden, it updates the database to unhide the thread, its associated posts, and comments.
24. **updateUserStatus.php**  
This PHP script handles a POST request to update the status of a user in the database. Using user ID and the new status ('active' or 'blocked') it then prepares and executes an SQL update statement to modify the user's status accordingly. If the update is successful, it returns a JSON response indicating success; otherwise, it returns an error message.
25. **user\_image.php**  
This PHP script retrieves the profile picture of the currently logged-in user from the database based on their username.
26. **User\_Profile\_Settings.php**  
This PHP script handles user profile settings functionality, allowing users to update their profile picture, username, and password. It retrieves user-specific information from the database and processes POST requests to update the profile settings accordingly.
27. **User\_Profile.php**  
This PHP script fetches user-specific information from the database such as user details, posts, comments, and threads associated with the currently logged-in user. It then dynamically generates HTML content to display the user's profile activity, including their posts, comments, and threads, under profile activity.