PrimeHub - detailed description

By Nawaf Alsuhaibani – 49293153

This Detailed Design Document for PrimeHub provides the complete description of the system.

After completing this document, the reader will be familiar with the site’s features and how the site work at a high-level.

First, each file or a page has its own unique feature starting from the client-side features ending at the server-side implementation. Using mysql as our database management system. The connection used is php PDO connection class.

The homepage <http://cosc360.ok.ubc.ca/49293153/client/index.php> checks weather you are logged in or not, if you are then you can see the changes at the top right. This is accomplished using php state validation. The contents are loaded directly from the database. Each thread has its own id which identify the member who wrote along with the thread’s contents and comments. After clicking any threads, each click goes to the server and change its click count, by doing so, the trending page is loaded based on the number of clicks. Also, the comment box of each thread is hidden using JavaScript along with php login validation method. The box will be available once you are logged in. As you can see, the comment section is structured in a way that you can add your own comment or reply to one. This is accomplished by adding a parent column to the database comment table. Once a comment is added, by using ajax, you don’t need to refresh the page to see your comment. The one limitation here due to time constrain is that once a comment is added you cannot delete it. This can be easily fixed by implementing the profile.php file but instead of threads changing it to comment. Also, under each thread you can see the user who wrote the thread and by clicking the username, this will take you to that user’s profile page which contains a brief summary of the user’s history, number of threads and links to their threads if there is any. This is done by using php by going and fetching the member’s information from the database. The create page is controlled by php, you must be logged in, and your state is valid. If either one of those are false, then you cannot create a category or a thread.

Next, the sign-up page. There are multiple layers of validation on this page. Starting with JavaScript, you cannot submit the form unless every field is filled with the required information. Furthermore; as you type the username and email, an ajax request is fired to check if the username or email is already taken. If one or both are taken you will be notified. Passwords are hashed using php md5 function. By uploading the profile avatar, a JavaScript function is fired to check multiple requirements. It checks if it is an allowable extension and the file is not larger that 10 MB. If everything is filled correctly, php validation also verifies the same requirements. The images are stored both on the server and the database. To avoid duplicate name which can lead to an error, each file that has been uploaded to the server, timestamp is added at the beginning of the filename. Once everything passes the validation layer, a prepared statement is used to insert into the database with the information that the user has entered. A separate table is used to save images. Linked by user id that is unique for each user. That concluded our sign-up page.

Login, it is a very simple login process where you provide username or email with passwords. JavaScript validation method checks if the fields are empty or not. php validation does a few verifications. All of which using prepared statements. First it checks if the user has requested a password recovery. If password recovery has been requested, the user must change the password to a new one and it cannot be the same as the old password. If there is no password recovery request. It also checks if the user account is active or not. if the user account is deactivated then the login process will terminate, and a message will be displayed accordingly. If the user is active, then the login process will continue, if the password or username/email is invalid, then a message will be displayed accordingly. Finally, If the validation process is complete and login completes, redirecting to the main page as a logged in user. Now the user can create a category or a thread.

The create.php file/page does an ajax request to fetch the categories list to choose where the thread goes along with validation of empty spaces in both sided. The limitations here is that a category cannot be edited/deleted. The thread cannot be edited.

The profile page, has one two features. First, changing password. The user has to provide the old password along with the new password. A JavaScript will check if the passwords are matched or not. Also, you cannot use the same password as this will return a no row affected when the prepared statement is executed. Second, view and a delete thread option. Bu clicking the delete button, an ajax request is fired calling a php function that takes that thread’s d and deleting it from the site. Upon success, an alert will pop up saying that the thread has been deleted.

Finally, when a password recovery is requested, the email that the user entered goes into validation. If the email exists, a temporary password is created using php and it is sent to that email. Also, the system makes a note that this user has requested a temporary password.

Logging in as an admin, there is three main features, first listing all users. By clicking list user button, an ajax request is fired and the list of all users if fetched into a table with their information along with 2 actions, both share similar functionality, if any of them is clicked, an ajax request happens, and an alert will pop up in case of success or failure. Second feature is similar to the first one, but this has to do with deleting threads. As mentioned, categories cannot be deleted. A search process takes an input and does a select sql statement with LIKE operator, fetching any users that has is related to a thread or by entering their username or name. Logout function as an admin or non-admin unset all the sessions.