**Design for Logic**

**If the database does not exists, create the website database**

Data Dictionaries

users

| **Field** | **Required** | **Type** | **FIeld Length** | **Key if applicable** |
| --- | --- | --- | --- | --- |
| username | Yes | VARCHAR | 25 | Primary |
| pword | Yes | VARCHAR | 255 |  |
| bio | Yes | VARCHAR | 50,000 |  |

Extra characters are given to the password and biography of the user as more characters will be introduced after sanitization for the input.

postsT

| **Field** | **Required** | **Type** | **FIeld Length** | **Key if applicable** |
| --- | --- | --- | --- | --- |
| dateCreated | Yes | DATETIME | n/a | Primary |
| votes | Yes | INT | n/a |  |
| title | Yes | VARCHAR | 144 |  |
| content |  | VARCHAR | 60000 |  |
| username |  | VARCHAR | 25 | Foreign |

Extra characters are given to the title and content again due to more characters being introduced from the sanitization of the input.

I realise that the votes field here is useless and should be replaced by a post id of sorts but by the time I realised this I had already finished my implementation and I don’t really have the time to go and fix it up right now.

ER Diagram



Pseudocode

PHP files

My variable naming is not amazing as the design changed and went here and there during the implementation so some of the variables names may not make a huge amount of sense

Where sql is used here it will refer to an sql statement in the sql section of the pseudocode

For example SQL (a) will refer to

*create user session variable rownum based on the pagenumber variable*  
  
I will not be including html in my php files as it is already covered by the design for the user interface, I will simply be saying where they will be outputted in the html.

about.php

1. **get probepage cookie**  
   1. user variable initialised and is set to the probepage cookie
2. **connect to database and do query**  
   1. connection variable initialised and set to the msqli\_connect function  
      which connects to the server via it’s domain and username.   
        
      my server doesn’t have a password as it wasn’t given one when it was created and i haven’t gotten round to giving it one
   2. using the mysqli\_select\_db function  
        
      give it the parameters connection variable and website for the database to select  
        
      if this fails output an error message
   3. initialise output variable  
        
      set it to function mysqli\_query  
        
      give it parameters connection variable and SQL (b)  
        
      if this fails then set the output variable to an error message from function mysql\_error where its parameter is connection variable
3. **echo the query**  
   1. while there are rows in the associated array of the query from the output variable  
      1. echo that rows bio field
   2. end while
4. **closing connection and freeing memory used by query**  
   1. call the msqli\_free\_result function to free memory used by the query  
      give it the parameter output variable
   2. call the mysqli\_close function to close the connection with the server  
      give it the parameter connection variable

feed.php

1. **get cookies and posted inputs**  
   1. initialise package variable and set it the the posted
   2. initialise the user variable and set it to the user cookie
   3. initialise the probepage variable and set it to the probepage cookie
2. **take the posted input and split it to get the actual inputs**  
   1. set the package variable to the package variable split on each comma
   2. initialise the pageno variable and set it to the 0th element of the package
   3. initialise the sortmethod variable and set it to the 1st element of the package
   4. initialise the page variable and set it to the 2nd element of the package variable
3. **connect to the server and then do the query**  
   1. connection variable initialised and set to the msqli\_connect function  
      which connects to the server via it’s domain and username.
   2. using the mysqli\_select\_db function  
        
      give it the parameters connection variable and website for the database to select  
        
      if this fails output an error message
   3. initialise the computedString variable  
      set it to the output of the mysqli\_query function  
      parameters being connection variable and SQL(a)  
        
      if this fails then set the computedString variable to an error message from function mysql\_error where its parameter is connection variable
   4. if page variable is set to home  
      1. if sortmethod variable is set to New  
         1. set computedString variable to mysqli\_query function  
            parameters being $connection and SQL(c), this executes an sql query who’s output is the functions output  
              
            if this fails then set the computedString variable to an error message from function mysql\_error where its parameter is connection variable
      2. end if
      3. else if the sort method variable is set to Old  
         1. set computedString variable to mysqli\_query function  
            parameters being $connection and SQL(d)  
              
            if this fails then set the computedString variable to an error message from function mysql\_error where its parameter is connection variable
      4. end else if
   5. else if page variable is set to profile  
      1. if sortmethod variable is set to New  
         1. set computedString variable to mysqli\_query function  
            parameters being $connection and SQL(e)  
              
            if this fails then set the computedString variable to an error message from function mysql\_error where its parameter is connection variable
      2. end if
      3. else if the sort method variable is set to Old  
         1. set computedString variable to mysqli\_query function  
            parameters being $connection and SQL(d)  
              
            if this fails then set the computedString variable to an error message from function mysql\_error where its parameter is connection variable
      4. end else if
   6. end else if
4. **output the query**  
   1. while there are rows in the associated array of computedString from the output variable  
      1. set the title variable to the title variable with all spaces replaced with \_955f\_  
           
         this is necessary for when it is outputted back again
      2. set the title variable to the output of the html\_entity\_decode function  
         with parameters title variable and a flag telling it to include both double and single quotes as characters to be replaced  
           
         this converts previously encoded html entities back to normal
      3. set the content variable to the title variable with all spaces replaced with \_955f\_  
           
         this is necessary for when it is outputted back again
      4. set the content variable to the output of the html\_entity\_decode function  
         with parameters title variable and a flag telling it to include both double and single quotes as characters to be replaced  
           
         this converts previously encoded html entities back to normal
      5. initialise and set the dateCreated variable to the title variable with all spaces replaced with \_955f\_  
           
         this is necessary for when it is outputted back again
      6. output the current rows rownum field, dateCreated variable, votes field, title variable, content variable, username variable  
           
         each separated by a space and with a new line at the end of the output
   2. end while
5. **closing connection and freeing memory used by query**  
   1. call the msqli\_free\_result function to free memory used by the query  
      give it the parameter computedString
   2. call the mysqli\_close function to close the connection with the server  
      give it the parameter connection variable

home.php

In the sidenav (orange expandable sidebar) where the username is meant to be the user cookie is echoed by php (outputted)

login.php

1. **If the posted username and password are empty set an error message**
   1. if username or password are empty
      1. initialise and set the error variable to the string Please enter a username and password
   2. end if
2. **Else if the posted username is not alphanumeric and is not empty set an error message**
   1. Else if the username is not alphanumeric and is not empty  
      1. initialise and set the error variable to the string Your username may only contain alphanumeric characters
   2. end else if
3. **Else connect to the database and execute a query  
     
   then use an if and else statement to determine what to do with the output  
     
   finally free memory used by the query and close the connection**
   1. else
      1. initialise and set the user variable to the posted username
      2. initialise and set the the pword to the posted password
      3. set the password variable to the function htmlentities to convert htmlentities to plain text  
         give it the parameters pword variable and flag ENT\_QUOTES to include double and single quotes
      4. set the password variable to the function addslahes which escapes any slashes  
         give it the parameter pword variable
      5. initialise and set the variable connection to msqli\_connect function  
         which connects to the server via it’s domain and username.
      6. using the mysqli\_select\_db function  
           
         give it the parameters connection variable and website for the database to select  
           
         if this fails output an error message
      7. initialise and set the cred variable to the mysqli\_query function  
           
         give it parameters connection variable and SQL(g)  
           
         if this fails set it to the error message outputted by the mysqli\_error function on the connection variable
      8. set cred variable to the associated array to the query from the cred variable
      9. if the COUNT(\*) field of the cred array is equal to one then that means that the inputs credentials are correct THEN  
         1. create and set the cookie user to the user variable and have it expire a day from the day the cookie was created, have its path set to the current domain and have its domain be the domain of the server
         2. create and set the cookie pageno to the 0 and have it expire a day from the day the cookie was created, have its path set to the current domain and have its domain be the domain of the server
         3. create and set the cookie sortmethod to the string New and have it expire a day from the day the cookie was created, have its path set to the current domain and have its domain be the domain of the server
         4. create and set the cookie probepage to the user variable and have it expire a day from the day the cookie was created, have its path set to the current domain and have its domain be the domain of the server
         5. change the current page to the home page via the header function
      10. end if
      11. else  
          1. set and initialize the error variable to the string Please enter a username and password
      12. end else
      13. call the msqli\_free\_result function to free memory used by the query  
          give it the parameter cred variable
      14. call the mysqli\_close function to close the connection with the server  
          give it the parameter connection variable
   2. end else

In the input for the username the value is the outputted username sent by post

Under the sign up button the error is echoed

notifications.php

In the sidenav (orange expandable sidebar) where the username is meant to be the user cookie is echoed by php (outputted)

profile.php

The title of the page in the tab is the probepage cookie concatenated with ‘s page giving for example, *Example’s page*.

In the sidenav (orange expandable sidebar) where the username is meant to be the user cookie is echoed by php (outputted)  
  
Where the username of the current page’s user is meant to be the probepage cookie is outputted

register.php

1. **If the posted username or password is empty and the posted username as a string is not the string 0 set an error message**  
   1. if posted username is empty or posted password is empty and the string value of the posted username is not the string 0 then  
      1. set and initialize the variable error to the string Please enter a username and password
   2. end if
2. **else if the username is not alphanumeric and is not empty set an error message**  
   1. else if the posted username is not alphanumeric and is not empty then  
      1. set and initialize the variable error to the string Your username may only contain alphanumeric characters
   2. end else if
3. **else if the username is too long then set an error message**  
   1. else if the length of posted username is longer than 25 characters then  
      1. set and initialize the variable error to the string Your username is too long
   2. end else if
4. **else if the passwords do not match then set an error message**  
   1. else if the posted password does not equal the posted re entered password then  
      1. set and initialize the variable error to the string Your passwords do not match
   2. end else if
5. **else if the password is too short and is not empty then set an error message**  
   1. else if the password is lesser than or equal to 6 characters in length then   
      1. set and initialize the variable error to the string Your password is too long
   2. end else if
6. **else if the password is too long then set an error message**
   1. else if the password is greater than 50 characters in length then

* + 1. set and initialize the variable error to the string Your password is too long
  1. end else if

1. **the username and password were not empty and the posted password and posted re entered password are the same then connect to the server and execute a sql query then do an if else on the outputted query finally free memory used by query and close the connection**
   1. else if the posted username and password are not empty and the posted password and re entered password are equal to each other then  
      1. set and initialise the variable user to the posted username
      2. set and initialise the variable pword to the posted password
      3. set and initialise the variable bio to the string Hello! My name is concatenate the user variable concatenate the string .I joined WorldTree on concatenate the current date in the form d/M/Y concatenate the string !
      4. set the pword variable to the output of the htmlentities function which converts html entities to plain text  
         give it the parameters pword and ENT\_QUOTES flag to include double and single quotes as characters to convert to plain text
      5. initialise and set the variable connection to msqli\_connect function  
         which connects to the server via it’s domain and username.
      6. using the mysqli\_select\_db function  
           
         give it the parameters connection variable and website for the database to select  
           
         if this fails output an error message
      7. set and initialise the variable exists to the mysqli\_query function with parameters connection variable and SQL(h) which sends a query to the server to be executed  
           
         if this fails set it to the error message outputted by the mysqli\_error function on the connection variable
      8. set exists variable to the array associated to the sql query
      9. if the COUNT(\*) field of exists is equal to 0 then that means that there are no users with this username then  
         1. set and initialise the variable exists to the mysqli\_query function with parameters connection variable and SQL(i) which sends a query to the server to be executed  
              
            if this fails set it to the error message outputted by the mysqli\_error function on the connection variable
         2. send the user to the login page via the header function
      10. end if
      11. else  
          1. set and initialise the error variable to the string That username already exists
      12. end else
      13. through the mysqli\_free\_result, give it the parameter exists variable  
            
          this frees the memory associated with the query to be used elsewhere
      14. through the mysqli\_close function, give it the parameter connection variable and close the connection to the server

in the username input on the page give it the value of the posted username

searchpage.php

1. **Get cookies and set variables**  
   1. initialise and set the variable search to the posted search
   2. initialise and set the error cookie to the string none
   3. initialise and set the variable error to a blank string
2. **if the search is empty set an error message and change the error cookie**  
   1. if the search is empty and the string of the search variable is not equal to the string 0  
      1. error variable is set to the string Please enter a search value <br>
      2. set the error cookie to the error variable
   2. end if
3. **else if the search values are not alphanumeric set an error message and change the error cookie**  
   1. else if the search variable is not alphanumeric   
      1. set the error variable to the string You may only search alphanumeric values <br>
      2. set the error cookie to the error variable
   2. end else if
4. **else make a connection with the server and do a query**
   1. else
      1. connection variable initialised and set to the msqli\_connect function  
         which connects to the server via it’s domain and username.   
           
         my server doesn’t have a password as it wasn’t given one when it was created and i haven’t gotten round to giving it one
      2. using the mysqli\_select\_db function  
           
         give it the parameters connection variable and website for the database to select  
           
         if this fails output an error message
      3. initialise the output variable  
         set it to the output of the mysqli\_query function  
         parameters being connection variable and SQL(a)  
           
         if this fails then set the output variable to an error message from function mysql\_error where its parameter is connection variable
      4. set computedString variable to mysqli\_query function  
         parameters being $connection and SQL(j)  
           
         if this fails then set the computedString variable to an error message from function mysql\_error where its parameter is connection variable
   2. end else

In the sidenav (orange expandable sidebar) where the username is meant to be the user cookie is echoed by php (outputted)

There is more php script in this file in the posts div, here is the pseudocode for it

1. **if the output variable is set then  
     
   if the number of rows in the output variable is greater than 0 then echo those rows  
     
   end if  
     
   else set an error msg and change the error cookie  
     
   end else  
     
   free memory used by the query and close the connection  
     
   output the error**
   1. if the output variable is set then  
      1. if, through the use of the mysqli\_num\_rows function with parameter output variable, the number of rows in the output is greater than 0 then  
         1. while there are rows in the associated array for the output do  
            1. initialise and set the bio variable to the current rows bio field
            2. set the bio variable to the bio variable with all its spaces replaced with the string \_955f\_  
                 
               this is necessary for the output
            3. output the current rows rownum field, username field and bio variable.  
                 
               rownum, username and bio are separated by a space with a new line, <br>, at the end of the string
         2. end while
      2. end if
      3. else  
         1. set the error variable to the string No results were found for your input
         2. set the error cookie to the error variable
      4. end else
      5. through the use of the mysqli\_free\_result function, free the memory used by the output variables query by setting it as a parameter
      6. call the mysqli\_close function to close the connection with the server  
         give it the parameter connection variable
   2. end if
   3. output the error variable

createpost.php

In the sidenav (orange expandable sidebar) where the username is meant to be the user cookie is echoed by php (outputted)

createpostI.php

In the sidenav (orange expandable sidebar) where the username is meant to be the user cookie is echoed by php (outputted)

postprocessing.php

1. **Receive variables from post and get cookies and set variables**  
   1. initialise and set the variable title as the string of the posted title
   2. initialise and set the variable content as the string of the posted content
   3. initialise and set the variable error to a blank string
   4. initialise and set the variable user to the user cookie
2. **If the length of the title variable is greater than 144 then set an error message**
   1. if the length of the title string is greater than 144 characters then  
      1. set the error variable to the string Your title must be shorter than or equal to 144 characters in length
   2. end if
3. **Else if the length of the content variable is longer than 50,000 characters than output an error message**  
   1. else if the length of the content string is greater than 50,000 characters then  
      1. set the error variable to the string Your post must be shorter than or equal to 50,000 characters in length
   2. end else if
4. **Else if the title variable is not empty or title is equal to the string 0 then connect to the server and execute an sql query then close the connection with the server and change the page to the home page**  
   1. else if the title is not empty or the title is equal to the string 0 then  
      1. set the variable title to the output of the function sanitize with the title variable as a parameter
      2. set the content variable to the output of the function sanitize with the content variable as a parameter
      3. connection variable initialised and set to the msqli\_connect function  
         which connects to the server via it’s domain and username.   
           
         my server doesn’t have a password as it wasn’t given one when it was created and i haven’t gotten round to giving it one
      4. using the mysqli\_select\_db function  
           
         give it the parameters connection variable and website for the database to select  
           
         if this fails output an error message
      5. call the function mysqli\_query  
           
         give it parameters connection variable and SQL (k)  
           
         if this fails then set the output variable to an error message from function mysqli\_error where its parameter is connection variable
      6. call the mysqli\_close function to close the connection with the server  
         give it the parameter connection variable
      7. change the page to the home page via the header function
   2. end else if
5. **Else if the title empty and not equal to the string “0” then set an error message**
   1. else if the title variable is empty and is not equal to 0 then  
      1. set the error variable to the string Please enter a value for your title
   2. end if
6. **Create a function called sanitize with parameter text variable and use it to make inputs safe for the database to handle**  
   1. declare function sanitize with parameter text variable as an input  
      1. set the text variable to the function htmlentities to convert htmlentities to plain text  
         give it the parameters text variable and flag ENT\_QUOTES to include double and single quotes
      2. set the text variable to the function addslahes which escapes any slashes  
         give it the parameter text variable
      3. return the text variable
   2. function end

In the sidenav (orange expandable sidebar) where the username is meant to be the user cookie is echoed by php (outputted)

settingsaccount.php

In the sidenav (orange expandable sidebar) where the username is meant to be the user cookie is echoed by php (outputted)

settingsaccountsubmit.php

1. **if the passwords do not match then set an error message**  
   1. else if the posted password does not equal the posted re entered password then  
      1. set and initialize the variable error to the string Your passwords do not match
   2. end else if
2. **else if the password is too short and is not empty then set an error message**  
   1. else if the password is lesser than or equal to 6 characters in length then   
      1. set and initialize the variable error to the string Your password is too long
   2. end else if
3. **else if the password is too long then set an error message**
   1. else if the password is greater than 50 characters in length then

* + 1. set and initialize the variable error to the string Your password is too long
  1. end else if

1. **else if password is not empty and the posted password and posted re entered password are the same then connect to the server and execute a sql query then set the error message so as to let the user know the password has been update, close the connection**
   1. else if the posted password is not empty and the posted password and re entered password are equal to each other then  
      1. set and initialise the variable user to the user cookie
      2. set and initialise the variable pword to the posted password
      3. set the pword variable to the output of the htmlentities function which converts html entities to plain text  
         give it the parameters pword and ENT\_QUOTES flag to include double and single quotes as characters to convert to plain text
      4. set the pword variable to the output of the addslashes function with the pword as the parameter. this escapes any slashes.
      5. initialise and set the variable connection to msqli\_connect function  
         which connects to the server via it’s domain and username.
      6. using the mysqli\_select\_db function  
           
         give it the parameters connection variable and website for the database to select  
           
         if this fails output an error message
      7. set and initialise the variable query to the mysqli\_query function with parameters connection variable and SQL(l) which sends a query to the server to be executed  
           
         if this fails set it to the error message outputted by the mysqli\_error function on the connection variable
      8. set and initialise error variable to the string Password updated
      9. through the mysqli\_close function, give it the parameter connection variable and close the connection to the server
   2. end else if

In the sidenav (orange expandable sidebar) where the username is meant to be the user cookie is echoed by php (outputted)

settingsdelete.php

In the sidenav (orange expandable sidebar) where the username is meant to be the user cookie is echoed by php (outputted)

settingsdeletesubmit.php

1. **set an error variable  
   if the first checkbox and second checkbox are checked then connect to the server and delete their account and posts that they have created  
   then delete all cookies and set the page to the login page**
   1. set and initialise the variable error to a blank string
   2. if deletea is posted and deleteb is posted then   
      1. set and initialise the variable user to the user cookie
      2. initialise and set the variable connection to msqli\_connect function  
         which connects to the server via it’s domain and username.
      3. using the mysqli\_select\_db function  
           
         give it the parameters connection variable and website for the database to select  
           
         if this fails output an error message
      4. call the mysqli\_query function with parameters connection variable and SQL(n) which sends a query to the server to be executed  
           
         if this fails set it to the error message outputted by the mysqli\_error function on the connection variable
      5. call the mysqli\_query function with parameters connection variable and SQL(o) which sends a query to the server to be executed  
           
         if this fails set it to the error message outputted by the mysqli\_error function on the connection variable
      6. through the mysqli\_close function, give it the parameter connection variable and close the connection to the server
      7. change the page to the login page
      8. if cookies have been set on the web server then  
         1. initialise and set the cookies variable to the array of cookies on the server created by splitting it on the semi colon
         2. for each cookie in the cookies array  
            1. initialise and set the variable parts to the array created from exploding the current cookie on it’s = sign
            2. initialise and set the variable name to the 0th element of the array created from parts and trim said element to get rid of all its whitespace at the start and end
            3. set the cookie with the same name as that within the name variable to blank and to have expired 1000 seconds before the time when this command is executed
            4. set the cookie with the same name as that within the name variable to blank and to have expired 1000 seconds before the time when this command is executed with a path the same as that of the domain
         3. end for
      9. end if
   3. end if
   4. else  
      1. set the error variable to the string <br><br>Please check both boxes to delete your account
   5. end else

In the sidenav (orange expandable sidebar) where the username is meant to be the user cookie is echoed by php (outputted)

under the submit button for deleting your account, echo the error variable

settingspreferences.php

1. **if bio has been posted and is lesser than or equal to 50,000 characters in length then   
     
   connect to the server and execute an sql query to update it and set msg to say that the bio has been updated  
     
   else set msg to say that that bio is too long**  
   1. if bio has been posted then   
      1. if the output of the function strlen on the parameter of the posted bio is lesser than or equal to 50000 characters in length then  
         1. initialise and set the variable bio to be the output of the sanitize function which acts on the parameter the posted bio
         2. initialise and set the user variable to the user cookie
         3. initialise and set the variable connection to msqli\_connect function  
            which connects to the server via it’s domain and username.
         4. using the mysqli\_select\_db function  
              
            give it the parameters connection variable and website for the database to select  
              
            if this fails output an error message
         5. call the mysqli\_query function with parameters connection variable and SQL(m) which sends a query to the server to be executed  
              
            if this fails set it to the error message outputted by the mysqli\_error function on the connection variable
         6. set and initialise the variable error to the string Bio updated
         7. through the mysqli\_close function, give it the parameter connection variable and close the connection to the server
      2. end if
      3. else  
         1. set and initialise the error variable to the string That bio is too long
      4. end else
   2. end if
2. **Create a function called sanitize with parameter text variable and use it to make inputs safe for the database to handle**  
   1. declare function sanitize with parameter text variable as an input  
      1. set the text variable to the function htmlentities to convert htmlentities to plain text  
         give it the parameters text variable and flag ENT\_QUOTES to include double and single quotes
      2. set the text variable to the function addslahes which escapes any slashes  
         give it the parameter text variable
      3. return the text variable
   2. function end

In the sidenav (orange expandable sidebar) where the username is meant to be the user cookie is echoed by php (outputted)

In the area between textarea for the bio if the bio has been posted echo the posted bio  
  
under the submit button, if the bio has been posted echo the error message

SQL for php files

for pages containing rownum

a)

create user session variable rownum based on the pagenumber variable

about.php

b)

select the bio field from users table where username is the same as the user variable

feed.php

c)

Select rownum, have it increment for each field and represent this field with the name rownum, title, dateCreated, votes, content, username from the postsT table

Order the select by dateCreated descending

Limit number of fields by 50 and have the outputted fields begin from row which is the same number as pagenumber.

d)

Select rownum, have it increment for each field and represent this field with the name rownum, title, dateCreated, votes, content, username from the postsT table

Order the select by dateCreated ascending

Limit number of fields by 50 and have the outputted fields begin from row which is the same number as pagenumber.

e)

Select rownum, have it increment for each field and represent this field with the name rownum, title, dateCreated, votes, content, username from the postsT table

Where the username field is = to the probepage variable

Order the select by dateCreated descending

Limit number of fields by 50 and have the outputted fields begin from row which is the same number as pagenumber.

f)

Select rownum, have it increment for each field and represent this field with the name rownum, title, dateCreated, votes, content, username from the postsT table

Where the username field is = to the probepage variable

Order the select by dateCreated ascending

Limit number of fields by 50 and have the outputted fields begin from row which is the same number as pagenumber.

login.php

g)

Select the number of fields from users table

where username is like the user variable and

password is like the password variable

register.php

h)

select the number of fields from users

where the username field is the same as the user variable

i)

insert into the users table where the fields are username, pword, bio

values user variable, password variable, bio variable respectively

searchpage.php

j)

Select rownum, have it increment for each field and represent this field with the name rownum, username, bio from the users table

where username is like the searchvariable with a wildcard at the end

postprocessing.php

k)

insert into the postsT table for the fields dateCreated,votes,title,content and username

values output of current\_timestamp which gives the current date time, 0, the title variable, content variable and user variable respectively

settingsacountsubmit.php

l)

update the users tables

set the pword field to the pword variable

where username field is the user variable

settingspreferences.php

m)

update users table

set the bio field to the bio variable

where username field is the user variable

settingsdeletesubmit.php

n)

Delete from the postsT table

where username is the user variable

o)

Delete from the users table

where username is the user variable

JavaScript Functions

openNav

1. **declare function open nav when called do**
   1. declare function openNav
2. **if the page is any of the login or registration pages, make the sidebar 100% and have the transition time be none then add the colour gradient for the background image**  
   1. if the current page is the login page or the index page or the registration page or the php login page or the php registration page then  
      1. get the element with id mySidenav and change its css width to 100%
      2. get the element with id mySidenav and change its css transition to none
      3. get the element with id mySidenav and change its css background image to a linear gradient from #6002ee to #ee6002
   2. end if
3. **else make the sidenav come out 350px**  
   1. else  
      1. get the element with id mySidenav and change its css width to 350px;
   2. end else
   3. end function

closeNav

1. **Declare function close nav and when called change the sidenavs width to 0**  
   1. declare function closeNav  
      1. get the element with id mySidenav and change its css width to 0
   2. end function

register

1. **Declare function register and when called change the current page to the registration page**
   1. declare function register  
      1. get the current windows location and change it to the /Register.html
   2. end function

dropdown

1. **Declare function dropdown and when called change open the dropdown menu**  
   1. declare function dropdown  
      1. get the element with id dropcontent and add to its class list the class show
   2. end function

profile

1. **Declare function profile and when called edit the element with grid container id’s css, gridTemplateAreas attribute to have the correct layout for the profile page**  
   1. declare function profile  
      1. get the element with id grid container and in its css change its gridTemplateAreas to equal   
           
         'header header header header header header header header'  
         'profile profile profile profile profile profile profile profile'  
         'posts posts posts posts posts posts posts util'
   2. end function

logout

1. **declare function logout and when called delete all cookies**  
   1. declare function logout  
      1. declare variable cookies which is set to the cookies on the website split on the semicolon creating an array
      2. for the length of the cookies array do  
         1. declare the variable cookie and set it to the current cookies element
         2. declare the variable eqPos and set it to the index of the = in the variable cookie string
         3. declare the variable name and set it to   
            1. if equal pos is greater than -1, the substring of cookie from its first character to eqPos, thereby getting the name of the cookie
            2. else variable cookie
         4. set the cookie with name variable name to equal nothing and to have expired on expires=Thu, 01 Jan 1970 00:00:00 GMT therefore deleting the cookie
      3. end for
   2. end function

sortnew

1. **declare the function sortnew and have it change the sortmethod cookie to new, empty the posts div and then call the request function with feed.php as its parameter**  
   1. declare function sortnew  
      1. set the sortmethod cookie to new
      2. get the element with id posts and set its innerHTML to nothing
      3. call the request function with parameter feed.php
   2. end function

sortold

1. **declare the function sortold and have it change the sortmethod cookie to old, empty the posts div and then call the request function with feed.php as its parameter**  
   1. declare function sortold  
      1. set the sortmethod cookie to new
      2. get the element with id posts and set its innerHTML to nothing
      3. call the request function with parameter feed.php
   2. end function

forward

1. **declare the function forward which gets the cookie pageno and adds 50 to it then empties the posts div and calls the request function with feed.php as its parameter**  
   1. declare function forward  
      1. declare variable pageno and set it to the string created from the using the replace function on the cookies for the site outputting the first substring match to the following regex  
           
         from start of string or anything, begin capture after <some amount of white space>, pageno <whitespace> = <whitespace>, end capture at semicolon, return value  
           
         therefore getting the pageno
      2. set pageno to itself, cast as a number and add 50
      3. set the cookie pageno to the variable pageno
      4. get the element with id posts and set its innerHTML to nothing
      5. call the request function with parameter feed.php
   2. end function

back

1. **declare the function back which gets the cookie pageno and subtracts 50 to it then empties the posts div and calls the request function with feed.php as its parameter**
   1. declare function forward  
      1. declare variable pageno and set it to the string created from the using the replace function on the cookies for the site outputting the first substring match to the following regex  
           
         from start of string or anything, begin capture after <some amount of white space>, pageno <whitespace> = <whitespace>, end capture at semicolon, return value
      2. set pageno to itself, cast as a number and subtract 50
      3. set the cookie pageno to the variable pageno
      4. get the element with id posts and set its innerHTML to nothing
      5. call the request function with parameter feed.php
   2. end function

feed

1. **declare function feed  
   if the pathname of the page is /profile.php then set the page cookie to profile  
   else if the pathname of the page is /home.php set the page cookie to home  
   request function with feed.php as its parameter**  
   1. declare feed function  
      1. if the pathname of the current page is equal to /profile.php then  
         1. set the page cookie to the string profile  
            set its path to the domain  
            set the domain to 127.0.0.1
      2. end if
      3. else if pathname of the current page is equal to /home.php then  
         1. set the page cookie to the string home  
            set its path to the domain  
            set the domain to 127.0.0.1
      4. end else if
      5. call the request function with parameter feed.php
   2. end function

reset

1. **declare rest function, when called set the pageno, sortmethod cookie and probepage to what they were originally**  
   1. declare reset function  
      1. set the pageno cookie equal to 0
      2. set the sortmethod cookie to the string New
      3. set the probepage cookie to the string created from the using the replace function on the cookies for the site outputting the first substring match to the following regex  
           
         from start of string or anything, begin capture after <some amount of white space>, user <whitespace> = <whitespace>, end capture at semicolon, return value  
           
         therefore setting the probepage cookie to the logged in users username

follows

1. **Declare the follows function which when called empties the posts div and sets it to an error message**  
   1. declare the follows function  
      1. get the element with id posts and set its innerHTML to nothing
      2. get the element with id posts and set its innerHTML to the string  
         The follows page is a work in progress
   2. end function

followers

1. **Declare the followers function which when called empties the posts div and sets it to an error message**  
   1. declare the followers function  
      1. get the element with id posts and set its innerHTML to nothing
      2. get the element with id posts and set its innerHTML to the string  
         The followers page is a work in progress
   2. end function

search

1. **Declare function search, get query from posts and set error variable to the error cookie**  
   1. declare search function  
      1. initialise and set the query variable to the element with id posts innerHTML  
           
         therefore getting the search results
      2. initialise and set the error variable to the string created from the using the replace function on the cookies for the site outputting the first substring match to the following regex  
           
         from start of string or anything, begin capture after <some amount of white space>, error <whitespace> = <whitespace>, end capture at semicolon, return value  
           
         therefore setting the error cookie to the error variable
2. **If the error variable is set to none then create a 2d array where each row is a record and each column is an array, then empty the posts div**  
   * 1. if the error variable is set to none then  
        1. initialise and set the variable array to the array outputted by splitting the query on each <br> or newline
        2. initialise and set the variable out to a blank array
        3. for element in array length minus 1 do  
           1. append to the out array the array created from splitting the current element on each space  
                
              therefore creating a 2d array containing each record in each element of the out array
        4. end for
        5. get the element with id posts and set its innerHTML to nothing
3. **for element in array minus 1 create that number of posts containing the searched users username as the title which is also a link to their profile page along with making the contents of the post the searched users bio**  
   * + 1. for element in length of array minus 1 do  
          1. initialise and set the variable parent to the element with id posts
          2. initialise and set the variable post to a newly created div
          3. give the post variable the attribute id which is set to post
          4. give the post variable the attribute class which is set to post
          5. initialise and set the variable vote to a newly created div
          6. give the vote variable the attribute id which is set to vote
          7. give the vote variable the attribute class which is set to vote
          8. initialise and set the variable user to a newly created div
          9. give the user variable the attribute id which is set to user
          10. give the user variable the attribute class which is set to title

initialise and set the variable useri to a newly created div

give the useri variable the attribute href which is set to profile.php

give the useri variable the attribute onclick which is set to the function probeprofile with parameter being the current element of out’s 1st element  
  
therefore setting the parameter to the name of the user that is currently being displayed here

initialise and set the variable userj to a newly created textnode containing the string that’s the current element of out’s 1st element  
  
therefore setting the textnode to the current records username

* + - * 1. append userj text to the useri link
        2. append the useri link to the user div
        3. initialise and set the variable content to a newly created div
        4. give the content variable the attribute id which is set to content
        5. give the content variable the attribute class which is set to content
        6. set the current element of out’s 2nd element to itself with each and every occurence of \_955f\_ with a space thereby making the bio what it should look like

initialise and set the variable contenti to the text node containing the current element of out’s 2nd element

* + - * 1. append the textnode contenti to the content div
        2. append to the parent variable the post variable
        3. append to the post variable the vote variable
        4. append to post variable the user variable
        5. append to post variable the content variable
      1. end for
    1. end if
  1. end function

request

1. **declare the request function with the parameter url  
   make a new xmlhttprequest**  
     
   declare function request with parameter url  
   1. initialise and set the variable httpRequest to a new XMLHttpRequest function
   2. when the httpRequest is ready declare an anonymous function
2. **try if the request is done and the request passed to   
   initialise and set a variable to the text outputted from the php  
   create a 2d array from the inputted text with each row being a record and each column being a field**  
   * 1. try, if the request is done then  
        1. if the request passes, meaning code 200 then  
           1. initialise and set the variable query to the response text from the httpRequest
           2. initialise and set the variable array to the array outputted by splitting the query on each <br> or newline
           3. initialise and set the variable out to a blank array
           4. for element in array length minus 1 do

append to the out array the array created from splitting the current element on each space  
  
therefore creating a 2d array containing each record in each element of the out array

* + - * 1. end for

1. **change the \_955f\_ in the title, content and dateCreated fields to spaces**  
   * + - 1. for element in array length minus 1 do

set the current element of out’s 1st element to itself with each and every occurence of \_955f\_ with a space thereby putting spaces back into the dateCreated field

set the current element of out’s 3rd element to itself with each and every occurence of \_955f\_ with a space thereby putting spaces back into the title

set the current element of out’s 4th element to itself with each and every occurence of \_955f\_ with a space thereby putting spaces back into the content

* + - * 1. end for

1. **Create the feed by filling the posts div with posts, containing their post number, title, contents, creator and the date they were created  
   you can click on the creator to go to their profile page**  
   * + - 1. initialise and set the variable counter to 0
         2. initialise and set the variable divs to an empty array
         3. for element in length of array minus 1 do

initialise and set the variable parent to the element with id posts

initialise and set the variable post to a newly created div

give the post variable the attribute id which is set to post

give the post variable the attribute class which is set to post

append the post variable to posts

initialise and set the variable vote to a newly created div

give the vote variable the attribute id which is set to vote

give the vote variable the attribute class which is set to vote

append the vote variable to post

for j from 0 to 2 inclusive do

initialise and set the divs counter element to a newly created div

give divs counter element the attribute id set to the string div concatenated with the number counter

if j is 0 then do

append to the vote div the divs counter element div

end if

else if j is 1 then do

initialise and set the variable votesi to a new text node containing the string   
current element of out’s 0th element therefore the rownumber

append to divs counter element div the votesi variable

append to vote divs counter element

end else if

else if js is 2 then

append to the vote div the divs counter element div

end else if

set counter to itself and add 1 to it

end for

initialise and set the variable title to a newly created div

give the title variable the attribute id which is set to title

give the title variable the attribute class which is set to title

initialise and set the variable titlei to a newly created textnode containing the string that’s the current element of out’s 3rd element  
  
therefore setting the textnode to the title of the post

append titlei variable text to the title

append the title variable to the post

initialise and set the variable content to a newly created div

give the content variable the attribute id which is set to content

give the content variable the attribute class which is set to content

initialise and set the variable contenti to the text node containing the current element of out’s 2nd element

append contenti variable to content

append the content variable to post

initialise and set the subbar variable to a newly created div element

give the subbar the attribute id set to subbar

give the subbar the class attribute set to subbar

initialise and set the createdby variable to a text node containing the string Created by

append the createdby variable to the subbar

initialise and create the username variable set to a newly created link element

give the username the attribute href and set that to profile.php

give the username the attribute onclick set to the function probeprofile which contains the parameter the current element of out’s 5th element thereby setting it to the post creators profile page

initialise and set the variable usernamehtml to a newly created textnode containing the string the current element of out’s 5th element

append to username the usernamehtml variable

append to subbar the username variable

initialise and set the variable datecreated to a newly created text node containing the string at concatenated with the current element of out’s 1st element thereby saying the date the post was created

append the datecreated variable to the subbar

append the subbar variable to post

* + - * 1. end for

1. **Declare the variable pageno and set it to the pageno cookie**  
   * + - 1. declare variable pageno and set it to the string created from the using the replace function on the cookies for the site outputting the first substring match to the following regex  
              
            from start of string or anything, begin capture after <some amount of white space>, pageno <whitespace> = <whitespace>, end capture at semicolon, return value  
              
            therefore getting the pageno
2. **if the number of posts is 50 and we’re on the first page then create a next page button to load the next page of posts**  
   * + - 1. if length of the out array is 50 and page number variable is 0 then

initialise and set the feednav variable to a newly created div

give the feednav variable the attribute id with it being set to feednav

initialise and set the variable forward to a newly created button

give it the forward variable the id attribute and set it to forward

give the forward variable the id attribute onclick and set it to the forward function

initialise and set the variable ftext to a newly created text node containing the string   
next >

append to forward the variable ftext

append to feednav the variable forward

* + - * 1. end if

1. **else if the number of posts is 50 then add a back button that will load the previous page of post and a forward button which will load the next page of posts**  
   * + - 1. else if the length of the out array is 50 then

initialise and set the feednav variable to a newly created div

give the feednav variable the attribute id with it being set to feednav

initialise and set the variable back to a newly created button

give it the back variable the id attribute and set it to back

give the backvariable the id attribute onclick and set it to the back function

initialise and set the variable btext to a newly created text node containing the string   
< prev

append to back the variable btext

append to feednav the variable back

initialise and set the variable forward to a newly created button

give it the forward variable the id attribute and set it to forward

give the forward variable the id attribute onclick and set it to the forward function

initialise and set the variable ftext to a newly created text node containing the string   
next >

append to forward the variable ftext

append to feednav the variable forward

* + - * 1. end else if

1. **else if the number of posts on the page is lesser than 50 and the page number is not equal to 0 then add a previous button to load the previous page of posts**  
   * + - 1. else if length of the out array is lesser than 50 and the pageno is not equal to 0 then

initialise and set the feednav variable to a newly created div

give the feednav variable the attribute id with it being set to feednav

initialise and set the variable back to a newly created button

give it the back variable the id attribute and set it to back

give the backvariable the id attribute onclick and set it to the back function

initialise and set the variable btext to a newly created text node containing the string   
< prev

append to back the variable btext

append to feednav the variable back

* + - * 1. end else if

1. **add a newline at the bottom of the posts div so that its easier to click on the next page button**  
   * + - 1. get the element with id posts and append to the end of its innerHTML a newline <br>
       1. end if
2. **else if there’s an error and the passed http request does not have a status code of 200 then alert an error message**  
   * + 1. else  
          1. create an alert containing the string There was a problem with the request.
       2. end else
     1. end try and if
3. **if the try fails then catch the error and send it to the console**  
   * 1. catch the error variable  
        1. output to console the string Caught Exception: concatenated with error.description
     2. end catch
   1. end anonymous function
4. **open and send a http request via post to the url asynchronously with the contents of the post being the pageno, sortmethod and page cookies put into an array as ajax doesn’t like sending more than one thing at a time**  
   1. using the httpRequest variable open an asynchronous post request with the url variable as the destination
   2. declare variable pageno and set it to the string created from the using the replace function on the cookies for the site outputting the first substring match to the following regex  
        
      from start of string or anything, begin capture after <some amount of white space>, pageno <whitespace> = <whitespace>, end capture at semicolon, return value  
        
      therefore getting the pageno cookie
   3. declare variable sortmehod and set it to the string created from the using the replace function on the cookies for the site outputting the first substring match to the following regex  
        
      from start of string or anything, begin capture after <some amount of white space>, sorthmethod <whitespace> = <whitespace>, end capture at semicolon, return value  
        
      therefore getting the sortmethod cookie
   4. declare variable page and set it to the string created from the using the replace function on the cookies for the site outputting the first substring match to the following regex  
        
      from start of string or anything, begin capture after <some amount of white space>, page <whitespace> = <whitespace>, end capture at semicolon, return value  
        
      therefore getting the page cookie
   5. initialise and set the package variable to an array composed of the pageno, sortmethod and page variable, with each respectively being the 0th, 1st and 2nd element of it
   6. using the httpRequest variable set the request header to specify that the content type is a query string
   7. using the httpRequest variable send the package variable by post with the package being encoded as a URI component
5. end function

probeprofile

1. **declare probeprofile and have its parameter be the variable profile  
   change the probepage cookie to the passed profile variable**  
   1. declare probeprofile variable and have its parameter be the profile variable  
      1. set the probepage cookie to the passed profile variable
   2. end function

about

1. **declare the request function with the parameter url  
   make a new xmlhttprequest**
   1. declare about function  
      1. initialise and set the variable httpRequest to a new XMLHttpRequest function
2. **try if the request is done and the request passed to   
   initialise and set a variable called query to the text outputted from the php  
   empty the posts div then fill it with the query variable**
   * 1. when the httpRequest is ready declare an anonymous function  
        1. try, if the request is done then  
           1. if the request passes, meaning code 200 then

initialise and set the variable query to the response text from the httpRequest

get the element with id posts and set its innerHTML to a blank string

get the element with id posts and set its innerHTML to query

* + - * 1. end if

1. **else if there’s an error and the passed http request does not have a status code of 200 then alert an error message**  
   * + - 1. else

create an alert containing the string There was a problem with the request.

* + - * 1. end else
      1. end try and if

1. **if the try fails then catch the error and send it to the console**  
   * + 1. catch the error variable  
          1. output to console the string Caught Exception: concatenated with error.description
       2. end catch
     1. end anonymous function
2. **open and send a http request via post to the url asynchronously**
   * 1. using the httpRequest variable open an asynchronous post request with the about.php file as the destination
   1. end function

when someone clicks off of the dropdown menu

1. **when the window is clicked pass an event to an anonymous function  
   if said event does not match the class dropbtn then close the dropdown menu**  
   1. when window is clicked, call an anonymous function and pass what was clicked on to the function  
      1. if the event target does not match the class .dropbtn then  
         1. initialise and set the variable dropdowns to the element of class dropdown-content
         2. initialise the variable i
         3. for i from 0 while i < length of dropdowns do  
            1. initialise and set the variable openDropdown to the current element of dropdowns
            2. if the class list of openDropdown variable contains the class show then

remove the class show from openDropdown’s class list

* + - * 1. end if
      1. end for
    1. end if
  1. end anonymous function