

CT4009 - Introduction to Web Development

Matej Kondrot, Assignment 001, s1804317

Web Page URL:

http://ct4009-matrot.studentsites.glos.ac.uk/CT4009_2018-19_001_1804317_07_Jan_2019/loginPage/login.html

Control Panel:

URL: ct4009-matrot.studentsites.glos.ac.uk Username: s1804317matrot Password: UniGlos2018

Functioning Pages:

- friendsPage/friends.html
- loginPage/login.html
- messagesPage/messages.html
- postPage/post.html
- registerPage/register.html
- reportsPage/reports.html
- statsPage/stats.html

These pages have been tested to work only on the Google Chrome browser. Firefox does not work due to restrictions with `localStorage`

Pages were constructed with Brackets, and uploaded via FileZilla.

Interesting CSS:

To create a layout, the website was designed with grid elements (Purdila, 2017). They were chosen as they are a relatively modern approach, tailored specifically to website layout. Each default `box` is encapsulated by the `wrapper`, which determines the width and height the boxes will fill. Through several iterations, it was determined that `grid-template-columns` and `grid-template-rows` are best left on `auto`, as this allows the most flexibility.

```
.wrapper {
  position: absolute;
  height: 95%;
  width: 100%;
  display: grid;
  grid-gap: 5px;
  grid-template-columns: auto;
  grid-template-rows: auto;
  background-color: #fff;
  color: #444;
}
```

To stop the boxes from shrinking when they lack content, their size was set with `min-height`, `max-height`, `min-width`, `max-width`. These sizes are mostly defined with viewport (Snook, 2011) units (`vh`, `vw`). This is because the site should accommodate any browser resolution, but using `%` only refers to its height within the wrapper, and causes a multitude of unexpected problems.

```
.header {  
  min-height: 20vh;  
  max-height: 20vh;  
  background-color: #3A775C;  
  grid-column: 1/4;  
  grid-row: 1;  
}
```

A final part of css was the decision to have two `.css` files for one `.html`. This is because global, often-referenced styles are in `styles.css`, while more niche styles specific to one page are in their individual files.

Interesting JS

Scrolling Messaging

When working on `messages.html`, messaging a user would cause messages to get appended to the bottom. This was fine, but a problem was that the user would have to scroll down with each message and, if they reloaded the page, they would have to scroll all the way to the bottom to find their most recent conversation. One solution for this was to append items to the top, but that seemed unnatural. An online solution showed how to have the chat window scroll to the bottom by default. (Black, 2008)

```
newContent2 = document.createTextNode(calcMsgTime(messageObject));  
newContent.appendChild(newContent2);  
newDiv.appendChild(newContent);  
var parentDiv = document.getElementById("messagesBox");  
parentDiv.appendChild(newDiv);  
var objDiv = document.getElementById("messagesBox");  
objDiv.scrollTop = objDiv.scrollHeight;
```

Dates

When implementing `messages.html`, each message has a date attached to it. To find out how to implement dates, an online solution was found. (Xul.fr, n.d.)

```
function calcMsgTime(object){
    var currentdate = new Date();
    var messageDate = new Date(object.date);
    if (currentdate - object.date > 604800000){
        return messageDate.toLocaleDateString();
    }
    else{
        var j = new Array("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday"); // i also found out the week starts on a sunday in js
        return (j[messageDate.getDay()]) + " " + messageDate.toLocaleTimeString('en-
GB', { hour: "numeric", minute: "numeric"});
    }
}
```

'Enter' to Send

To make up for the small 'Send' button, an easy way to send messages was to have the user hit the enter key. To do this, an online solution was found. (Sachleen, 2012) It checks each key press, and if it is the enter key, then it sends the same input as the 'Send' button.

```
document.getElementById("txtMessage").onkeypress = function(e) {
    var event = e || window.event;
    var charCode = event.which || event.keyCode;
    if ( charCode == '13' ) {
        if (document.getElementById("txtMessage").value != ""){
            if (selectedUser != undefined){
                sendMessage()
            }
        }
        return false;
    }
}
```

Changing the Tab

On `messages.html`, the user can tab between their friends and see their ongoing conversations. Each time they change the tab, the previous messages must be deleted and new ones loaded. To do this, an online solution was found. It gets all the elements of a given name and, while there are any of that element left, systematically removes them.

```
function removeElementsByClass(className){
    var elements = document.getElementsByClassName(className);
    while(elements.length > 0){
        elements[0].parentNode.removeChild(elements[0]);
    }
}
```

Website User Manual

- Use Google Chrome
- To create an account on SocMed, first navigate from the login page to the registration page.

- Here, you can set up a new account.
 - It must have a First Name and Last Name
 - Your username must be unique
 - You must have a valid e-mail address
 - Your password must be at least six characters and match twice
 - You can upload a profile photo and/or write a biography, though this is optional.
- After you create an account, navigate back to the login page and log in.
- You will be on the main page, and can make your first post to your wall.
- You must write text, and you may select an image to go along with it.
- You can navigate to your messages, on the header, to send someone a friend request.
 - Just type their unique username into the left textbox and hit add.
 - Alternatively, navigate to 'friends' to see if you've received any requests of your own.
 - You will be able to accept or reject them using the buttons on the sidebar.
- Once you have friends on your account, you can message them and you can see each other's wall posts. You can comment on their posts or report them.
- When you have finished using the site, log out.

Website Administrator Manual

- Use Google Chrome
- To create a sample userbase, go into the register webpage, hit `f12` to open up the console, then type `createAccounts()`.
 - These accounts don't have any posts associated with them however
 - To remove all sample data and start from a blank slate, enter `localStorage.clear()`
 - Alternatively, wait for any real users to start using your system.
- To log into your administrative account, enter `Administrator` as your login, and `okokok89` as the password.
- Once you are logged in, you are able to see reported post on your wall.
 - You can dismiss these posts or ban the offending user.
 - Banned users are removed from the website, and their names can't be used again.
- You can also view user stats by navigating to users in the header.
- Here, you will see all the users and their number of posts. To narrow your search, enter a user's username into the search bar.
- You have the option to ban any user from this menu.
- When you have finished using the site, log out.

References

Black (2008). *Scroll to bottom of div?*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/270612/scroll-to-bottom-of-div> [Accessed 7 Jan. 2019].

GEOCHET (2012). *Removing elements by class name?*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/4777077/removing-elements-by-class-name> [Accessed 7 Jan. 2019].

Purdila, A. (2017). *5 Quick Resources for Mastering CSS Grid Layout*. [online] Web Design Envato Tuts+. Available at: <https://webdesign.tutsplus.com/tutorials/5-quick-resources-for-mastering-css-grid-layout--cms-29919> [Accessed 7 Jan. 2019].

Sachleen (2012). *How to detect when the user presses Enter in an input field*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/11365632/how-to-detect-when-the-user-presses-enter-in-an-input-field> [Accessed 7 Jan. 2019].

Snook, J. (2011). *Sizing with CSS3's vw and vh units*. [online] Snook.ca. Available at: https://snook.ca/archives/html_and_css/vm-vh-units [Accessed 7 Jan. 2019].

Xul.fr. (n.d.). *The Date object in JavaScript*. [online] Available at: <https://www.xul.fr/javascript/date.php> [Accessed 7 Jan. 2019].

Page Print Screens

Login



SOC MED

Social Media for Medicine Societies

Log in:

Username:

Password:

Don't have an account? [Register](#)

Register

SOCMED

Social Media for Medicine Societies

Register Here

First Name:

Last Name:

User Name:

Email:

Password:

Confirm Password:

Profile Picture: No file chosen

Bio: (optional)

[Back to login](#)

Post

SOCMED

| [messages](#) | [post](#) | [friends](#) | [log out](#) |

No file chosen



Ville Sinki | ville
Sample Text lorem ipsum etc..

1/7/2019, 2:18:15 PM

Friends

SOCMED

[messages](#) | [post](#) | friends | [log out](#) |

Requests

Friends



Ville Sinki | ville
No bio
Remove friend

Messages

SOCMED

[messages](#) | [post](#) | [friends](#) | [log out](#) |

CONTACTS


MESSAGES | Ville Sinki

Enter Username

Add

ville | Ville Sinki

test message
Monday 14:19



Send

Reports

SOCMED - ADMIN

| [reports](#) | [users](#) | [log out](#) |

Reported Posts



Ville Sinki | ville
Sample Text lorem ipsum etc..

[Ban User](#) [Dismiss Report](#)
1/7/2019, 2:18:15 PM

Ville Sinki | ville
comment

Users

SOCMED - ADMIN

| [reports](#) | [users](#) | [log out](#) |



I really love grass!
Number of posts in last
[Day](#) | [Week](#) | [All Time](#) : 0
[Ban User](#)



Joshua Ramsay | cooliodj
No bio
Number of posts in last
[Day](#) | [Week](#) | [All Time](#) : 0
[Ban User](#)



Ville Sinki | ville
No bio
Number of posts in last
[Day](#) | [Week](#) | [All Time](#) : 1
[Ban User](#)

Page Source Code

HTML

friends.html

```
<!DOCTYPE html>
<html>
<head>
  <link rel = "stylesheet" href="../css/styles.css" type="text/css"/>
  <link rel="stylesheet" type="text/css" href="friends.css">
  <script type="text/javascript" src="friends.js"></script>
  <meta charset="UTF-8">
</head>
```



```

<body>
  <div class="wrapper">
    <div class="box header">
      <center>
        <br>
        <h1> SOCMED</h1>
      </center>
      <p>| <a href=../messagesPage/messages.html> messages </a> | <a
href=../postPage/post.html> post </a> | friends | <a href="" onmousedown="logout()">
log out </a> | </p>
    </div>
    <div id="friendsListBox" class="box friends">
      <h1>Friends</h1>
    </div>
    <div id="friendRequestsBox" class="box friendsrequests">
      <h1>Requests</h1>
      <div id="insertBeforeFriendRequests"></div><!-- I could've done appendChild
in my js code but I decided to leave it as it worked --->
    </div>
  </div>
</body>
</html>

```

login.html

```

<html>
  <head>
    <title> SocMed </title>
    <meta charset="utf-8">
    <link rel = "stylesheet" href=../css/styles.css" type="text/css"/>
    <link rel="stylesheet" type="text/css" href="login.css">
    <script type="text/javascript" src="login.js"></script>
  </head>
  <body>
    <div class="wrapper">
      <div class="box header">
        <center> <h1> SOCMED</h1>
        <h3> Social Media for Medicine Societies</h3>
      </center></div>
      <div class="box wall">
        <center>
          <h4> Log in: </h4>
          <form onsubmit="checkLogin()" id="formUserLogin"> <!--compares with
accounts already created-->
            <label> Username: </label>
            <input type="text" id = "txtUserName" name="userName"
required="required"><br><br>

            <label> Password: </label>
            <input type="password" id = "txtPassword" name="password"
required="required"><br><br>
            <input type="submit" id = "btnLogin" value="Login">
          </form>
        </center>
      </div>
    </div>
  </body>
</html>

```

```

        <h4> Don't have an account? <a
href="../registerPage/register.html">Register</a></h4>
    </center>
</div>
</div>
</body>
</html>

```

messages.html

```

<!DOCTYPE html>
<html>
<head>
    <link rel = "stylesheet" href="../css/styles.css" type="text/css"/>
    <link rel="stylesheet" type="text/css" href="messages.css">
    <script type="text/javascript" src="messages.js"></script>
    <meta charset="UTF-8">
    <title> SOCMED: Messaging</title>
</head>
<body>
    <div class="wrapper">
        <div class="box header">
            <center>
                <br>
                <h1> SOCMED</h1>
            </center>
            <p>| messages | <a href=../postPage/post.html> post </a> | <a
href=../friendsPage/friends.html> friends </a> | <a href="" onmousedown="logout()"> log
out </a> | </p>
        </div>
        <div class="box messagestitle">MESSAGES | <span id="selectedUserNameMessages">
</span></div>
        <div id="contactsBox" class="box contacts">CONTACTS <br>
            <input type="text" id = "txtUsername" name="username" placeholder="Enter
Username">
            <input type="submit" id="submitUsr" value="Add" onmousedown="addUser()">
            <div id="insertFriendBefore"></div> <!-- Because there is no
document.body.insertAfter, only insertBefore -->
        </div>
        <div id="messagesBox" class="box messages">
        </div>
        <div class="box textbox">
            <textarea type="text" id="txtMessage" name="message" length="200px"
required="required"></textarea>
        </div>
        <div class="box send">
            <center>
                <button id="msgSendButton" onmousedown="sendMessage()" class="button"
disabled="true">Send</button>
            </center>
        </div>
    </div>
</body>

```

```
</html>
```

posts.html

```
<!DOCTYPE html>
<html><head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <link rel = "stylesheet" href="../css/styles.css" type="text/css"/>
  <link rel="stylesheet" type="text/css" href="post.css">
  <script type="text/javascript" src="post.js"></script>
  <meta charset="UTF-8">
</head>
<body>
  <div class="wrapper">
    <div class="box header">
      <center>
        <br>
        <h1> SOC MED</h1>
      </center>
      <p>| <a href="../messagesPage/messages.html"> messages </a> | post | <a
href="../friendsPage/friends.html"> friends </a> | <a href="" onmousedown="logout()">
log out </a> | </p>
    </div>
    <div class="box textbox">
      <textarea type="text" id="txtMessage" name="message" length="200px"
required="required"></textarea>
    </div>
    <div class="box imagebox">
      <input type="file" id="imgPpf" name="pfp" accept="image/png, image/jpeg"
onchange="readURL(this);">
      <img src="" id="displayedUploadedImg" style="display: none;">
    </div>
    <div class="box postbox">
      <button onmousedown="postwall()" class="button">Post to wall</button>
    </div>
    <div id="postwallBox" class="box wall">
      <div id="insertPostBefore"></div>
    </div>
  </div>

</body></html>
```

register.html

```
<!DOCTYPE html>
<html>
<head>
  <link rel = "stylesheet" href="../css/styles.css" type="text/css"/>
  <link rel="stylesheet" type="text/css" href="register.css">
  <meta charset="utf-8">
  <title> SocMed: Register </title>
```

```

<script type="text/javascript" src="register.js"></script>
</head>
<body>
  <div class="wrapper">
    <div class="box header"><center> <h1> SOCMED</h1>
    <h3> Social Media for Medicine Societies</h3>
    </center></div>
    <div class = "box wall">
      <center>
        <form id="formUserRegistration" >
          <h4> Register Here</h4>
          <label> First Name: </label>
          <input type="text" id = "txtFirstName" name="firstName"
required="required"> <br> <br>
          <label> Last Name: </label>
          <input type="text" id = "txtLastName" name="lastName" required="required">
<br> <br>
          <label> User Name: </label>
          <input type="text" id = "txtUserName" name="userName" required="required">
<br> <br>
          <label> Email: </label>
          <input type="email" id = "txtEmail" name="email" required="required"> <br>
<br>
          <label> Password: </label>
          <input type="password" id = "txtPassword" name="password"
required="required"> <br> <br>
          <label> Confirm Password: </label>
          <input type="password" id = "txtConfirmPassword" name="confirmPassword"
required="required"> <br> <br>
          <label>Profile Picture:</label>
          <input type="file" id="imgPpf" name="pfp" onchange="readURL(this);"
accept="image/png, image/jpeg"><br><br>
          <img src="" id="displayedUploadedImg" style="display: none;">
          <label> Bio: (optional)</label>
          <input type="text" id = "txtBio" name="bio"> <br><br>
          <input type="submit" id = "btnRegister" value="Register"
onmousedown="checkFields()">
          <br><br>
          <a href=../loginPage/login.html> Back to login </a>
        </form>
      </center>
    </div>
  </div>
</body>
</html>

```

reports.html

```

<!DOCTYPE html>
<html>
<head>
  <link rel = "stylesheet" href="../css/styles.css" type="text/css"/>
  <link rel="stylesheet" type="text/css" href="reports.css">

```

```

<script type="text/javascript" src="reports.js"></script>
<meta charset="UTF-8">
</head>
<body>
  <div class="wrapper">
    <div class="box header">
      <center>
        <br>
        <h1> SOCMED - ADMIN</h1>
      </center>
      <p>| reports | <a href=../statsPage/stats.html> users </a> | <a href=""
onmousedown="logout()"> log out </a> | </p>
    </div>
    <div class="box heading">
      <h1>Reported Posts</h1>
    </div>
    <div id="postWallBox" class="box wall">
      <div id="insertPostBefore"></div>
    </div>
  </div>
</body>
</html>

```

stats.html

```

<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href=../css/styles.css" type="text/css"/>
  <link rel="stylesheet" type="text/css" href="stats.css">
  <script type="text/javascript" src="stats.js"></script>
  <meta charset="UTF-8">
  <title> SOCMED: Messaging</title>
</head>
<body>
  <div class="wrapper">
    <div class="box header">
      <center>
        <br>
        <h1> SOCMED - ADMIN</h1>
      </center>
      <p>| <a href=../reportsPage/reports.html> reports </a> | users | <a
href="" onmousedown="logout()"> log out </a> | </p>
    </div>
    <div id="adminUsersBox" class="box contacts">USERS <br>
      <input type="text" id="txtUsername" name="username" placeholder="Enter
Username">
      <button onmousedown="searchUser()" id="submitUsr">Search</button>
    </div>
  </div>
</body>
</html>

```

CSS

styles.css

```
html, body{
  min-height: 100%;
  /*    set to 100% as otherwise the box grid would sometimes collapse*/
  overflow-x: hidden;
  /*    a small, insignificant, portion of the site is off screen due to margins. to
  avoid horizontal scrolling, the excess is cut off*/
}

.wrapper {
  /*    the wrapper is the main grid which holds all the elements in place.*/
  position: absolute;
  height: 95%;
  width: 100%;
  display: grid;
  grid-gap: 5px;
  grid-template-columns: auto;
  grid-template-rows: auto;
  background-color: #fff;
  color: #444;
}

.box {
  /*    the base properties of all grid element*/
  background-color: #3A775C;
  color: #fff;
  border-radius: 5px;
  font-size: 150%;
}

h1{
  /*    lowers the extra space taken up by <h1>, reducing clutter*/
  margin:0px;
  padding:0px;
}

p{
  /*    reduces clutter*/
  margin:0px;
  padding:0px;
}

textarea {
  /*    resizes the text area to be (almost) the entire grid element in which it
  resides.*/
  resize: none;
  width:99%;
  height:95%;
}
```

```
::-webkit-scrollbar {  
  /* removes the scrollbar so it doesn't take up space*/  
  display: none;  
}
```

friends.css

```
.header {  
  min-height: 20vh;  
  max-height: 20vh;  
  background-color: #3A775C;  
  grid-column: 1/3;  
  grid-row: 1;  
}  
  
.friendsrequests {  
  /* 78vh is often used instead of 80 to make up for the margins*/  
  min-height: 78vh;  
  max-height: 78vh;  
  min-width: 20vw;  
  max-width: 20vw;  
  grid-column: 1;  
  grid-row: 2;  
  overflow-y: scroll;  
}  
  
.friends {  
  min-height: 78vh;  
  max-height: 78vh;  
  min-width: 80vw;  
  max-width: 80vw;  
  grid-column: 2;  
  grid-row: 2;  
  overflow-y: scroll;  
}  
  
.container {  
  /*container for the messages and notifications.*/  
  border: 2px solid #28513f;  
  background-color: #30624c;  
  border-radius: 5px;  
  padding: 10px;  
  margin: 10px 0;  
  margin-bottom: 0;  
}  
  
.commentbox{  
  /*alternative theme for sent messages, appears when the 2nd user sends a message.*/  
  border: 2px solid #30624c;  
  background-color: #28513f;  
  border-radius: 5px;  
  padding: 10px;  
}
```

```

.user {
  /*    user names have their own styling*/
  font-size: 120%;
  color: #aaa
}
/*Base container code taken from w3schools tutorial on chat messages*/
/* Clear floats */
.container::after {
  /* the '::after' function is executed at the end of every container element*/
  content: "";
  clear: both;
  display: table;
}

/* Style images */
.container img {
  float: left;
  max-width: 30%;
  max-height: 40vh;
  width: auto;
  margin-right: 20px;
  border-radius: 3%;
}
.profilePicture{
  height: 150px;
  width: 150px;
}
.removeFriend{
  cursor:pointer;
  color: #aaa
}

```

login.css

```

.header {
  min-height: 20vh;
  max-height: 20vh;
  background-color: #3A775C;
  grid-column: 1;
  grid-row: 1;
}

.wall{
  font-size: 100%;
  min-height: 75vh;
  max-height: 75vh;
  grid-column: 1;
  grid-row: 2;
  overflow-y: scroll;
}

```

messages.css


```

.header {
  min-height: 20vh;
  max-height: 20vh;
  background-color: #3A775C;
  grid-column: 1/4;
  grid-row: 1;
}

.contacts {
  grid-column: 1 ;
  grid-row: 2/5;
  overflow-y: scroll;
}

.messagestitle{
  background-color: #3A775C;
  grid-column: 2/4 ;
  grid-row: 2 ;
}

.messages {
  min-height: 65vh;
  max-height: 65vh;
  background-color: #3A775C;
  grid-column: 2/4 ;
  grid-row: 3 ;
  overflow-y: scroll;
}

.textbox {
  background-color: #fff;
  grid-column: 2/3;
  grid-row: 4;
}

.send {
  background-color: #3A775C;
  grid-column: 3;
  grid-row: 4
}

.button{
/*   the button unfortunately doesn't fill the entire box*/
  background-color: #32946F;
  border: none;
  color: white;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
}

.friend{
  border: 2px solid #28513f;

```

```
background-color: #30624c;
border-radius: 5px;
padding: 5px;
margin: 5px;
cursor: pointer;
}

/**/
/*base containers and styling from w3schools*/
/**/
/* Chat containers */
.container {
border: 2px solid #28513f;
background-color: #30624c;
border-radius: 5px;
padding: 10px;
margin: 10px 0;
}

/* Darker chat container */
.darker {
border-color: #30624c;
background-color: #28513f;
}

/* Clear floats */
.container::after {
content: "";
clear: both;
display: table;
}

/* Style images */
.container img {
float: left;
max-width: 60px;
width: 100%;
margin-right: 20px;
border-radius: 50%;
}

/* Style the right image */
.container img.right {
float: right;
margin-left: 20px;
margin-right: 0;
}

/* Style time text */
.time-right {
float: right;
color: #aaa;
}
```

```

/* style time text */
.time-left {
  float: left;
  color: #999;
}

/**/
/**/
.profilePicture{
  max-height:100px;
  max-width:100px;
}

```

posts.css

```

.header {
  min-height: 20vh;
  max-height: 20vh;
  background-color: #3A775C;
  grid-column: 1/5;
  grid-row: 1;
}
.textbox{
  min-width: 70vw;
  max-width: 70vw;
  min-height: 14vh;
  grid-column: 1/3;
  grid-row: 2;
}
.imagebox{
  min-width: 20vw;
  max-width: 20vw;
  grid-column: 3/4;
  grid-row: 2;
}
.postbox{
  min-width: 10vw;
  max-width: 10vw;
  grid-column: 4;
  grid-row: 2;
}
.wall{
  min-height: 60vh;
  max-height: 60vh;
  grid-column: 1/5;
  grid-row: 3;
  overflow-y: scroll;
}
.container {
  border: 2px solid #28513f;
  background-color: #30624c;
  border-radius: 5px;
}

```

```

padding: 10px;
margin: 10px 0;
margin-bottom: 0;
}

/*darker version of the container used for comments*/
/*in hindsight could possibly be 'container darker'*/
.commentbox{
border: 2px solid #30624c;
background-color: #28513f;
border-radius: 5px;
padding: 10px;

}

.time-right {
float: right;
color: #aaa;
}

.user {
font-size: 120%;
color: #aaa

}

/* Clear floats */
.container::after {
content: "";
clear: both;
display: table;
}

/*Formats the size of any posted images*/
.postImageBox img{
/* float: left;*/
max-width: 30%;
max-height: 40vh; /* vh used as the image itself can define the % height of the
post */

/* width: auto;*/
/* margin-right: 20px;*/
/* border-radius: 3%;*/
}
/*
.postImageBox{
max-width:900px;
max-height:900px;
}*/

```

```

.header {
    min-height: 20vh;
    max-height: 20vh;
    background-color: #3A775C;
    grid-column: 1;
    /* header cannot be added to styles.css as each page has a different number of grid-
columns*/
    grid-row: 1;
}

.wall{
    font-size: 100%;
    min-height: 75vh;
    max-height: 75vh;
    grid-column: 1;
    grid-row: 3;
    overflow-y: scroll;
}

```

reports.css

```

html, body{
    min-height: 100%;
    overflow-x: hidden;
}

.header {
    min-height: 20vh;
    max-height: 20vh;
    background-color: #3A775C;
    grid-column: 1;
    grid-row: 1;
}

.heading{
    min-height: 10vh;
    max-height: 10vh;
    background-color: #3A775C;
    grid-column: 1;
    grid-row: 2;
}

}
*/
/*wall contents emulate that of a normal user wall, but only show posts flagged as
reported*/
.wall{
    min-height: 60vh;
    max-height: 60vh;
    grid-column: 1;
    grid-row: 3;
}

```

```

        overflow-y: scroll;
    }
    .container {
        border: 2px solid #28513f;
        background-color: #30624c;
        border-radius: 5px;
        padding: 10px;
        margin: 10px 0;
        margin-bottom: 0;
    }
    .commentbox{
        border: 2px solid #30624c;
        background-color: #28513f;
        border-radius: 5px;
        padding: 10px;

    }
    .time-right {
        float: right;
        color: #aaa;
    }

    .user {
        font-size: 120%;
        color: #aaa

    }

    /* Clear floats */
    .container::after {
        content: "";
        clear: both;
        display: table;
    }

    /* Style images */
    .container img {
        float: left;
        max-width: 30%;
        max-height: 40vh;
        width: auto;
        margin-right: 20px;
        border-radius: 3%;
    }

```

stats.css

```

html, body{
    min-height: 100%;
    overflow-x: hidden;
}

```

```
.header {
  min-height: 20vh;
  max-height: 20vh;
  background-color: #3A775C;
  grid-column: 1;
  grid-row: 1;
}

.contacts {
  min-height: 78vh;
  max-height: 78vh;
  grid-column: 1 ;
  grid-row: 2;
  overflow-y: scroll;
}

.messagestitle{
  background-color: #3A775C;
  grid-column: 2/4 ;
  grid-row: 2 ;
}

.messages {
  min-height: 65vh;
  max-height: 65vh;
  background-color: #3A775C;
  grid-column: 2/4 ;
  grid-row: 3 ;
  overflow-y: scroll;
}

.textbox {
  background-color: #fff;
  grid-column: 2/3;
  grid-row: 4;
}

.send {
  background-color: #3A775C;
  grid-column: 3;
  grid-row: 4
}

.button{
  background-color: #32946F;
  border: none;
  color: white;
  /* padding: 100% 32px;*/
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
}
```

```
.friend{
  border: 2px solid #28513f;
  background-color: #30624c;
  border-radius: 5px;
  padding: 5px;
  margin: 5px;
  cursor: pointer;
}

/* Chat containers */
.container {
  border: 2px solid #28513f;
  background-color: #30624c;
  border-radius: 5px;
  padding: 10px;
  margin: 10px 0;
}

/* Darker chat container */
.darker {
  border-color: #30624c;
  background-color: #28513f;
}

/* Clear floats */
.container::after {
  content: "";
  clear: both;
  display: table;
}

/* Style images */
.container img {
  float: left;
  max-width: 30%;
  max-height: 40vh;
  width: auto;
  margin-right: 20px;
  border-radius: 3%;
}

/* Style the right image */
.container img.right {
  float: right;
  margin-left: 20px;
  margin-right: 0;
}

/* Style time text */
.time-right {
  float: right;
  color: #aaa;
}
```



```

/* style time text */
.time-left {
  float: left;
  color: #999;
}
.profilePicture{
  height: 150px;
  width: 150px;
}

```

JS

friends.js

```

var accountList = {};
var loggedInUsername;
if (JSON.parse(localStorage.getItem("saveData")) !== null){ // if an account has
already been registered, it loads the account list
  accountList = JSON.parse(localStorage.getItem("saveData"));
}
if (localStorage.getItem("loggedInData") !== null){ // checks if an account was
previously logged in
  loggedInUsername = localStorage.getItem("loggedInData");
}
window.onload = function () { // ensures that the html loads before the javascript
  addFriendRequestElements();
  addFriendsList();
};
function logout(){
  localStorage.removeItem("selectedUserData"); //removes the log in data
  localStorage.removeItem("loggedInData"); //removes the log in data
  window.location.replace("../loginPage/login.html");
}
function respondRequest(userName, match){ // whether the friend request is accepted or
rejected
  if (match == true){
    delete accountList[loggedInUsername].friendRequests[userName]; // removes the
friend request
    accountList[loggedInUsername].friends[userName] = {messages: {}};
    accountList[userName].friends[loggedInUsername] = {messages: {}};
  }
  else{
    delete accountList[loggedInUsername].friendRequests[userName];
  }
  localStorage.setItem("saveData", JSON.stringify(accountList));
  location.reload();
}
function addFriendRequestElements(){ //adds friend request divs
  var parentDiv = document.getElementById("friendRequestsBox");
  for (addedUserName in accountList[loggedInUsername].friendRequests){
    var newDiv = document.createElement("div");

```

```

        var newContent = document.createTextNode([accountList[addedUserName].firstName
+ " " + accountList[addedUserName].lastName]);
        newDiv.appendChild(newContent);
        newDiv.appendChild(document.createElement("br"));
        newContent = document.createTextNode([addedUserName]);
        newDiv.appendChild(newContent);
        newDiv.appendChild(document.createElement("br"));
        newContent = document.createElement("button");
        newContent.setAttribute("onmousedown", `respondRequest('${addedUserName}',
true)`); // gives the button onmousedown="respondRequest(addedUserName, true)"
        var newContent2 = document.createTextNode("Accept");
        newContent.appendChild(newContent2);
        newDiv.appendChild(newContent);
        newContent = document.createElement("button");
        newContent.setAttribute("onmousedown", `respondRequest('${addedUserName}',
false)`);
        newContent2 = document.createTextNode("Reject");
        newContent.appendChild(newContent2);
        newDiv.appendChild(newContent);
        newDiv.setAttribute("class", "container");
        var friendRequestQuantity =
object.keys(accountList[loggedInUsername].friendRequests).length;
        newDiv.setAttribute("id", [friendRequestQuantity]);
        if (friendRequestQuantity > 1){ // checks if you only have 1 friend
            var currentDiv = document.getElementById([friendRequestQuantity-1]);
            parentDiv.insertBefore(newDiv, currentDiv);
        }
        else{ // if you only have 1 friend, then it inserts the div below the submit
user button in the body
            var currentDiv = document.getElementById("insertFriendBefore");
            parentDiv.insertBefore(newDiv, currentDiv);
        }
    }
}

function addFriendsList(){ //adds friend divs
    var parentDiv = document.getElementById("friendsListBox");
    for (userName in accountList[loggedInUsername].friends){
        var userFriend = accountList[userName];
        var newDiv = document.createElement("div");
        newDiv.setAttribute("class", "container");
        var newContent = document.createElement("img");
        newContent.setAttribute("class", "profilePicture");
        newContent.setAttribute("alt", "Avatar");
        if (userFriend.profilePicture == "pic1.png"){
            newContent.setAttribute("src", "../images/pic1.png"); //default image
        }
        else{
            dataImage =
localStorage.getItem(userFriend.userName+userFriend.profilePicture);
            newContent.setAttribute("src", "data:image/png;base64,"+dataImage);
        }
        newDiv.appendChild(newContent);
    }
}

```

```

        newContent = document.createElement("div");
        newContent.setAttribute("class", "user");
        var newContent2 = document.createTextNode(userFriend.firstName + " " +
userFriend.lastName + " | " + userFriend.userName);
        newContent.appendChild(newContent2);
        newDiv.appendChild(newContent);
        newContent = document.createElement("p");
        newContent2 = document.createTextNode(userFriend.bio);
        newContent.appendChild(newContent2);
        newDiv.appendChild(newContent);
        newContent = document.createElement("span");
        newContent.setAttribute("onmousedown",
`removeFriend('${userFriend.userName}')`);
        newContent.setAttribute("class", "removeFriend");
        newContent2 = document.createTextNode("Remove friend");
        newContent.appendChild(newContent2);
        newDiv.appendChild(newContent);
        parentDiv.appendChild(newDiv);
    }
}
function removeFriend(username){
    var checkRemove = confirm("Are you sure you want to remove "+username+" from your
friends list?"); // prompt the user
    if(checkRemove){
        delete accountList[loggedInUsername].friends[username];
        delete accountList[username].friends[loggedInUsername];
        localStorage.setItem("saveData", JSON.stringify(accountList));
        location.reload();
    }
}

```

login.js

```

var accountList = {};
var bannedUsers = [];
var admin = {
    username: "administrator",
    password: "okokok89"
};
var loggedInUsername;
if (JSON.parse(localStorage.getItem("saveData")) !== null){ // if an account has
already been registered, it loads the account list
    accountList = JSON.parse(localStorage.getItem("saveData"));
}
if (localStorage.getItem("loggedInData") !== null){ // checks if an account was
previously logged in
    loggedInUsername = localStorage.getItem("loggedInData");
}
if (localStorage.getItem("bannedUsersData") !== null){ // checks the list of banned
usernames
    bannedUsers = localStorage.getItem("bannedUsersData");
}
if (loggedInUsername !== undefined){

```

```

    if (loggedInUsername != admin.username){
        redirectLoggedInUser();
    }
    else{
        redirectAdmin();
    }
}
function checkLogin(){ // checks log in fields
    var inputUserName = document.getElementById("txtUserName").value;
    var inputPassword = document.getElementById("txtPassword").value;
    var userNameMatched = false;
    for (x in accountList){ // checks if the inputted username matches any usernames in
the object
        if(inputUserName.toLowerCase() == accountList[x].userName.toLowerCase()){ // no
case sensitive discrepancies will occur as it will compare two strings in full upper
case
            userNameMatched = true;
            var userName = inputUserName.toLowerCase();
        }
    }
    if (userNameMatched == true){
        if (inputPassword != ""){ // checks if it has a value, otherwise the line below
will give an error
            if (inputPassword == accountList[userName].password){
                login(userName);
            }
            else{
                alert("Invalid password");
            }
        }
        else{
            alert("Invalid password");
        }
    }
    else{
        if (inputUserName == admin.username && inputPassword == admin.password){
            login(inputUserName);
        }
        else{
            if (bannedUsers.indexOf(inputUserName.toLocaleLowerCase()) > -1){
                alert("User banned");
            }
            else{
                alert("Invalid username");
            }
        }
    }
}

function login(userName){ // logs in if the account details were entered correctly
localStorage.setItem("LoggedInData", userName);
if (userName == admin.username){
    redirectAdmin();
}

```

```

    }
    else{
        redirectLoggedInUser();
    }
}

function redirectLoggedInUser(){
    window.location.replace("../postPage/post.html");
}

function redirectAdmin(){
    window.location.replace("../reportsPage/reports.html"); // redirects the user to
the admin page
}

```

messages.js

```

var accountList = {};
var loggedInUsername;
var selectedUser;
if (JSON.parse(localStorage.getItem("saveData")) !== null){ // if an account has
already been registered, it loads the account list
    accountList = JSON.parse(localStorage.getItem("saveData"));
}
if (localStorage.getItem("loggedInData") !== null){ // checks if an account was
previously logged in
    loggedInUsername = localStorage.getItem("loggedInData");
}
if (localStorage.getItem("selectedUserData") !== null){ // checks last selected user
    selectedUser = localStorage.getItem("selectedUserData");
}
function logout(){
    localStorage.removeItem("loggedInData"); //removes the log in data
    localStorage.removeItem("selectedUserData"); //removes the log in data
    window.location.replace("../loginPage/login.html");
}
window.onload = function () { // ensures that the html loads before the javascript
    addFriendElementsLoaded(); // adds added friends to the contacts list
    addMessageElementsLoaded(); // adds sent messages for selected user
    if (selectedUser !== undefined){ // selects the last selected friend if it's not
undefined i.e. when the page is refreshed
        if (isFriends(selectedUser)){
            selectFriend(selectedUser); //the user can only be select if he is friends
with the logged in user
        }
    }
    document.getElementById("txtMessage").onkeypress = function(e) { // detects if
enter is pressed in the textarea - from stackoverflow
        var event = e || window.event;
        var charCode = event.which || event.keyCode;
        if ( charCode == '13' ) {
            if (document.getElementById("txtMessage").value != ""){

```

```

        if (selectedUser !== undefined){
            sendMessage()
        }
    }
    return false;
}
};

function isFriends(user){ // returns true if the logged in user is friends with the
inputted username
    for (users in accountList[loggedInUsername].friends){
        if (accountList[user].userName == accountList[users].userName){
            return true;
        }
    }
    return false;
}

function addUser(){ // called when the "add" user button is pressed
    var addedUserName = document.getElementById("txtUsername").value;
    if(addedUserName !== loggedInUsername){ // checks that you're not trying to add
yourself
        var userNameMatched = false;
        for (x in accountList){ //checks that the user exists in the database
            if(addedUserName.toLowerCase() == accountList[x].userName.toLowerCase()){
                userNameMatched = true;
                var userName = addedUserName.toLowerCase();
            }
        }
        if (userNameMatched == true){
            if (accountList[loggedInUsername].friends[userName] === undefined){ //
checks that you haven't already added them
                accountList[userName].friendRequests[loggedInUsername] =
loggedInUsername;
                localStorage.setItem("saveData", JSON.stringify(accountList));
                document.getElementById("txtUsername").value = "";
                alert("Friend request sent!")
            }
            else{
                alert("You've already added them");
                document.getElementById("txtUsername").value = "";
            }
        }
        else{
            alert("Username not found");
            document.getElementById("txtUsername").value = "";
        }
    }
    else{
        alert("You can't add yourself");
        document.getElementById("txtUsername").value = "";
    }
}

```

```

}

function addFriendElementsLoaded(){ // creates a div with the class "friend" and an id
equivalent to the user's number of friends when they were added (1st friend = 1)
    var parentDiv = document.getElementById("contactsBox");
    for (addedUserName in accountList[loggedInUsername].friends){
        var newDiv = document.createElement("div");
        var newContent = document.createTextNode([addedUserName+" |
"+accountList[addedUserName].firstName + " " + accountList[addedUserName].lastName]);
        newDiv.appendChild(newContent);
        newDiv.setAttribute("class", "friend");
        newDiv.setAttribute("onmousedown", `selectFriend('${addedUserName}')`); //
gives the div onmousedown="selectFriend(addedUserName)"
        var friendQuantity = Object.keys(accountList[loggedInUsername].friends).length;
        newDiv.setAttribute("id", [friendQuantity]);
        if (friendQuantity > 1){ // checks if you only have 1 friend
            var currentDiv = document.getElementById([friendQuantity-1]);
            parentDiv.insertBefore(newDiv, currentDiv);
        }
        else{ // if you only have 1 friend, then it inserts the div below the submit
user button in the body
            var currentDiv = document.getElementById("insertFriendBefore");
            parentDiv.insertBefore(newDiv, currentDiv);
        } // I realised that I could've just done appendChild after writing this
    }
}

function sendMessage(){
    var messageData = document.getElementById("txtMessage").value;
    var currentdate = new Date();
    if (messageData != ""){ // ensures the message data isn't just blank
        if (selectedUser != undefined){ // ensures that a contact is selected
            messageNumber =
Object.keys(accountList[loggedInUsername].friends[selectedUser].messages).length+1; //
the ordered value of the message (e.g. 3rd message in the chat = 3)
            accountList[loggedInUsername].friends[selectedUser].messages[messageNumber]
= {
                sent: "out", // in/out - sending/receiving
                date: currentdate, // records the computer clock's date and time
                data: messageData,
                user: selectedUser // the other user in the conversation
            }
            accountList[selectedUser].friends[loggedInUsername].messages[messageNumber]
= { // records it in the receiver's account data
                sent: "in",
                date: currentdate,
                data: messageData,
                user: loggedInUsername
            }
        }
        document.getElementById("txtMessage").value = "";
        messageData = ""
        localStorage.setItem("saveData", JSON.stringify(accountList));
    }
}

```

```

        addMessageElement(messageNumber);
    }
}

function addMessageElement(messageNumber){
    var messageObject =
accountList[loggedInUsername].friends[selectedUser].messages[messageNumber];
    var newDiv = document.createElement("div");
    newDiv.setAttribute("class", "container msg");
    var loggedInUserMessage = false;
    if (messageObject.sent == "out"){ //sets sent messages to have the "darker" class
affected by css styling
        var loggedInUserMessage = true; // the message is being sent by the logged in
user
    }
    var newContent = document.createElement("img");
    newContent.setAttribute("alt", "Avatar");
    newContent.setAttribute("class", "profilePicture");
    if (loggedInUserMessage){
        newDiv.setAttribute("class", "container darker msg");
        if (accountList[loggedInUsername].profilePicture == "pic1.png"){
            newContent.setAttribute("src", "../images/pic1.png");
        }
        else{
            dataImage =
localStorage.getItem([loggedInUsername+accountList[loggedInUsername].profilePicture]);
            newContent.setAttribute("src", "data:image/png;base64,"+dataImage);
        }
        newContent.setAttribute("class", "right");
    }
    else{
        if (accountList[messageObject.user].profilePicture == "pic1.png"){
            newContent.setAttribute("src", "../images/pic1.png");
        }
        else{
            dataImage =
localStorage.getItem([messageObject.user+accountList[messageObject.user].profilePicture
]);
            newContent.setAttribute("src", "data:image/png;base64,"+dataImage);
        }
    }
    newDiv.appendChild(newContent);
    newContent = document.createElement("p");
    var newContent2 = document.createTextNode(messageObject.data);
    newContent.appendChild(newContent2);
    newDiv.appendChild(newContent);
    newContent = document.createElement("span");
    if (loggedInUserMessage){
        newContent.setAttribute("class", "time-left");
    }
    else{
        newContent.setAttribute("class", "time-right");
    }
}

```



```

newContent2 = document.createTextNode(calcMsgTime(messageObject));
newContent.appendChild(newContent2);
newDiv.appendChild(newContent);
var parentDiv = document.getElementById("messagesBox");
parentDiv.appendChild(newDiv);
var objDiv = document.getElementById("messagesBox");
objDiv.scrollTop = objDiv.scrollHeight;
}

function calcMsgTime(object){ //calculates what time to display on messages (>7 days
date)
    var currentdate = new Date();
    var messageDate = new Date(object.date);
    if (currentdate - object.date > 604800000){ // week in milliseconds
        return messageDate.toLocaleDateString(); // returns a date
    }
    else{
        var j = new Array("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday");
        return (j[messageDate.getDay()]) + " " + messageDate.toLocaleTimeString('en-
GB', { hour: "numeric", minute: "numeric"}); // returns a day & time
    }
}

function addMessageElementsLoaded(){
    if (selectedUser !== undefined){ // code only runs if there is a selected user
        if (accountList[loggedInUsername].friends[selectedUser] !== undefined){
            for (messageNumber in
accountList[loggedInUsername].friends[selectedUser].messages){
                var messageObject =
accountList[loggedInUsername].friends[selectedUser].messages[messageNumber];
                var newDiv = document.createElement("div");
                newDiv.setAttribute("class", "container msg");
                var loggedInUserMessage = false;
                if (messageObject.sent == "out"){ //sets sent messages to have the
"darker" class affected by css styling
                    var loggedInUserMessage = true; // the message is being sent by the
logged in user
                }
                var newContent = document.createElement("img");
                newContent.setAttribute("alt", "Avatar");
                newContent.setAttribute("class", "profilePicture");
                if (loggedInUserMessage){
                    newDiv.setAttribute("class", "container darker msg");
                    if (accountList[loggedInUsername].profilePicture == "pic1.png"){
                        newContent.setAttribute("src", "../images/pic1.png"); //default
image
                    }
                    else{
                        dataImage =
localStorage.getItem([loggedInUsername+accountList[loggedInUsername].profilePicture]);
                        newContent.setAttribute("src",
"data:image/png;base64,"+dataImage);

```

```

        }
        newContent.setAttribute("class", "right");
    }
    else{
        if (accountList[messageObject.user].profilePicture == "pic1.png"){
            newContent.setAttribute("src", "../images/pic1.png");
        }
        else{
            dataImage =
localStorage.getItem([messageObject.user+accountList[messageObject.user].profilePicture
]);
            newContent.setAttribute("src",
"data:image/png;base64,"+dataImage);
        }
    }
    newDiv.appendChild(newContent);
    newContent = document.createElement("p");
    var newContent2 = document.createTextNode(messageObject.data);
    newContent.appendChild(newContent2);
    newDiv.appendChild(newContent);
    newContent = document.createElement("span");
    if (loggedInUserMessage){
        newContent.setAttribute("class", "time-left");
    }
    else{
        newContent.setAttribute("class", "time-right");
    }
    newContent2 = document.createTextNode(calcMsgTime(messageObject));
    newContent.appendChild(newContent2);
    newDiv.appendChild(newContent);
    var parentDiv = document.getElementById("messagesBox");
    parentDiv.appendChild(newDiv);
}
}
var objDiv = document.getElementById("messagesBox");
objDiv.scrollTop = objDiv.scrollHeight;
}

function selectFriend(userName){ //username is the userName of the selected user
    document.getElementById("msgSendButton").disabled = false;
    if(selectedUser != userName){ // function only runs if the selected user is not
currently selected
        selectedUser = userName;
        localStorage.setItem("selectedUserData", selectedUser);
        removeElementsByClass("msg")
        addMessageElementsLoaded();
    }
    document.getElementById("selectedUserNameMessages").innerHTML =
[accountList[selectedUser].firstName] + " " + [accountList[selectedUser].lastName];
}

```

```
function removeElementsByClass(className){ // removes all elements with a certain class
name - from stackoverflow
    var elements = document.getElementsByClassName(className);
    while(elements.length > 0){
        elements[0].parentNode.removeChild(elements[0]);
    }
}

function createAccounts(){ //for testing purposes
    accountList = {
        lemmy57: {
            firstName: "Matej",
            lastName: "Kondrot",
            userName: "lemmy57",
            email: "kondrot@yahoo.com",
            password: "ilovelemon1",
            profilePicture: "pic1.png",
            bio: "I really love lemon!",
            friends: {},
            friendRequests: {},
            posts: {}
        },
        dabaws1221: {
            firstName: "Martin",
            lastName: "Varbanov",
            userName: "dabaws1221",
            email: "varbanov@yahoo.co.uk",
            password: "meGrass22",
            profilePicture: "pic1.png",
            bio: "I really love grass!",
            friends: {},
            friendRequests: {},
            posts: {}
        },
        cooliodj: {
            firstName: "Joshua",
            lastName: "Ramsay",
            userName: "CoolioDJ",
            email: "ramsay@gmail.com",
            password: "heyokay5",
            profilePicture: "pic1.png",
            bio: "No bio",
            friends: {},
            friendRequests: {},
            posts: {}
        },
        ville: {
            firstName: "Ville",
            lastName: "Sinki",
            userName: "Ville",
            email: "ville@gmail.co.uk",
            password: "123456",
            profilePicture: "pic1.png",
```

```

        bio: "No bio",
        friends: {},
        friendRequests: {},
        posts: {}
    }
}
localStorage.setItem("saveData", JSON.stringify(accountList));
}

```

posts.js

```

var accountList = {};
var reportedPosts = [];
var loggedInUsername;
if (JSON.parse(localStorage.getItem("saveData")) !== null){ // if an account has
already been registered, it loads the account list
    accountList = JSON.parse(localStorage.getItem("saveData"));
}
if (localStorage.getItem("loggedInData") !== null){ // checks if an account was
previously logged in
    loggedInUsername = localStorage.getItem("loggedInData");
}
if (localStorage.getItem("reportedPostsData") !== null){ // loads reported posts
    reportedPosts = JSON.parse(localStorage.getItem("reportedPostsData"));
}
window.onload = function () { // ensures that the html loads before the javascript
    createPostWallElements();
    createPostCommentElements();
};

function logout(){
    localStorage.removeItem("selectedUserData"); //removes the log in data
    localStorage.removeItem("loggedInData"); //removes the log in data
    window.location.replace("../loginPage/login.html");
}

function postWall(){
    var postData = document.getElementById("txtMessage").value;
    if (postData !== ""){
        var postImage =
document.getElementById("imgPpf").value.split(/(\\|\/)/g).pop(); //file name and format
        var currentdate = new Date();
        var postQuantity = Object.keys(accountList[loggedInUsername].posts).length;
        accountList[loggedInUsername].posts[postQuantity+1] = {
            data: postData,
            image: postImage,
            date: currentdate,
            username: loggedInUsername,
            comments: {}
        }
        if(accountList[loggedInUsername].posts[postQuantity+1].postImage !== ""){ // if
there is an image
            var imgData = getBase64Image(displayedUploadedImg);

```

```

localStorage.setItem([loggedInUsername+accountList[loggedInUsername].posts[postQuantity
+1].image], imgData); //F details.profilePicture is a string equal to the username and
the name/format of the saved file, e.g. lemmy57dog.png
    }
    localStorage.setItem("saveData", JSON.stringify(accountList));
    createPostWallElements();
    document.getElementById("txtMessage").value = "";
    postData = document.getElementById("txtMessage").value;
    location.reload();
}
}

function createPostWallElements(){
    removeElementsByClass("post");
    var parentDiv = document.getElementById("postWallBox");
    var visiblePostUsers = Object.keys(accountList[loggedInUsername].friends);
    visiblePostUsers.push(loggedInUsername);
    var unorderedPostsArray = [];
    for (userName of visiblePostUsers){
        for (i=1; i<Object.keys(accountList[userName].posts).length+1;i++){
            unorderedPostsArray.push(accountList[userName].posts[i]);
        }
    }
    var originalPostArrayLength = unorderedPostsArray.length;
    for (x=0;x<originalPostArrayLength;x++){
        var newestPost = unorderedPostsArray[0];
        var newestPostZ = 0;
        var newestPostDate = new Date(newestPost.date) // converts from string to date
        for (z=0;z<unorderedPostsArray.length;z++){
            if (z < unorderedPostsArray.length-1){
                var nextPost = unorderedPostsArray[z+1];
                var nextPostDate = new Date(nextPost.date) // will cause an error if
nextPost is undefined
            }
            if(z < unorderedPostsArray.length-1){
                if (newestPostDate - nextPostDate <= 0){
                    newestPost = nextPost;
                    newestPostZ = z+1;
                    newestPostDate = new Date(newestPost.date)
                }
            }
            else{
                unorderedPostsArray.splice(newestPostZ, 1); //removes the newest post
from the array
            }
        }
        var newDiv = document.createElement("div");
        newDiv.setAttribute("class", "post");
        newDiv.setAttribute("id", [newestPost.username+newestPost.date]);
        var newContent = document.createElement("div");
        newContent.setAttribute("class", "container");
        if (newestPost.image != ""){ //if an image has been uploaded

```

```

        var newContent2 = document.createElement("div");
        newContent2.setAttribute("class", "postImageBox");
        var newContent3 = document.createElement("img");
        newContent3.setAttribute("alt", "image")
        dataImage = localStorage.getItem([newestPost.username+newestPost.image]);
        newContent3.setAttribute("src", "data:image/png;base64,"+dataImage);
        newContent2.appendChild(newContent3);
        newContent.appendChild(newContent2);
    }
    var newContent2 = document.createElement("div");
    newContent2.setAttribute("class", "user");
    var newContent3 =
document.createTextNode(accountList[newestPost.username].firstName + " " +
accountList[newestPost.username].lastName + " | " + newestPost.username);
    newContent2.appendChild(newContent3);
    newContent.appendChild(newContent2);
    newContent2 = document.createElement("p");
    newContent3 = document.createTextNode(newestPost.data);
    newContent2.appendChild(newContent3);
    newContent.appendChild(newContent2);
    newContent2 = document.createElement("div");
    newContent2.setAttribute("class", "time-right");
    newContent3 = document.createTextNode([new
Date(newestPostDate)].toLocaleString()); // from JSON string to date object to string
    newContent2.appendChild(newContent3);
    newContent3 = document.createElement("br");
    newContent2.appendChild(newContent3);
    newContent3 = document.createElement("button");
    var newContent4 = document.createTextNode("Report Post");
    newContent3.appendChild(newContent4);
    newContent3.setAttribute("onmousedown", `reportPost('${newestPost.username}',
'${newestPost.date}')`);
    newContent2.appendChild(newContent3);
    newContent.appendChild(newContent2);
    newDiv.appendChild(newContent);
    newContent = document.createElement("div");
    newContent.setAttribute("class", "commentbox");
    newContent2 = document.createElement("textarea");
    newContent2.setAttribute("type", "text");
    newContent2.setAttribute("id",
["txtComment"+newestPost.username+newestPost.date]); // the id of the textarea will be
a very long string consisting of "textComment", followed by the username, followed by
the computer clock date/time (as a string)
    newContent2.setAttribute("name", "comment");
    newContent2.setAttribute("required", "required");
    newContent.appendChild(newContent2);
    newContent2 = document.createElement("button");
    newContent2.setAttribute("class", "button");
    newContent2.setAttribute("onmousedown", `postComment('${newestPost.username}',
'${newestPost.date}')`);
    newContent3 = document.createTextNode("Post Comment");
    newContent2.appendChild(newContent3);
    newContent.appendChild(newContent2);

```

```

        newDiv.appendChild(newContent);
        parentDiv.insertBefore(newDiv, insertPostBefore);
    }
}

function postComment(username, date){ //initializes the variable data for the comment
    for (x in accountList[username].posts){
        if (accountList[username].posts[x].date == date){ // finds the post that you're
commenting on
            var postCommentData =
document.getElementById(["txtComment"+username+date]).value;
            if (postCommentData != ""){
                var postCommentQuantity =
Object.keys(accountList[username].posts[x].comments).length;
                accountList[username].posts[x].comments[postCommentQuantity+1] = {
                    username: loggedInUsername,
                    data: postCommentData
                }
                document.getElementById(["txtComment"+username+date]).value = "";
                postCommentData = "";
                createPostCommentElements();
            }
        }
    }
    localStorage.setItem("saveData", JSON.stringify(accountList));
}

function createPostCommentElements(){ // creates the elements for the comments on the
page
    var visiblePostUsers = Object.keys(accountList[loggedInUsername].friends);
    visiblePostUsers.push(loggedInUsername);
    var unorderedPostsArray = [];
    for (userName of visiblePostUsers){
        for (i=1; i<Object.keys(accountList[userName].posts).length+1;i++){
            unorderedPostsArray.push(accountList[userName].posts[i]);
        }
    }
    removeElementsByClass("comment");
    for (z=0;z<unorderedPostsArray.length;z++){ // array of all relevant posts
        var parentDiv =
document.getElementById(unorderedPostsArray[z].username+unorderedPostsArray[z].date);
        var postCommentQuantity = Object.keys(unorderedPostsArray[z].comments).length;
        for (i=postCommentQuantity; i>0; i--){
            var newDiv = document.createElement("div");
            newDiv.setAttribute("class", "commentbox comment");
            var newContent = document.createElement("div");
            newContent.setAttribute("class", "user");
            var newContent2 =
document.createTextNode(accountList[unorderedPostsArray[z].comments[i].username].firstN
ame + " " + accountList[unorderedPostsArray[z].comments[i].username].lastName + " | " +
unorderedPostsArray[z].comments[i].username);
            newContent.appendChild(newContent2);
            newDiv.appendChild(newContent);

```

```

        newContent = document.createElement("p");
        newContent2 =
document.createTextNode(unorderedPostsArray[z].comments[i].data);
        newContent.appendChild(newContent2);
        newDiv.appendChild(newContent);
        parentDiv.appendChild(newDiv);
    }
}
}

function reportPost(userName, date){
    var alreadyReported = false;
    for (z of reportedPosts){
        if (z.username == userName && z.date == date){
            alreadyReported = true;
        }
    }
    if (alreadyReported == false){
        for (userPosts in accountList[userName].posts){
            if(accountList[userName].posts[userPosts].date == date){
                reportedPosts.push(accountList[userName].posts[userPosts]);
            }
        }
        alert("Post has been reported");
    }
    else{
        alert("Post has already been reported");
    }
    localStorage.setItem("reportedPostsData", JSON.stringify(reportedPosts));
}

function readURL(input){ // displays uploaded image - from stackoverflow
    if (input.files && input.files[0]) {
        var reader = new FileReader();
        reader.onload = function (e) {
            document.getElementById('displayedUploadedImg').src = e.target.result;
        }
        reader.readAsDataURL(input.files[0]);
    }
}

function getBase64Image(img){ //converts an image to base64 string - from stackoverflow
    var canvas = document.createElement("canvas");
    canvas.width = img.width;
    canvas.height = img.height;
    var ctx = canvas.getContext("2d");
    ctx.drawImage(img, 0, 0);
    var dataURL = canvas.toDataURL("image/png");
    return dataURL.replace(/^data:image\/(png|jpg);base64/, "");
}

function removeElementsByClass(className){ // removes all elements with a certain class
name - from stackoverflow
    var elements = document.getElementsByClassName(className);

```



```

    while(elements.length > 0){
        elements[0].parentNode.removeChild(elements[0]);
    }
}

```

register.js

```

var accountList = {};
var bannedUsers = [];
if (localStorage.getItem("bannedUsersData") !== null){ // checks the list of banned
usernames
    bannedUsers = localStorage.getItem("bannedUsersData");
}
if (JSON.parse(localStorage.getItem("saveData")) !== null){ // if an account has
already been registered, it loads the account list
    accountList = JSON.parse(localStorage.getItem("saveData"));
}

function checkFields(){ //checks that mandatory fields have been filled in the registry
form
    var inputFirstName = document.getElementById("txtFirstName").value;
    var inputLastName = document.getElementById("txtLastName").value;
    var inputUserName = document.getElementById("txtUserName").value;
    var inputEmail = document.getElementById("txtEmail").value;
    var inputPassword = document.getElementById("txtPassword").value;
    var inputConfirmPassword = document.getElementById("txtConfirmPassword").value;
    var inputProfilePicture = document.getElementById("imgPpf").value;
    var inputBio = document.getElementById("txtBio").value;
    if (inputFirstName !== "" && inputLastName !== "" && inputUserName !== ""){ //
ensures that name fields aren't blank
        var userNameTaken = false;
        var emailTaken = false;
        for (x in accountList){ // checks if the username or email has been taken by
iterating through the object
            if(inputUserName.toLowerCase() == accountList[x].userName.toLowerCase()){
// no case sensitive discrepancies will occur as it will compare two strings in full
upper case
                userNameTaken = true;
            }
            if(inputEmail.toLowerCase() == accountList[x].email.toLowerCase()){
                emailTaken = true;
            }
        }
        if(userNameTaken == false){
            if (inputUserName != "administrator"){
                if (bannedUsers.indexOf(inputUserName.toLowerCase()) < 0){
                    if (emailTaken == false){
                        if(inputUserName.indexOf(' ') == -1){ // ensures the username
doesn't contain a space
                            if (inputEmail.match(/^w+([\.-]?\w+)*@w+([\.-]?\w+)*
(\.\w{2,3})+$/)){ // ensures that the email has valid formatting - from w3resource email
validation

```

```

        if(inputPassword === inputConfirmPassword){ // checks
if the passwords match
            if(inputPassword.length >= 6){ // checks if the
password has atleast 6 characters
                if(/\d/.test(inputPassword) === true){ //
checks if the password contains a number
                    var registeredAccount = {
                        firstName: inputFirstName,
                        lastName: inputLastName,
                        userName:
inputUserName.toLocaleLowerCase(),
                        email: inputEmail,
                        password: inputPassword,
                        profilePicture:
inputProfilePicture.split(/(\\|\/)/g).pop(), // takes only the file name, no file path
                        bio: inputBio,
                        friends: {},
                        friendRequests: {},
                        posts: {}
                    };
                    if(inputProfilePicture === ""){ // if no
picture is uploaded, sets it to a stock image
                        registeredAccount.profilePicture =
"pic1.png";
                    }
                    if(inputBio === ""){ // if no bio is set,
sets it to "no bio"
                        registeredAccount.bio = "No bio";
                    }
                    registerAccount(registeredAccount);
                }
            }
        }
        else{
            alert("Password must contain atleast 1
number");
        }
    }
    else{
        alert("Password must contain atleast 6
characters");
    }
}
else{
    alert("Passwords don't match");
}
}
else{
    alert("Invalid Email");
}
}
else{
    alert("Username cannot contain whitespace");
}
}
}

```

```

        else{
            alert("Email taken");
        }
    }
    else{
        alert("User has been banned");
    }
}
else{
    alert("Please pick a different username");
}
}
else{
    alert("Username taken");
}
}
}

function registerAccount(details){
    var saveData = accountList; // fetches currently saved accounts
    saveData[details.userName.toLowerCase()] = details; // adds a new entry to the
object
    accountList = saveData; // updates the account list
    localStorage.setItem("saveData", JSON.stringify(saveData)); // writes the save data
to local storage
    if(details.profilePicture != "pic1.png"){ // if a profile picture was set
        var imgData = getBase64Image(displayedUploadedImg);
        localStorage.setItem([details.userName+details.profilePicture], imgData); //
details.profilePicture is a string equal to the username and the name/format of the
saved file, e.g. lemmy57dog.png
    }
}

function readURL(input){ // displays image to be converted to base64 string, but the
image is hidden - from stackoverflow
    if (input.files && input.files[0]) {
        var reader = new FileReader();
        reader.onload = function (e) {
            document.getElementById('displayedUploadedImg').src = e.target.result;
        }
        reader.readAsDataURL(input.files[0]);
    }
}

function getBase64Image(img){ //converts an image to base64 string - from stackoverflow
    var canvas = document.createElement("canvas");
    canvas.width = img.width;
    canvas.height = img.height;
    var ctx = canvas.getContext("2d");
    ctx.drawImage(img, 0, 0);
    var dataURL = canvas.toDataURL("image/png");
    return dataURL.replace(/^data:image\/(png|jpg);base64/, "");
}

```

```

function createAccounts(){ //for testing purposes
  accountList = {
    lemmy57: {
      firstName: "Liam",
      lastName: "Gerwahrts",
      userName: "lemmy57",
      email: "liam@yahoo.com",
      password: "ilovelemon1",
      profilePicture: "pic1.png",
      bio: "I really love lemon!",
      friends: {},
      friendRequests: {},
      posts: {}
    },
    dabaws1221: {
      firstName: "Martin",
      lastName: "Varbanov",
      userName: "dabaws1221",
      email: "varbanov@yahoo.co.uk",
      password: "meGrass22",
      profilePicture: "pic1.png",
      bio: "I really love grass!",
      friends: {},
      friendRequests: {},
      posts: {}
    },
    cooliodj: {
      firstName: "Joshua",
      lastName: "Ramsay",
      userName: "cooliodj",
      email: "ramsay@gmail.com",
      password: "heyokay5",
      profilePicture: "pic1.png",
      bio: "No bio",
      friends: {},
      friendRequests: {},
      posts: {}
    },
    ville: {
      firstName: "ville",
      lastName: "Sinki",
      userName: "ville",
      email: "ville@gmail.co.uk",
      password: "123456",
      profilePicture: "pic1.png",
      bio: "No bio",
      friends: {},
      friendRequests: {},
      posts: {}
    }
  }
  localStorage.setItem("saveData", JSON.stringify(accountList));
}

```

```
}
```

reports.js

```
var accountList = {};  
var reportedPosts = [];  
var bannedUsers = [];  
var admin = {  
  username: "administrator",  
  password: "okokok89"  
};  
var loggedInUsername;  
if (JSON.parse(localStorage.getItem("saveData")) !== null){ // if an account has  
  already been registered, it loads the account list  
  accountList = JSON.parse(localStorage.getItem("saveData"));  
}  
if (localStorage.getItem("loggedInData") !== null){ // checks if an account was  
  previously logged in  
  loggedInUsername = localStorage.getItem("loggedInData");  
}  
if (localStorage.getItem("bannedUsersData") !== null){ // checks the list of banned  
  usernames  
  bannedUsers = localStorage.getItem("bannedUsersData");  
}  
if (localStorage.getItem("reportedPostsData") !== null){ // loads reported posts  
  reportedPosts = JSON.parse(localStorage.getItem("reportedPostsData"));  
}  
function logout(){  
  localStorage.removeItem("loggedInData"); //removes the log in data  
  localStorage.removeItem("selectedUserData"); //removes the log in data  
  window.location.replace("../loginPage/login.html");  
}  
window.onload = function () { // ensures that the html loads before the javascript  
  createPostWallElements();  
  createPostCommentElements();  
  if (loggedInUsername != "administrator"){ // prevents non-admin users from  
    accessing this page  
    logout();  
  }  
}  
function createPostWallElements(){  
  removeElementsByClass("post");  
  var parentDiv = document.getElementById("postWallBox");  
  var unorderedPostsArray = [];  
  for (x of reportedPosts){  
    unorderedPostsArray.push(x);  
  }  
  var originalPostArrayLength = unorderedPostsArray.length;  
  for (x=0;x<originalPostArrayLength;x++){  
    var newestPost = unorderedPostsArray[0];  
    var newestPostZ = 0;  
    var newestPostDate = new Date(newestPost.date) // converts from string to date  
    for (z=0;z<unorderedPostsArray.length;z++){
```

```

        if (z < unorderedPostsArray.length-1){
            var nextPost = unorderedPostsArray[z+1];
            var nextPostDate = new Date(nextPost.date) // will cause an error if
nextPost is undefined
        }
        if(z < unorderedPostsArray.length-1){
            if (newestPostDate - nextPostDate <= 0){
                newestPost = nextPost;
                newestPostZ = z+1;
                newestPostDate = new Date(newestPost.date)
            }
        }
        else{
            unorderedPostsArray.splice(newestPostZ, 1); //removes the newest post
from the array
        }
    }
    var newDiv = document.createElement("div");
    newDiv.setAttribute("class", "post");
    newDiv.setAttribute("id", [newestPost.username+newestPost.date]);
    var newContent = document.createElement("div");
    newContent.setAttribute("class", "container");
    if (newestPost.image != ""){ //if an image has been uploaded
        var newContent2 = document.createElement("div");
        newContent2.setAttribute("class", "postImageBox");
        var newContent3 = document.createElement("img");
        newContent3.setAttribute("alt", "image")
        dataImage = localStorage.getItem([newestPost.username+newestPost.image]);
        newContent3.setAttribute("src", "data:image/png;base64,"+dataImage);
        newContent2.appendChild(newContent3);
        newContent.appendChild(newContent2);
    }
    var newContent2 = document.createElement("div");
    newContent2.setAttribute("class", "user");
    var newContent3 =
document.createTextNode(accountList[newestPost.username].firstName + " " +
accountList[newestPost.username].lastName + " | " + newestPost.username);
    newContent2.appendChild(newContent3);
    newContent.appendChild(newContent2);
    newContent2 = document.createElement("p");
    newContent3 = document.createTextNode(newestPost.data);
    newContent2.appendChild(newContent3);
    newContent.appendChild(newContent2);
    newContent2 = document.createElement("span");
    newContent2.setAttribute("class", "time-right");
    newContent3 = document.createElement("button");
    newContent4 = document.createTextNode("Ban User");
    newContent3.appendChild(newContent4);
    newContent3.setAttribute("onmousedown", `banUser('${newestPost.username}',
'${newestPost.date}')`);
    newContent2.appendChild(newContent3);
    newContent3 = document.createElement("button");
    newContent4 = document.createTextNode("Dismiss Report");

```

```

        newContent3.appendChild(newContent4);
        newContent3.setAttribute("onmousedown",
`dismissReport('${newestPost.username}', '${newestPost.date}')`);
        newContent2.appendChild(newContent3);
        newContent3 = document.createElement("br");
        newContent2.appendChild(newContent3);
        newContent3 = document.createTextNode([new
Date(newestPostDate)].toLocaleString()); // from JSON string to date object to string
        newContent2.appendChild(newContent3);
        newContent.appendChild(newContent2);
        newDiv.appendChild(newContent);
        parentDiv.insertBefore(newDiv, insertPostBefore);
    }
}
function createPostCommentElements(){ // creates the elements for the comments on the
page
    var unorderedPostsArray = [];
    for (x of reportedPosts){
        unorderedPostsArray.push(x);
    }
    removeElementsByClass("comment");
    for (z=0;z<unorderedPostsArray.length;z++){ // array of all relevant posts
        var parentDiv =
document.getElementById(unorderedPostsArray[z].username+unorderedPostsArray[z].date);
        var postCommentQuantity = Object.keys(unorderedPostsArray[z].comments).length;
        for (i=postCommentQuantity; i>0; i--){
            var newDiv = document.createElement("div");
            newDiv.setAttribute("class", "commentbox comment");
            var newContent = document.createElement("div");
            newContent.setAttribute("class", "user");
            var newContent2 =
document.createTextNode(accountList[unorderedPostsArray[z].comments[i].username].firstN
ame + " " + accountList[unorderedPostsArray[z].comments[i].username].lastName + " | " +
unorderedPostsArray[z].comments[i].username);
            newContent.appendChild(newContent2);
            newDiv.appendChild(newContent);
            newContent = document.createElement("p");
            newContent2 =
document.createTextNode(unorderedPostsArray[z].comments[i].data);
            newContent.appendChild(newContent2);
            newDiv.appendChild(newContent);
            parentDiv.appendChild(newDiv);
        }
    }
}

function banUser(userName, date){
    var banCheck = confirm("Are you sure you want to ban "+userName+"?"); // prompts
the admin
    if (banCheck){ //if yes is clicked
        delete accountList[userName];
        bannedUsers.push(userName.toLocaleLowerCase()); //must be all lower case as
string comparisons are case sensitive
    }
}

```

```

        localStorage.setItem("bannedUsersData", bannedUsers);
        localStorage.setItem("saveData", JSON.stringify(accountList));
        alert("User banned");
        dismissReport(userName, date);
    }
}

function dismissReport(userName, date){
    for (z of reportedPosts){
        if (z.username == userName){
            if (z.date == date){
                var y = reportedPosts.indexOf(z); //gets the index number of the post
                reportedPosts.splice(y, 1); //dismisses the reported post
                localStorage.setItem("reportedPostsData",
JSON.stringify(reportedPosts));
                location.reload();
            }
        }
    }
}

function removeElementsByClass(className){ // removes all elements with a certain class
name - from stackoverflow
    var elements = document.getElementsByClassName(className);
    while(elements.length > 0){
        elements[0].parentNode.removeChild(elements[0]);
    }
}

```

stats.js

```

var accountList = {};
var bannedUsers = [];
var admin = {
    username: "administrator",
    password: "okokok89"
};
var loggedInUsername;
if (JSON.parse(localStorage.getItem("saveData")) !== null){ // if an account has
already been registered, it loads the account list
    accountList = JSON.parse(localStorage.getItem("saveData"));
}
if (localStorage.getItem("loggedInData") !== null){ // checks if an account was
previously logged in
    loggedInUsername = localStorage.getItem("loggedInData");
}
if (localStorage.getItem("bannedUsersData") !== null){ // checks the list of banned
usernames
    bannedUsers = localStorage.getItem("bannedUsersData");
}
function logout(){
    localStorage.removeItem("loggedInData"); //removes the log in data
    localStorage.removeItem("selectedUserData"); //removes the log in data
    window.location.replace("../loginPage/login.html");
}

```



```

}
window.onload = function () { // ensures that the html loads before the javascript
    if (loggedInUsername != "administrator"){ // prevents non-admin users from
        accessing this page
            logout();
        }
        addUserList(accountList);
        for (userNames in accountList){
            statsPost(userNames, "day");
        }
        document.getElementById("txtUsername").onkeypress = function(e) { // detects if
enter is pressed in the textarea - from stackoverflow
            var event = e || window.event;
            var charCode = event.which || event.keyCode;
            if ( charCode == '13' ) {
                searchUser();
            }
        }
    }
}

function addUserList(userList){ //adds user divs, userList is an object containing the
usernames as key values
    var parentDiv = document.getElementById("adminUsersBox");
    removeElementsByClass("container");
    for (lowerCaseUserName in userList){ //I coded this before setting all usernames to
have lowercase in registration
        var userName = accountList[lowerCaseUserName].userName;
        var newDiv = document.createElement("div");
        newDiv.setAttribute("class", "container");
        var newContent = document.createElement("img");
        newContent.setAttribute("class", "profilePicture");
        newContent.setAttribute("alt", "Avatar");
        if (accountList[lowerCaseUserName].profilePicture == "pic1.png"){
            newContent.setAttribute("src", "../images/pic1.png"); //default image
        }
        else{
            dataImage =
localStorage.getItem(userName+accountList[lowerCaseUserName].profilePicture);
            newContent.setAttribute("src", "data:image/png;base64,"+dataImage);
        }
        newDiv.appendChild(newContent);
        newContent = document.createElement("div");
        newContent.setAttribute("class", "user");
        var newContent2 =
document.createTextNode(accountList[lowerCaseUserName].firstName + " " +
accountList[lowerCaseUserName].lastName + " | " + userName);
        newContent.appendChild(newContent2);
        newDiv.appendChild(newContent);
        newContent = document.createElement("p");
        newContent2 = document.createTextNode(accountList[lowerCaseUserName].bio);
        newContent.appendChild(newContent2);
        newDiv.appendChild(newContent);
        newContent2 = document.createElement("br");

```

```

newContent.appendChild(newContent2);
newContent2 = document.createElement("small");
var newContent3 = document.createTextNode("Number of posts in last");
newContent2.appendChild(newContent3);
newContent3 = document.createElement("br");
newContent2.appendChild(newContent3);
newContent3 = document.createElement("button");
newContent3.setAttribute("onmousedown", `statsPost('${userName}', 'day')`);
newContent3.setAttribute("id", [userName+"statsDay"])
newContent3.setAttribute("disabled", "true");
var newContent4 = document.createTextNode("Day");
newContent3.appendChild(newContent4);
newContent2.appendChild(newContent3);
newContent3 = document.createElement("button");
newContent3.setAttribute("onmousedown", `statsPost('${userName}', 'week')`);
newContent3.setAttribute("id", [userName+"statsWeek"])
var newContent4 = document.createTextNode("Week");
newContent3.appendChild(newContent4);
newContent2.appendChild(newContent3);
newContent3 = document.createElement("button");
newContent3.setAttribute("onmousedown", `statsPost('${userName}', 'alltime')`);
newContent3.setAttribute("id", [userName+"statsAlltime"])
var newContent4 = document.createTextNode("All Time");
newContent3.appendChild(newContent4);
newContent2.appendChild(newContent3);
newContent3 = document.createTextNode(" : ")
newContent2.appendChild(newContent3);
newContent3 = document.createElement("span");
newContent3.setAttribute("id", [userName+"statsPosts"])
newContent2.appendChild(newContent3);
newContent.appendChild(newContent2);
newContent2 = document.createElement("br");
newContent.appendChild(newContent2);
newContent2 = document.createElement("button");
newContent3 = document.createTextNode("Ban User");
newContent2.appendChild(newContent3);
newContent2.setAttribute("onmousedown", `banUser('${userName}')`);
newContent.appendChild(newContent2);
newDiv.appendChild(newContent);
parentDiv.appendChild(newDiv);
statsPost(lowerCaseUserName, "day");
}
}

function banUser(userName, date){
    var banCheck = confirm("Are you sure you want to ban "+userName+"?"); // prompts
the admin
    if (banCheck){ //if yes is clicked
        delete accountList[userName];
        if (typeof bannedUsers == "string"){ // with only 1 value in the array,
saving/loading in local storage seems to convert the value to a string, which causes
bannedUsers.push() to produce an error
            bannedUsers = bannedUsers.split(" ");
        }
    }
}

```

```

        bannedUsers.push(userName.toLocaleLowerCase()); //must be all lower case as
string comparisons are case sensitive
        localStorage.setItem("bannedUsersData", bannedUsers);
        localStorage.setItem("saveData", JSON.stringify(accountList));
        alert("User banned");
        location.reload();
    }
}

function removeElementsByClass(className){ // removes all elements with a certain class
name - from stackoverflow
    var elements = document.getElementsByClassName(className);
    while(elements.length > 0){
        elements[0].parentNode.removeChild(elements[0]);
    }
}

function statsPost(userName, sort){
    if(sort == "day"){
        var postsDay = 0; //reset the value to 0
        var postDate;
        var currentdate = new Date();
        for (z in accountList[userName].posts){
            postDate = new Date(accountList[userName].posts[z].date);
            if (currentdate - postDate <= 86400000){ // milliseconds in a day
                postsDay++; //the value of postsDay will be the number of posts that
meet this conditional statement (posted within the last day)
            }
        }
        document.getElementById([userName+"statsPosts"]).innerHTML = postsDay;
        document.getElementById([userName+"statsDay"]).disabled = true; //disables the
clicked button and enables the other buttons
        document.getElementById([userName+"statsWeek"]).disabled = false;
        document.getElementById([userName+"statsAlltime"]).disabled = false;
    }
    else if(sort == "week"){
        var postsWeek = 0;
        var postDate;
        var currentdate = new Date();
        for (z in accountList[userName].posts){
            postDate = new Date(accountList[userName].posts[z].date);
            if (currentdate - postDate <= 604800000){ // milliseconds in a week
                postsWeek++;
            }
        }
        document.getElementById([userName+"statsPosts"]).innerHTML = postsWeek;
        document.getElementById([userName+"statsDay"]).disabled = false;
        document.getElementById([userName+"statsWeek"]).disabled = true;
        document.getElementById([userName+"statsAlltime"]).disabled = false;
    }
    else if(sort == "alltime"){
        var postsAlltime = 0;
        for (z in accountList[userName].posts){
            postsAlltime = object.keys(accountList[userName].posts).length;

```

```

    }
    document.getElementById([userName+"statsPosts"]).innerHTML = postsAlltime;
    document.getElementById([userName+"statsDay"]).disabled = false;
    document.getElementById([userName+"statsWeek"]).disabled = false;
    document.getElementById([userName+"statsAlltime"]).disabled = true;
  }
}

function searchUser(){
  var searchInput = document.getElementById("txtUsername").value;
  var searchUsersObj = {};
  if (searchInput != ""){ // search won't perform when no value is inputted
    for (userNames in accountList){
      if (userNames.indexOf(searchInput) > -1){
        searchUsersObj[userNames] = true;
      }
    }
    addUserList(searchUsersObj);
  }
  else{
    addUserList(accountList); //when there is no value for the search, it just
    shows all users
  }
}

```