

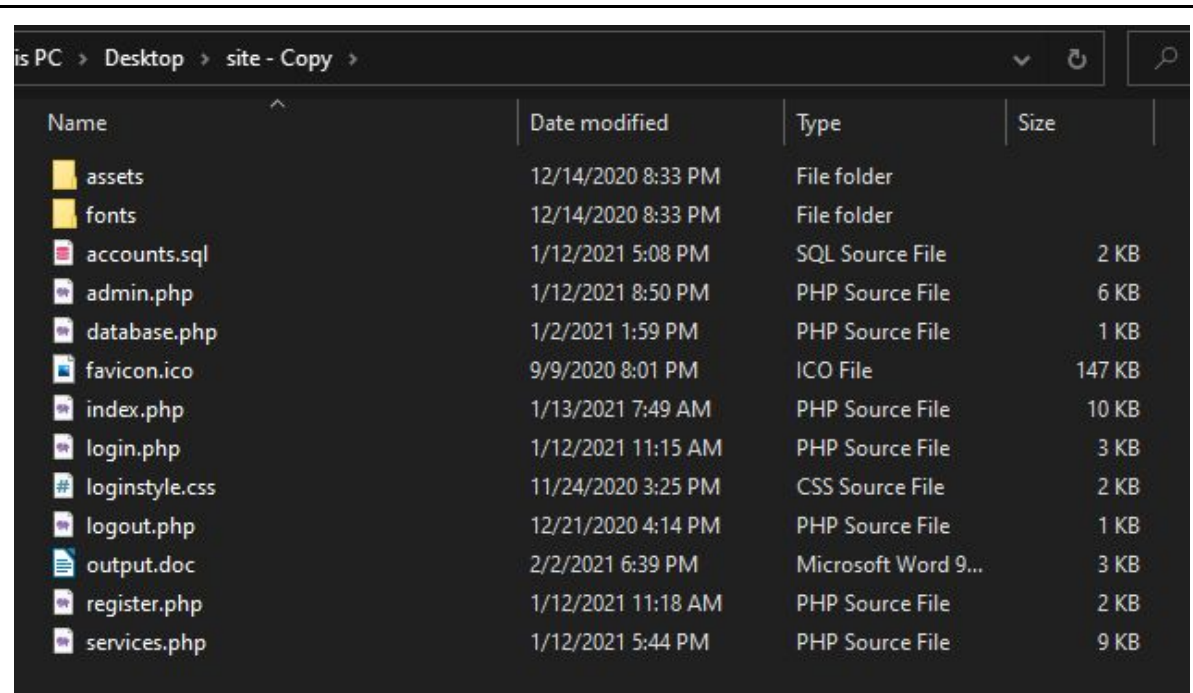
## Criterion E: Product development

### Advanced techniques used to address the client's requirements:

- Manipulation of HyperText Markup Language (HTML)
- Cascading style sheets (CSS)
- Proficient use of techniques to enable easy navigation (such as menus or buttons, internal and external links)
- Loops
- Parameter passing

## Organization of the website (Folders)

This indicates how the website is structured:



| Name           | Date modified      | Type                | Size   |
|----------------|--------------------|---------------------|--------|
| assets         | 12/14/2020 8:33 PM | File folder         |        |
| fonts          | 12/14/2020 8:33 PM | File folder         |        |
| accounts.sql   | 1/12/2021 5:08 PM  | SQL Source File     | 2 KB   |
| admin.php      | 1/12/2021 8:50 PM  | PHP Source File     | 6 KB   |
| database.php   | 1/2/2021 1:59 PM   | PHP Source File     | 1 KB   |
| favicon.ico    | 9/9/2020 8:01 PM   | ICO File            | 147 KB |
| index.php      | 1/13/2021 7:49 AM  | PHP Source File     | 10 KB  |
| login.php      | 1/12/2021 11:15 AM | PHP Source File     | 3 KB   |
| loginstyle.css | 11/24/2020 3:25 PM | CSS Source File     | 2 KB   |
| logout.php     | 12/21/2020 4:14 PM | PHP Source File     | 1 KB   |
| output.doc     | 2/2/2021 6:39 PM   | Microsoft Word 9... | 3 KB   |
| register.php   | 1/12/2021 11:18 AM | PHP Source File     | 2 KB   |
| services.php   | 1/12/2021 5:44 PM  | PHP Source File     | 9 KB   |

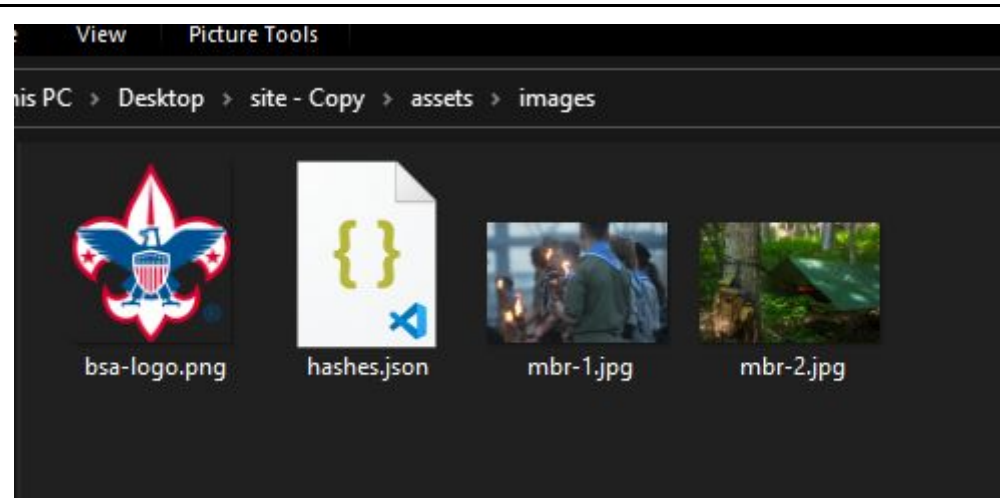
The PHP web pages, CSS file of the login system and favicon of the website are placed in the root directory, and assets and fonts are sorted into 2 folders. I left what I had modified or created from scratch in the root directory, and the favicon there as it is required to be there for browser compatibility.

| Name          | Date modified    | Type               | Size   |
|---------------|------------------|--------------------|--------|
| 300.ttf       | 9/7/2020 7:58 PM | TrueType font file | 125 KB |
| 300italic.ttf | 9/7/2020 7:58 PM | TrueType font file | 114 KB |
| 500.ttf       | 9/7/2020 7:58 PM | TrueType font file | 129 KB |
| 500italic.ttf | 9/7/2020 7:58 PM | TrueType font file | 115 KB |
| 600.ttf       | 9/7/2020 7:58 PM | TrueType font file | 129 KB |
| 600italic.ttf | 9/7/2020 7:58 PM | TrueType font file | 115 KB |
| 700.ttf       | 9/7/2020 7:58 PM | TrueType font file | 129 KB |
| 700italic.ttf | 9/7/2020 7:58 PM | TrueType font file | 115 KB |
| 800.ttf       | 9/7/2020 7:58 PM | TrueType font file | 129 KB |
| 800italic.ttf | 9/7/2020 7:58 PM | TrueType font file | 115 KB |
| 900.ttf       | 9/7/2020 7:58 PM | TrueType font file | 129 KB |
| 900italic.ttf | 9/7/2020 7:58 PM | TrueType font file | 115 KB |
| italic.ttf    | 9/7/2020 7:58 PM | TrueType font file | 115 KB |
| regular.ttf   | 9/7/2020 7:58 PM | TrueType font file | 129 KB |
| style.css     | 9/7/2020 7:58 PM | CSS Source File    | 3 KB   |

On the fonts folder of the root directory, are the font files used in the login system.

| Name         | Date modified      | Type        | Size |
|--------------|--------------------|-------------|------|
| bootstrap    | 12/14/2020 8:33 PM | File folder |      |
| dropdown     | 12/14/2020 8:33 PM | File folder |      |
| formoid      | 12/14/2020 8:33 PM | File folder |      |
| images       | 12/18/2020 9:45 AM | File folder |      |
| mobirise     | 12/14/2020 8:33 PM | File folder |      |
| parallax     | 12/14/2020 8:33 PM | File folder |      |
| popper       | 12/14/2020 8:33 PM | File folder |      |
| smoothscroll | 12/14/2020 8:33 PM | File folder |      |
| socicon      | 12/14/2020 8:33 PM | File folder |      |
| tether       | 12/14/2020 8:33 PM | File folder |      |
| theme        | 12/14/2020 8:33 PM | File folder |      |
| touchswipe   | 12/14/2020 8:33 PM | File folder |      |
| web          | 12/14/2020 8:33 PM | File folder |      |

Above are the assets used by mobirise web builder, in a folder titled assets at the root directory. It contains resources such as images used in the website within an images folder; Bootstrap files; folders with scripts to run for aesthetical improvements, compatibility with different devices, form sending service; icons, Mobirise images and fonts and CSS in their corresponding folders as well.



All images are included in the images folder within the aforementioned assets folder.

## Technique 1: Manipulation of HyperText Markup Language (HTML)

My index and services pages are derived from a website building service called Mobirise, which has extended libraries of royalty-free images, icons, and code for navbars, buttons, stylesheets, javascript, and forms. The index, services pages were modified templates form Mobirise, with the icons and images sourced from their royalty-free library.

```
<?php
session_start();      *PHP portion of webpage

require 'database.php';

if(!empty($_POST['email']) && !empty($_POST['password'])):

    $records = $conn->prepare('SELECT id,email,password,verified,admin FROM accounts WHERE email = :email');
    $records->bindParam(':email', $_POST['email']);
    $records->execute();
    $results = $records->fetch(PDO::FETCH_ASSOC);

    $message = '';

    if(count($results) > 0 && password_verify($_POST['password'], $results['password']))(

        $_SESSION['user_id'] = $results['id'];
        $_SESSION['admin'] = $results['admin'];
        $verified = $results['verified'];
        if($verified == 1){
            if ($_SESSION['admin'] == 1){
                header("Location: admin.php");
            }
            else {
                header("Location: services.php");
            }
        }
        else {
            $message = 'Your login is pending. Please wait for account approval';
        }
    }
    else{
        $message = 'There was an error login in, please double check the email and password.';
    }
}

endif;
?>
```

Login authentication and redirection taken then modified from Stackoverflow [3]

The login, signup pages were created from scratch, with the PHP code for authentication and registration sourced from Stackoverflow. In the sign-in page, a piece of PHP code is to authenticate the users.

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1">
  <link rel="shortcut icon" href="assets/images/bsa-logo.png" type="image/x-icon">
  <link rel="icon" href="favicon.png" type="image/x-icon">
  <title>Troop1453 Login</title>
  <link rel="stylesheet" href="loginstyle.css">
  <link href="https://fonts.googleapis.com/css?family=Ubuntu" rel="stylesheet">
</head>

<body>
  <div class="main">
    <p class="sign" align="center">Sign in</p>
    <?php if(!empty($message)): ?>
      <p><?=$message ?></p>
    <?php endif; ?>

    <form action="login.php" method="POST">

      <form class="form1">
        <input class="un " type="text" align="center" placeholder="Email" name="email">
        <input class="pass" type="password" align="center" placeholder="Password" name="password">
        <input type="submit" class="submit" align="center">
        <p class="forgot" align="center"><a href="register.php">Register A New Troop1453 Account</p>
      </form>

    </div>
  </body>
</html>

```

\*HTML portion of webpage

Asset inclusion

Text located above form

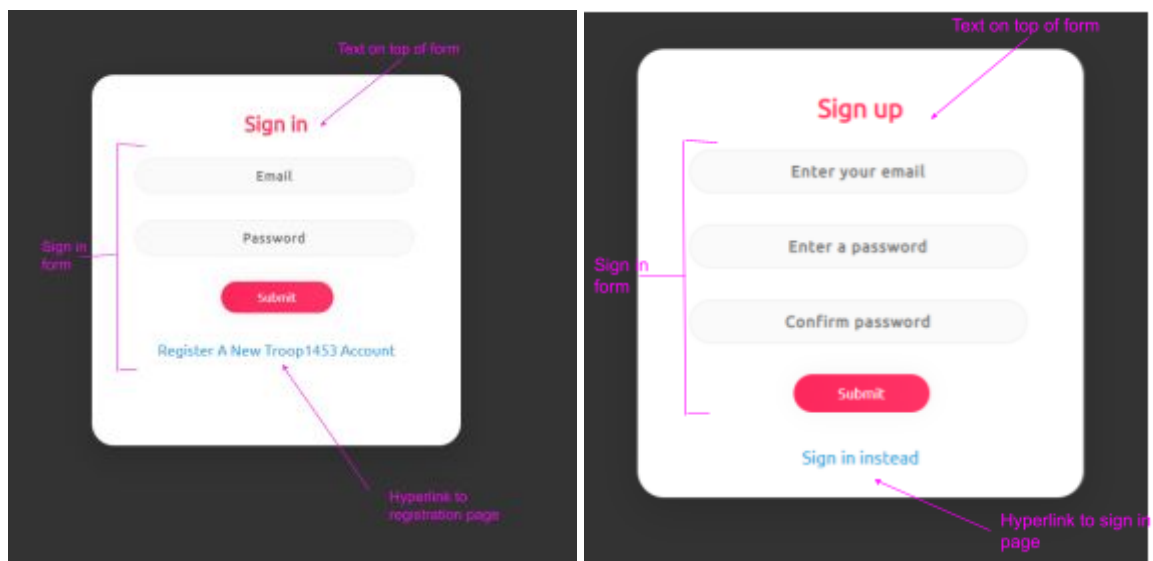
Text input fields for email password

Submit button

Sign-in form

Button to redirect to registration page

HTML code for a form and piece of text on top, and redirection hyperlink, details annotated.



The sign in and sign up pages rendered in a browser.

```

<?php
require 'database.php';

$message = '';

if (!empty($_POST['email']) && !empty($_POST['password']) && $_POST['password']==$_POST['cpassword']):

    $sql = "INSERT INTO accounts (email, password) VALUES (:email, :password)";
    $stmt = $conn->prepare($sql);

    $stmt->bindParam(':email', $_POST['email']);
    $stmt->bindParam(':password', password_hash($_POST['password'], PASSWORD_BCRYPT));

    if( $stmt->execute() ):
        $message = 'Account Created Successfully';
    else:
        $message = 'There Was An Issue Creating Your Account';
    endif;

endif;
?>

```

**\*PHP portion of webpage**

Code taken from Stackoverflow and modified, to create users and insert them into a database [3]

In the registration page, a piece of PHP code is to insert new users data into the database. Details annotated.

```

<!DOCTYPE html>
<html>
<head>
<title>Troop1453 Registration</title>
<link rel="stylesheet" href="loginstyle.css">
<link href="https://fonts.googleapis.com/css?family=Ubuntu" rel="stylesheet">
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="shortcut icon" href="assets/images/bsa-logo.png" type="image/x-icon">
<link rel="icon" href="favicon.png" type="image/x-icon">
<link rel="stylesheet" href="path/to/font-awesome/css/font-awesome.min.css">
</head>

<body>
<div class="main">
<p class="sign" align="center">Sign up</p>

<?php if(!empty($message)): ?>
    <p>= $message ?&gt;&lt;/p&gt;
&lt;?php endif; ?&gt;

&lt;form action="register.php" method="POST"&gt;

&lt;form class="form1"&gt;
    &lt;input class="un" type="text" align="center" placeholder="Enter your email" name="email"&gt;
    &lt;input class="pass" type="password" align="center" placeholder="Enter a password" name="password"&gt;
    &lt;input class="pass" type="password" align="center" placeholder="Confirm password" name="cpassword"&gt;
    &lt;input type="submit" class="submit" align="center"&gt;
    &lt;p class="forgot" align="center"&gt;&lt;a href="login.php"&gt;Sign in instead&lt;/p&gt;
&lt;/form&gt;

&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre

*HTML portion of the page



Asset inclusion



Registration form



Text input fields for email password and password confirmation



Submit form button



Button to redirect to sing in page


```

HTML code for a form and a redirection button. Details annotated



### **Citations**

"10 Open-Source Login Page Templates Built With HTML & CSS." Speckyboy Design Magazine, 5 Jan. 2021, [speckyboy.com/login-pages-html5-css/](https://speckyboy.com/login-pages-html5-css/).

Adams, David. "Secure Login System with PHP and MySQL." CodeShack, 27 Jan. 2021, [codeshack.io/secure-login-system-php-mysql/](https://codeshack.io/secure-login-system-php-mysql/).

"FREE WEBSITE BUILDER SOFTWARE." Best Free Website Builder Software 2021, 10 Mar. 2021, [mobirise.com/](https://mobirise.com/).

"Mobirise Free Icons." Free Icon Font - Pixel-Perfect Line Vector Icons, [mobiriseicons.com/](https://mobiriseicons.com/).



## Technique 2: Cascading style sheets (CSS)

I had created a custom CSS file, titled `loginstyle.css` that is shared by the sign-up and sign-in pages, instead of using Mobirise CSS as they did not have templates explicitly specifically for sign-in pages. CSS is used in order to make the pages more visually appealing.

```
<link rel="stylesheet" href="loginstyle.css">
<link href="https://fonts.googleapis.com/css?family=Roboto:400,700,900">
```

The stylesheet is included in the pages using the above command.

```
body {
  background-color: #333333;
  font-family: 'Ubuntu', sans-serif;
}

.main {
  background-color: #FFFFFF;
  width: 400px;
  height: 400px;
  margin: 7em auto;
  border-radius: 1.5em;
  box-shadow: 0px 11px 35px 2px rgba(0, 0, 0, 0.14);
}

.sign {
  padding-top: 40px;
  color: #FF3366;
  font-family: 'Ubuntu', sans-serif;
  font-weight: bold;
  font-size: 23px;
}

.un {
  width: 76%;
  color: rgb(38, 50, 56);
  font-weight: 700;
  font-size: 14px;
  letter-spacing: 1px;
  background: rgba(136, 126, 126, 0.04);
  padding: 10px 20px;
  border: none;
  border-radius: 20px;
  outline: none;
  box-sizing: border-box;
  border: 2px solid rgba(0, 0, 0, 0.02);
  margin-bottom: 50px;
  margin-left: 46px;
  text-align: center;
  margin-bottom: 27px;
  font-family: 'Ubuntu', sans-serif;
}

form form1 {
  padding-top: 40px;
}

.pass {
  width: 76%;
  color: rgb(38, 50, 56);
  font-weight: 700;
  font-size: 14px;
  letter-spacing: 1px;
  background: rgba(136, 126, 126, 0.04);
  padding: 10px 20px;
  border: none;
  border-radius: 20px;
  outline: none;
  box-sizing: border-box;
  border: 2px solid rgba(0, 0, 0, 0.02);
  margin-bottom: 50px;
  margin-left: 46px;
  text-align: center;
  margin-bottom: 27px;
  font-family: 'Ubuntu', sans-serif;
}

.un:focus, .pass:focus {
  border: 2px solid rgba(0, 0, 0, 0.18) !important;
}

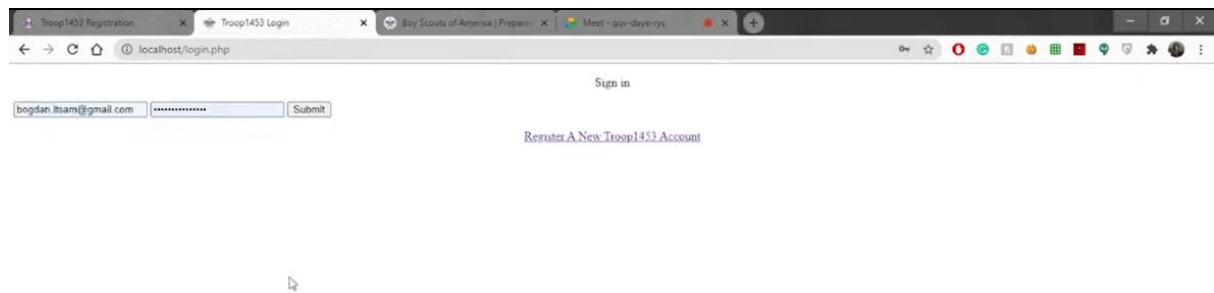
.submit {
  cursor: pointer;
  border-radius: 5em;
  color: #fff;
  background: linear-gradient(to right, #fc275c, #ff3366);
  border: 0;
  padding-left: 40px;
  padding-right: 40px;
  padding-bottom: 10px;
  padding-top: 10px;
  font-family: 'Ubuntu', sans-serif;
  margin-left: 35%;
  font-size: 13px;
  box-shadow: 0 0 28px 1px rgba(0, 0, 0, 0.04);
}

.forgot {
  text-shadow: 0px 0px 3px rgba(117, 117, 117, 0.12);
  color: #1490cc;
  padding-top: 15px;
}

a {
  text-shadow: 0px 0px 3px rgba(117, 117, 117, 0.12);
  color: #1490cc;
  text-decoration: none
}
```

As shown here, I have created multiple class attributes for the elements of both pages such as the forms, text, buttons, and background, which changes the colours of the background, text, buttons, font, button shape, alignment, and padding.

We may visualize the change the CSS file makes if we take it out of the equation, and we can see that without the CSS the page looks very bland:

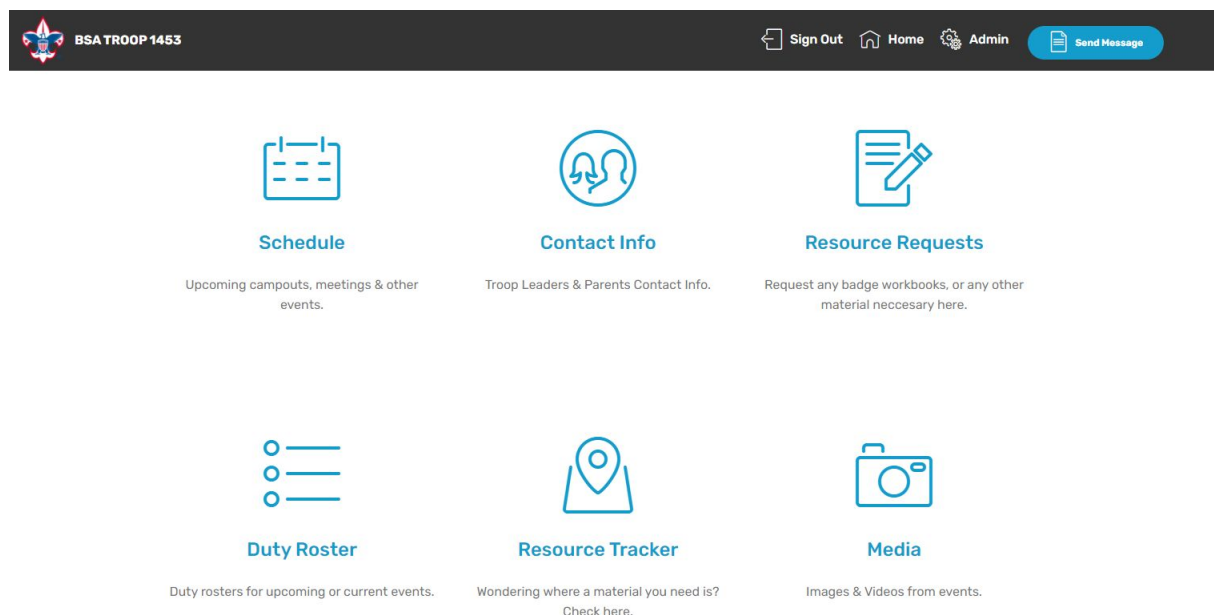


### **Citations:**

"10 Open-Source Login Page Templates Built With HTML & CSS." Speckyboy Design Magazine, 5 Jan. 2021, speckyboy.com/login-pages-html5-css/.

"W3.CSS Tutorial." W3.CSS Home, www.w3schools.com/w3css/default.asp.

### **Technique 3: Proficient use of techniques to enable easy navigation (such as menus or buttons, internal and external links)**



On the services page, we see icons and text with hyperlinks to various google services, and on the navbar a button only accessible to those who have admin status.

```

<?php
    *PHP part of the webpage
    session_start();
    if ( !isset ( $_SESSION['user_id'])) {
    header('Location: login.php');
    }
    if ( $_SESSION['admin']==1){
        $adminlink = '<li class="nav-item">
            <a class="nav-link link text-white text-primary display-4" href="admin.php">
                <span class="mbri-setting mbr-iconfont mbr-iconfont-btn"></span>
                Admin
            </a>
        </li>';
    } else {
        $adminlink='';
    }
}
?>

```

Mobirise button class

Code used to check admin status and display admin page hyperlink button [3]

Mbri specifies the image,

```

</li>
<li class="nav-item">
<a class="nav-link link text-white text-primary display-4" href="index.php">
    <span class="mbri-home mbr-iconfont mbr-iconfont-btn"></span>
    Home
</a>
</li>

```

Mobirise button class

Snippet of navbar code for buttons

```

<div class="card p-3 col-12 col-md-6 col-lg-4">
    <div class="card-img pb-3">
        <a href="https://drive.google.com/drive/folders/1jBGzw2CSA9DrI-xAC_AGPnISLnVJzh3Y?usp=sharing" target="_blank"><span class="mbr-iconfont mbr-edit"></span>
        </div>
    <div class="card-box">
        <h4 class="card-title py-3 mbr-fonts-style display-5">
            Resource Requests</h4>
        <p class="mbr-text mbr-fonts-style display-7">
            Request any badge workbooks, or any other material necessary here.
        </p>
    </div>
</div>
</div>

```

Snippet of service hyperlinks

Throughout the navbar, we may see a logout button, which if clicked is redirected to a logout.php page that destroys your session, and redirects you to the login page.

```

<a class="nav-link link text-white text-primary display-4" href="logout.php">
    <span class="mbri-logout mbr-iconfont mbr-iconfont-btn"></span>
    Sign Out
</a>

```

Redirection link to logout

### Citations:

Adams, David. "Secure Login System with PHP and MySQL." CodeShack, 27 Jan. 2021, codeshack.io/secure-login-system-php-mysql/.

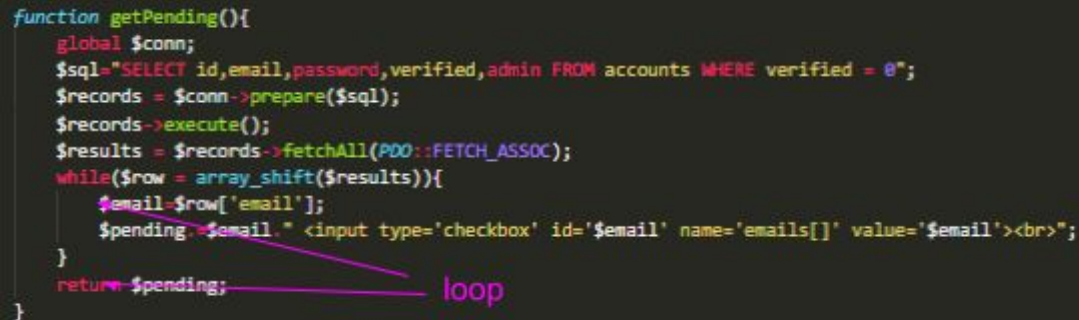
"FREE WEBSITE BUILDER SOFTWARE." Best Free Website Builder Software 2021, 10 Mar. 2021, mobirise.com/.

"HTML Element Reference." HTML Reference, www.w3schools.com/tags/.

#### Technique 4: Loops

There is a loop in database.php within the getPending function which fetches the records of all unverified accounts, and it does a loop of the function until no further results are available to be displayed.

```
function getPending(){
    global $conn;
    $sql="SELECT id,email,password,verified,admin FROM accounts WHERE verified = 0";
    $records = $conn->prepare($sql);
    $records->execute();
    $results = $records->fetchAll(PDO::FETCH_ASSOC);
    while($row = array_shift($results)){
        $email=$row['email'];
        $pending.=$email." <input type='checkbox' id='$email' name='emails[]' value='$email'><br>";
    }
    return $pending;
}
```



#### Citations

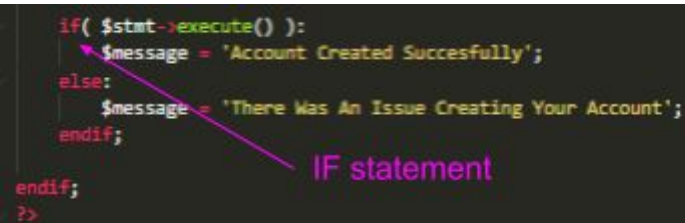
Hermans, Philo. "How to Implement Email Verification for New Members." Code Envato Tuts+, Envato Tuts, 9 Oct. 2020, code.tutsplus.com/tutorials/how-to-implement-email-verification-for-new-members--net-3824.

PHP While Loop, www.w3schools.com/php/php\_looping\_while.asp.

#### Technique 5: Parameter passing

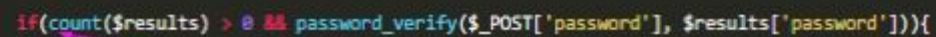
In registration page, if the account was created successfully, a message comes up prompting that the account was created successfully, if not, it prompts that there was a problem creating the account, after checking the execution with an IF statement.

```
if( $stmt->execute() ):
    $message = 'Account Created Succesfully';
else:
    $message = 'There Was An Issue Creating Your Account';
endif;
endif;
?>
```



On the sign-in page, there are few lines of PHP code is used to verify if the user has created an account with an IF statement,

```
if(count($results) > 0 && password_verify($_POST['password'], $results['password']))){
```



in addition to his verification status and admin status, which is done by this statement to check if the verification and or status is set to 1 in the database if so redirecting them to the according pages, which is done after this SQL query. If the account is not verified, it will prompt that the account is to be verified, or if there is an issue logging in

```

if ($SESSION['admin'] == 1){
    header("location: admin.php");
}
else {
    header("location: services.php");
}

```

On the top PHP code of admin.php, we have a line of code that checks for permission to access the admin panel, if the user does not have an admin value set to 1 but is verified, he is redirected to the services page, and if nor is he verified, then he is redirected to the login page.

```

if ($SESSION['admin'] == 0){
    header('location: services.php');
}

```

```

$result->getPending();
if ($result==""){
    $result="<div style='background-color:green; color:white; padding:10px'>There are currently no pending accounts to verify </div>";
}

```

On admin.php, the aforementioned get pending function, followed by an IF statement that is instructed to say "There are currently no pending accounts to verify" with inline styling if the function's result is empty.

### **Citations:**

Adams, David. "Secure Login System with PHP and MySQL." CodeShack, 27 Jan. 2021, [codeshack.io/secure-login-system-php-mysql/](https://codeshack.io/secure-login-system-php-mysql/).

Hermans, Philo. "How to Implement Email Verification for New Members." Code Envato Tuts+, Envato Tuts, 9 Oct. 2020, [code.tutsplus.com/tutorials/how-to-implement-email-verification-for-new-members--net-3824](https://code.tutsplus.com/tutorials/how-to-implement-email-verification-for-new-members--net-3824).

PHP If...else...elseif Statements, [www.w3schools.com/php/php\\_if\\_else.asp](https://www.w3schools.com/php/php_if_else.asp).

**Word count: 822**