Scott's Website

Author: Scott Seburn
Date: 11/15/23

// FLATIRON SCHOOL

LAMP Stack

LAMP STACK Information

Description:

LAMP stack is an open-source software stack commonly used for web development. LAMP stands for Linux, Apache, MySQL, and PHP. Together, these components form an environment for building and hosting dynamic websites and web applications. Linux provides the operating system foundation, Apache handles web server functionalities, MySQL manages the database, and PHP enables server-side scripting for dynamic content generation.

System Specs:

Operating System: Fedora Linux v.38

Apache Version: Apache/2.4.58

Database: 10.5.22-MariaDB MariaDB Server

PHP: PHP Version 8.2.12

Index.php

Index.php Information

Description:

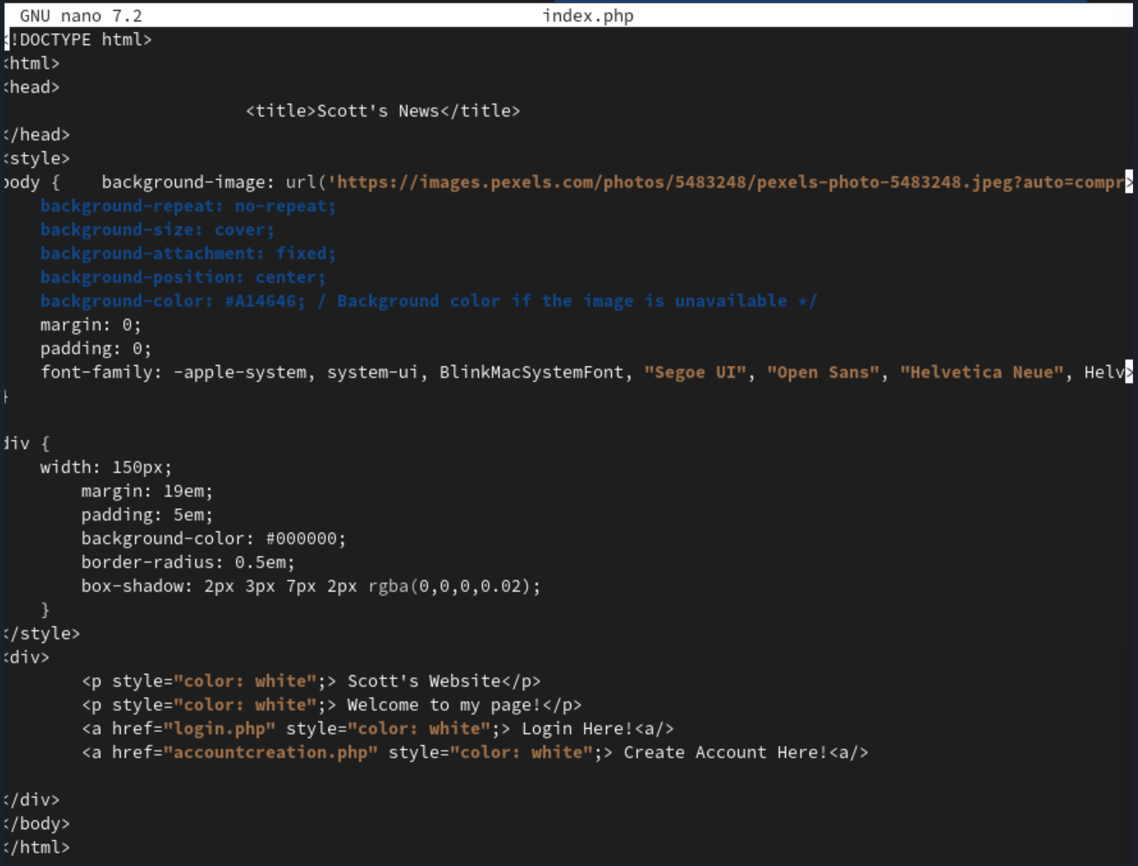
This is my index page! Also known as an home page, this page is used to navigate to either the login page or account creation page when traffic comes in from the internet. The general point of an index page it give an overview of the content of a website, to which then they can explore further on deeper pages. However, my page is in a crude form while undergoing construction, so the most basic features are laid out.

Key Points about the Page:

- Used to navigate to deeper page layers
- "Home page" that all internet traffic sees when first interacting with your website.

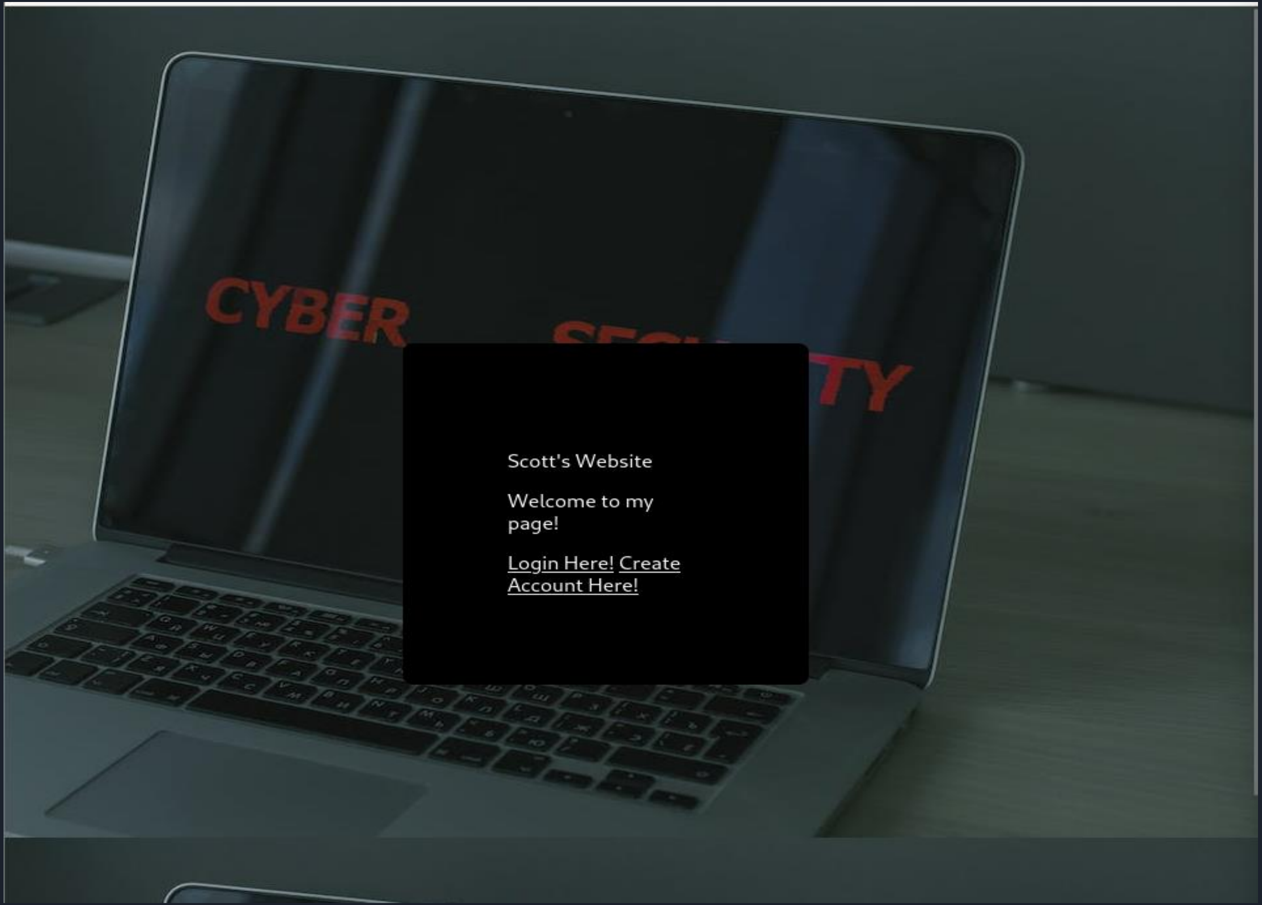
Index.php Code Screenshot

- Defines the document as html.
- The styles define the appearances of the web page.
- The content paragraphs welcome you to the page and provides links for the login and account creation pages.



```
 GNU nano 7.2                                                    index.php
<!DOCTYPE html>
<html>
<head>
                        <title>Scott's News</title>
</head>
<style>
body {    background-image: url('https://images.pexels.com/photos/5483248/pexels-photo-5483248.jpeg?auto=compr>
   background-repeat: no-repeat;
   background-size: cover;
   background-attachment: fixed;
   background-position: center;
   background-color: #A14646; / Background color if the image is unavailable */
   margin: 0;
   padding: 0;
   font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helv>
}

div {
   width: 150px;
       margin: 19em;
       padding: 5em;
       background-color: #000000;
       border-radius: 0.5em;
       box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
   }
</style>
<div>
       <p style="color: white";> Scott's Website</p>
       <p style="color: white";> Welcome to my page!</p>
       <a href="login.php" style="color: white";> Login Here!<a/>
       <a href="accountcreation.php" style="color: white";> Create Account Here!<a/>

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

PHP Webpage Screenshot



Scott's Website

Welcome to my page!

Login Here! Create Account Here!

Database Information

Database Information

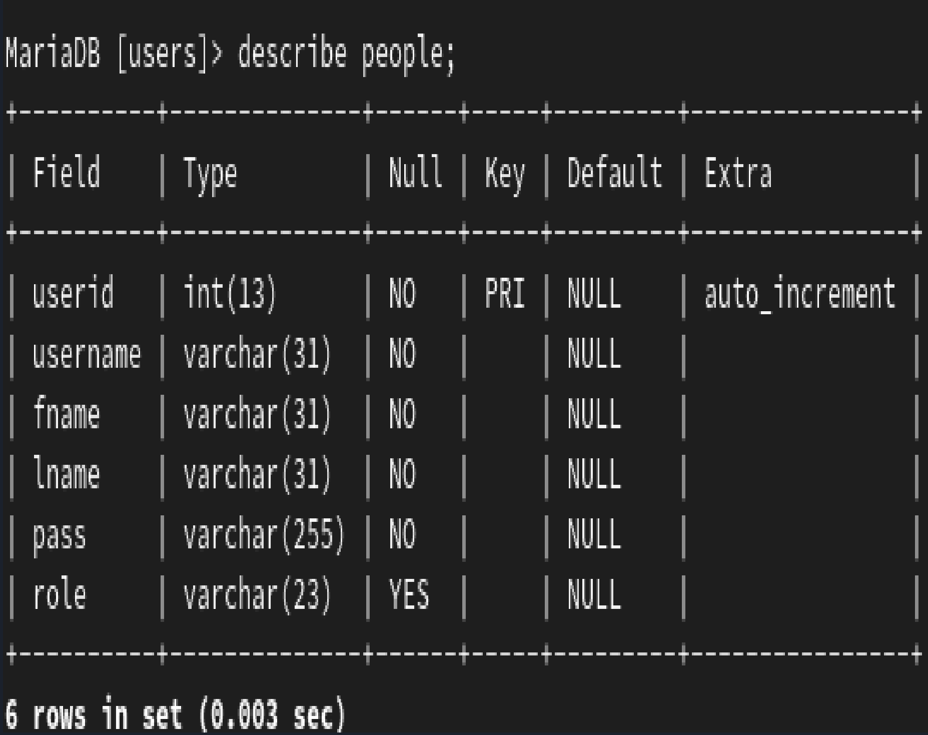
Description:

This database was a required component of my website in that it helps stash and access account credentials. Without this database, my website would almost be unusable, because anyone who would visit my page would be unable to have their own account. This database ensures that everyone who visits the site and creates an account will have their own private page.

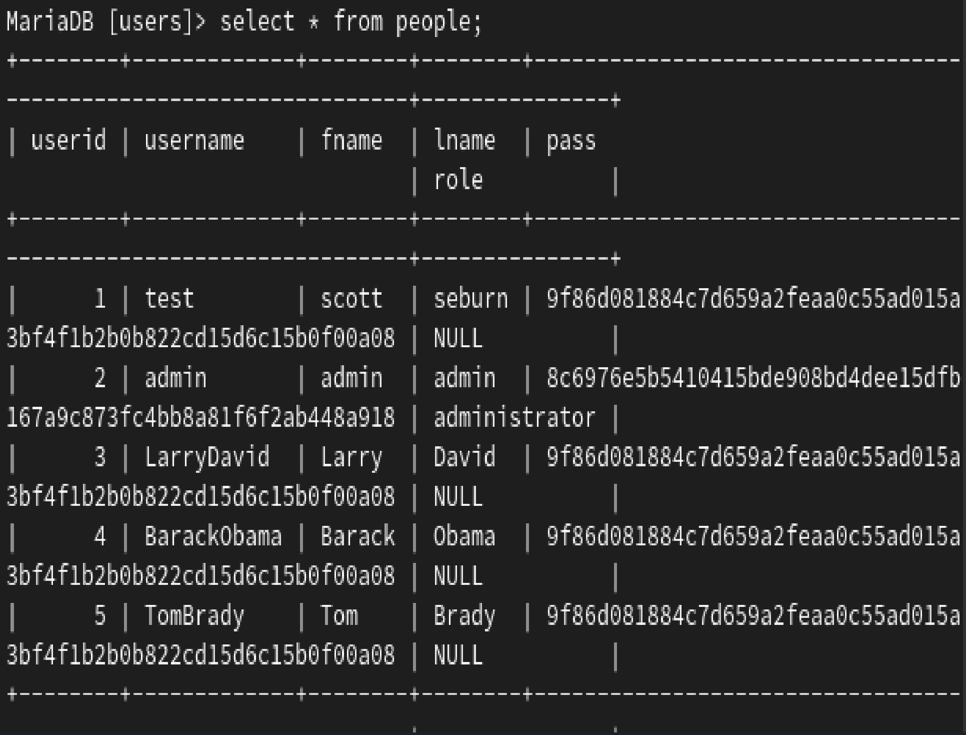
Key Points about the Page:

- During the database install I had to completely scrap my Fedora due to messing up the password change command.
- I was unable to access mariadb from that point on, leaving restarting as the only option.
- The one real problem with this was when I moved my files to a drop box and then into my new fedora, whenever I would to my website it would tell me "access denied" so I had to turn SElinux to "permissive mode," so I could acces the files.

Database Screenshot



```
MariaDB [users]> describe people;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| userid   | int(13)      | NO   | PRI | NULL    | auto_increment |
| username | varchar(31)  | NO   |     | NULL    |                |
| fname    | varchar(31)  | NO   |     | NULL    |                |
| lname    | varchar(31)  | NO   |     | NULL    |                |
| pass     | varchar(255) | NO   |     | NULL    |                |
| role     | varchar(23)  | YES  |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
6 rows in set (0.003 sec)
```



```
MariaDB [users]> select * from people;
+--------+-------------+--------+--------+----------------------------------
--------------------------------+---------------+
| userid | username    | fname  | lname  | pass
                                | role          |
+--------+-------------+--------+--------+----------------------------------
--------------------------------+---------------+
|      1 | test        | scott  | seburn | 9f86d081884c7d659a2feaa0c55ad015a
3bf4f1b2b0b822cd15d6c15b0f00a08 | NULL          |
|      2 | admin       | admin  | admin  | 8c6976e5b5410415bde908bd4dee15dfb
167a9c873fc4bb8a81f6f2ab448a918 | administrator |
|      3 | LarryDavid  | Larry  | David  | 9f86d081884c7d659a2feaa0c55ad015a
3bf4f1b2b0b822cd15d6c15b0f00a08 | NULL          |
|      4 | BarackObama | Barack | Obama  | 9f86d081884c7d659a2feaa0c55ad015a
3bf4f1b2b0b822cd15d6c15b0f00a08 | NULL          |
|      5 | TomBrady    | Tom    | Brady  | 9f86d081884c7d659a2feaa0c55ad015a
3bf4f1b2b0b822cd15d6c15b0f00a08 | NULL          |
+--------+-------------+--------+--------+----------------------------------
```

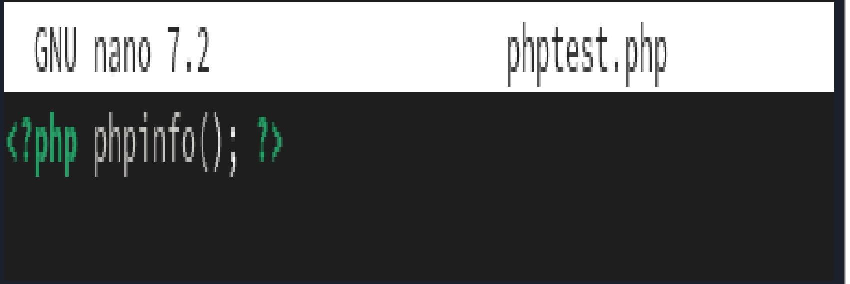
phptest.php

phptest.php Information

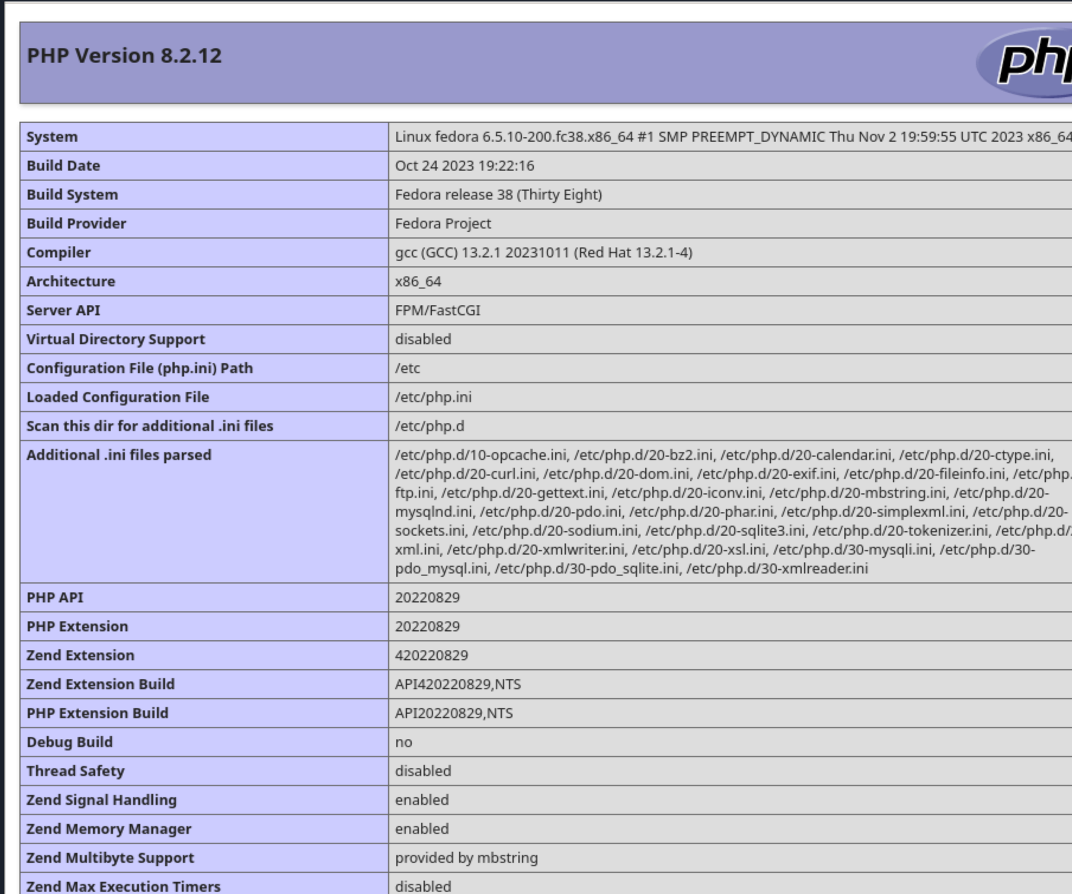
Description:

Phptest.php is run to utilize the `phpinfo() function.` The `phpinfo()` function is used to display comprehensive information about the PHP environment and configuration settings. When called within a PHP script, it generates a detailed report that includes information such as PHP version, server information, system paths, configuration directives, installed extensions, and more. This function is often employed for debugging purposes or to gather specific details about the PHP installation on a server.

phptest.php Code Screenshot



```
GNU nano 7.2                    phptest.php

<?php phpinfo(); ?>
```



PHP Version 8.2.12

System	Linux fedora 6.5.10-200.fc38.x86_64 #1 SMP PREEMPT_DYNAMIC Thu Nov 2 19:59:55 UTC 2023 x86_64
Build Date	Oct 24 2023 19:22:16
Build System	Fedora release 38 (Thirty Eight)
Build Provider	Fedora Project
Compiler	gcc (GCC) 13.2.1 20231011 (Red Hat 13.2.1-4)
Architecture	x86_64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/10-opcache.ini, /etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php. ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-mbstring.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-simplexml.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sodium.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/ xml.ini, /etc/php.d/20-xmlwriter.ini, /etc/php.d/20-xsl.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini, /etc/php.d/30-xmlreader.ini
PHP API	20220829
PHP Extension	20220829
Zend Extension	420220829
Zend Extension Build	API420220829,NTS
PHP Extension Build	API20220829,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
Zend Max Execution Timers	disabled

Connect.php page

Connect Page Information

Description:

This code is the ground work on which the rest of the website runs. Put simply, this code establishes a secure connection to my MySQL database using PDO with specified parameters and options, and it ensures proper error handling in case the connection cannot be established.

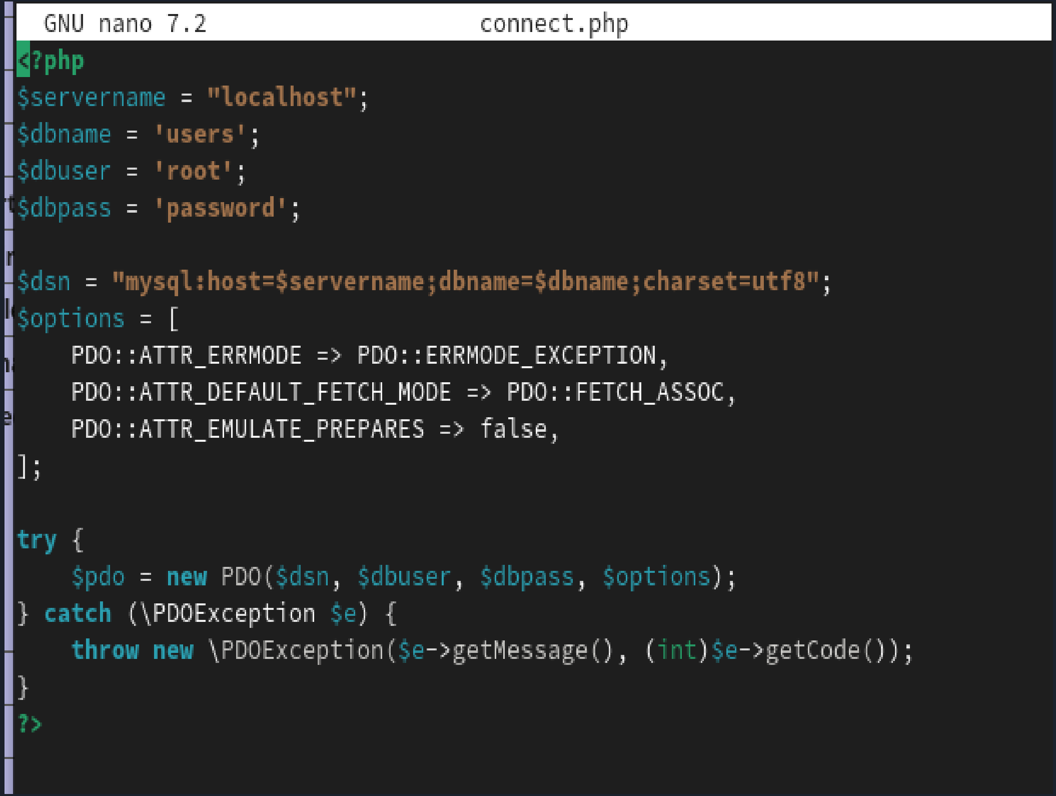
Key Points about the Page:

-This page is vital to the website running smoothly.

-It allows for fetching data from the database.

PHP Code Screenshot

- Variables are defined.
- Specifies the type of database, host, database name, and character set.
- Configure PDO connection and set error handling.
- Attempts to create new PDO instance.



```php
<?php
$servername = "localhost";
$dbname = 'users';
$dbuser = 'root';
$dbpass = 'password';

$dsn = "mysql:host=$servername;dbname=$dbname;charset=utf8";
$options = [
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES => false,
];

try {
    $pdo = new PDO($dsn, $dbuser, $dbpass, $options);
} catch (\PDOException $e) {
    throw new \PDOException($e->getMessage(), (int)$e->getCode());
}
?>
```

Account Creation Web Page

Account Creation Page Information

Description:

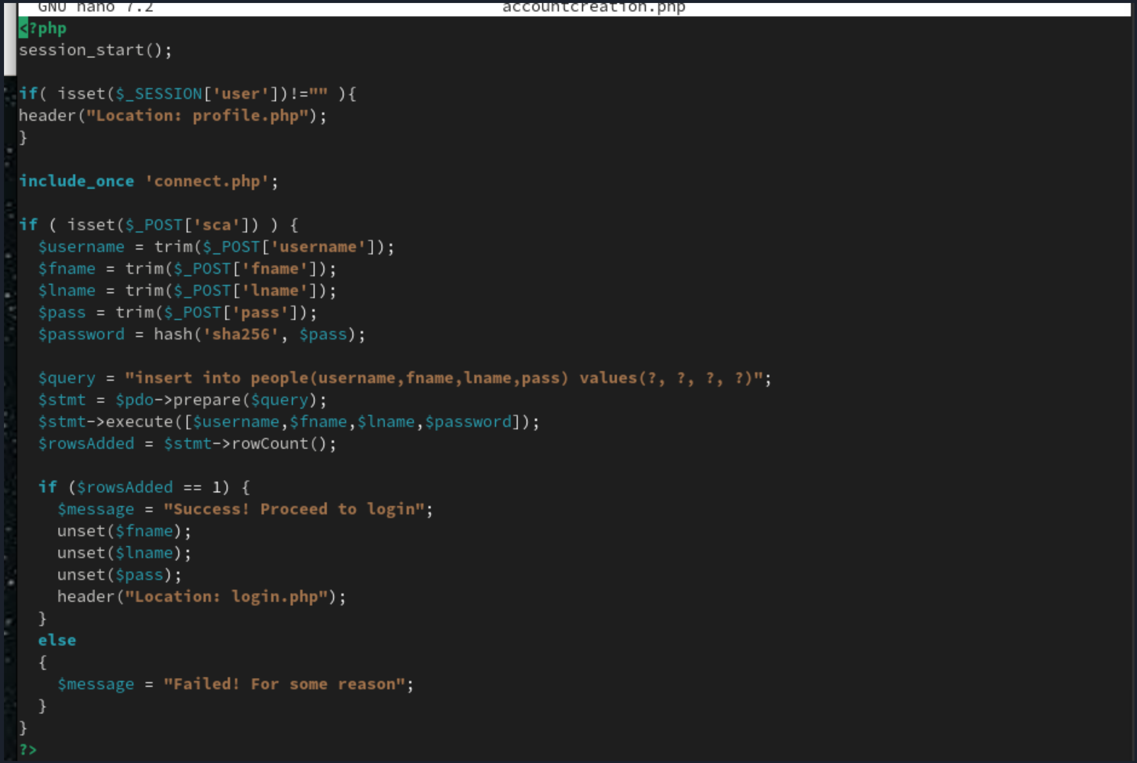
This page is vital to registering users to the website so that they may access their own personal, private page. The code of this page creates a user registration form with PHP handling form submission, connecting to a database, and processing the registration. The HTML section contains the form and styling, while JavaScript provides client-side interactivity for the password input field.

Key Points about the Page:

-The password length requirement code was very tricky.

PHP Code Screenshot

- **PHP Section:**
- `session_start();`: Initializes or resumes a session.
- `if( isset($_SESSION['user'])!="" ){...}`: Checks if a user is already logged in (as indicated by the 'user' session variable) and redirects them to the 'profile.php' page if true.
- `include_once 'connect.php';`: Includes the ('connect.php') file.
- Form Submission Handling:
  - Retrieves and sanitizes form data (username, first name, last name, and password) submitted via POST.
  - Hashes the password using SHA-256.
  - Prepares and executes an SQL query to insert user data into the 'people' table.
  - Checks the number of rows affected (`$rowsAdded`) to determine if the registration was successful.
  - If successful, sets a success message, clears form input values, and redirects to the login page. If unsuccessful, sets a failure message.

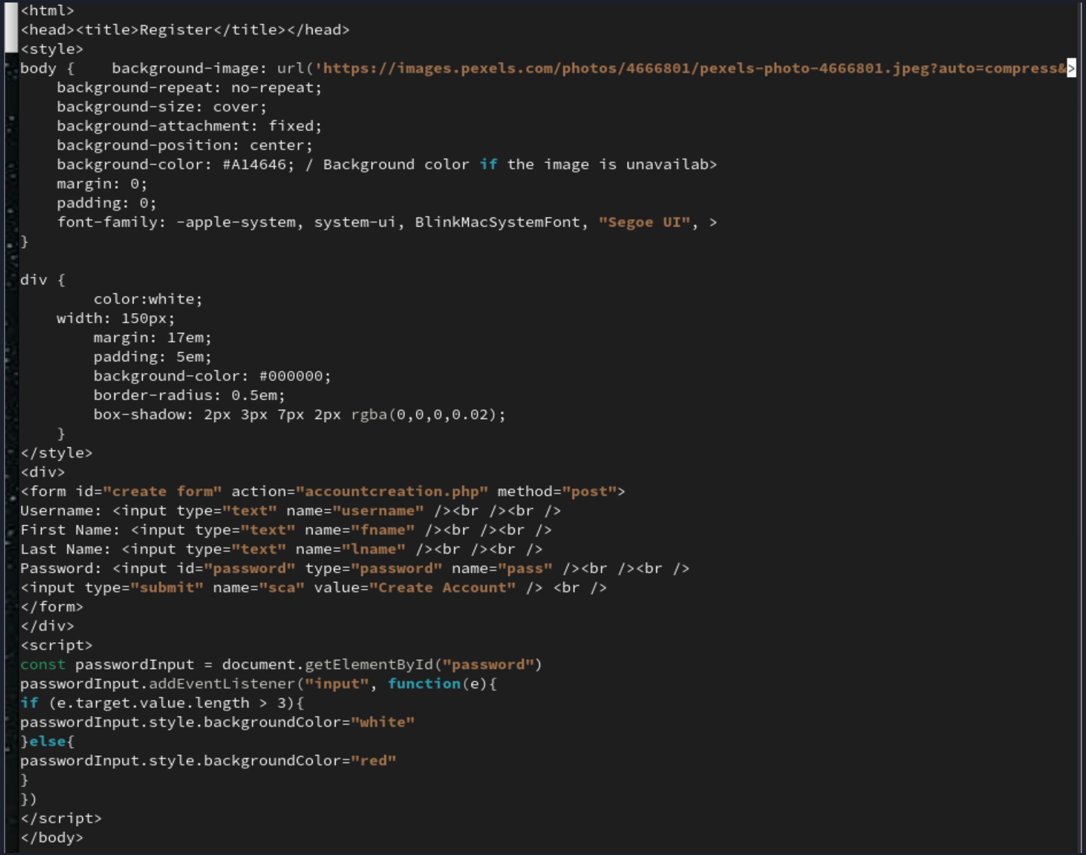


```
  GNU nano 7.2                                        accountcreation.php
<?php
session_start();

if( isset($_SESSION['user'])!="" ){
header("Location: profile.php");
}

include_once 'connect.php';

if ( isset($_POST['sca']) ) {
  $username = trim($_POST['username']);
  $fname = trim($_POST['fname']);
  $lname = trim($_POST['lname']);
  $pass = trim($_POST['pass']);
  $password = hash('sha256', $pass);

  $query = "insert into people(username,fname,lname,pass) values(?, ?, ?, ?)";
  $stmt = $pdo->prepare($query);
  $stmt->execute([$username,$fname,$lname,$password]);
  $rowsAdded = $stmt->rowCount();

  if ($rowsAdded == 1) {
    $message = "Success! Proceed to login";
    unset($fname);
    unset($lname);
    unset($pass);
    header("Location: login.php");
  }
  else
  {
    $message = "Failed! For some reason";
  }
}
?>
```

HTML, Java Script, and CSS Code Screenshot

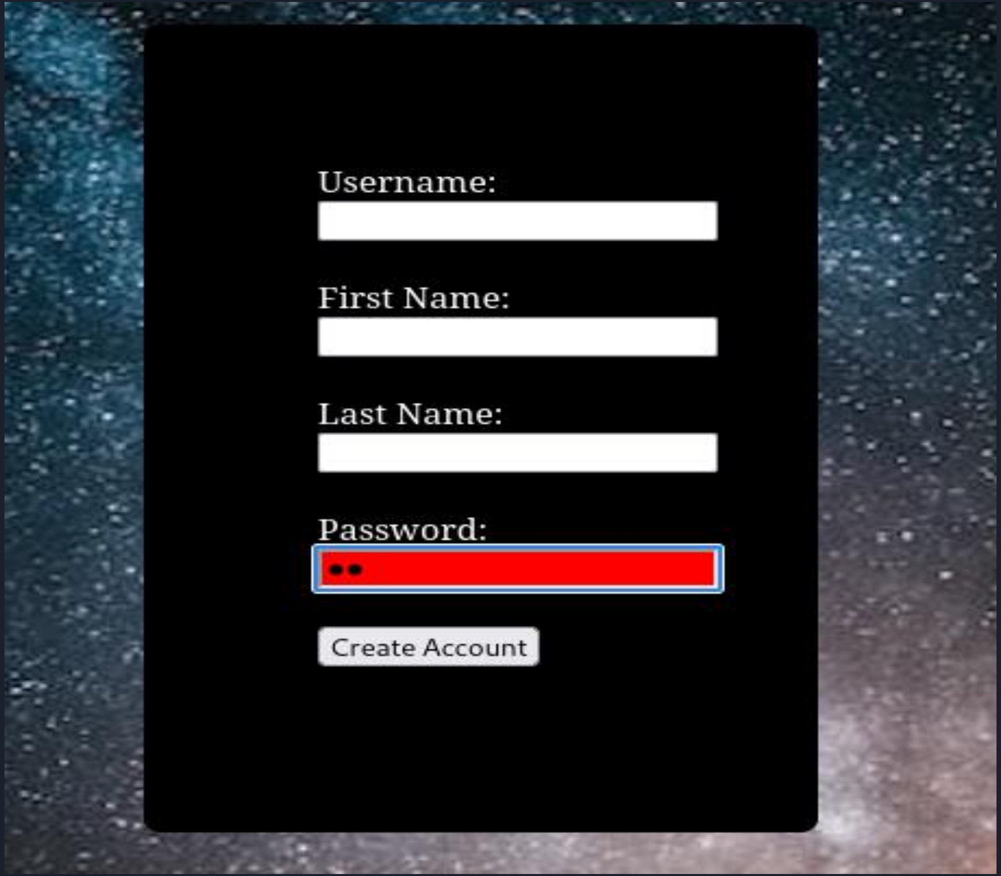
HTML Section:

- Styling:
  - Sets the background image, color, and font for the entire page.
  - Styles a `<div>` element with white text, a black background, and a box shadow.
- `<div>` Element:
  - Contains the registration form.
  - Includes fields for username, first name, last name, and password.
  - Submits the form to the same page ('accountcreation.php') using the POST method.
- JavaScript:
  - Adds an event listener to the password input field.
  - Changes the background color of the password input based on the length of the entered password.



```
<html>
<head><title>Register</title></head>
<style>
body {    background-image: url('https://images.pexels.com/photos/4666801/pexels-photo-4666801.jpeg?auto=compress&>
    background-repeat: no-repeat;
    background-size: cover;
    background-attachment: fixed;
    background-position: center;
    background-color: #A14646; / Background color if the image is unavailab>
    margin: 0;
    padding: 0;
    font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", >
}

div {
        color:white;
    width: 150px;
        margin: 17em;
        padding: 5em;
        background-color: #000000;
        border-radius: 0.5em;
        box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
</style>
<div>
<form id="create form" action="accountcreation.php" method="post">
Username: <input type="text" name="username" /><br /><br />
First Name: <input type="text" name="fname" /><br /><br />
Last Name: <input type="text" name="lname" /><br /><br />
Password: <input id="password" type="password" name="pass" /><br /><br />
<input type="submit" name="sca" value="Create Account" /> <br />
</form>
</div>
<script>
const passwordInput = document.getElementById("password")
passwordInput.addEventListener("input", function(e){
if (e.target.value.length > 3){
passwordInput.style.backgroundColor="white"
}else{
passwordInput.style.backgroundColor="red"
}
})
</script>
</body>
```

Account Creation Webpage Screenshot



Username:

First Name:

Last Name:

Password: ••

Login Webpage

Login Page Information

Description:

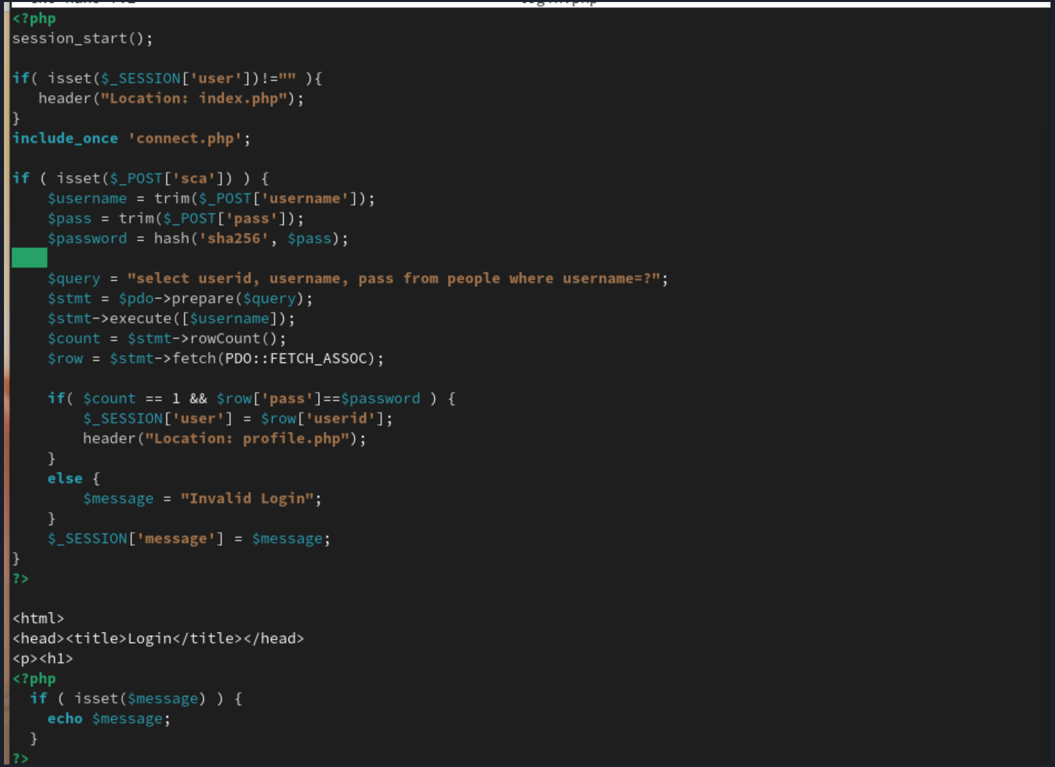
This page allows users who come from the index to login to their individual, private, pages. Upon form submission, it checks the entered credentials against the database. If the login is successful, it sets a session variable and redirects to the profile page. Otherwise, it displays an "Invalid Login" message.

Key Points about the Page:

-This page is pretty straight forward, both technically and user experience.

PHP Code Screenshot

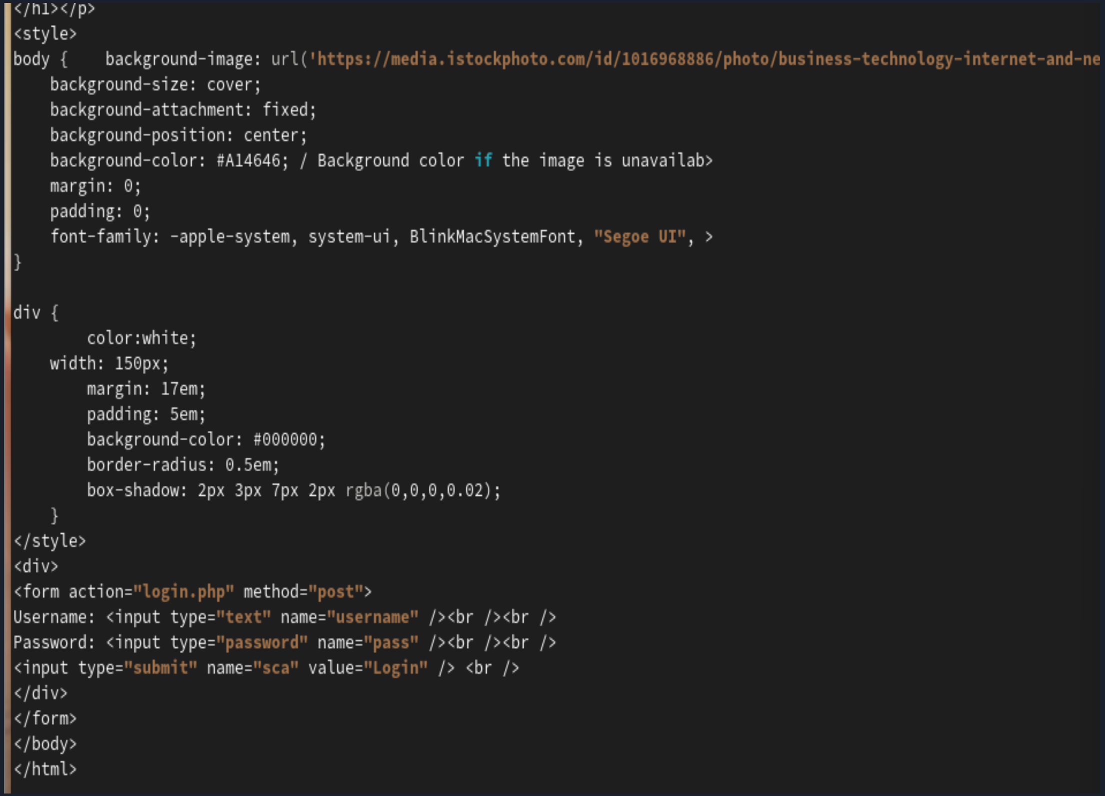
- **PHP Section:**
- `session_start();`: Initializes or resumes a session.
- `if( isset($_SESSION['user'])!="" ){...}`: Checks if a user is already logged in (as indicated by the 'user' session variable) and redirects them to the 'index.php' page if true.
- `include_once 'connect.php';`: Includes the file ('connect.php') file.
- Form Submission Handling:
    - Retrieves and sanitizes form data (username and password) submitted via POST.
    - Hashes the entered password using SHA-256.
    - Prepares and executes an SQL query to select user data from the 'people' table based on the provided username.
    - Checks the result: if a user is found with the correct password, it sets the 'user' session variable and redirects to the 'profile.php' page. Otherwise, it sets an "Invalid Login" message.



```php
<?php
session_start();

if( isset($_SESSION['user'])!="" ){
   header("Location: index.php");
}
include_once 'connect.php';

if ( isset($_POST['sca']) ) {
    $username = trim($_POST['username']);
    $pass = trim($_POST['pass']);
    $password = hash('sha256', $pass);

    $query = "select userid, username, pass from people where username=?";
    $stmt = $pdo->prepare($query);
    $stmt->execute([$username]);
    $count = $stmt->rowCount();
    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    if( $count == 1 && $row['pass']==$password ) {
        $_SESSION['user'] = $row['userid'];
        header("Location: profile.php");
    }
    else {
        $message = "Invalid Login";
    }
    $_SESSION['message'] = $message;
}
?>

<html>
<head><title>Login</title></head>
<p><h1>
<?php
  if ( isset($message) ) {
    echo $message;
  }
?>
```

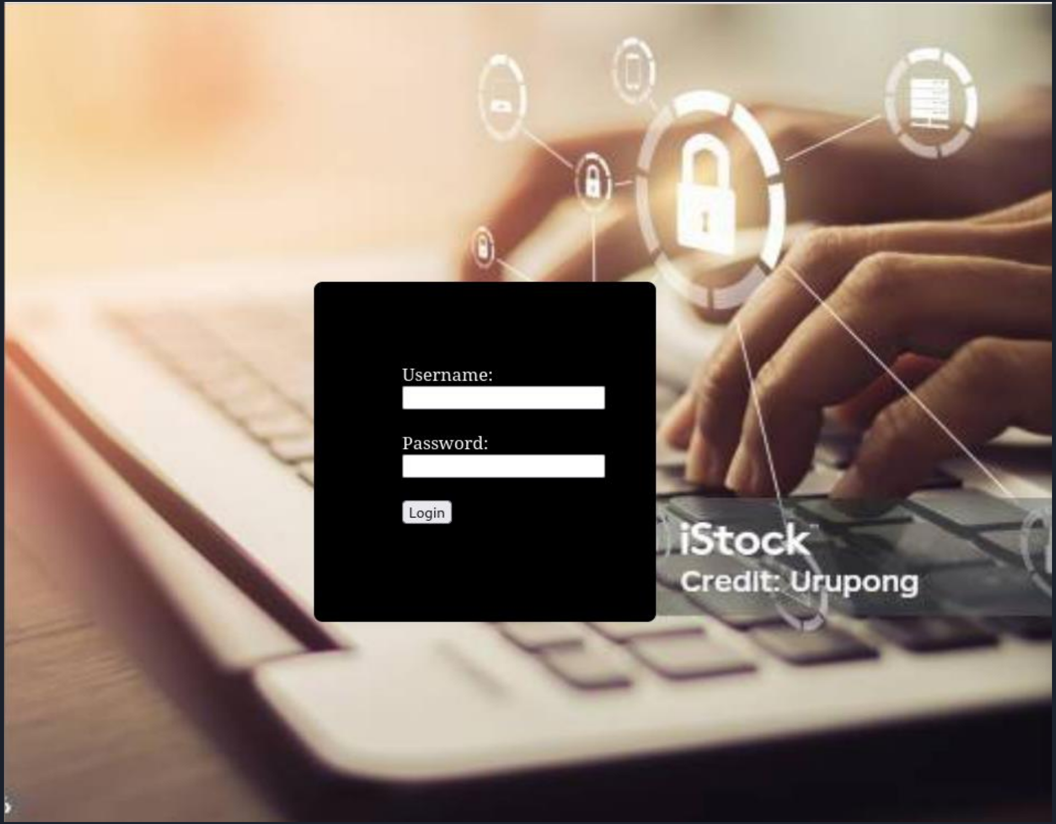
HTML & CSS Code Screenshot

- **HTML Section:**
- Styling:
  - Sets the background image, color, font, and styling for a `<div>` element.
- `<div>` Element:
  - Contains the login form.
  - Styled with white text, a black background, border-radius, and a box shadow.



```
</h1></p>
<style>
body {    background-image: url('https://media.istockphoto.com/id/1016968886/photo/business-technology-internet-and-ne
    background-size: cover;
    background-attachment: fixed;
    background-position: center;
    background-color: #A14646; / Background color if the image is unavailab>
    margin: 0;
    padding: 0;
    font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", >
}

div {
        color:white;
    width: 150px;
        margin: 17em;
        padding: 5em;
        background-color: #000000;
        border-radius: 0.5em;
        box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
</style>
<div>
<form action="login.php" method="post">
Username: <input type="text" name="username" /><br /><br />
Password: <input type="password" name="pass" /><br /><br />
<input type="submit" name="sca" value="Login" /> <br />
</div>
</form>
</body>
</html>
```

PHP Webpage Screenshot



Username:

Password:

Profile.php page

Profile Page Information

Description:

The user's home page that they have accessed through either the login or account creation page. This page, when completed, is where the user can navigate to shop, do banking, look at social media, watch videos, or play games all depending on the type of website.

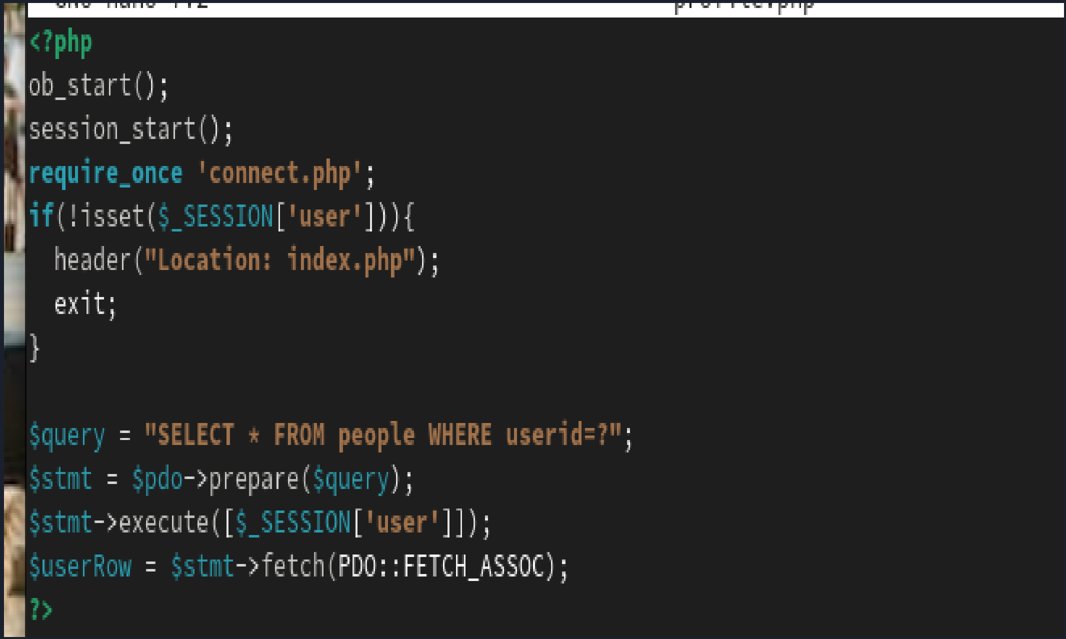
Key Points about the Page:

-Once the user is logged in, this page fetches and displays their information. The page has styling and includes options for an "administrator" user to access an "EDIT" page and for any user to log out. Output buffering is used to manage the order in which content is sent to the browser.

PHP Code Screenshot

PHP Section:

- ob_start();: Turns on output buffering.
- session_start();: Initializes or resumes a session.
- require_once 'connect.php';: Includes the ('connect.php') file.
- Check User Session:
  - Redirects to the 'index.php' page if the user is not logged in.
  - If the user is logged in, it proceeds to fetch user data from the database based on the 'userid' stored in the session.

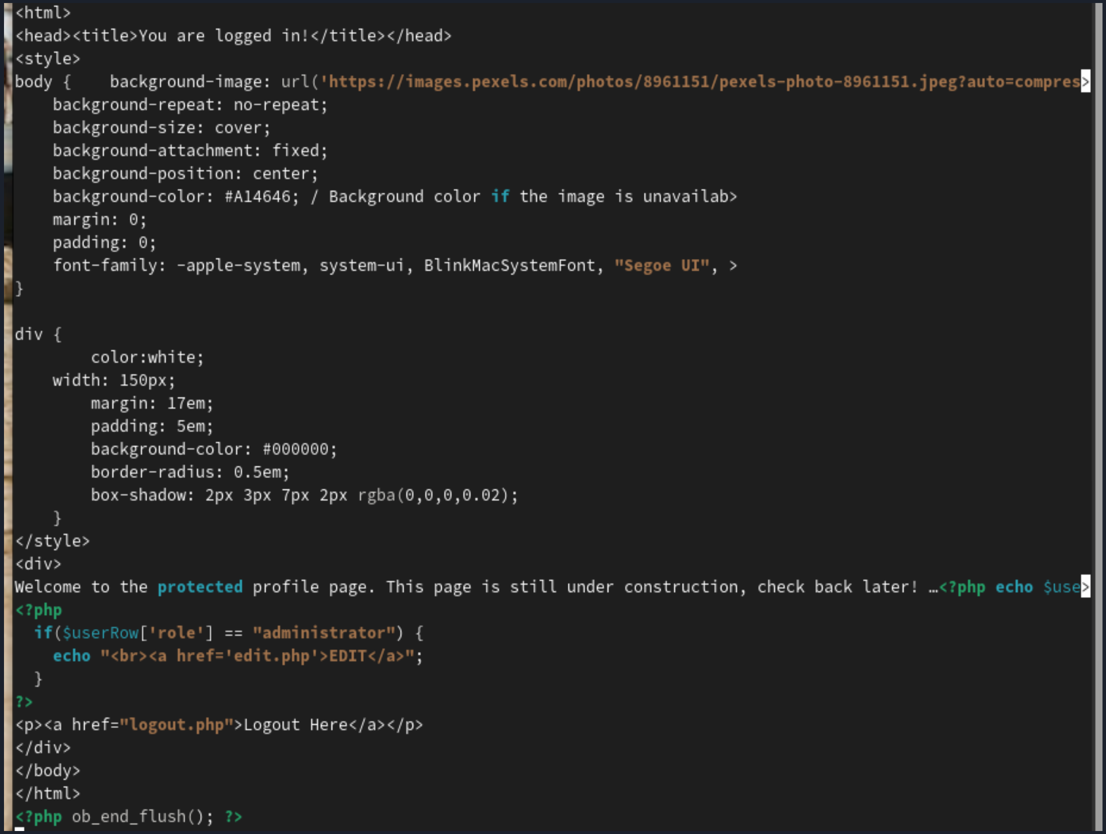


```php
<?php
ob_start();
session_start();
require_once 'connect.php';
if(!isset($_SESSION['user'])){
  header("Location: index.php");
  exit;
}

$query = "SELECT * FROM people WHERE userid=?";
$stmt = $pdo->prepare($query);
$stmt->execute([$_SESSION['user']]);
$userRow = $stmt->fetch(PDO::FETCH_ASSOC);
?>
```

HTML & CSS Code Screenshot

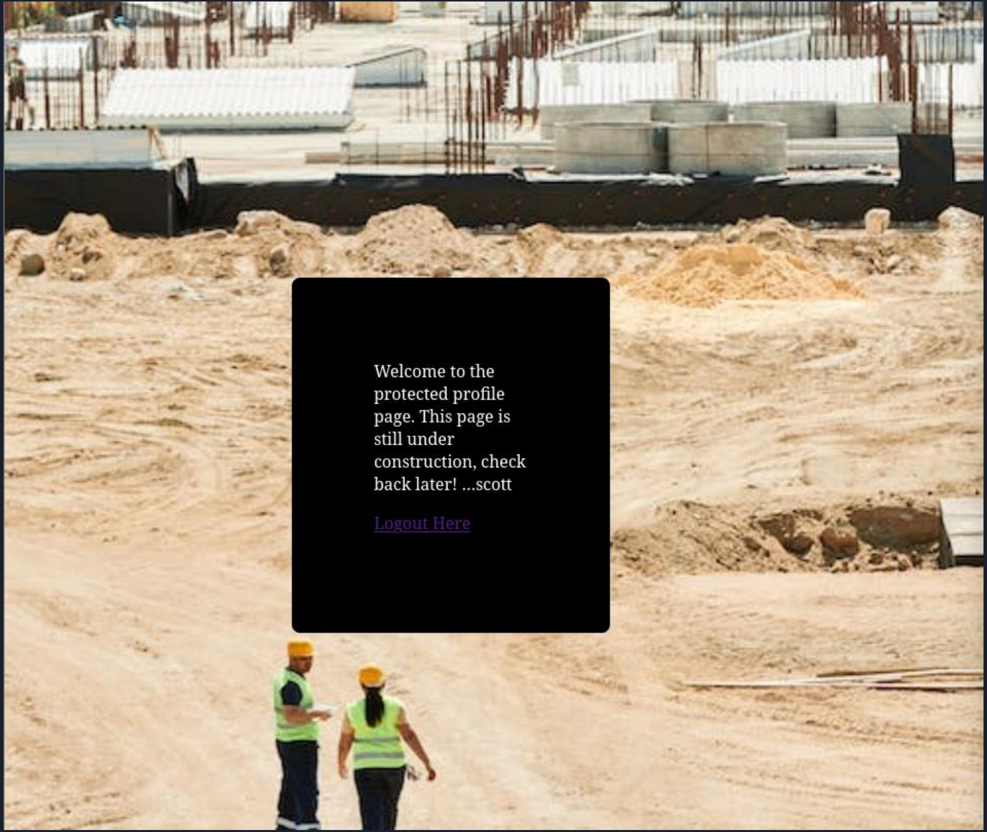
HTML Section:

- Styling:
  - Sets the background image, color, font, and styling for a `<div>` element.
- `<div>` Element:
  - Contains a welcome message, telling the user that the page is under construction.
  - Displays user information retrieved from the database.
  - If the user has an "administrator" role, it provides a link to an "EDIT" page.
  - Includes a logout link that directs to the 'logout.php' page.
- `<?php ob_end_flush(); ?>`: Flushes the output buffer, sending the buffered output to the browser. This is typically done at the end of the script.



```
<html>
<head><title>You are logged in!</title></head>
<style>
body {    background-image: url('https://images.pexels.com/photos/8961151/pexels-photo-8961151.jpeg?auto=compres>
    background-repeat: no-repeat;
    background-size: cover;
    background-attachment: fixed;
    background-position: center;
    background-color: #A14646; / Background color if the image is unavailab>
    margin: 0;
    padding: 0;
    font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", >
}

div {
        color:white;
    width: 150px;
        margin: 17em;
        padding: 5em;
        background-color: #000000;
        border-radius: 0.5em;
        box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
</style>
<div>
Welcome to the protected profile page. This page is still under construction, check back later! …<?php echo $use>
<?php
  if($userRow['role'] == "administrator") {
    echo "<br><a href='edit.php'>EDIT</a>";
  }
?>
<p><a href="logout.php">Logout Here</a></p>
</div>
</body>
</html>
<?php ob_end_flush(); ?>
```

Profile Page Screenshot



Welcome to the protected profile page. This page is still under construction, check back later! ...scott

Logout.php page

Logout Page Information

Description:

This page is important because it allows the user to logout and exit their browsing session. Without this page, there would be no clear, safe, way to end a browsing session on a web page.

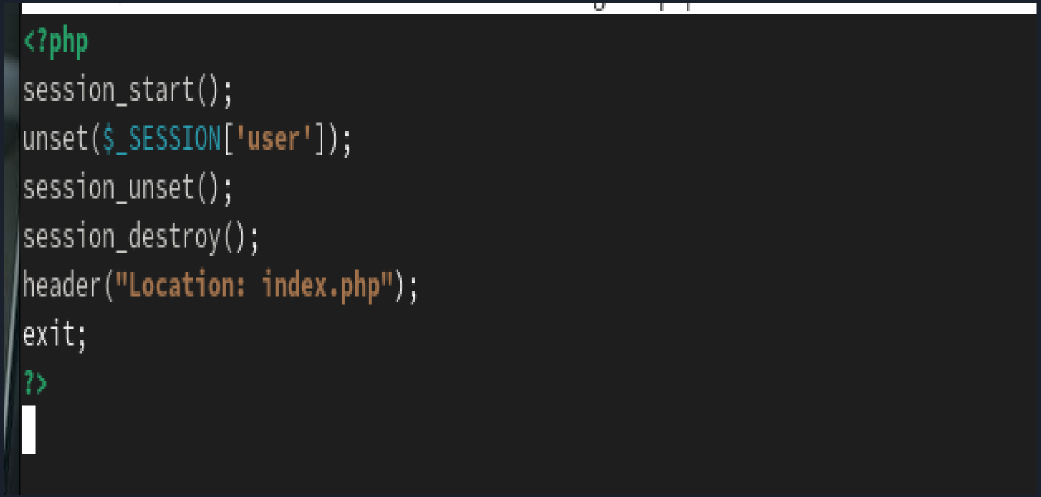
Key Points about the Page:

-This page performs the necessary steps to log out a user.

-It unsets the specific 'user' session variable, clears all session variables, destroys the session, redirects the user to the 'index.php' page, and terminates the script.

PHP Code Screenshot

- `session_start();`: Initializes or resumes a session.
- `unset($_SESSION['user']);`: Unsets the 'user' session variable. This removes the specific variable from the session data.
- `session_unset();`: Unsets all session variables. This function is used to ensure that all session variables are removed.
- `session_destroy();`: Destroys the session. This removes all session data from the server, effectively ending the user's session.
- `header("Location: index.php");`: Redirects the user to the 'index.php' page after the session is destroyed.
- `exit;`: Terminates the script immediately, preventing any further code execution.



```php
<?php
session_start();
unset($_SESSION['user']);
session_unset();
session_destroy();
header("Location: index.php");
exit;
?>
```

404 Error Page

404 Error Page Page Information

Description:

This page represents that the user has reached a domain they requested, however, there is no provided information on that page. Thus, this issue prompts the 404 not found page.

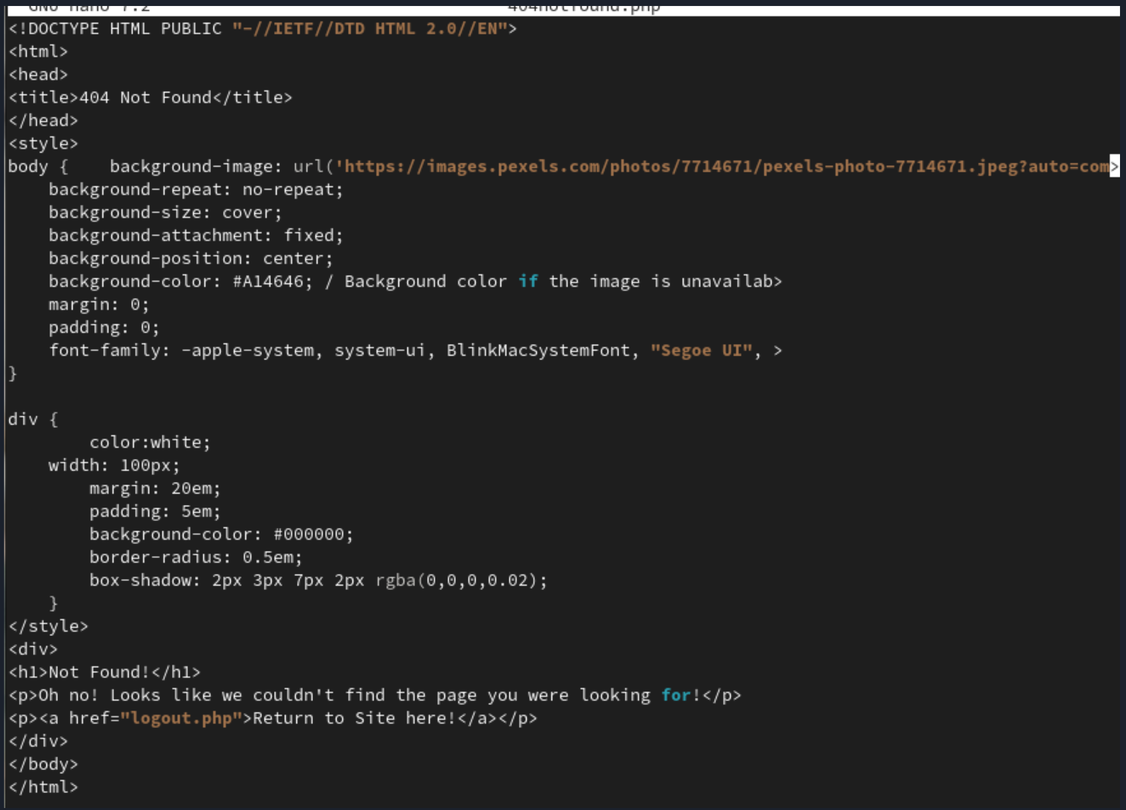
Key Points about the Page:

-This page is important in letting users know that where they wanted to go has no information available to them.

-It's also important that this page allows users to navigate back to the index page.

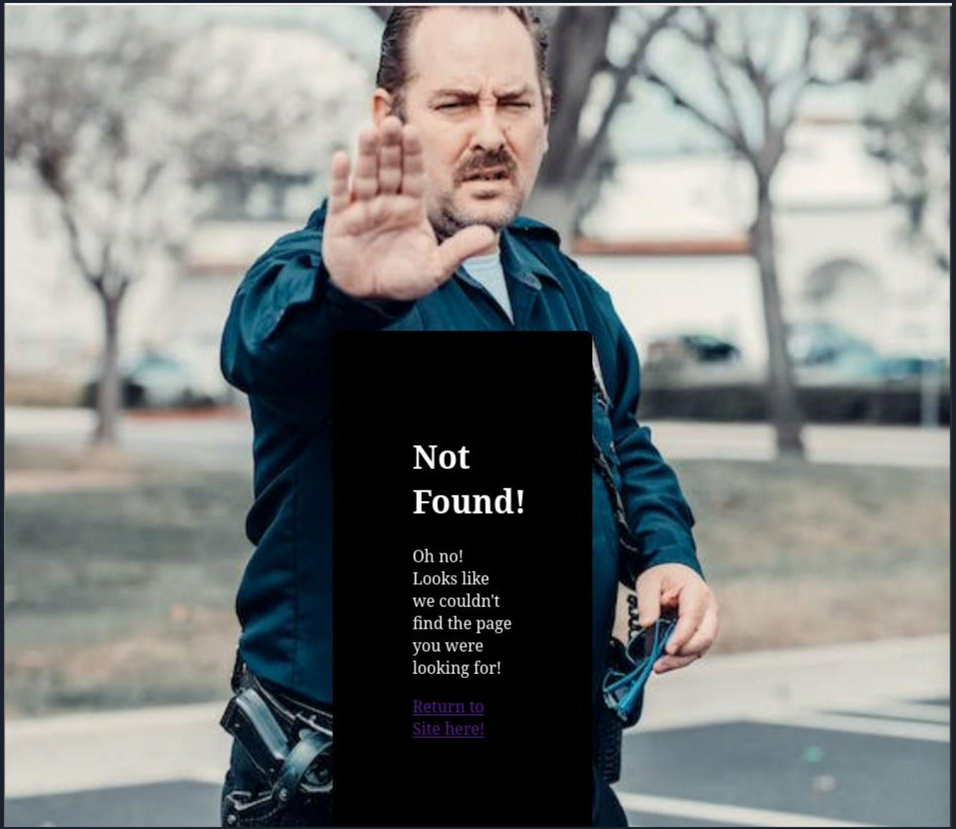
PHP Code Screenshot

- **HTML Section:**
- `<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">`: Declares the HTML document type and version.
- `<html>`: Begins the HTML document.
- `<head>`: Contains metadata about the HTML document.
- `<title>404 Not Found</title>`: Sets the title of the document to "404 Not Found."
- `<style>`: Defines the internal CSS styles for the page.
- An `<h1>` heading with the text "Not Found!" and two `<p>` paragraphs with error-related messages.
- Provides a link (`<a>`) to "Return to Site here!" that directs to the 'logout.php' page.



```
  GNU nano 7.2                                    404notfound.php
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
<head>
<title>404 Not Found</title>
</head>
<style>
body {    background-image: url('https://images.pexels.com/photos/7714671/pexels-photo-7714671.jpeg?auto=com>
    background-repeat: no-repeat;
    background-size: cover;
    background-attachment: fixed;
    background-position: center;
    background-color: #A14646; / Background color if the image is unavailab>
    margin: 0;
    padding: 0;
    font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", >
}

div {
        color:white;
    width: 100px;
        margin: 20em;
        padding: 5em;
        background-color: #000000;
        border-radius: 0.5em;
        box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
</style>
<div>
<h1>Not Found!</h1>
<p>Oh no! Looks like we couldn't find the page you were looking for!</p>
<p><a href="logout.php">Return to Site here!</a></p>
</div>
</body>
</html>
```

PHP Webpage Screenshot



Not Found!

Oh no!
Looks like we couldn't find the page you were looking for!

Return to Site here!

Questions?