Development:

Firstly I set up my website using Unicontroller. Originally I was going to use XAMPP, but it didn’t work. Because I am using Unicontroller I can host my website locally while developing my website. This program allow me to create a database for my project while also allowing. After I have completed this project I will move my project to a global host so that users will be able to access it.

A screenshot of a computer

Description automatically generated

Firstly I created a folder called “coursework”. All of the files for the website will go into this folder. I also created a page called “index.html”. In the future this page will become the homepage for the website.

A screen shot of a computer code

Description automatically generated

When I go to the url “127.0.0.1/computingcoursework” it will automatically send me to “127.0.0.1/computingcoursework.html”. The next thing that I have to do is to add buttons that will allow the user to go the page that they want to go to.

The pages that will have to be linked to the homepage are the userlogin page, the sitterlogin page, the user create account page and the sitter create account page.

A white background with black dots

Description automatically generated

To add the method that allows the user to go to the next page I added buttons that are programmed to send the user to a certain url.

A computer screen shot of text

Description automatically generated

In this webpage I tried put each of the buttons in certain places in the page. To do this I created containers for each of the buttons, I also created classes in the stylesheet , then I assigned classes to each of the containers. But currently it isn’t working. The reason why the classes that I assigned to the containers aren’t working is because I haven’t created the classes in the stylesheet yet

A screenshot of a computer program

Description automatically generated

This is the code for the stylesheet. So now each of the buttons should be moved into a specific place that is determined by the class that I have assigned to their containers. Putting the code into a separate stylesheet means that I will be able to use these classes on other pages without having to rewrite the code.

A white rectangular object with black text

Description automatically generated

Currently the class top left and centre left are working, but the classes top right, centre right and signup are not working. To correct these problems I will check through the code in the stylesheet to make sure that there are no problems with the syntax of the code

A screen shot of a computer program

Description automatically generated

The problems that I found in the code of the stylesheet were that I had misspelt orange, which caused the buttons to not have the correct colour, I also had wrong value for left in the topright class, which caused the sitter login button to be too far to the right

A screenshot of a computer

Description automatically generated

Now all of the buttons are in the correct place and the webpage looks right I will have to validate that all of the buttons go to the correct webpage when they are clicked. To do this I will create a page for user login, sitter login, sitter signup and user signup. Although these pages won’t have any functionality at this point it will be a good way to check if the buttons send the user to the correct page.

A screenshot of a computer

Description automatically generated

Here is the screenshot of one of the pages that I will use to validate the functionality of the buttons, when I click on the user login button I will be sent to this page and the text in the body will tell me that I am on the correct page. I have created a webpage like this for all of the pages that the buttons will send the user to

When I clicked on the user login page it sent me to this page

A black text on a white background

Description automatically generated

This shows that the url in the user login button is correct because it sent me to the user login webpage

When I clicked on the sitter login page it sent me to this page.

A black text on a white background

Description automatically generated

When I clicked on the sitter signup page it sent me to this page.

A close-up of a sign up page

Description automatically generated

When I clicked on the user signup page it sent me to this page

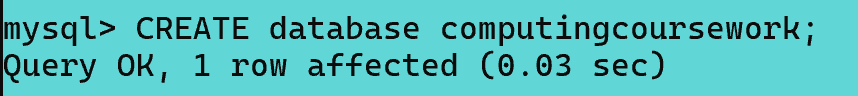
A black text on a white background

Description automatically generated

Now that I have validated that all of the buttons send the user to the correct page I will start developing the next stage of the project. Now I will create a page that will connect the program to the database.

To do this I will create a page called “connection.php”, which will connect the website to the database.

Firstly I need to create the database.In this SQL statement I created a database, called “computingcoursework” which will contain all of the tables that I will need for the project



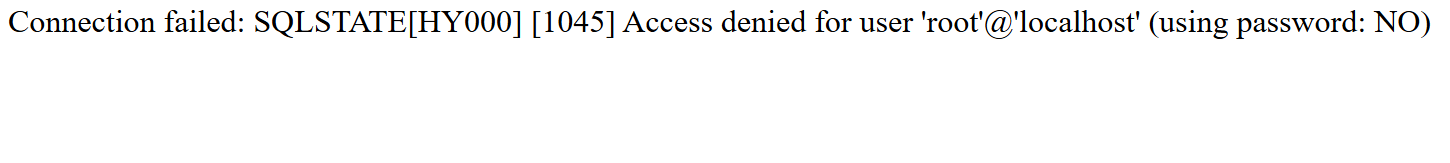
Now I have to create the connection.php page.

A screenshot of a computer program

Description automatically generated

This page will allow the website to connect to the database. To validate that this page works I will create a button that goes on index.html and when it is pressed it will send me to connection.php. If the page works the page connection.php should say “Connected successfully”, but if the page doesn’t work it should say “Connected successfully”. Although the button that I am creating won’t be on the final version of the “index.html” page it will be a good way to test if the connections page is working.

When I click on the button connection.php page it sends me to this page.



This because this page doesn’t say “Connected successfully” it means that there is a problem in the code. The problem in the code of the connection.php page is that the password variable at the top of the page is blank, which means when the SQL statement is entered into the database, the database rejects it because the database has a password, which is “123”. To fix this problem I will change the value of the password variable from “ “ to “123”.

A screenshot of a computer program

Description automatically generated

Now when I click on the button that sends me to the connection.php page the webpage is this.

A white background with black dots

Description automatically generated

This test shows that the connection.php page now works correctly.

The next thing that I will do is create a way for the user to create an account. To do this I will create two pages, one that allows user to sign up as clients, and another page that allows sitters to sign up. I will also have to create two tables, one of the tables will hold the sitters data, and the other table will hold all of the clients data.

This is the sql statement that I will use to create the table:

CREATE TABLE users

(userid INT(4) UNSIGNED AUTO\_INCREMENT PRIMARY KEY, clientname VARCHAR(30)

NOT NULL, firstname VARCHAR(30) NOT NULL, surname VARCHAR(30) NOT NULL,

email VARCHAR(30) NOT NULL, emergencycontact VARCHAR(30) NOT NULL)"

Here is the description of the table “users”:

A blue screen shot of a computer screen

Description automatically generated

To create the table that holds all of the cat sitters data I will do the same thing that I did to create the table for the client table.

Here is the SQL statement that I used to create the sitter table:

CREATE TABLE sitters

(sitterid INT(4) UNSIGNED AUTO\_INCREMENT PRIMARY KEY, sittername VARCHAR(30)

NOT NULL, firstname VARCHAR(30) NOT NULL, surname VARCHAR(30) NOT NULL,

email VARCHAR(30) NOT NULL, emergencycontact VARCHAR(30) NOT NULL);

Here is the description of the sitters table:

A blue paper with black text

Description automatically generated

Now I will create the page “users.php”. This page will allow the user to enter their username, first name, surname, email and emergency contact information, then it will send this data to another page that will enter these details into the users table.

Firstly I will create all of the places that the user will input data into the users page to make sure that they all work. The users.php webpage will also assign each of the inputs to a variable, so that they can later be inserted into the users table.

Current code for users.php:

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Now that I know that all of the inputs for this page appear correctly I will code the part of the page that sends the data inputted into the fields to another page.

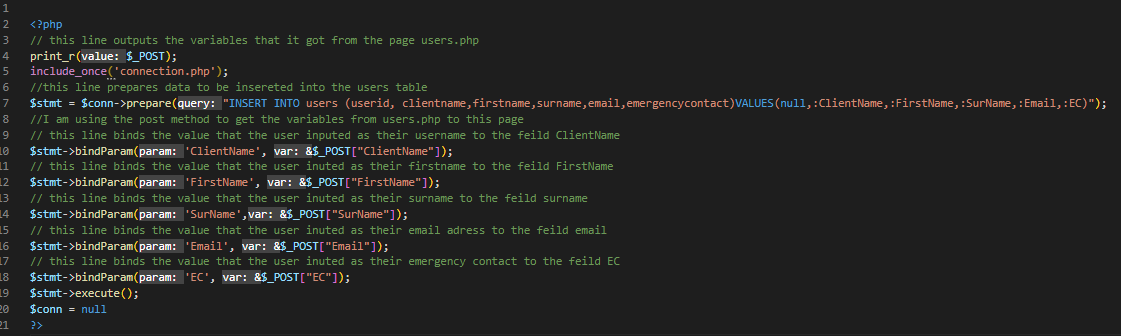
A computer screen shot of a program code

Description automatically generated

The code that I added selects all of the values from the users table and prepares to add it to the table, and it also connects the page to the users table.

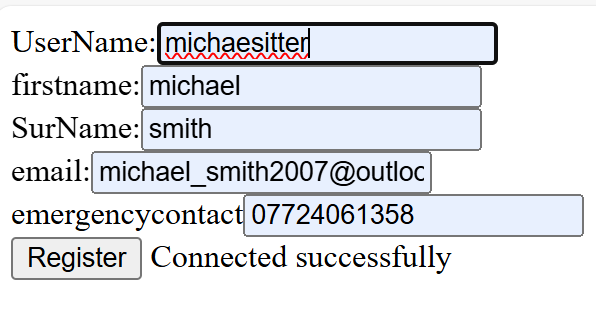
Now I will have to create the page that actually inputs the data from the fields into the table. This webpage will be called “adduser.php”.

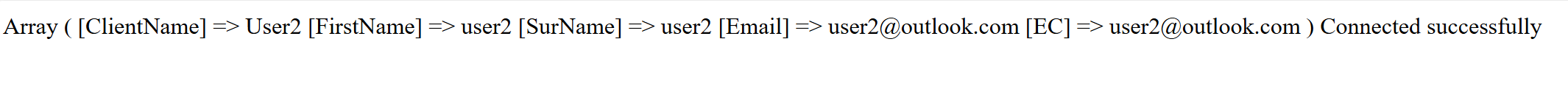
Here is the screenshot to “addusers.php”:



The page “addusers.php” inserts data into the users table. To do this it first connects to the database, then it prepares to insert the data into the table. To get the data from the inputs from users.php it uses the post method, which allows it to get the variables from users.php. I am also usint the $\_POST method so it outputs the variables that it got from users.php to the page addusers.php. Now I will input data into users.php and when the data is sent to addusers.php it will be outputted to the webpage.

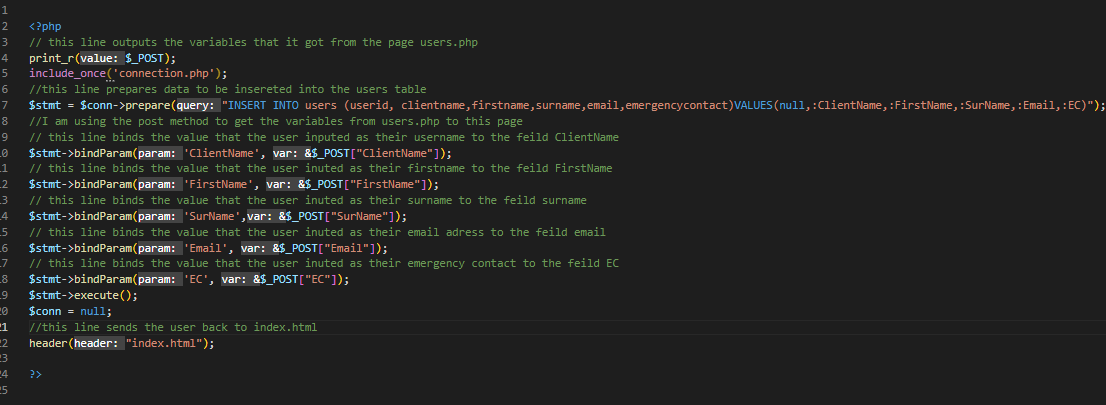
Here is the screenshot of the data that I inputted into users.php:

  
here is the screenshot of the page addusers.php after data has been inputted:



This shows that all of the data that is being inputted into the page users.php is being sent to the page adduser.php. Although in the current form of this page it is easy to tell wether or not the page is working correctly there is currently no way for the user to get to another page. To fix this problem I will add a section of code that will send the user to another page.

Here is the screenshot of the edited code:



To send the user to another page after they have created their account I added the line “header(“index.html”); this line will send the user back to the index.html page. I chose this page to send the user back to because the index.html page allows the user to do a variety of actions.

To validate this page I will enter some data into the fields. If the two webpages both work then I will be able to check the users table and data should be added. If the something in one of the two pages doesn’t work when I check to see if data has been added to the users table there should be nothing there

Here is a screenshot of the data being entered into the database:

A close-up of a contact list

Description automatically generated

A blue background with black text

Description automatically generated

This result shows me that the two pages can work together to enter data into the database. Now I will have to validate whether or not the webpages will be able to recognise if a field that should have data in it doesn’t have data in it.

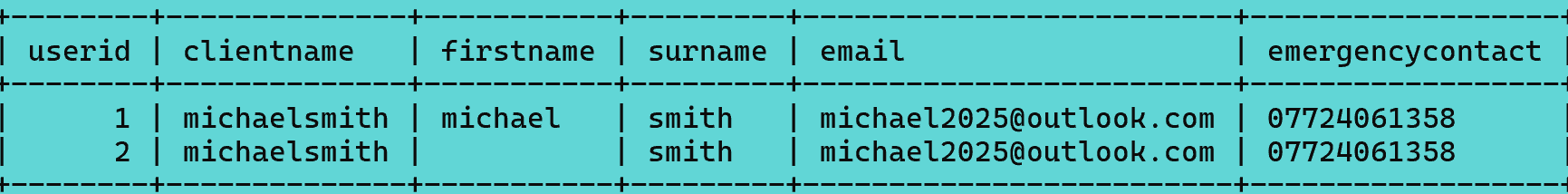
Here is a screenshot of the data being entered into the table.

A close-up of a contact us

Description automatically generated

In this screenshot I have left the first name field blank. This means that the data shouldn’t be added to the database because first name field isn’t allowed to be null.

Here is the result from my test:



This test shows me that the program has no way to tell whether or not a field that is meant to have data in it is null or not. This is a problem because if a user creates an account without entering their email or emergency contact there will be no way to contact that user if there is a problem.

To try and fix this problem I added this segment of code to the page:

A screen shot of a computer code

Description automatically generated

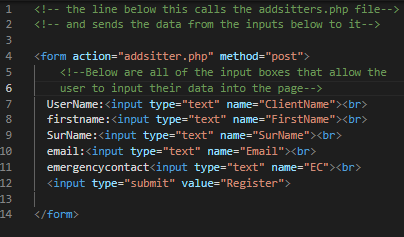
Currently I only have it check to see if the firstname variable has data entered into it. If this code works I will use this method for all of the variables that are inputted

//finish the area that makes sure that all of the variables are filled

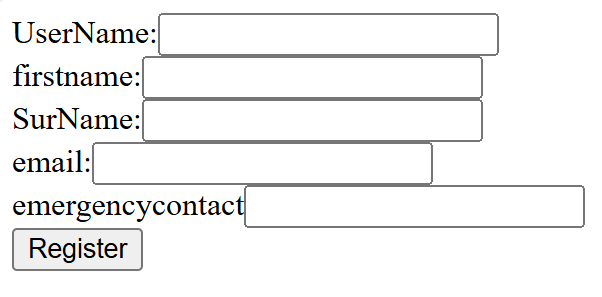
Now that I have created the page that allows the client to create an account I will have to create the page that allows the sitter to create an account. This page will be almost exactly the same as the page that allowed the clients to create an account on the website. But instead of inserting data into the users table it will insert data into the sitters table.

Firstly I will code the page so that it only has inputs. This will make sure that all of the inputs are on the page, and there are no errors in the code of the input boxes.

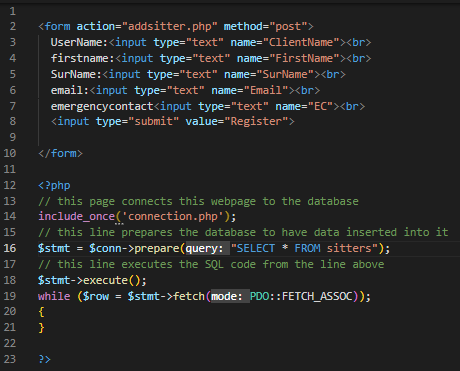
Here is the screenshot of the current code:



Here is the screenshot of the current page.



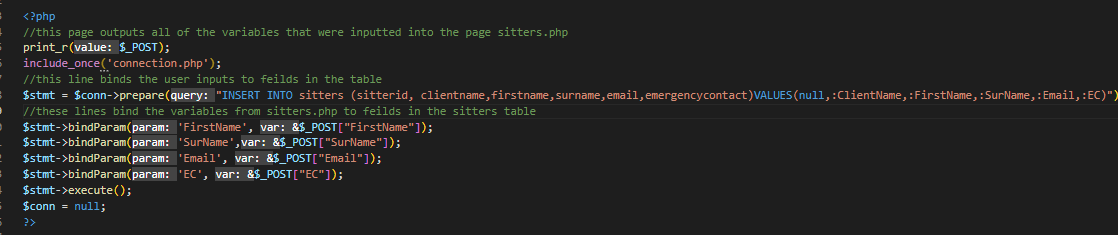
Now that I know that all of the currect inputs are put onto the page I will now code the section that will connect to addsitters.php, which is the page that will actually inert the data from this page into the database.



The code that I have added allows the page sitters.php to connect to the database. It also prepares the table “sitters” to have data inputed into it. Now when data is inputted into this page it should be able to send that data to “addsitter.php”, which is a page that I will create in the future.

Now I will create “addsitters.php”, which is a page that will insert the data from the page “sitters.php” into the database.

Here is the screenshot of addsitters.php:

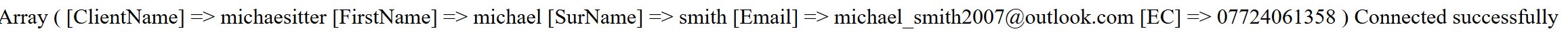


Now that I have created addsitter.php and sitters.php I will be able to validate wether or not sitters are able to create an account for the website. To do this I will enter some data into the webpage “sitters.php”, and when I press the button “submit” the data should be added to the database.

Here is a screenshot of the data that I entered into the “sitters.php” page:



here is the screenshot of the addsitters.php page:



All of the variables from the page sitters.php have been attached to the correct field from the table sitters. The next thing that I will have to do is to add a segment of code to the page addsitters.php that will send the user back to index.html.

A screen shot of a computer program

Description automatically generated

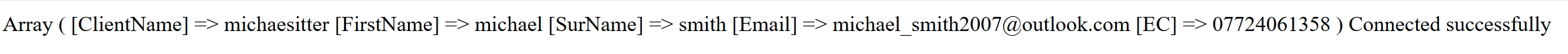
Now when I enter data into the webpage and click the button submit, I should be sent to the webpage addsitters, which will enter all of the details that I put into the inputs on the sitters.php page into the sitters table. After addsitters.php does that I should be rerouted to the index.html page.

Screenshot of data being entered into the sitters.php page:

A screenshot of a computer

Description automatically generated

After I click the register button I get sent to this webpage:



This tells me that there is an error in the code of the page addsitters.php.

I looked through the code of addsitters.php and realised that the line of code that binds the variable clientname to the field ClientName was missing, so I added that line into the code.

A computer screen shot of text

Description automatically generated

Now when I click the register button I get sent to this page:

As a way to check if the code is working up to a certain point I added this line of code to the addusers.php page:



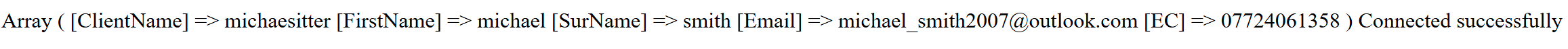
This line of code outputs the sentence “this is still working” if the code reaches that point. If I see that the sentence isn’t outputted to the webpage I will know that the code hasn’t reached that point and there is an error in the code above it.

So now the code on the addusers.php page looks like this

A screenshot of a computer program

Description automatically generated

When I click the submit button I get sent to this page:



This tells me that there is an issue in the code that comes before this point, because otherwise the sentence “this is still working would be outputted to screen”.

I moved the line print\_r($\_POST); around the code to see where which line had a problem in it. This would tell me which line had a problem in it because when it was put after the line with a problem it would stop outputting the variables from the inputs in the page “sitters.php”. After I did this I found the the line “$stmt->execute();” had a problem in it. Because this line had a problem it meant that the lien that prepares to enter the variables into the database has a problem in it. So I looked through the line and found that I had named a field clientname instead of sitter name.

Here is the changed code:



Now after I input the data into the page “sitters.php” I get sent to this page:



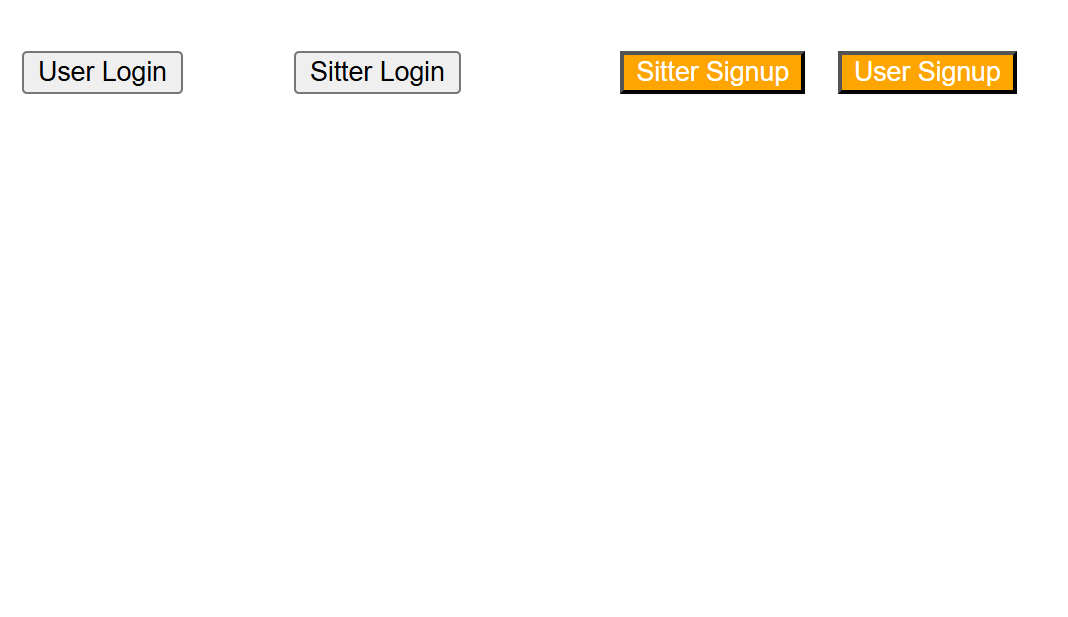
This tells me that all of the code is working properly, otherwise I wouldn’t have been redirected to the index.html page.

Now to validate that this page is working the way that it should I will input data into the page “sitters.php” and then I will check to see what page I am redirected to, and then I will check in the table “sitters” to see if the data that I had inputed into the sitters.php page has been entered into it.

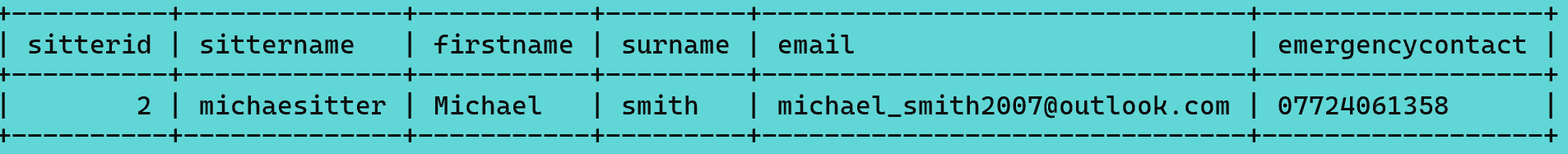
Screenshot of the data being entered into the sitters.php page:



Here is a screenshot of the page that I am redirected to:



Here is a screenshot of the sitters table after I have clicked the register button on sitters.php:



These tests validate the sitters.php page and the addsitter.php page because it shows that both of the pages work together because I am redirected to index.html and the data that I inputted into sitters.php has been entered into the database. This is because if the sitters.php page didn’t work properly the page addsitter.php wouldn’t be able to get the inputs from the sitters.php page, so then the addsitters.php page wouldn’t be able to input the data into the database, and if it had a problem with that the line “$stmt->execute();” wouldn’t work, so the code would stop on the addusers.php page and the user wouldn’t be redirected to the index.php page.

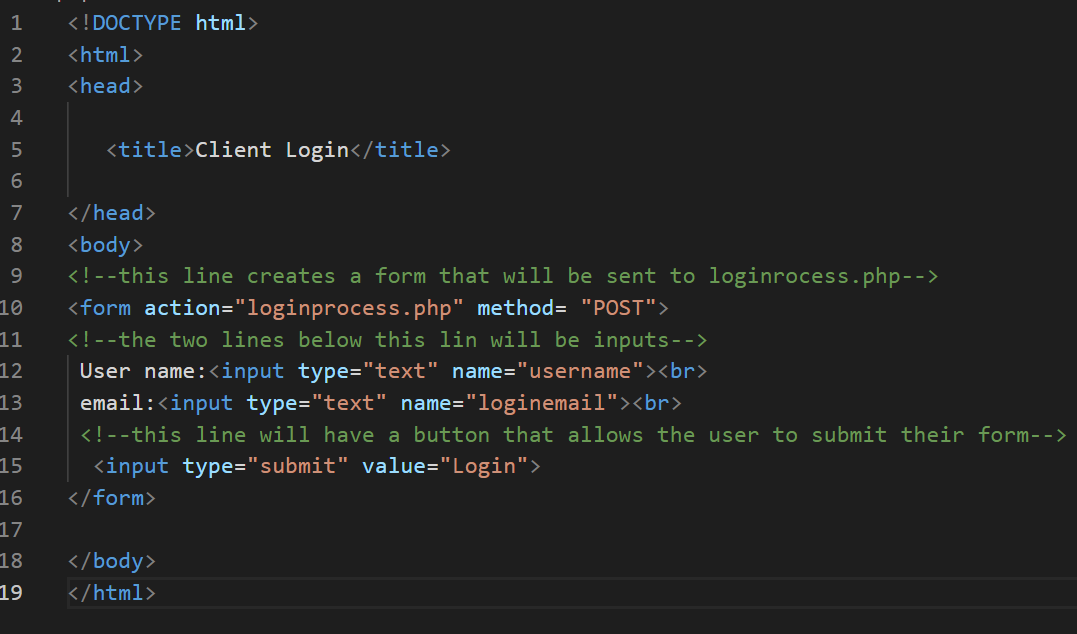
The next thing that I will do to validate this page is that I will try to enter data into the database, but I will leave some inputs empty. The addsitters.php page should be able to tell that some of the data isn’t being inputted and it shouldn’t allow the data to be inputted into the database.

Finish sitters.php/ addsitter/php

The next thing that I will do is to create the page that Allows the user to login to their account. For this I will create a page for sitters to login to and a page for clients to login to. I will also create a homepage for clients and a homepage for sitters. This is because sitters and clients will have to be able to do different things, so if there was only one page for both sitters and clients there would be things that they woudn’t be able to do that they would have to do.

Firstly I will create the userlogin.php page. This page will be the page that allows the client to login to the the account that they have created. To start off with I will create the inputs on the page that the user will enter their details into.

Here is a screenshot of the code for that page:



Here is the screenshot of the page:

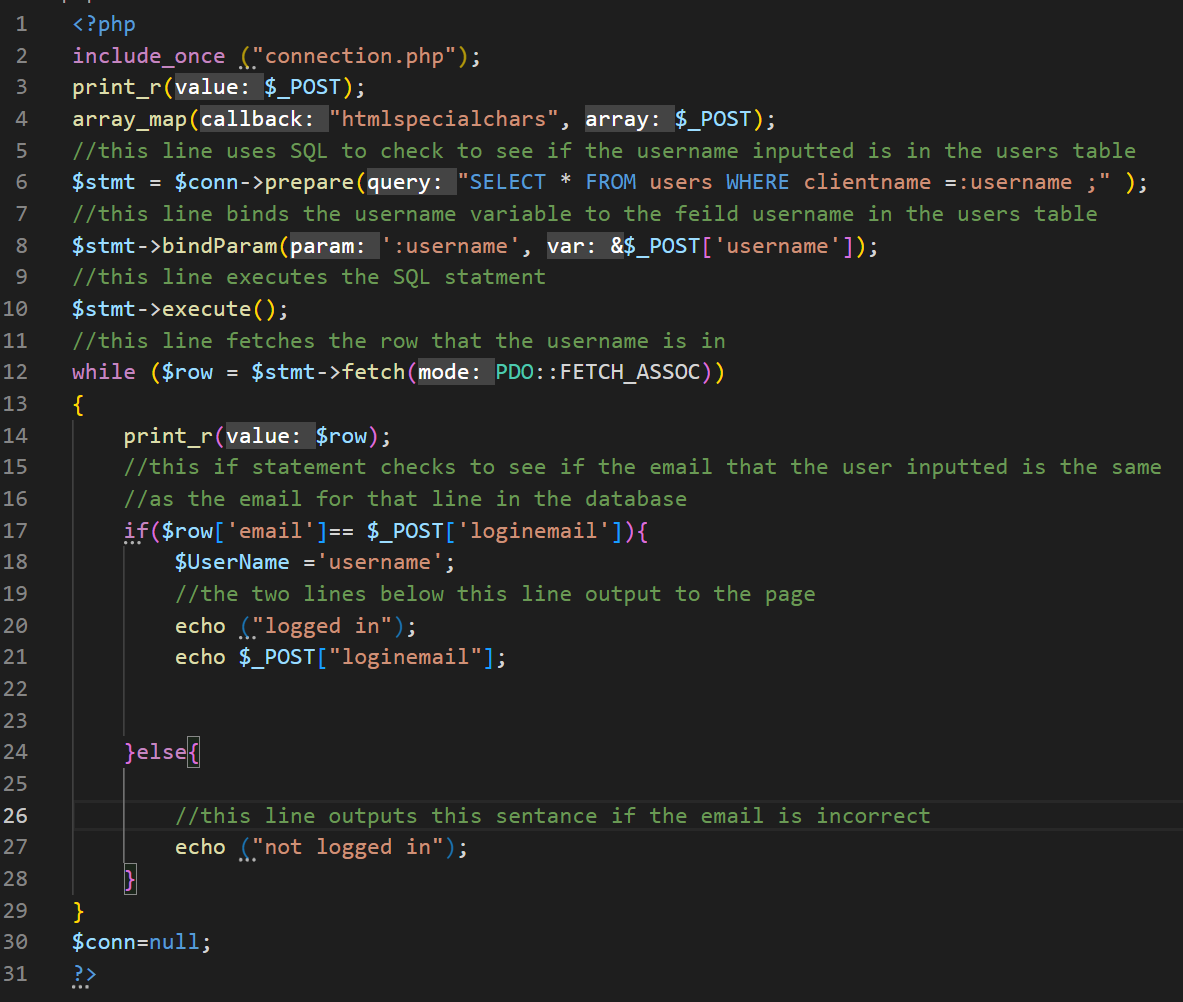


Currently the login page doesn’t look good. This is because there is no colour in the page and the inputs are not in the center of this page. To fix this problem I will create a style sheet and assign classes to the different section of this page, so that they will look nicer.

Create stylesheet:

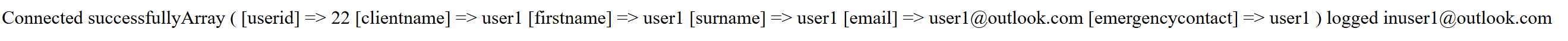
The next thing that I will do is to create the login process page. This page is the page that actually checks the database to see if the data that is inputted into the login.php page is correct. To do this is searches for a username that matches the username that has been inputted by the user, and it checks the value in the field email in that row. If both inputs are correct it should send the user to the homepage.php page, but if one or both of the inputs are incorrect it should not send them to the homepage.php page, and it should clear all of the inputs and allow the user to input their data into the page again.

Here is the code for login process:



When I enter correct data into the fields of login.php the page loginprocess.php should say “logged in” and it should output the email that I entered into login.php.

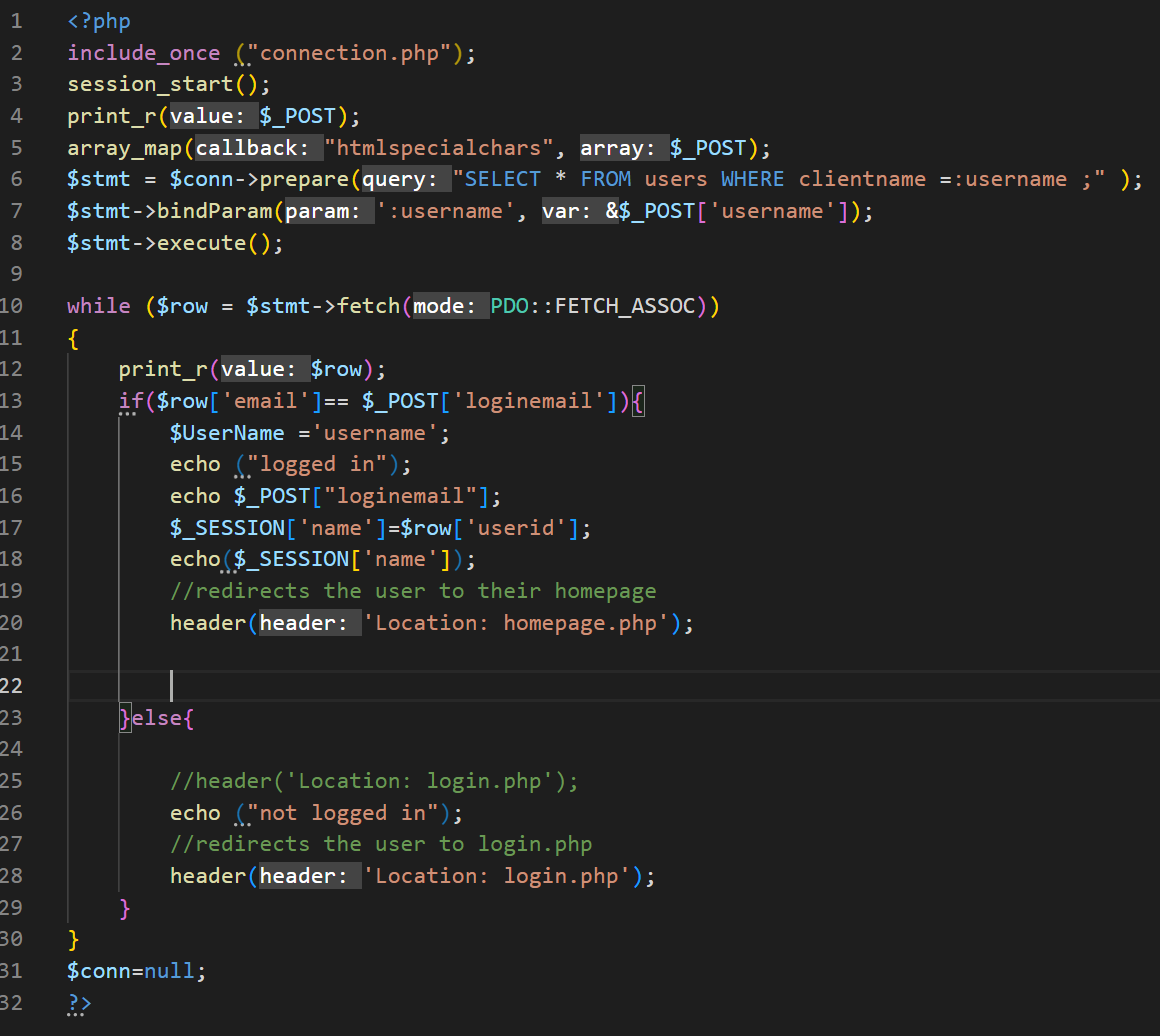
Here is a screenshot of the data that I inputted into the login.php page:



This shows me that both of the pages are working correctly because it says “logged in”. If one of the pages didn’t work it would have said “not logged in”.

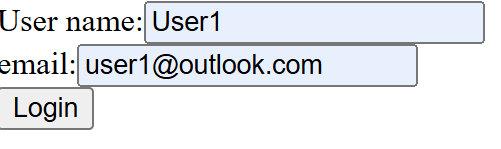
Now I have to add a way that will send a client to their homepage. This is important because otherwise there would be no way for the user to do things on the website. Firstly I will create a page called homepage.php, then I will add a line of code to loginprocess.php, which will redirect the user to their homepage. I will also have to add a line of code to the loginprocess.php page that redirects the user to the login.php page if they entered incorrect login details, so that they can enter their details again. I will also have to add a segment of code that creates a session. This is important because it ensures that the user will only see data that is meant for them.

Here is the loginprocess.php page with the lines of code that redirect the user to their homepage:

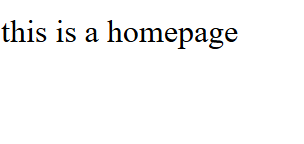


Now when I enter the correct details I should get sent to the user homepage, but when I enter the incorrect details I should get sent back to the login.php page.

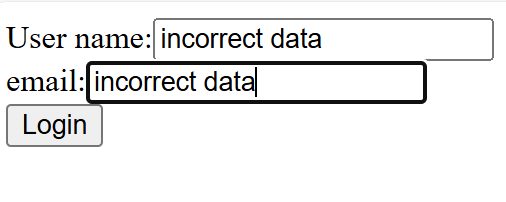
Here is a screenshot of me entering the correct details:



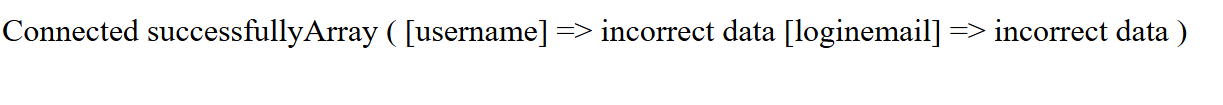
Here is the screenshot of the page that I get sent to:



Here is a screenshot of me entering incorrect data:



Here is a screenshot of the page that I get sent to:



This tells me that there is a problem with the code in loginproces.php because all of the variables from login.php are being sent to the page loginprocess.php but for some reason it doesn’t redirect the user to login.php.

The problem that I found was that if the username was incorrect the code would search through the user table looking for the inputted username, but when it couldn’t find the inputted username it would have no way to check to see if the email that was inputted matched the email in that row, so it would never be able to go through the while loop, which is where all of the redirecting lines of code are. To fix this problem I added a line of code to the end of the page that redirects the user to the login page. This shouldn’t affect the rest of the code because it is at the end of the page, so if any other actions need to be carried out they will be carried out before it.

Here is the screenshot of the code on the page loginprocess.php:

A screen shot of a computer program

Description automatically generated

Here is a screenshot of the data that I entered into login.php:

A close-up of a computer screen

Description automatically generated

Here is the screenshot of the page that I got redirected to:

A white rectangular object with black lines

Description automatically generated

This shows that both pages are now working because when I enter the incorrect data I get sent to the correct page.

The next thing that I will have to do to this page is to add sessions. This is an important feature because it allows the website to know which user is logged in, so that it can show the user all of their data, and it also stops other users from accessing other users data.

A screen shot of a computer program

Description automatically generated

Now at the start of loginprocess.php a session is created by the line “session\_start();”, then a variable called $\_SESSION is set to the userid of the row that the correct username and email are in. This will allow the website to validate which user is on the website because it will be able to check the variable SESSION and match it to a row in the users database.

To validate the pages login.php and loginprocess.php I will enter data into the database and will check the outputs. For this test I will create a page called homepage.php, which will display the SESSION variable for the user. The page displays the SESSION variable to show that the segment of the code that creates sessions is working. If I enter the correct data into the inputs on the page login.php when the loginprocess.php page check the inputs I should re redirected back to login.php.

Here is a screenshot of the code for homepage.php:

A screenshot of a computer program

Description automatically generated

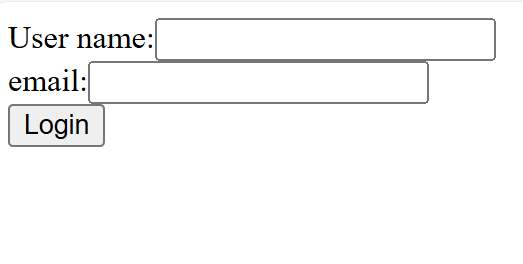
Currently This page is being used as a way to check to see if the pages login.php and loginprocess.php are working properly. Later on I will develop this page further so that it will be able to meat all of the clients needs.

Here is a screenshot of me entering the correct data to login.php:

A screenshot of a computer

Description automatically generated

Here is a screenshot of the page that I get sent to when I enter the correct data.



For some reason when I enter the correct data into the input boxes it redirects me to the login page. I think this is because the line “header(“location : login.php);” is at the end of the code, so no matter what the user inputs they get redirected to the login.php page, this line overrides all of the other lines that try to redirect the user to another age.

To fix this problem I created a variable that is normally set to 1, but if the username that the user inputted into the page login.php is correct the value of check is changed to 2. There I added an if statement that checks after the while loop in the code that checks to see if the value for check has been changed from 1. If the value hasn’t been changed it sets the value for check to 0. I also added another while loop that checks to see if the check value is zero, and if it is it has a line of code that redirects the user to the login.php page. This solves the problem because unless the check value has been changed to 0 there is no way for the user to be redirected to the wrong page.

Here is a screenshot of the code for loginprocess.php:

A screen shot of a computer program

Description automatically generated

So when I enter the correct data into login.php I should be redirected to the page homepage.php.

Here is a screenshot of me entering the correct data:

A close-up of a computer screen

Description automatically generated

And here is the screenshot of the page that I get sent to:

A white background with black text

Description automatically generated

This test validates the two pages because it shows that they can work together. This is because the homepage displays the correct userid, which means that the segment of code in the page loginprocess.php that creates the session is working, otherwise it wouldn’t output the correct data, and it shows that the problem that I get sent back to the login page even if I enter the correct data has been fixed because I have been sent to the correct page in the website.

Another thing that I will have to do to validate these two pages is to check to see how it handles incorrect data. For this I will enter an incorrect email with a correct username, and I will do another test where I enter a incorrect username and an incorrect email. I won’t have to do a test where I have an incorrect username and a correct email because the code in loginprocess.php scans through the users table looking for a correct username, and then it checks the row that it is in to see if the username matches the inputted email. So if I entered an incorrect username the code would never check the email field of any row.

Here is a screenshot of the enter a correct username but an incorrect email:

A screenshot of a computer

Description automatically generated

Here is a screenshot of the page that I got sent to:

A screenshot of a computer

Description automatically generated

This shows that the pages work when an incorrect email but a correct username because it sends me to the correct page.

Now I will test to see what happens when both inputs are incorrect.

Here is a screenshot of me entering incorrect data:

A screenshot of a computer

Description automatically generated

Here is a screenshot of the page that I got sent to after entering these details:

A white background with black text

Description automatically generated

For some reason the value of check wasn’t being changed, which meant that I wasn’t being redirected to the login.php page when I inputted the incorrect data. I think this was because the if statement that changed the value of the variable check from 1 to 0 was inside of the while loop that checks to see if the username and password are correct. To fix this problem I moved that if statement outside of the while loop.

Here is a screenshot of the if statement that changes the value of the variable check from 1 to 0:

A screen shot of a computer

Description automatically generated

Here is a screenshot of the new code for the page loginprocess.php:

A screen shot of a computer program

Description automatically generated

Now when I input incorrect data into the login.php page I should be redirected to the login.php page from the loginprocess.php page.

Here is a screenshot of me entering incorrect data into the login.php page:

A screenshot of a computer

Description automatically generated

And here is a screenshot of the page that I get redirected to:

A screenshot of a computer

Description automatically generated

These tests show that the pages login.php and loginprocess.php can tell wether or not the correct data has been inputted into login.php, and it can send the user to the correct page if they enter the correct details and it will send the user to the wrong page when they enter the incorrect data.

Currently the login.php page looks very bland and it isn’t user friendly. So the next thing that I will do to make the webpage look more appealing is to create a style sheet that will change how the page will look.

To make the webpage look more appealing to the user I will use CSS to change the layout of the page and to add colour the the webpage. I will also use some of the classes from the style sheet that I created for the page “index.html” so that I don’t have to rewrite all of the classes.

Here is a screenshot of the new code for login.php:

A screen shot of a computer program

Description automatically generated

I imported the stylesheet that I used for index.html. This is because I wanted to use the class centre and if I imported the stylesheet it meant that I didn’t have to rewrite the class centre. I also created a class called button, which sets the background of the button to navt and the colour of the text to white. I added another class called orange background, which sets the background to orange for what ever it is attached to. I assigned the body of this webpage the class “orangebackground”. This is because I wanted the background of the webpage to be orange, and I wanted the text inside of the webpage to be white, so if I gave the body of the webpage the class orange background all of the contense inside of the body of the webpage will be effected by this class. I created a container for all of the inputs on this webpage, and then I assigned this container the class centre. This is because when I assigned the container this class all of the contense inside of this container were effected by this class. Finally I assigned the submit button the class button. This means that the colour of the button will be navy and the text colour will be white.

Here is a screenshot of the webpage login.php:

A screenshot of a computer

Description automatically generated

Now that I have validated that all of the code inside of login.php and loginprocess.php I will create the pages that allow a cat sitter to sign into an account that they have created.

Firstly I will create the page “sitterlogin.php” . The sitterlogin.php page should allow a cat sitter to enter their username and their email. Then the inputs should be sent to a page called sitterloginprocess.php, which will check wether or not the details that the sitter has inputted are correct.

The first thing that I did on the sitterlogin.php page was to create the inputs. I did this to check if the inputs were in the correct place.

Here is a screenshot of the page sitterlogin.php:

A screen shot of a computer program

Description automatically generated

Here is a screenshot of the sittterlogin.php page:

A screenshot of a computer

Description automatically generated

Now I will create the sitterloginprocess.php page. This page take the variavles that has been given to it by the sitterlogin.php page and it will check the username variable given to it against all of the values in the field sittername, and if it finds a matching value it will check the email field of that row to see if the email that the sitter inputted matches. If the values match the sitter will be sent to the sitterhomepage. Otherwise the sitter will be sent back to the sitterlogin.php page. Because the code for sitterloginprocess.php is very similar to the code from loginprocess.php I will be able to use lots of the same code to create sitterloginprocess.php.

Here is a screenshot of the code for the page sitterloginprocess.php:

A screenshot of a computer program

Description automatically generated

Here is a screenshot of me entering data into sitterlogin.php

A close-up of a computer screen

Description automatically generated

When I input data into the page sitterlogin.php I get sent to this page:



This shows me that the sitterlogin.php page and the sitterloginprocess.php are working because when I entered the correct data it outputted “logged in” to the screen. The next thing that I will do is add segments of code that send the user to different webpages, depending on if they have entered the correct details or the incorrect details. This is necessary because otherwise the user will be stuck on the sitterloginprocess.php page.

Here is a screenshot of the edited code for the page sitterloginprocess.php:

A screen shot of a computer program

Description automatically generated

So now when I enter correct data I should be redirected to sitter homepage.

Sitterhomepage.php is a page that I will create in the future, but now I will create the page so that I can test to see if the page sitterlogin.php redirects the user to the correct page. I will develop the page sitterhomepage.php in the future so that the user will be able to use it as a homepage.

Here is a screenshot of the current code for the page sitterhomepage.php:

A black screen with white text

Description automatically generated

Here is a screenshot of me entering correct data into sittelogin.php:

A close-up of a computer screen

Description automatically generated

Here is a screenshot of the page that I got sent to:

A white background with black text

Description automatically generated

This shows me that the code that redirects the user to a page is working because when I inputted the correct data I got redirected to the sitterhomepage.

The next thing that I will have to do for this page is to add a section of code that starts a session. This is important because the session allows the website to know which user is accessing a page, so that it can make sure that the user sees all of their data, and no one else’s data.

Here is a screenshot of the edited code for the page sitterloginprocess.php:

A screenshot of a computer program

Description automatically generated

To create the session I used the line “session\_start();”, then I created a variable and set its value to the value in the field sitterid in the row that the correct email and password are int, then the page outputs the variable SESSION.

To test to see if the code that creates the session is working I changed the code in the page sitterhomepage.php to this :

A screen shot of a computer program

Description automatically generated