Task 2 Developing the Solution

In terms of certain rules regulations and requirements, I have taken relevant ones into consideration and now I have designed an application that accommodates many of the requiresments I outlined in task 1. It ranges between anything accessibility wise through to non-copyrighted images being used on the website. I have kept colour schemes to black and white with the occasional blue text on forms. This all makes for a professional looking site and easy to read for all users.

When sourcing external links and videos for the further learning resources I made sure that the sites they navigated too were entirely trusted and not in any way malicious or misleading. The sites were strictly educational – some with some fun learning and others straight to the point. This gives the users an extra tool in case they want to learn more externally to GibJohn Tutoring.

When coding my digital application, I used bootstrap and Blade Templating (Laravel). To make this happen I used XAMPP – this is a very clever application that has a php server and database built into it – and many other features (which I will talk about later on).Lots of bootstrap is entirely public and opensource however whenever I took a code snippet from a website, I have made sure to reference them to give them the credit and validate for the purpose of the exam to show that I have used sources. We were given full internet access so there is no reason why this should not be allowed providing I reference the relevant sites/sources. And this is what programming and coding is all about, collaborating and coding by using different and sometimes pre coded code to help you along.

I have used three / four Languages including HTML, CSS, PHP and an occasional bit of JavaScript to create both the frontend and backend to my GibJohn digital solution.

All Images used are entirely in line with ethical and legal guidelines and regulations – completely diverse and fits in with today's societal expectations.

Prototype for the proposed solution

GibJohn Tutoring has secured a new contract with my company to develop a digital solution to implement onto an already existing digital system. GibJohn Tutoring already offers face to face tutoring sessions, access to learning resources and support to develop understanding in different subjects of Education.

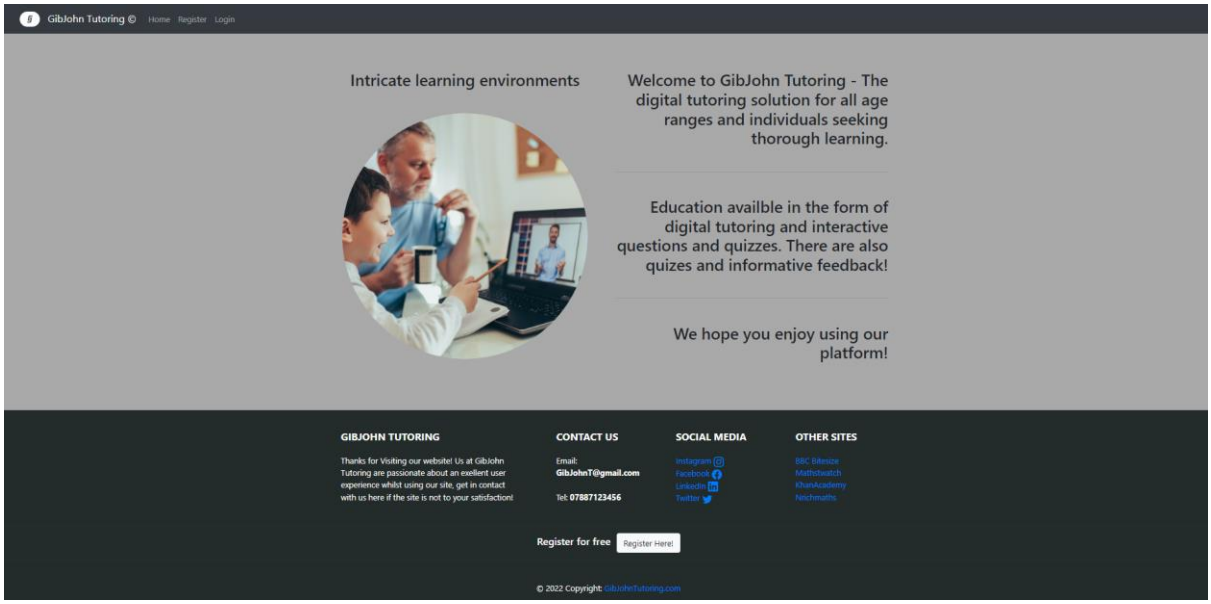
GibJohn Tutoring's requirements for us was as follows:

- Provide interactive teaching and learning resources in a range of subjects.
- Provide digital access to content to encourage wider learning.
- Support Assessment and monitoring of learner progress.

GibJohn also had some recommendations based off their own independent market research. This included:

- Collaborative teaching and learning tools.
- Accessibility features to support a wide range of users.
- A learning reward system
- Gamified learning.

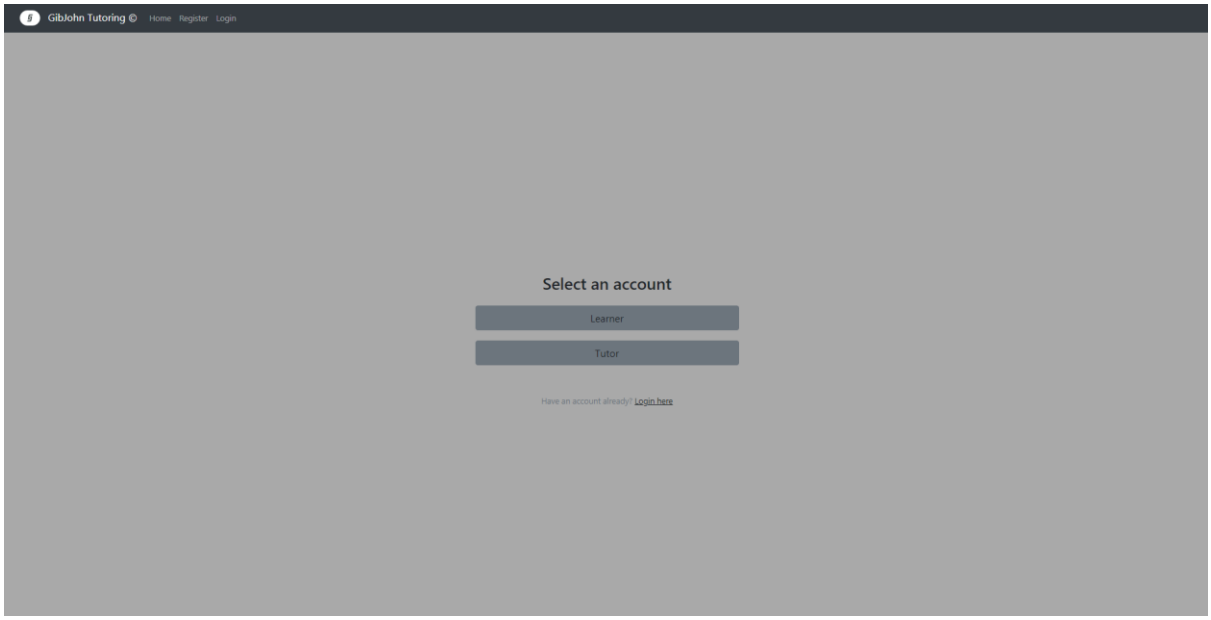
Homepage



I have gone for a very simplistic homepage that is easy to look at with a little bit of information about the site. In the footer below you can see a description of GibJohn Tutoring, links to social media and external websites, contact information, copyrighted site and a register here button.

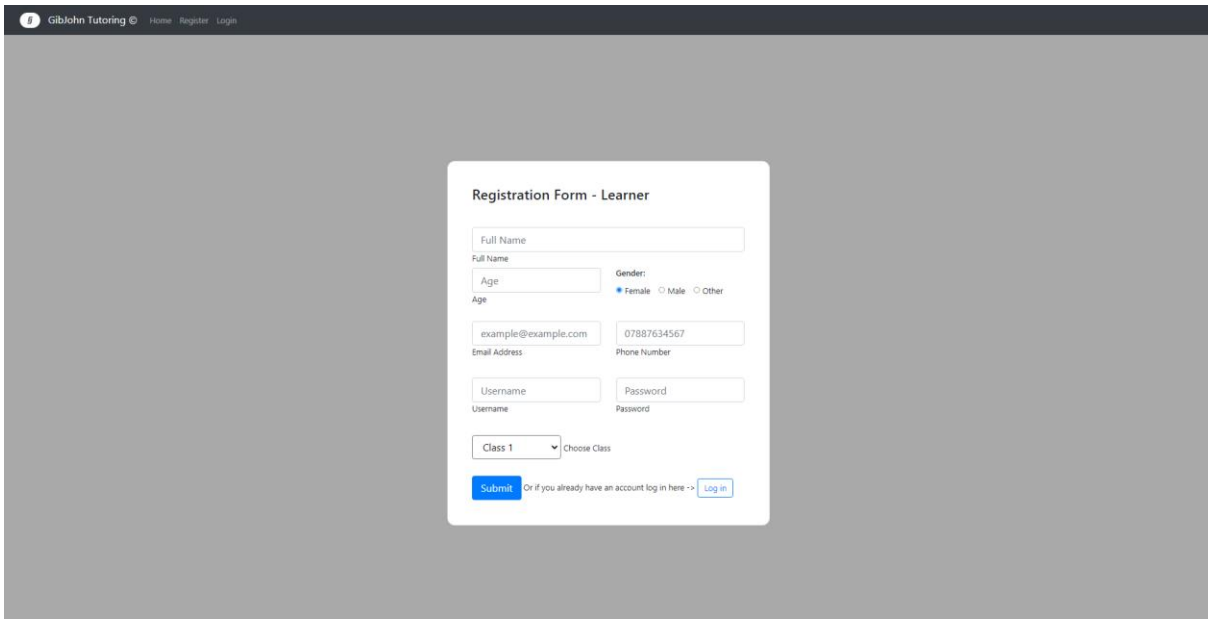
All the homepage is doing is welcoming users to the website and giving them a general bit of information about the site.

Register Screen (Option)



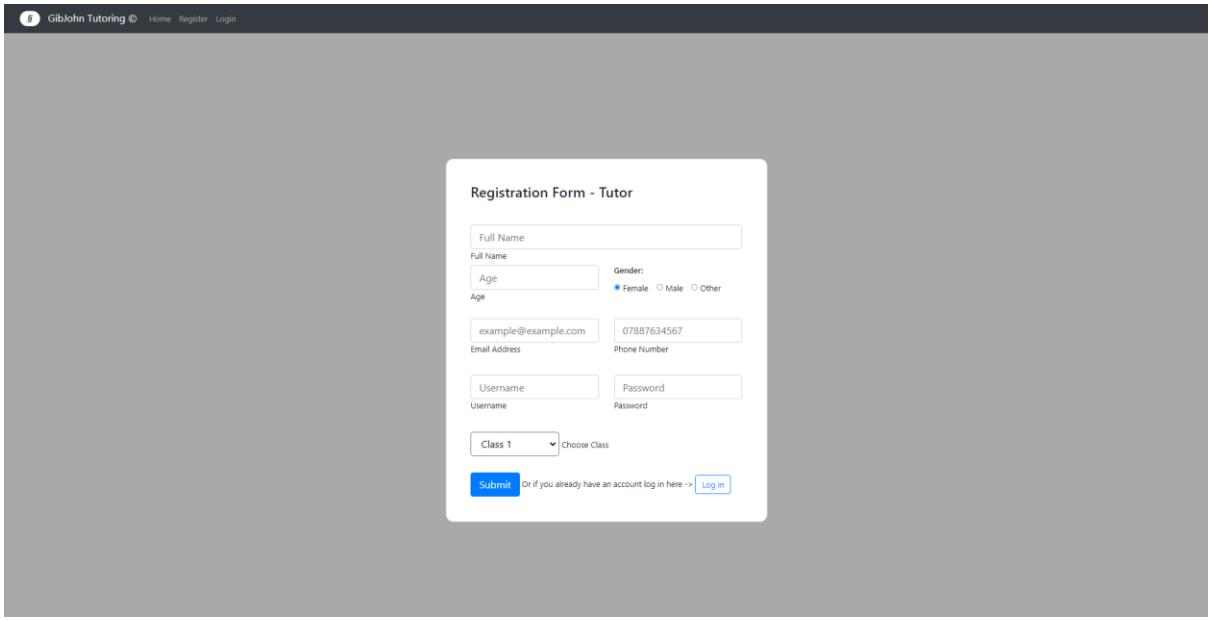
This is the register screen in which the users can choose either a Tutor or learner account to register as and if they already have an account it will redirect to the login option screen. Both tutor and learner option screens are the same in layout and simplistic design. However Just noticing now that the grey text is hard to read however is completely readable when you are on the page itself.

Register Form Learner



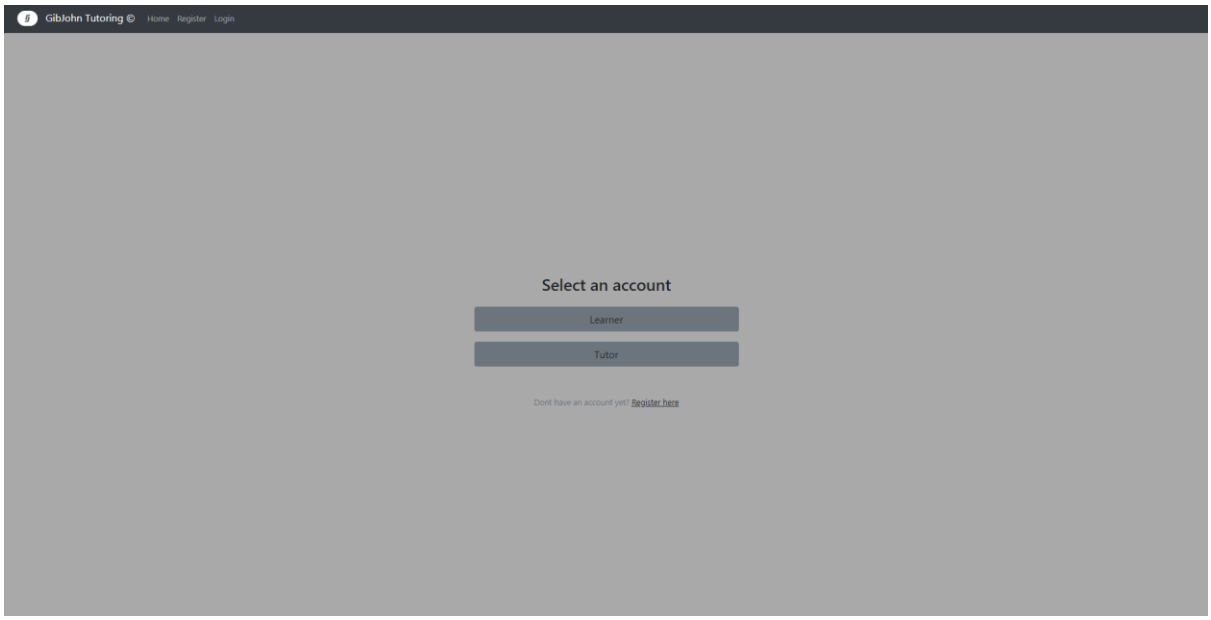
This is the register form for a learner, as you can see it asks for all relevant fields and the most important one the class. This determines the content they see. All their details are sent POST to a MYSQL Database and stored ready to login. Once the user has clicked submit then they are redirected to the login screen (Learner).

Register Form Tutor



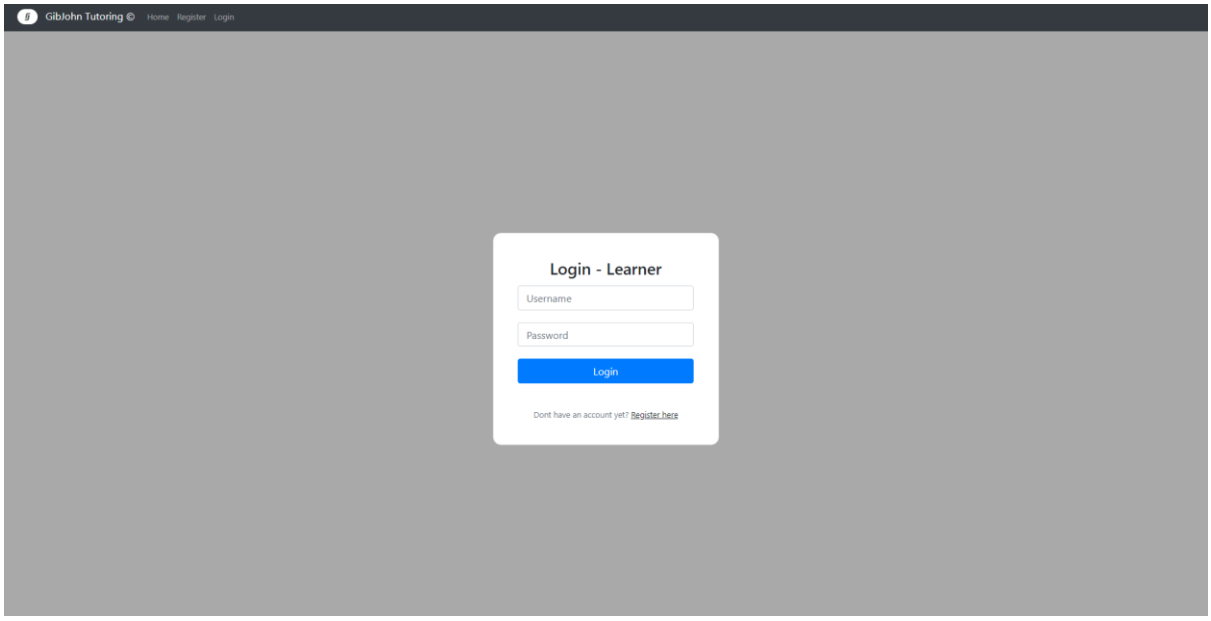
This is the register form for a learner, as you can see it asks for all relevant fields and the most important one the class. This determines the content they see. All their details are sent to a MYSQL Database and stored ready to login. Once the user has clicked submit then they are redirected to the login screen (Learner)

Login Screen (Option)



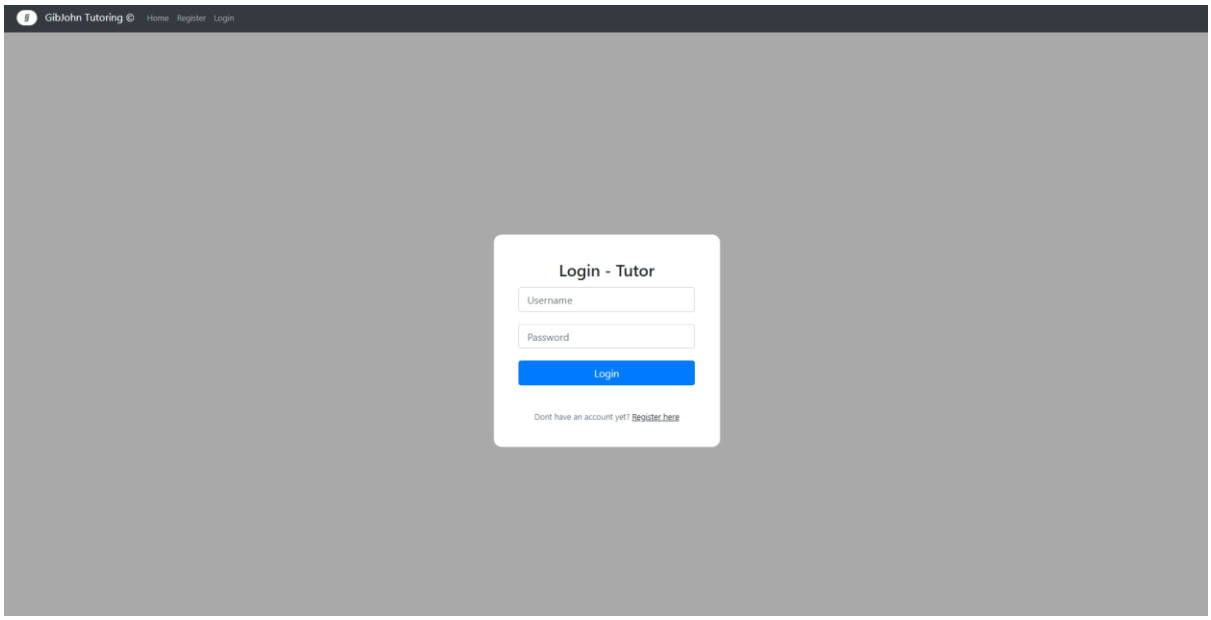
This is the login option screen that asks the user to login with either a learner or a tutor account. Once the users have registered the system redirects them to the relevant forms depending on their tutor or learner login. There is an option to register below if the user hasn't created an account.

Login Learner



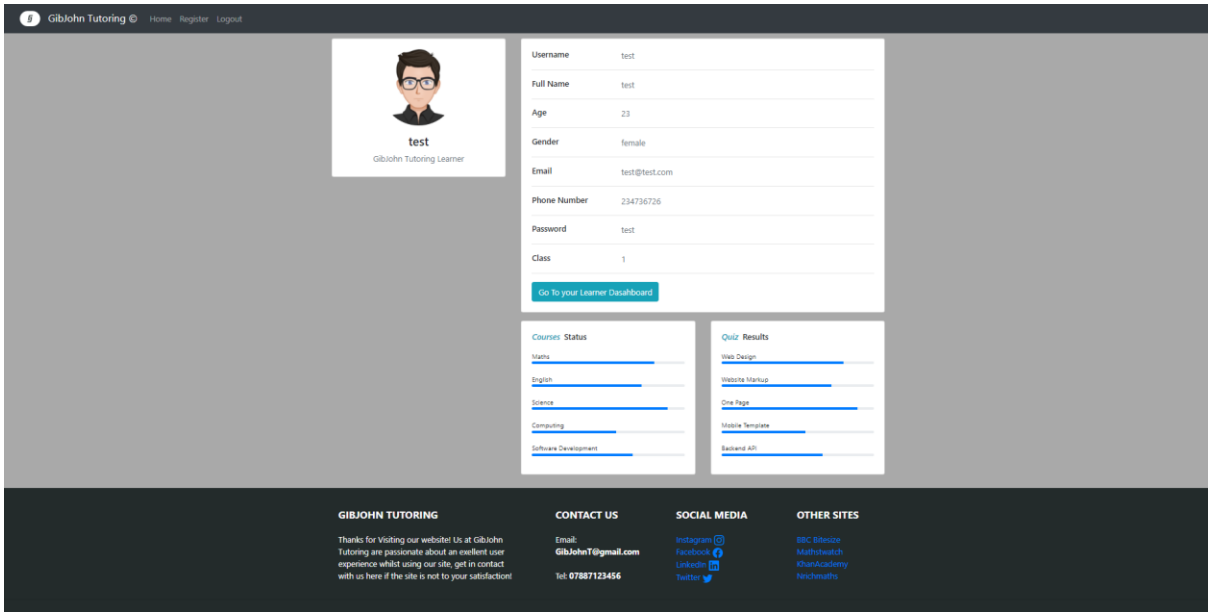
This is the form in which a user can login as a learner and navigate further into the site itself. When the button is clicked the data is checked against the learners' registered details and a match is either made or not. IF it is the learner is redirected to the learner information/user data page. If not, it directs to error pages.

Login Tutor



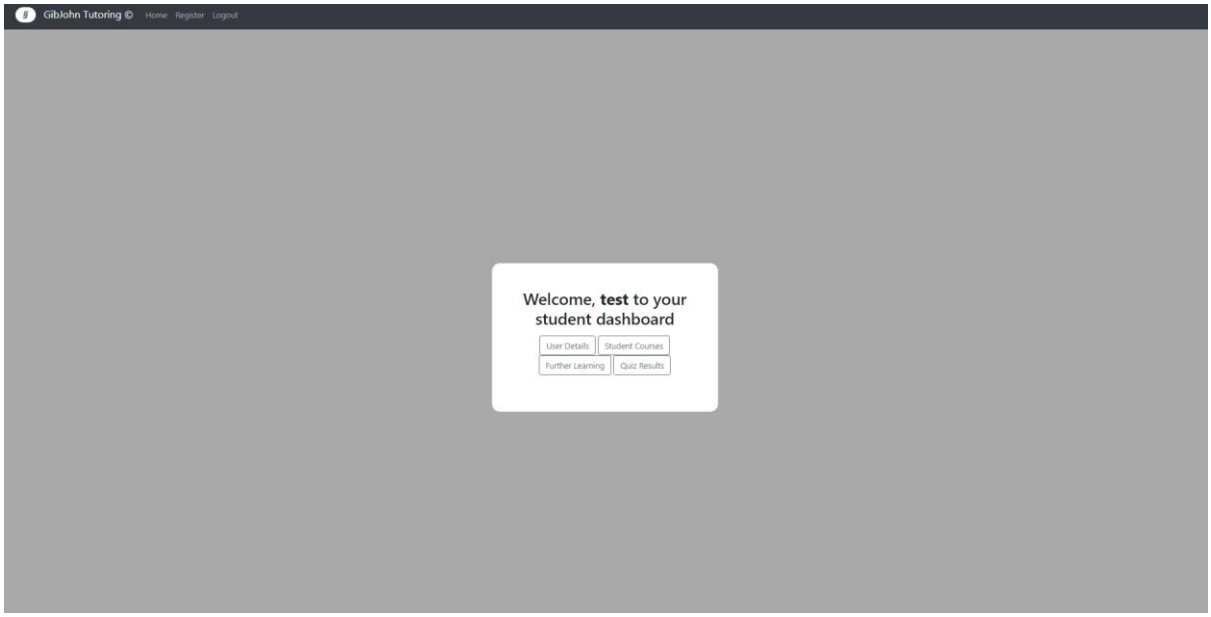
This is the form in which a user can login as a learner and navigate further into the site itself. When the button is clicked the data is checked against the learners' registered details and a match is either made or not. IF it is the learner is redirected to the learner information/user data page. If not, it directs to error pages.

Learner Details

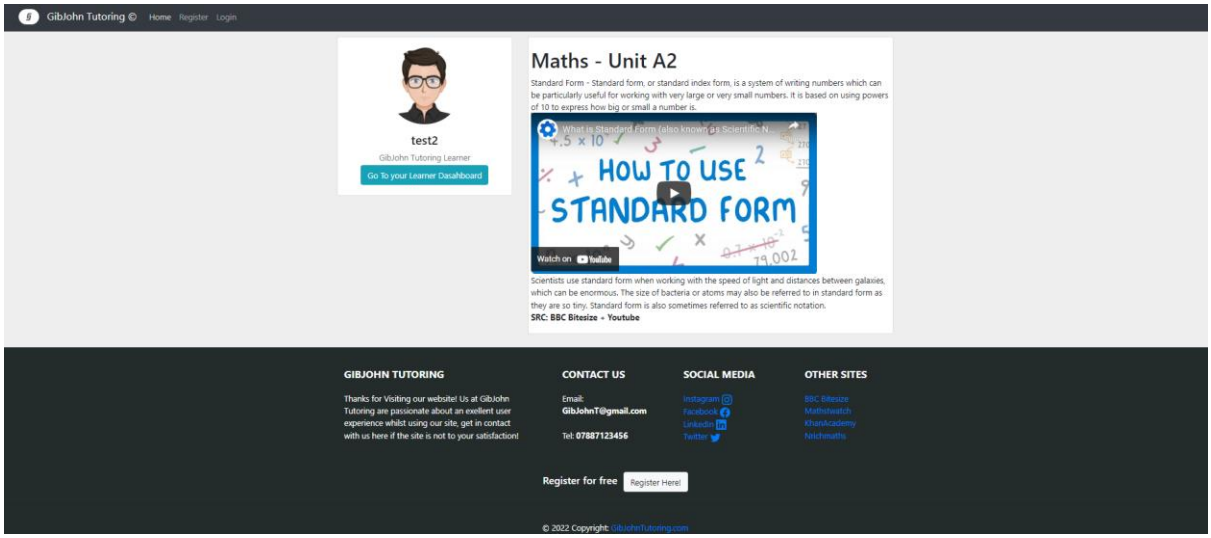


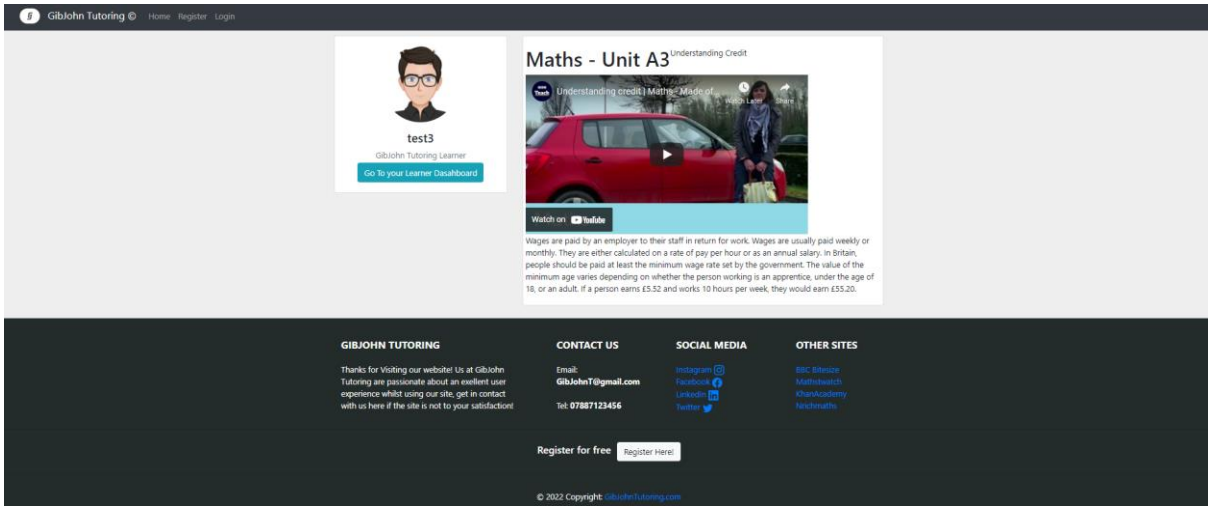
This is the learner details page. As the name implies it displays all the user details, however I did plan on having them editable however it was a very complicated process that unfortunately did not work out in time so if there was another iteration of this digital solution it would be implemented in. The user details are queried in the database and drawn out via the user ID of the session that is created once the user is logged in. The picture image is of a male, and it does not change depending on the gender which again would be something I would add to the next iteration of this digital solution. The 2 progress boxes are just there to give an indication of progress made- however they do not provide relevant data as they are just hard coded and the progress bars in the actual quiz results and student progress will be produced from values in the database. I would implement these 2 progress boxes with real data in the next version of the application.

Learner Dashboard



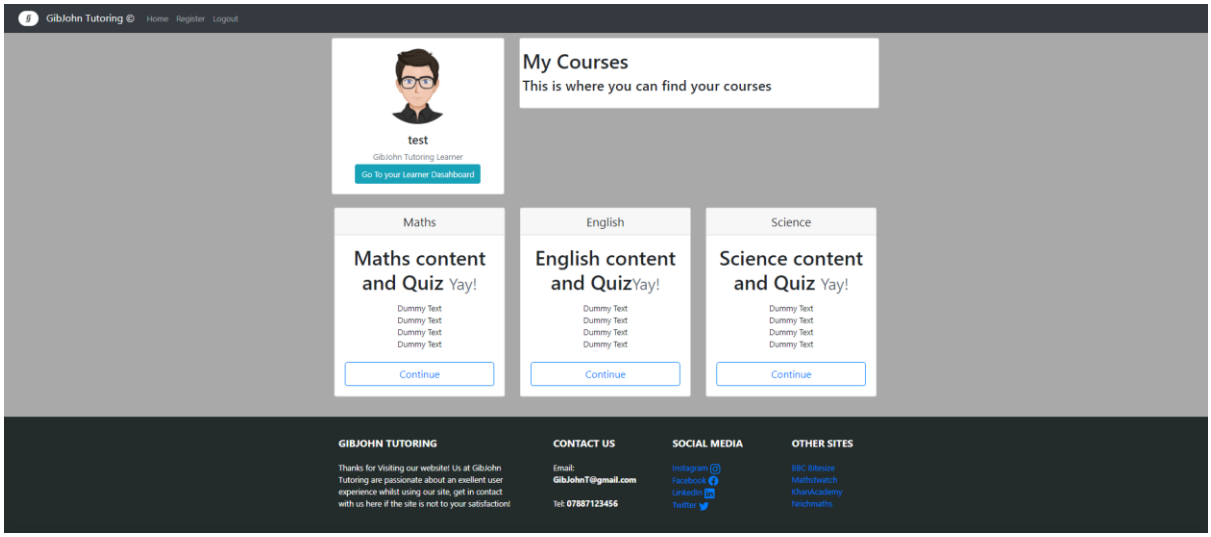
This is just the central navigation point for the user (learner) to gain access to different parts of the webpage. The Session learner name is used to personalise the page slightly. The 4 pages, user details, courses, quiz results and further learning are all separate webpages. I have used bootstrap buttons with <A> tags to make the 4 buttons that continue the greyscale theme and when hovered over go a darker grey.





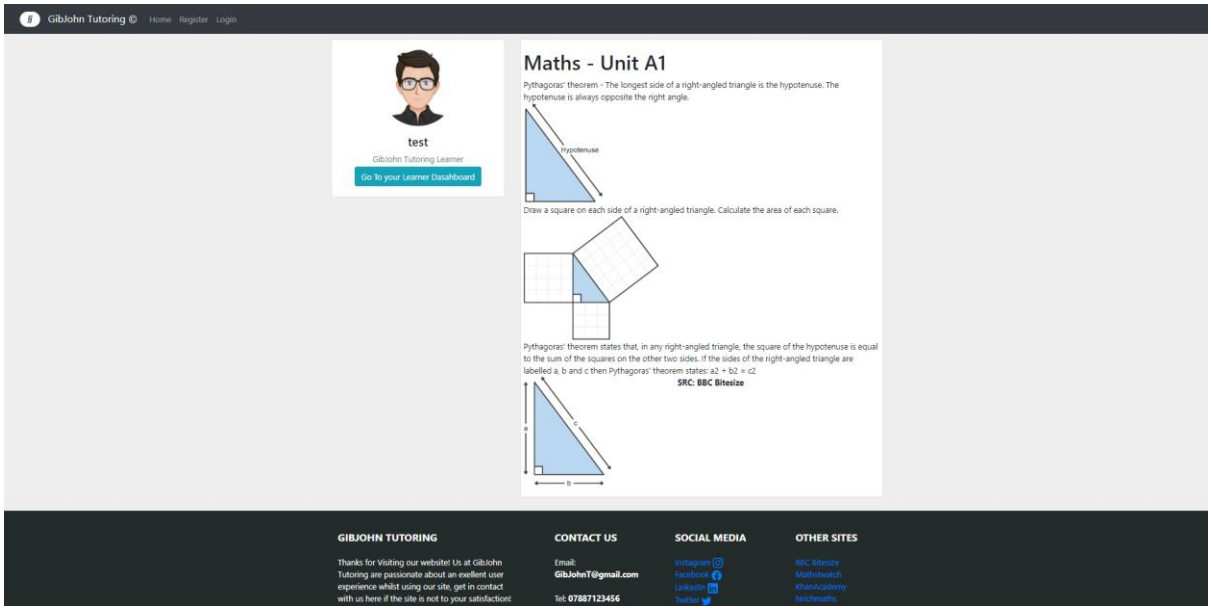
These are the 3 maths pages that I have hard coded and all using blade templating to make my code more efficient and easier. Here is a screenshot of how the system displays the correct content depending on the learner's class.

Learner Courses

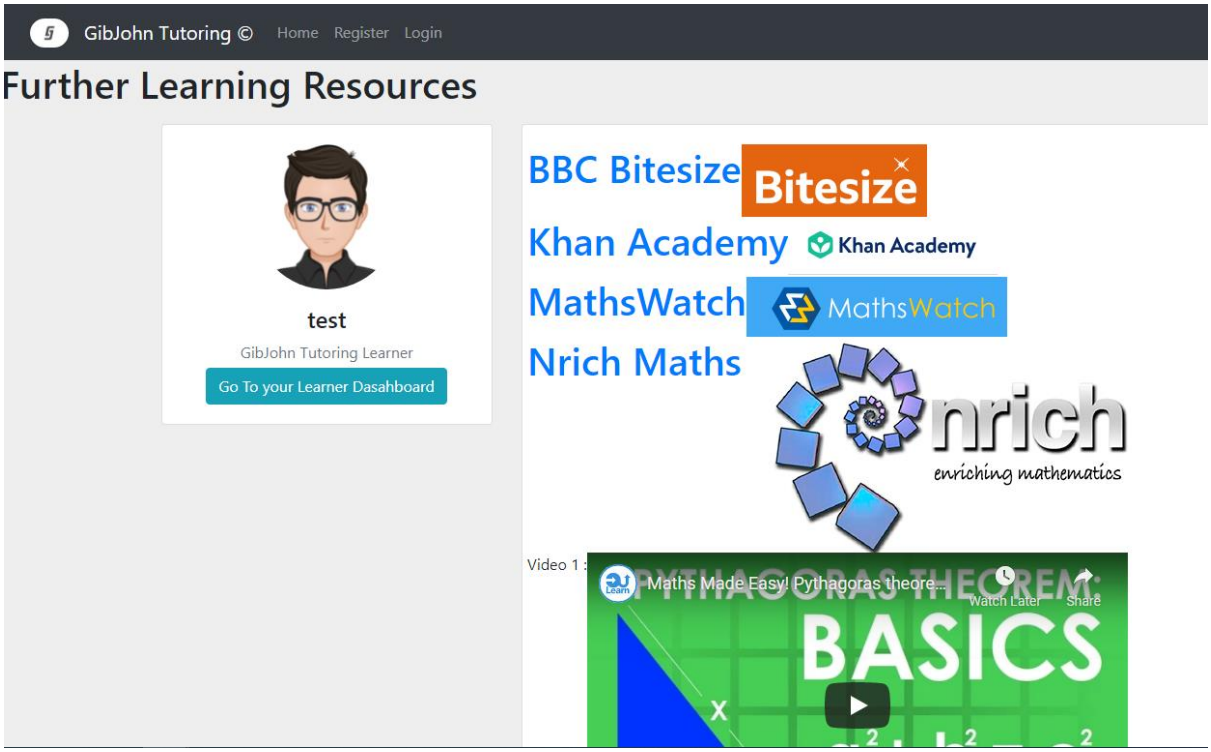


This is the learner courses page. I have included maths English and science as courses and depending on the class of either 1,2,3 the user gets presented with different content for maths, English and science. This is a bootstrap template and I used it due to its simplicity yet professional looking layout. The buttons navigate to a redirect page for each and depending on the learner's class they go to different content pages (as mentioned above). The next few screenshots are of three content pages for each subject – one for each class selected on the register form. These follow with the current colour scheme and is not horribly contrasting to the grey background colour. You can visibly see and make out what each element of the page is about. It follows a Z pattern and is slightly different to my initial thought of the courses page.

Learner Courses Maths 1,2,3 (Classes 1,2,3)



Further Learning Resources (Learner and Tutor page)



This page is where learners can find links to external sources (sites and videos) in order to progress with their learning. This is obviously entirely changeable – all the links can be edited, and things can be deleted/added. If there was a version 2 of the software, I may personalise the further learning for each student. The example above had hyperlinks and videos embedded into the content.

php > mathsredirect.php

```php
<?php
// We need to use sessions, so you should always start sessions using the below code.
session_start();
// If the user is not logged in redirect to the login page...
if (!isset($_SESSION['loggedin'])) {
    //header('Location: index.html');
    exit;
}

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "tlevelexam";


//this is the database connection method usiing the variables defined above.

$conn = mysqli_connect($servername, $username, $password, $dbname)
    or die("Connection Failed");
$querey = mysqli_query ($conn, 'SELECT * FROM registered_learners');

//this is a querey in which I used to get the LearnerID out of, so I can assign it to a variable and it can be used in other queries as shown below.

$querey1 = mysqli_query($conn, 'SELECT * FROM registered_learners WHERE LearnerUsername like "%'. $_SESSION['name'] . '%"');
$roww = mysqli_fetch_array($querey1);
$id = $roww["LearnerID"];

//this qurey below grabs the learner class depending on the learnerID from the $id Learner Id assigned above.

$querey = mysqli_query($conn, 'SELECT LearnerClass FROM registered_learners WHERE LearnerID like "%'. $id . '%"');
$row = mysqli_fetch_array($querey);
if ($row) {

    print($row['LearnerClass']);
    $lc = $row["LearnerClass"];
    //learnerclass is now assigned $lc
} else {
    print("NOT FOUND");
}
```

You will possibly notice that I used “” marks around the numbers and that is because for some unknown reason integers wouldn’t be picked up in the queries above and varchar (string) values were being picked up. The LIKE % $variable % worked only with strings when the values were compared within the database, so I set the data type to varchar (255).

```php
// this is the actual logic behind it - just if and elseif statements.
if($lc == "1"){
  header("Location:/Projects/Blade-test/test_BladeOne/BladeOne/maths1learner.php");
} else if($lc == "2") {
  header("Location:/Projects/Blade-test/test_BladeOne/BladeOne/maths2learner.php");
} else {
header("Location:/Projects/Blade-test/test_BladeOne/BladeOne/maths3learner.php");
}
```

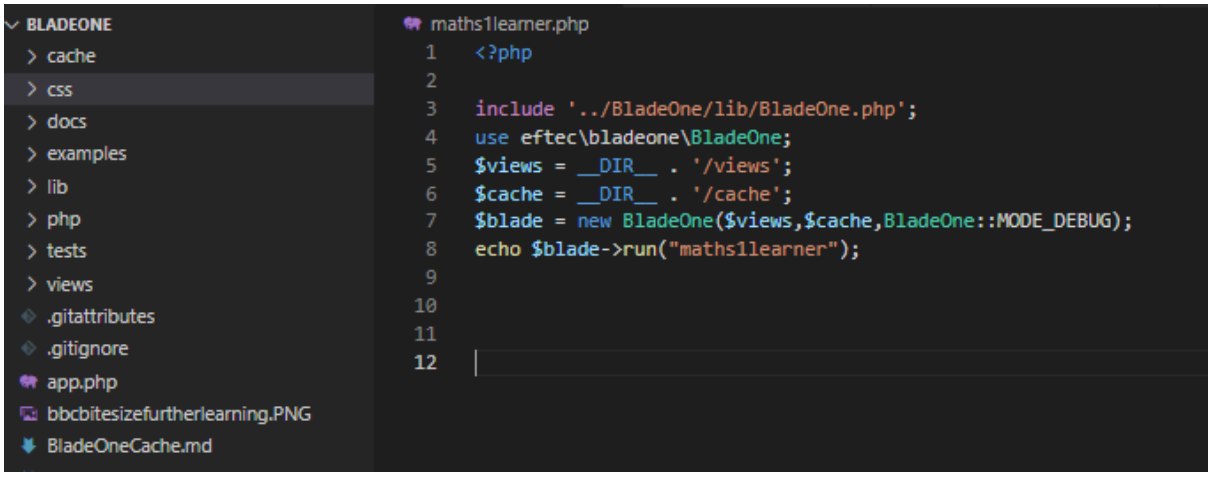
The actual logic is this code ^ here. It is just going through and checking the learner class of 1,2,3 against the variable learner lass set and then redirecting to the correct webpage corresponding to the outcome of the sequence. This is the maths redirect. Below is a picture of the actual code for the maths1learner page.

This is a screenshot of the Maths 1 learner view.

views > maths1learner.blade.php

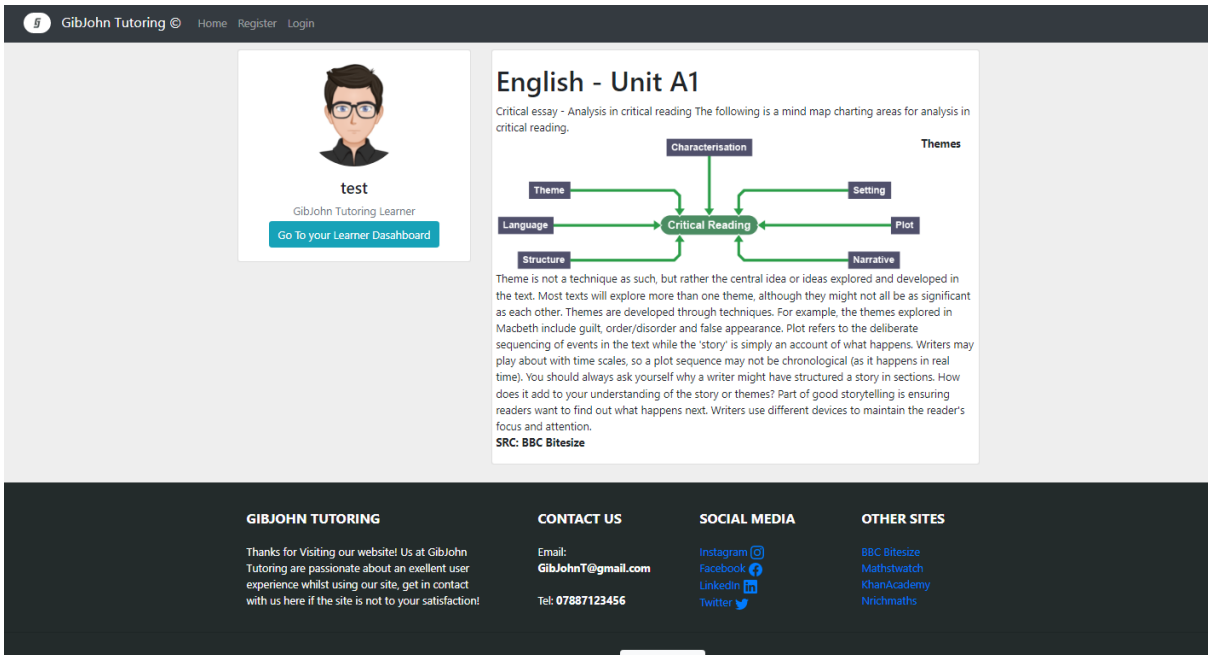
```php
<?php
// We need to use sessions, so you should always start sessions using the below code.
session_start();
// If the user is not logged in redirect to the login page...
if (!isset($_SESSION['loggedin'])) {
  //header('Location: index.html');
  exit;
}
?>
<!-- this is the template part that extends that app.blade.php ( index basically, but mine is app instead )  -->
@extends('app')

@section('content')
<!-- This is the content part and this bit changes for each page. All content is between this section adn @endsection. -->


<div class="container">
    <div class="main-body">
    <div class="padding">
          <div class="row gutters-sm">
            <div class="col-md-4 mb-3">
              <div class="card">
                <div class="card-body">
                  <div class="d-flex flex-column align-items-center text-center">
                    <img src="https://bootdey.com/img/Content/avatar/avatar7.png" alt="UserImage" class="rounded-circle" width="150">
                    <div class="mt-3">
                      <h4><?=$_SESSION['name']?></h4>
                      <p class="text-secondary mb-1">GibJohn Tutoring Learner</p>
                      <a class="btn btn-info " target="__blank" href="learnerdashboard.php">Go To your Learner Dasahboard</a>
                    </div>
                  </div>
                </div>
              </div>
            </div>
            <div class="col-md-8">
              <div class="card mb-3">
                <div class="card-body">
                  <div class="row">
                        <h1>Maths - Unit A1</h1>
                        Pythagoras' theorem -
                        The longest side of a right-angled triangle is the hypotenuse. The hypotenuse is always opposite the right angle.
                        <img src="maths1learnerimage1.png" alt="UserImage">
                        Draw a square on each side of a right-angled triangle. Calculate the area of each square.
                        <img src="maths1learnerimage2.png" alt="UserImage">
                        Pythagoras' theorem states that, in any right-angled triangle, the square of the hypotenuse is equal to the sum of the squares on the other two sides.

                        If the sides of the right-angled triangle are labelled a, b and c  then Pythagoras' theorem states: a2 + b2 = c2
                        <img src="maths1learnerimage3.png" alt="UserImage">
                        <strong>SRC: BBC Bitesize</strong>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
    </div>
</div>
</div>

<!-- This is  all the html for the bootstrap template. This is Maths 1 Unit A1. -->


@endsection
```

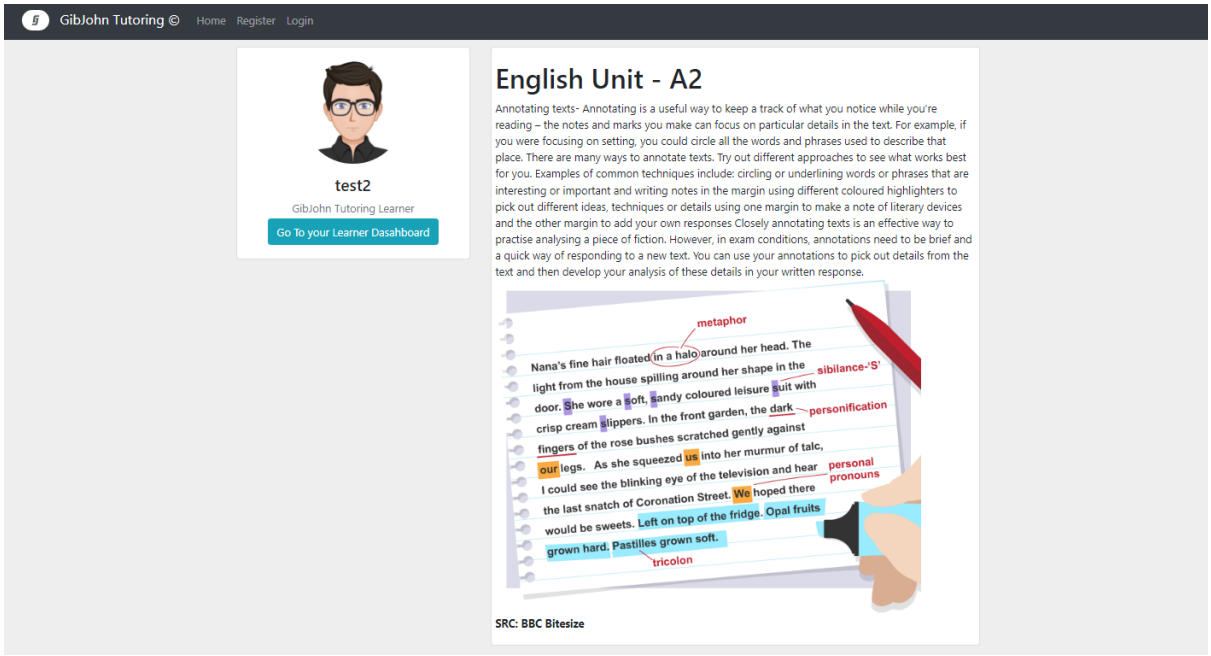
This is using blade templating which really speeds up pages, makes them more consistent and looks a lot more professional. There is much less code required as you don't have to copy over all the headers and footers and differ ant elements from your main webpage. For my app page (which acts as the main template for the content pages) I have made it so there is a header section, a content section and a footer section which automatically clone across when you type @extends('app') @section('content) and @endsection and the extends app extends both header and footer onto the page and the content is changeable. This took a bit of time to set up as you must install Blade One through GitHub over a file destination. With the blade templates you have a views folder which is where the live pages are, and you also need to write this section of code for each view and change the filename so blade can process and run the webpage. If this is not done blade has no source to run the webpage and does not pick it up.

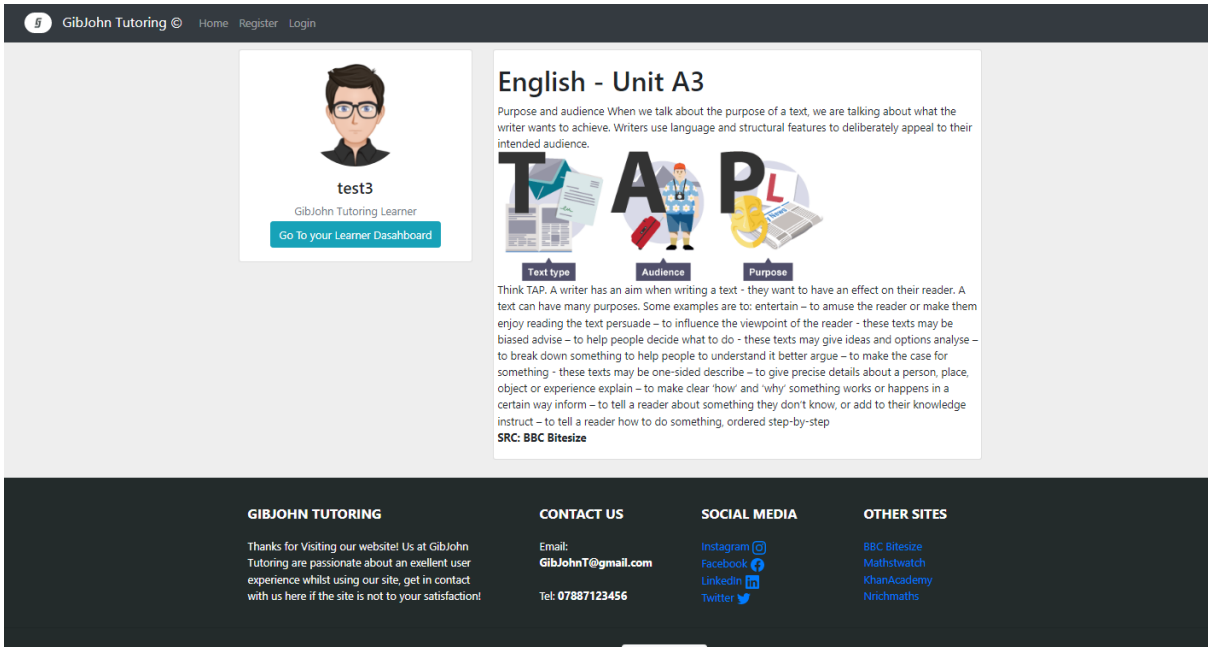


This is how blade creates the html and prints out the webpage in the views folder. Its variables can be stored and outputted here.

Learner Courses English 1,2,3 (Classes 1,2,3)

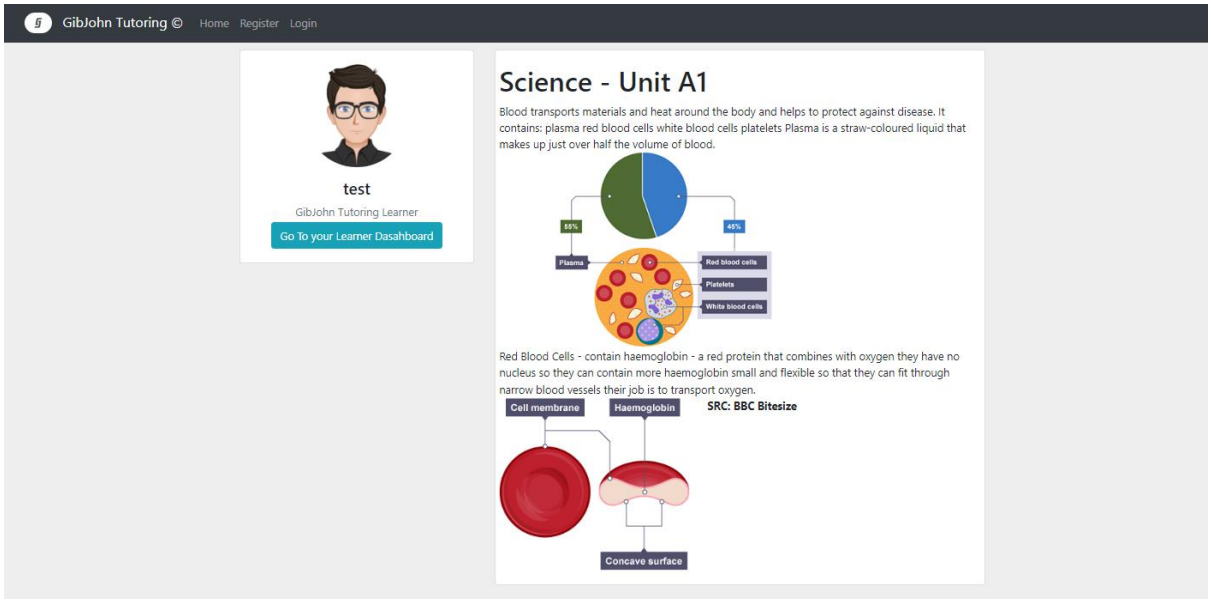


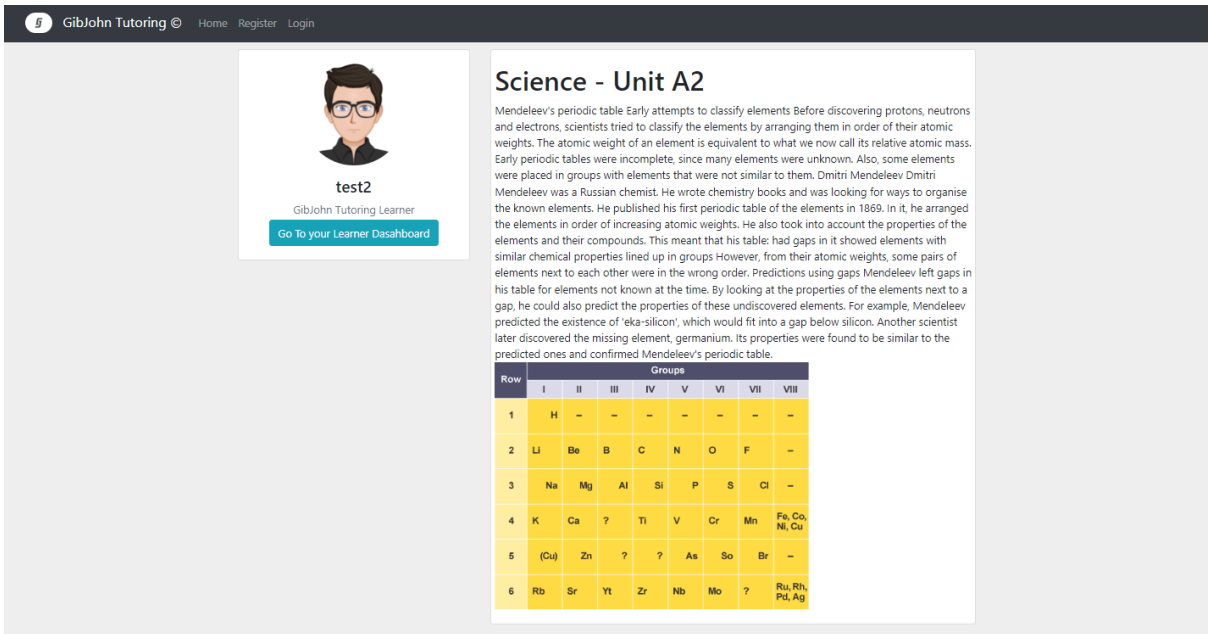


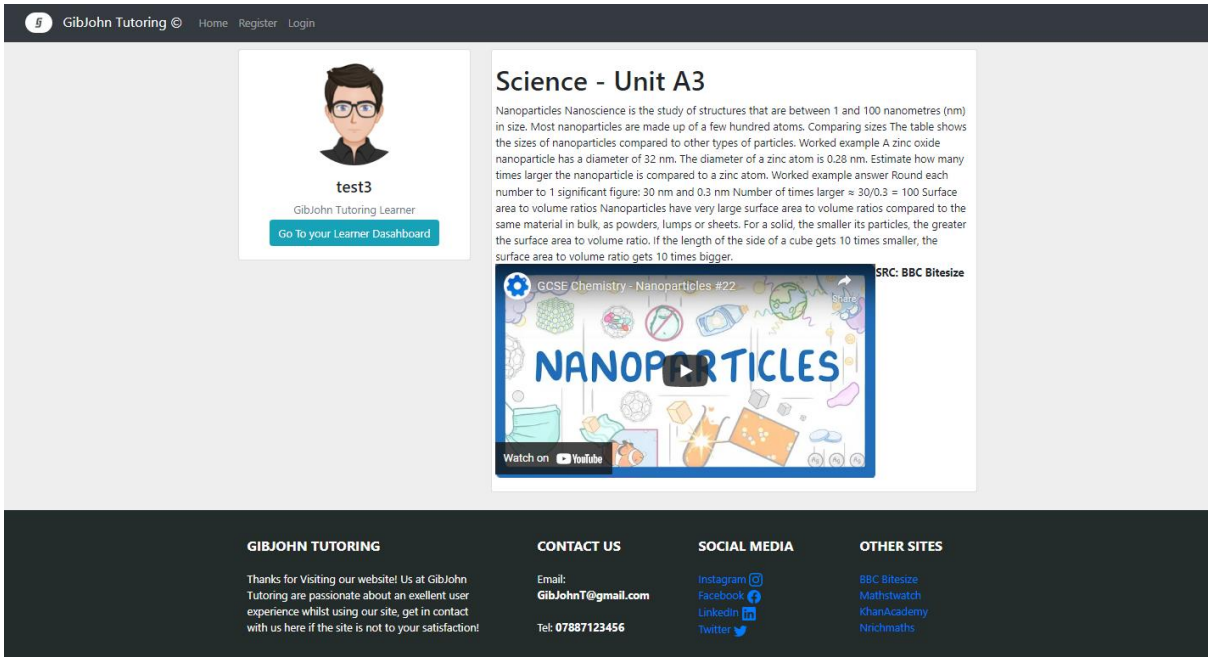


These are the 3 English content pages for the different learners. They are hard coded and are sourced from BBC Bitesize. They are all templated off the 'app' as I have named it. There is not much to add about these pages they are relatively basic and are quite self-explanatory.

Learner Courses Science 1,2,3 (Classes 1,2,3)

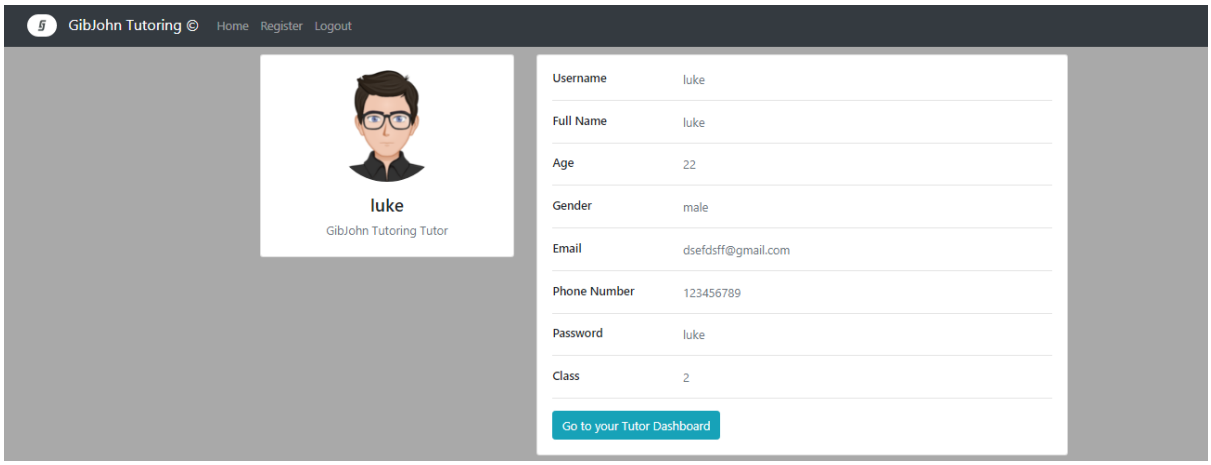






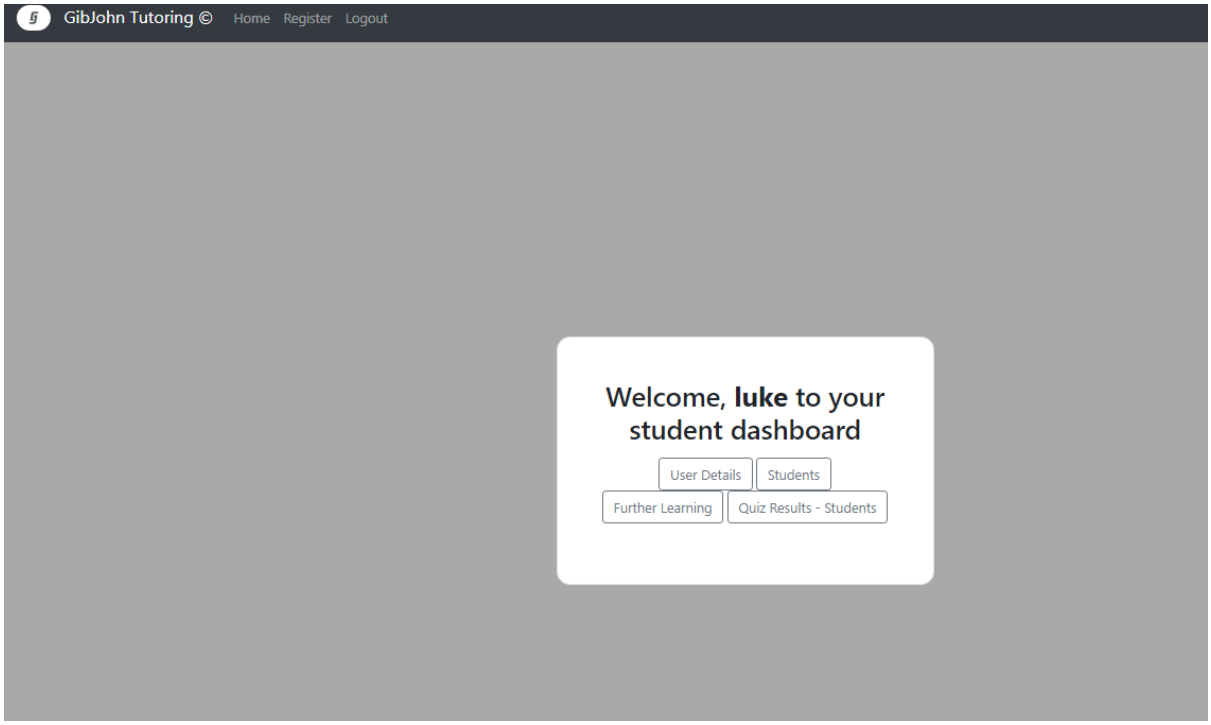
These are the 3 Science content pages for the different learners. They are hard coded and are sourced from BBC Bitesize. They are all templated off the ‘app’ as I have named it. There is not much to add about these pages they are relatively basic and are quite self-explanatory.

Tutor User Details



This is the tutor details page where the tutors’ details are drawn out of the database. This is a mirror copy of the learner user details only it query’s a different table in the database.

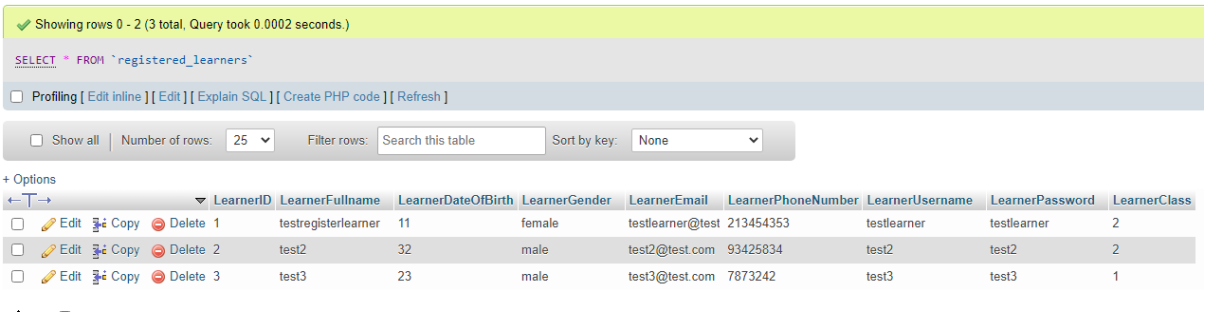
Tutor Dashboard

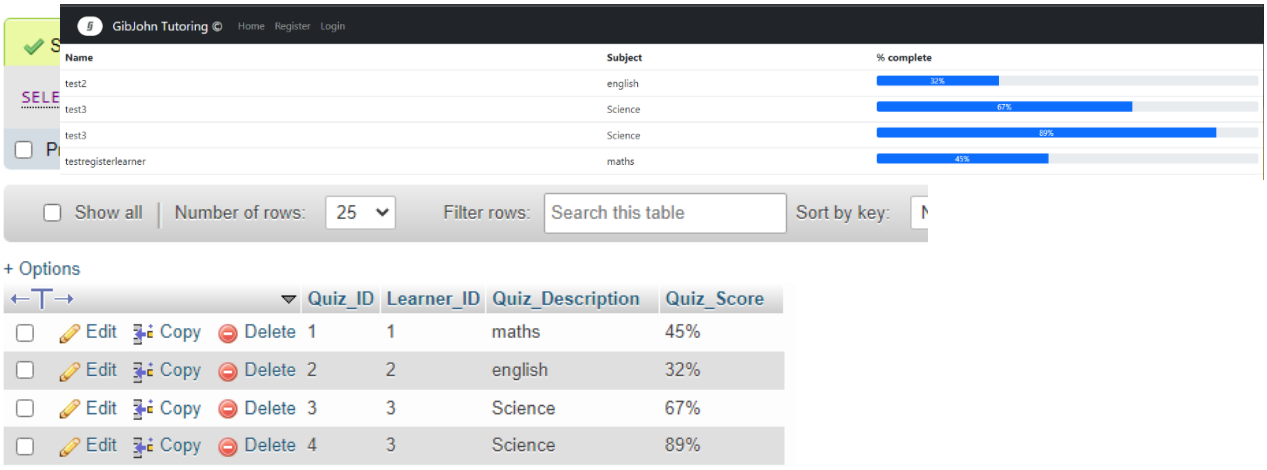


This is the tutor dashboard – it contains: user details, further learning and quiz results. The students and students quiz results have not been fully programmed yet. The quiz system is explained later in the document, but I can get the results out of the database and display them with progress bars, and I can do this for both learner and tutor views, however I have not managed to implement a quiz system that records and sends learner quiz results to the database. If there was a version 2 this would be implemented and have this functionality.

Learner Quizzes and progress.

I had planned to make a quiz at the end of each resource/content piece for the learners however I have run out of time due to fixing and figuring out other programming errors. I have developed along with research a way that results are drawn out of the database and shown in a progress bar in a table. This is a snippet of an example – It is missing the tutor session and bootstrap template as I was only trying to show you that I had tried to make something that supported and recorded student progress. The code and images for reference is shown below.



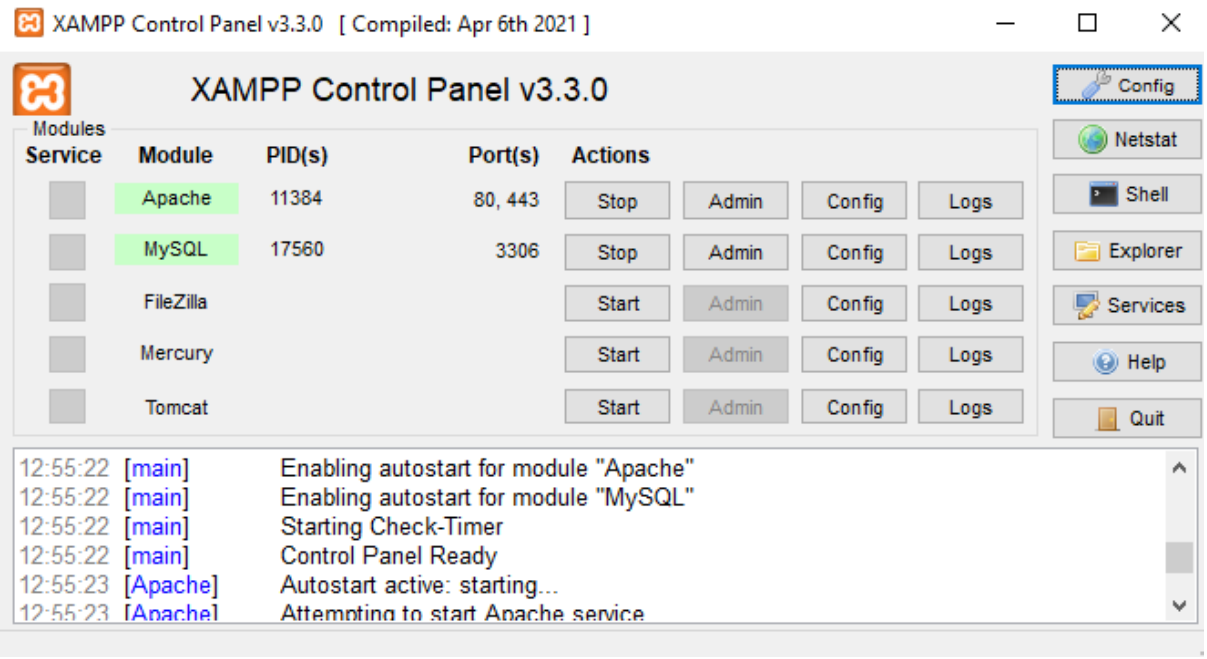



```
$sql = "

SELECT
registered_tutors.TutorID, registered_tutors.TutorFullname, registered_learners.LearnerFullname, quiz_results.Quiz_Description, quiz_results.Quiz_Score
from
registered_tutors
left join registered_learners on registered_learners.LearnerClass
left join quiz_results on quiz_results.Learner_ID = registered_learners.LearnerID
where
registered_tutors.TutorID = 1

order by registered_learners.LearnerFullname;

";
```

This Query joins 2 tables together the registered learners and the quiz results by the learner's class and learner ID. The table it makes is ordered by the learner's name alphabetically. As you can see an example of the learner id that was joined - 2 quizzes with ids 3 and 4 both appearing for science with 67% and 89% have the same learner ID of 3 so leaner 3 did two quizzes. And the learners 1 and 2 only did 1 quiz each.

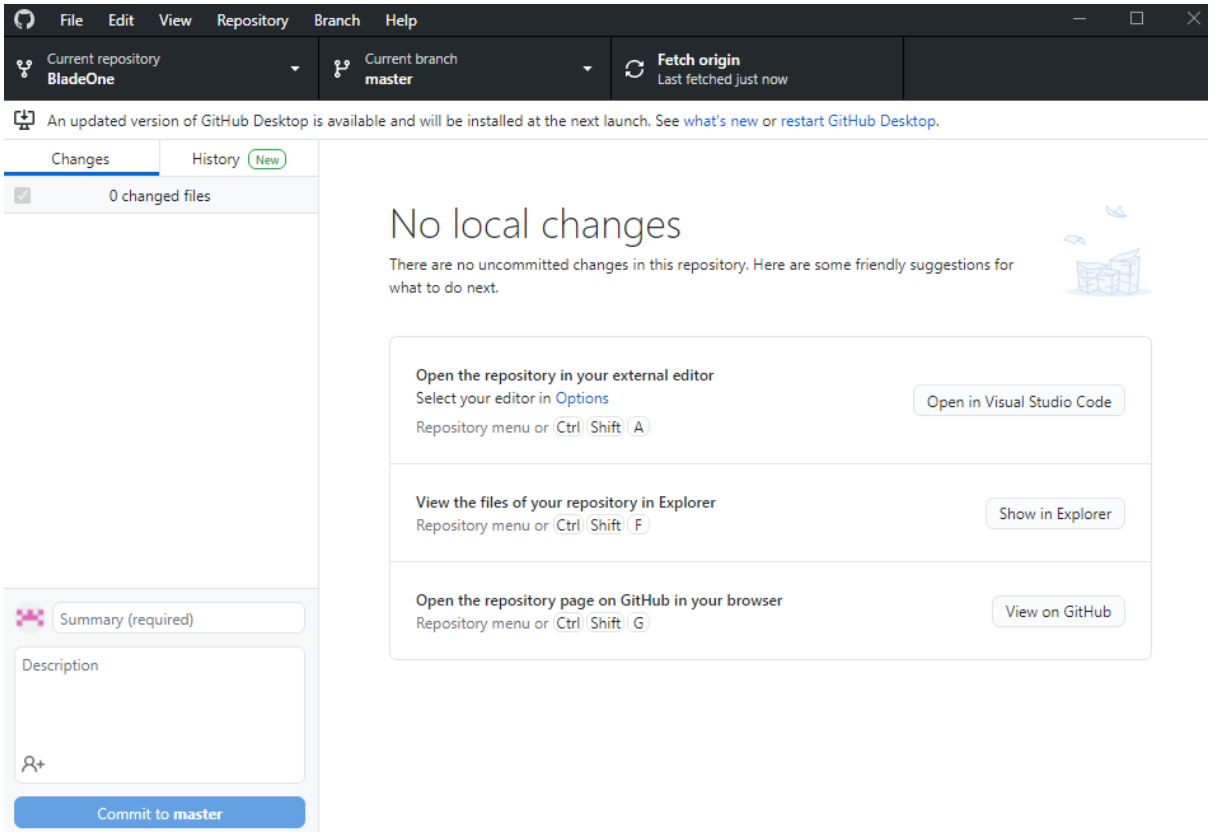
So, this code works, I just need to add a quiz system – like the register system – that takes data and posts them to the database. Then the results can be requested and viewed by both learner and tutor

Xampp



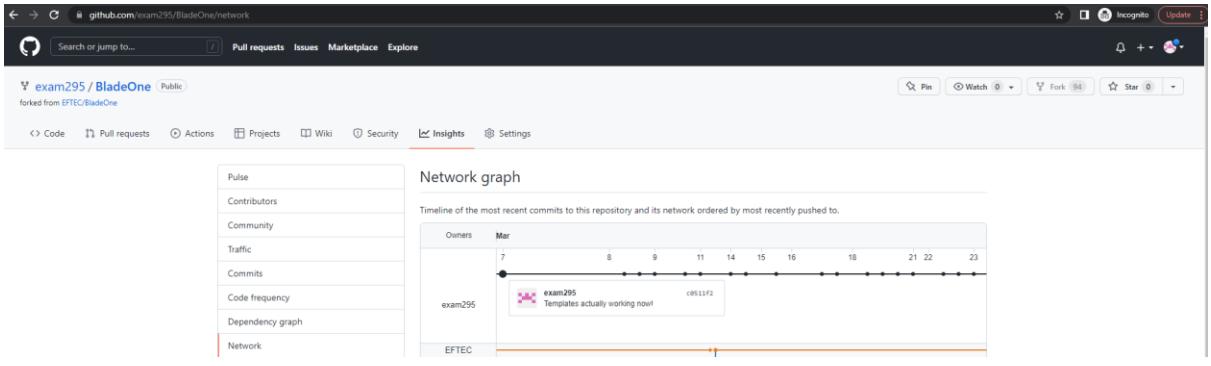
Xampp is a cross platform application that includes a range of software like: Apache, MYSQL, PHP Servers, and Perl. In simple terms it's a local (APACHE) web server on the computer its running on. I used Xampp due to its simplicity and high-quality performance at running back-end processes. The php and Apache server all interlink so you have a MYSQL Database that interlinks with the webserver, enabling me to write php and it be interpreted through Apache and if needed to perform operations with and within the MYSQL (phpMyAdmin) Database. Xampp was very easy to setup – just install and start up….

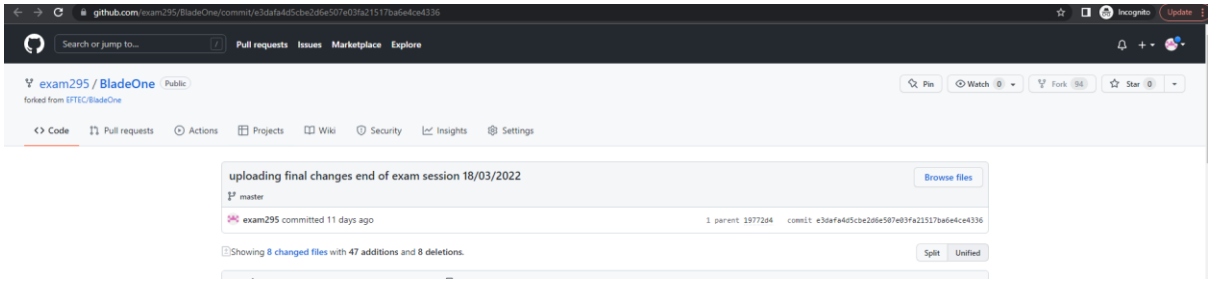
GitHub + GitHub Desktop



I used GitHub as it was a very efficient code hosting service and it allowed me to manage my commits and have full version control access to my development at any point during the journey.

I can view the entire commit history and see the changes I have made and progressed with through out the course of the past 4 weeks. The network graph would look more interesting that just 1 commit after another if I changed to another branch instead of master however, I only needed 1 branch for this exam.





Just an example image of one of the commits I did – 8 changed files with 47 additions and 8 deletions. If I was to scroll all the way down, you would see the full extend of file changes and code changes line for line.

Bootstrap

I have used plenty of bootstrap throughout my development as it is well designed and has plenty of predesigned templates and classes that are all open source and ready to be customised. All the templates look very professional, and with minimal styling code. Bootstrap Grid systems are well designed and laid out. Much easier than CSS grid systems.

Other Website Sources Used throughout this project.

Source Name	Source Description	Source Link
Bootstrap Footer	This is where I got inspiration for my bootstrap footer.	https://mdbootstrap.com/docs/b4/jquery/navigation/footer/
Bootstrap Icons	This is where I got my social media icons from.	https://icons.getbootstrap.com/
Bootstrap Buttons	This is where I used code to get bootstrap buttons from.	https://getbootstrap.com/docs/4.0/components/buttons/
Login System	This is where I used some sections of code and edited it personally to work with my database and table names.	https://codeshack.io/secure-login-system-php-mysql/
Bootstrap Forms	These sites are where I looked to gain more information on bootstrap forms.	https://getbootstrap.com/docs/4.6/components/forms/ https://getbootstrap.com/docs/5.0/forms/overview/
Bootstrap Navigation	This I where I got inspiration from for my bootstrap navigation bar.	https://www.w3schools.com/bootstrap5/bootstrap_navbar.php
Hex Colour CSS	This Is a tool I used to get the colours for my website	https://www.color-hex.com/color/232b2b

Testing

I am aware there is a testing test log template we can use; however, I have used my own template as I believe it includes what the Pearson testing template has and a bit more detail. I have included image evidence for all the tests that require it.

Testing Plans

Test Ref.	Type of Test	Purpose of Test	Expected Outcome	Test Data
1	Functional test	To test if the application takes in all the register data and records it in the Database (Learner)	On click of the submit button the data is posted to the database and the user is redirected to the login page.	Form Data: `LearnerFullname, LearnerDateOfBirth, LearnerGender, LearnerEmail, LearnerPhoneNumber, LearnerUsername, LearnerPassword, LearnerClass`

Test Ref.	Type of Test	Purpose of Test	Expected Outcome	Test Data

2	Functional test	To test if the application takes in all the register data and records it in the Database (Learner)	On click of the submit button the data is posted to the database and the user is redirected to the login page.	Form Data:
				`LearnerFullname,`
				`LearnerDateOfBirth,`
				`LearnerGender,`
				`LearnerEmail,`
				`LearnerPhoneNumber,`
				`LearnerUsername,`
				`LearnerPassword,`
				`LearnerClass`

Test Ref.	Type of Test	Purpose of Test	Expected Outcome	Test Data
3	Functional test	To test if program can login a user that has signed up (Learner)	Login details compared to ones in the database already.	Form Data: `LearnerUsername, LearnerPassword.`

Test Ref.	Type of Test	Purpose of Test	Expected Outcome	Test Data
4	Functional test	To test if program can login a user that has signed up (Learner).	Login details should be compared to the ones in the database already and login if details correct.	Form Data: `LearnerUsername, LearnerPassword.`

Test Ref.	Type of Test	Purpose of Test	Expected Outcome	Test Data
5	Functional Test	To test what the application does when incorrect data is submitted in the login page (learner + tutor)	IT should navigate to an error page informing the user of what they need to do.	Form Data: `LearnerUsername, LearnerPassword.`

Test Ref.	Type of Test	Purpose of Test	Expected Outcome	Test Data
6	Functional test	To test if the session is around on all of the pages once logged in.	I have included session start () at the top of my document on all relevant pages after login so it should be there.	Check all pages for a visible session and its relevant data (f12 and check network cookie/session area).

Test Ref.	Type of Test	Purpose of Test	Expected Outcome	Test Data
7	Functional test	To test if the correct data is displayed for the user logged in (Learner).	The session started should provide enough information for the queries I have written to draw the correct data out for each user.	Form Data: `LearnerUsername, LearnerPassword.`

Test Ref.	Type of Test	Purpose of Test	Expected Outcome	Test Data
8	Functional Test	To test if the correct data is displayed for the user logged in (Tutor).	The session started should provide enough information for the queries I have written to draw the correct data out for each user.	Form Data: `LearnerUsername, LearnerPassword.`

Test Ref.	Type of Test	Purpose of Test	Expected Outcome	Test Data
9	Functional Test	Learner Courses are different depending on the class value. (Checking the logical operation).	IF a learner class is 1 then they see A1 content, IF a learner class is 2 they see A2 Content, IF a learner class is 3 they can see A3.	Learner Class and the if statement in the maths, English and science redirect pages.

Test Ref.	Type of Test	Purpose of Test	Expected Outcome	Test Data
10	Functional Test	Test if the progress bar values on the tutor view of students are drawing in the correct values.	The progress bars will use values from the database and will represent 1 student each. They are in a table, so the student and progress bar are in the same row.	Quiz Result

Test Ref.	Type of Test	Purpose of Test	Expected Outcome	Test Data
11	Functional Test	Test incorrect or missing data in the register forms.	Should redirect to an error page.	Email – ed.com (no@) Age – eeeee Etc.

Test Ref.	Type of Test	Purpose of Test	Expected Outcome	Test Data
12	Usability – non functional	An easy-to-use program that is fast and efficient.	Should be useable and very quick. /Responsive as it is not a large program that uses lots of resources.	N/A

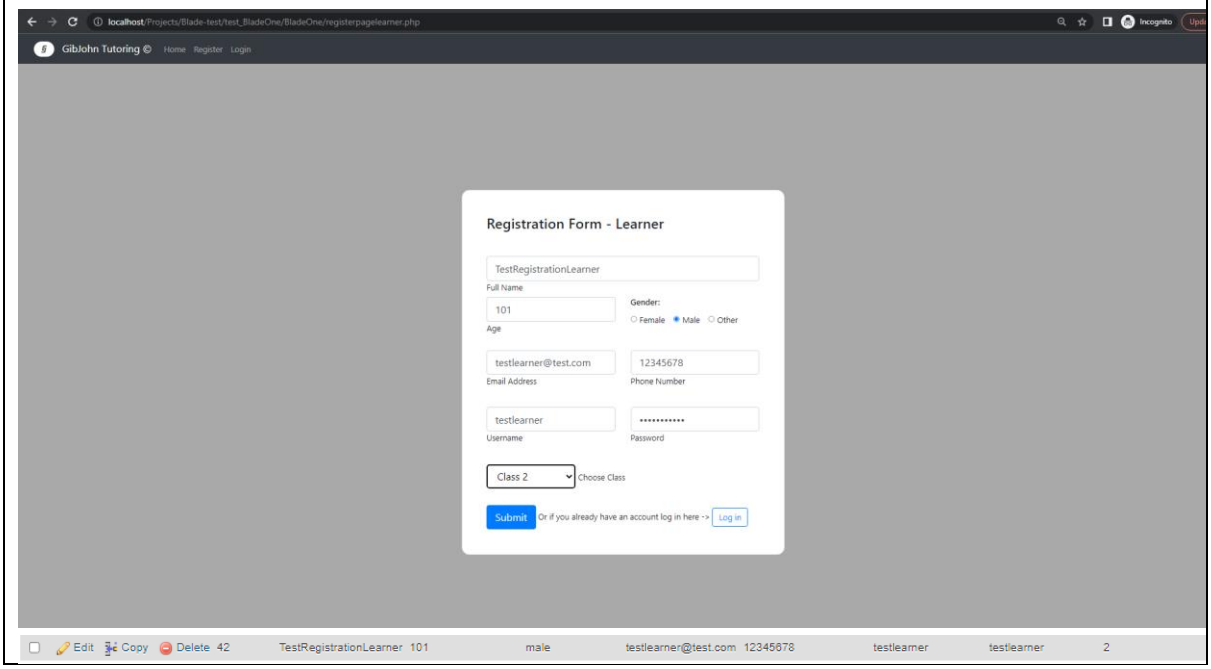
Test Ref.	Type of Test	Purpose of Test	Expected Outcome	Test Data
13	Non-Functional	The program should have a professional looking interface that follows good user interface and experience principles.	The interface is greyscale which should be readable for most users and the user interface should keep a consistent format across all the pages. Templating will help with this.	N/A

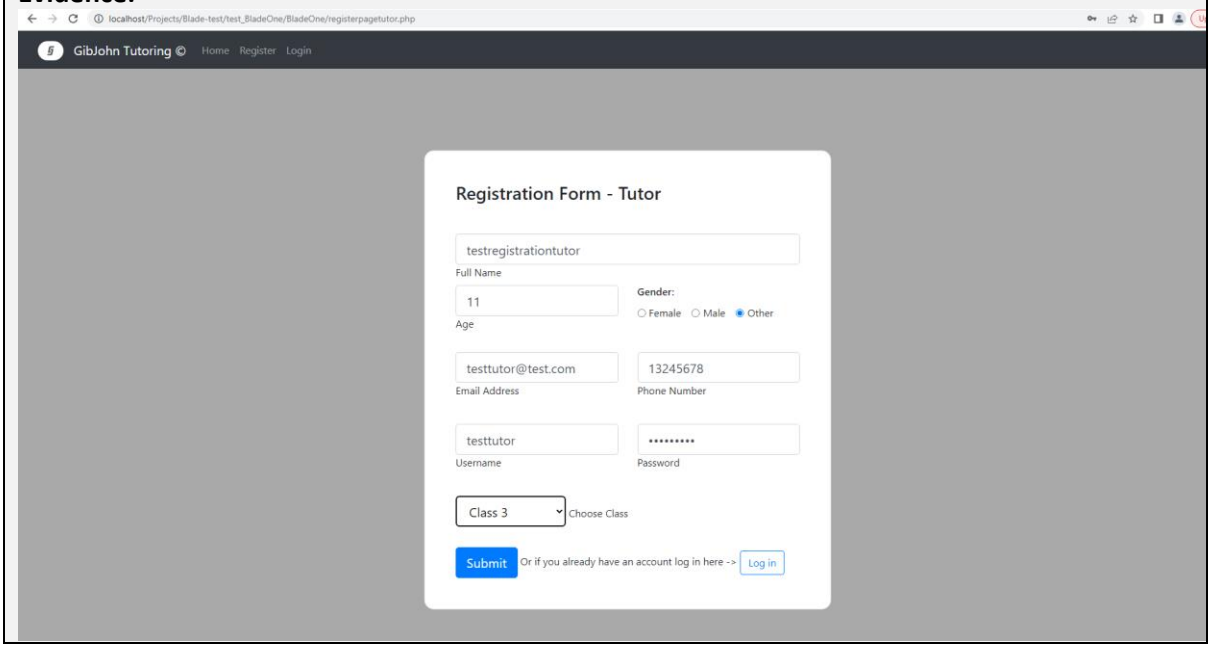
Test Ref.	Type of Test	Purpose of Test	Expected Outcome	Test Data
14	Functional	To test if the application can be used the same on mobile devices and smaller devices than the average 24-inch computer screen.	The bootstrap should automatically size adjust depending on the size. Not entirely sure of what every page will look like.	N/A

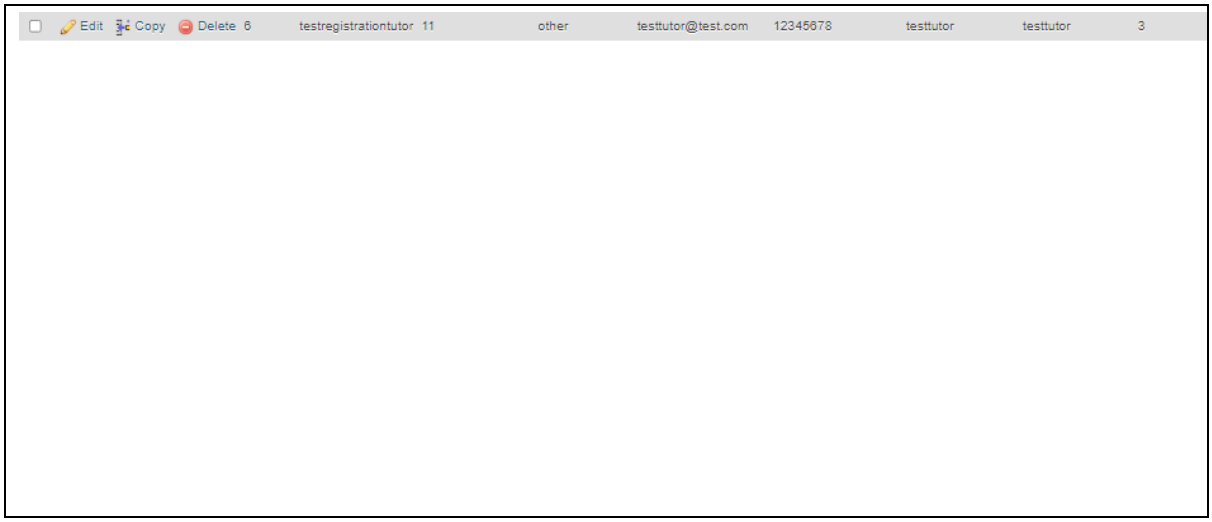
Test Ref.	Type of Test	Purpose of Test	Expected Outcome	Test Data
15	Functional	Test if the applications templates work and extend the different relevant sections.	The templates should be filled with header footer and content sections. Non views can generate html and the echo just prints out the webpages in views.	@extends('app') @section('content') @endsection

Testing Logs

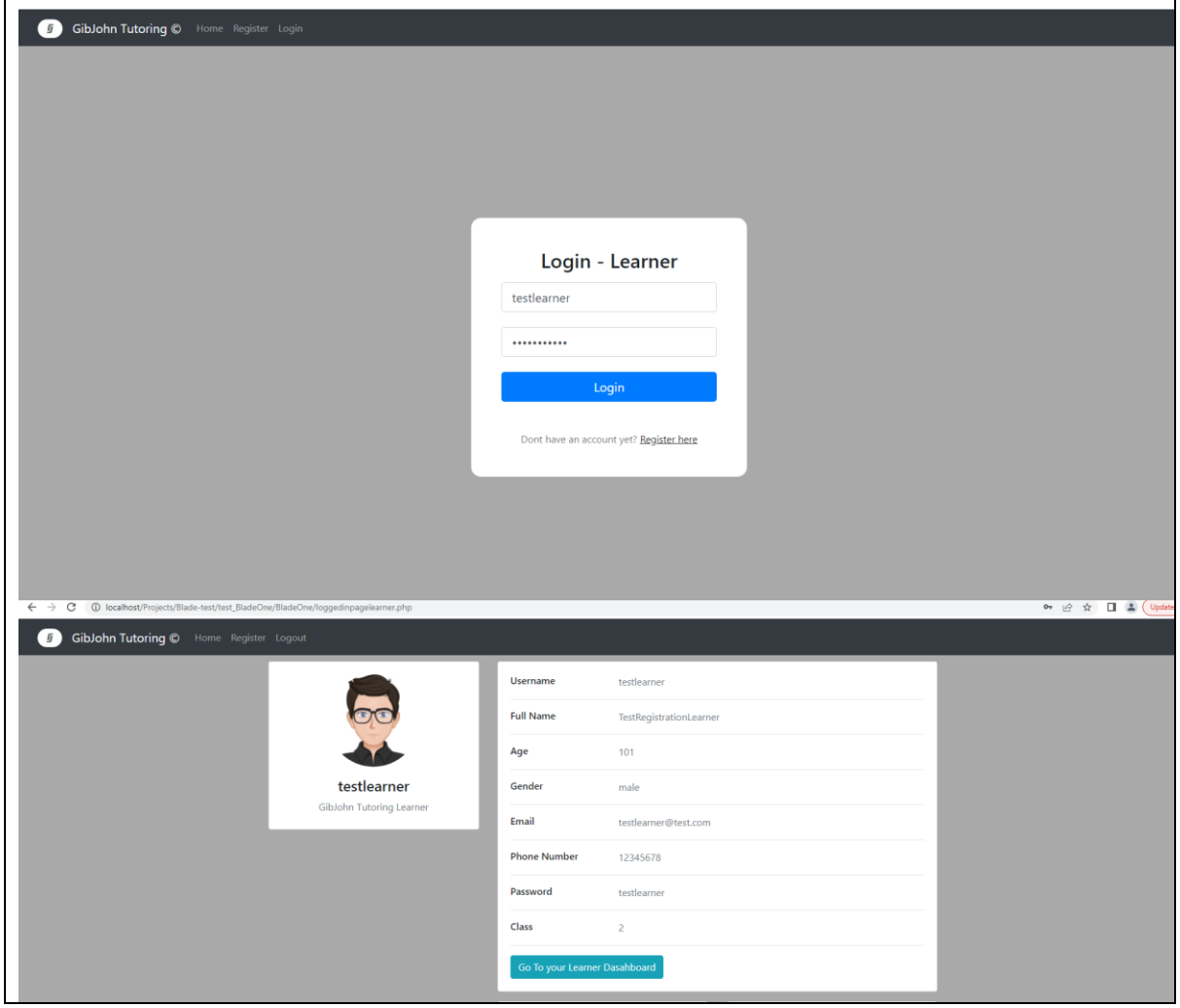
Test Reference:	1	**Name of Tester:**	Ed	**Date of Test:**	29/03/2022
Purpose of Test:	To test if the application takes in all the register data and records it in the Database (Learner)				
Test Type:	Functional TEST				
Test Data:	Form Data: `LearnerFullname, LearnerDateOfBirth, LearnerGender, LearnerEmail, LearnerPhoneNumber, LearnerUsername, LearnerPassword, LearnerClass`				
Expected Outcome:	On click of the submit button the data is posted to the database and the user is redirected to the login page.				
Actual Outcome:	As Expected.				
Pass/Fail/Skipped:	PASS				
Comments and/or Recommendations:	N/A				
Evidence:					



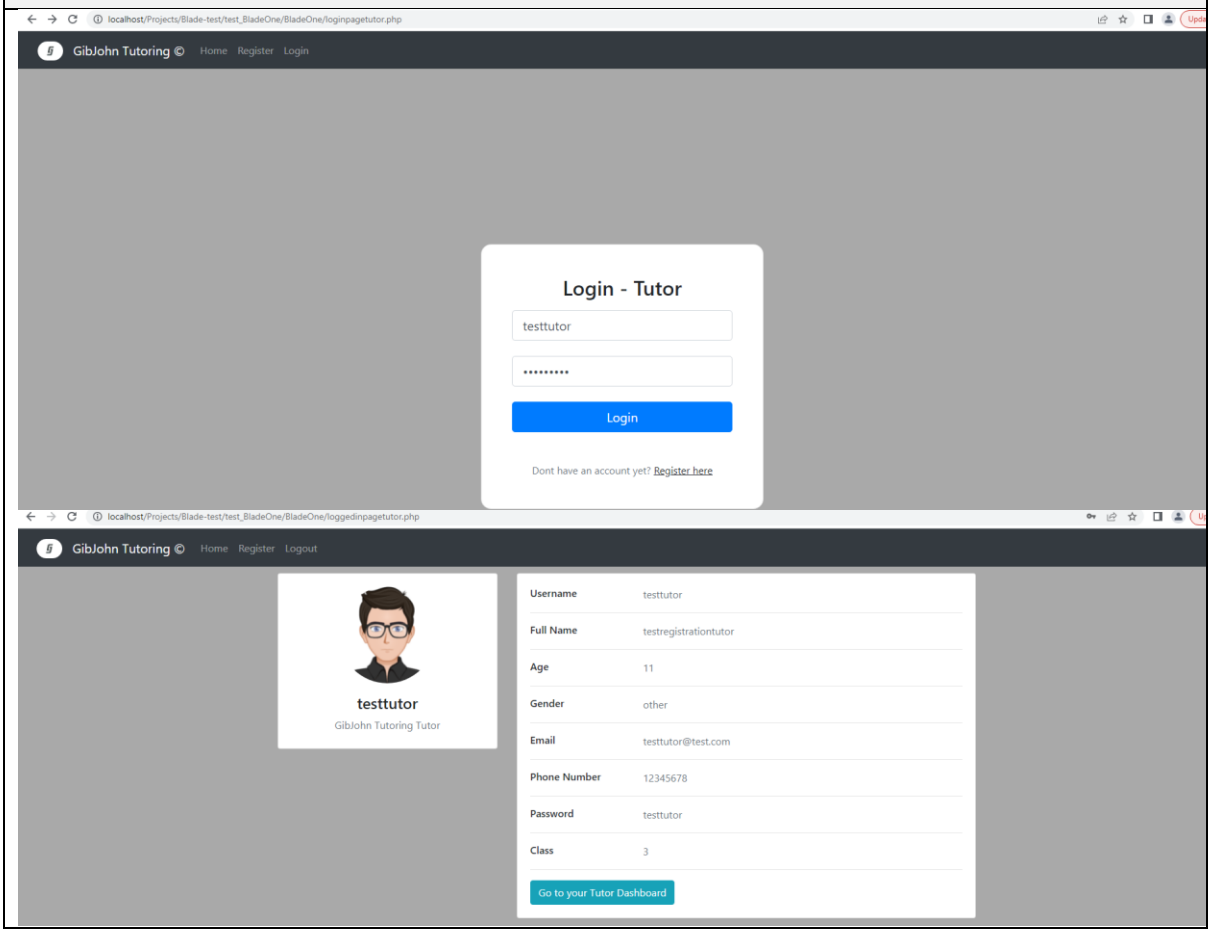
Test Reference:	2	**Name of Tester:**	Ed	**Date of Test:**	29/03/2022
Purpose of Test:	To test if the application takes in all the register data and records it in the Database (Tutor)				
Test Type:	Functional test				
Test Data:	Form Data: `LearnerFullname, LearnerDateOfBirth, LearnerGender, LearnerEmail, LearnerPhoneNumber, LearnerUsername, LearnerPassword, LearnerClass`				
Expected Outcome:	On click of the submit button the data is posted to the database and the user is redirected to the login page.				
Actual Outcome:	As Expected.				
Pass/Fail/Skipped:	PASS				
Comments and/or Recommendations:	N/A				
Evidence:					



Test Reference:	3	**Name of Tester:**	Ed	**Date of Test:**	29/03/2022
Purpose of Test:	To test if program can login a user that has signed up (Learner)				
Test Type:	Functional test				
Test Data:	Form Data: `LearnerUsername, LearnerPassword.`				
Expected Outcome:	Login details compared to ones in the database already and the user is logged in				
Actual Outcome:	As Expected – All works.				
Pass/Fail/Skipped:	PASS				
Comments and/or Recommendations:					
Evidence:					

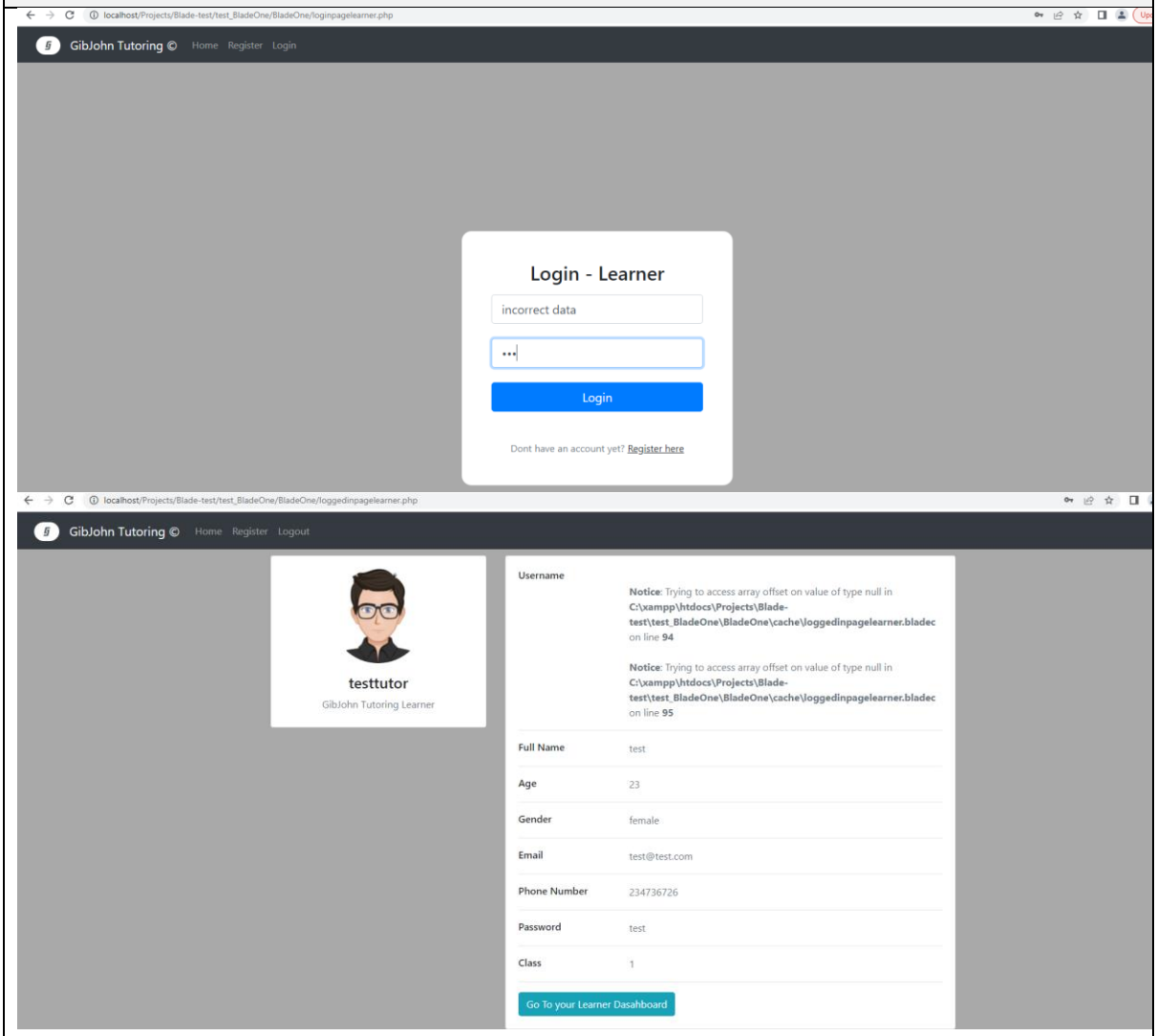

GibJohn Tutoring ©
Home
Register
Login
Login - Learner
testlearner
Login
Dont have an account yet? Register here
localhost/Projects/Blade-test/test_BladeOne/BladeOne/loggedinpagelearner.php
GibJohn Tutoring ©
Home
Register
Logout
testlearner
GibJohn Tutoring Learner
Username
testlearner
Full Name
TestRegistrationLearner
Age
101
Gender
male
Email
testlearner@test.com
Phone Number
12345678
Password
testlearner
Class
2
Go To your Learner Dasahboard

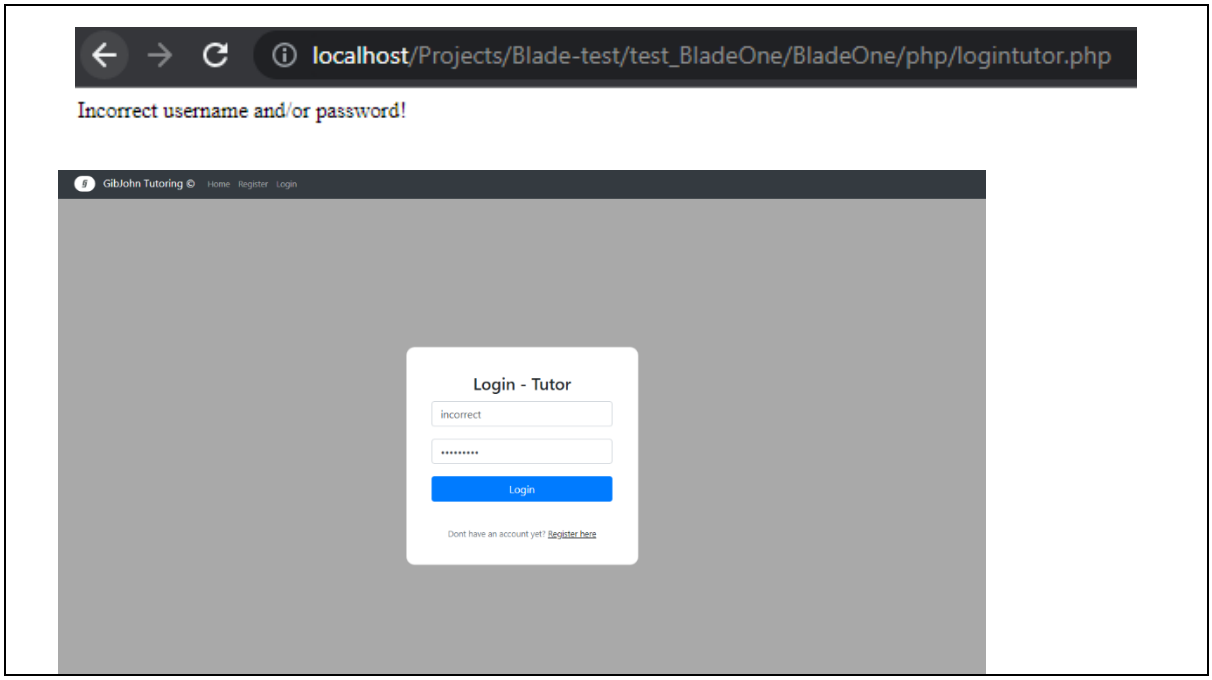
Test Reference:	4	**Name of Tester:**	Ed	**Date of Test:**	29/03/2022
Purpose of Test:	To test if program can login a user that has signed up (Tutor).				
Test Type:	Functional				
Test Data:	Form Data: `LearnerUsername, LearnerPassword.`				
Expected Outcome:	Login details should be compared to the ones in the database already and login if details correct.				
Actual Outcome:	As Expected – all working				
Pass/Fail/Skipped:	PASS				
Comments and/or Recommendations:	N/A				
Evidence:					



Test Reference:	5	**Name of Tester:**	Ed	**Date of Test:**	29/03/2022
Purpose of Test:	To test what the application does when incorrect data is submitted in the login page (learner + tutor)				
Test Type:	Functional				
Test Data:	Form Data: `LearnerUsername, LearnerPassword`				
Expected Outcome:	It should navigate to an error page informing the user of what they need to do.				
Actual Outcome:	For the Learner it proceeds to the learner details screen and does not error for some unknown reason. Somehow it logs in. I think the session in this case had not been ended. For the tutor it redirects to a page where it says incorrect name or password.				
Pass/Fail/Skipped:	Fail				
Comments and/or Recommendations:	Need to end the previous user session and add the redirect to error page for the learner.				

Evidence:



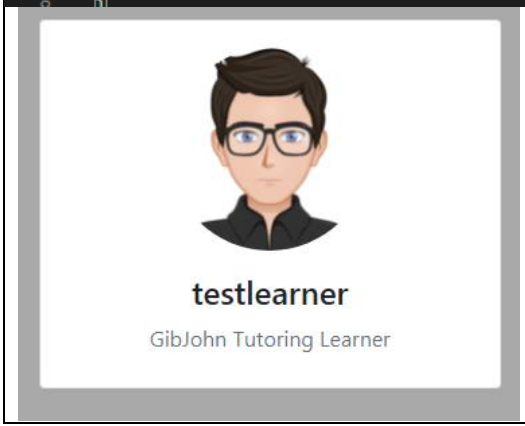

localhost/Projects/Blade-test/test_BladeOne/BladeOne/php/logintutor.php
Incorrect username and/or password!
GibJohn Tutoring ©
Home
Register
Login
Login - Tutor
incorrect
Login
Dont have an account yet? Register here

Test Reference:	6	Name of Tester:	Ed	Date of Test:	29/03/2022
Purpose of Test:	To test if the session is around on all of the pages once logged in.				
Test Type:	Functional – White Box testing				
Test Data:	Check all pages for a visible session and its relevant data (f12 and check network cookie/session area).				
Expected Outcome:	I have included session start () at the top of my document on all relevant pages after login so it should be there.				
Actual Outcome:	The Session is available over all relevant webpages once logged in. This will work wherever Session_start is, up until the session is ended/destroyed.				
Pass/Fail/Skipped:	PASS				
Comments and/or Recommendations:	N/A				
Evidence:					

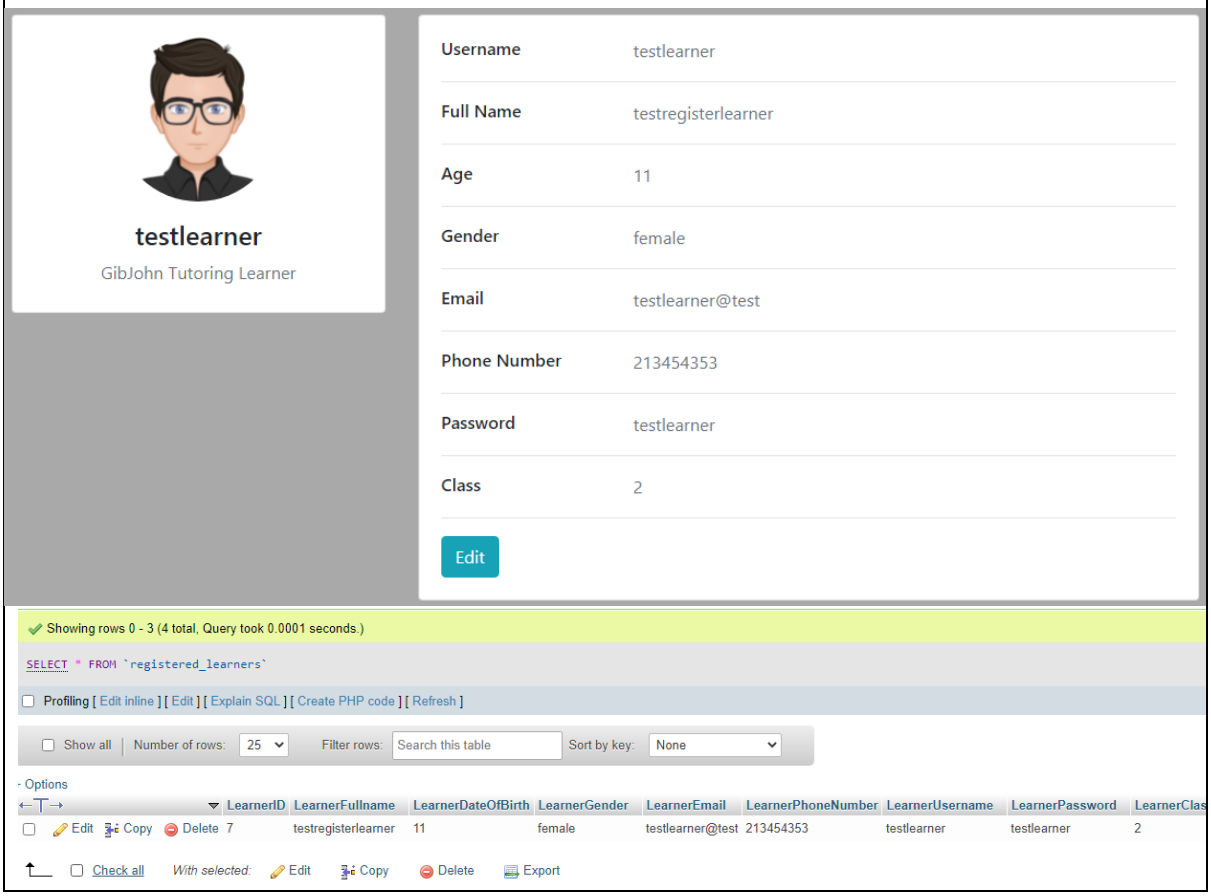
```php
if ($_POST['password'] === $password) {
    // Verification success! User has logged-in!
    // Create sessions, so we know the user is logged in,= it acts like cookies but remembers a bit the data on the server.
    session_regenerate_id();
    $_SESSION['loggedin'] = TRUE;
    $_SESSION['name'] = $_POST['username'];
    $_SESSION['id'] = $id;
```

views > loggedinpagelearner.blade.php

```php
<?php
// We need to use sessions, so you should always start sessions using th
session_start();
// If the user is not logged in redirect to the login page...
if (!isset($_SESSION['loggedin'])) {
    //header('Location: index.html');
    exit;
```

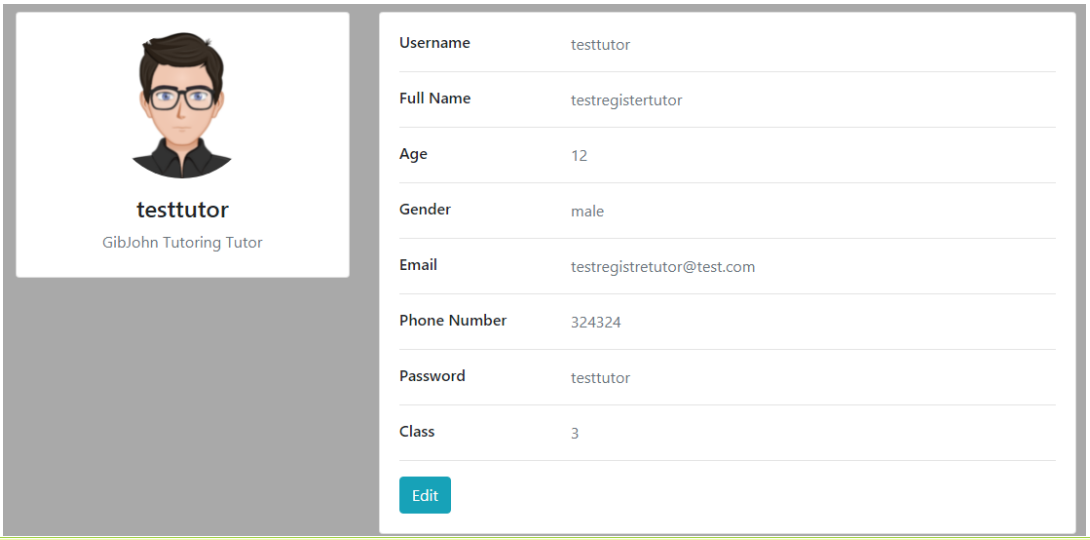


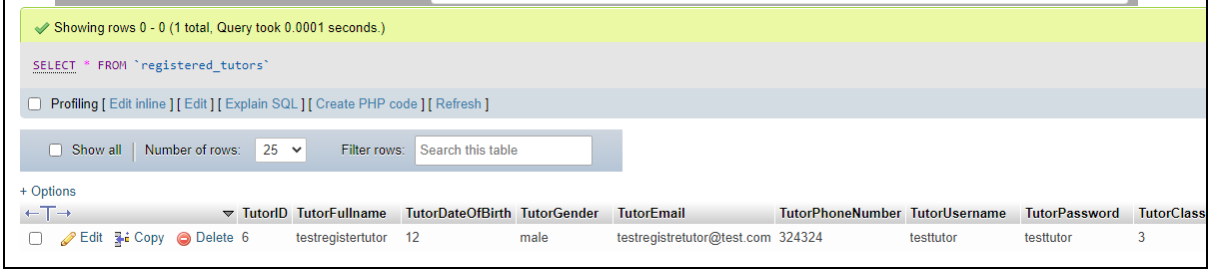
Test Reference:	7	Name of Tester:	Ed	Date of Test:	29/03/2022
Purpose of Test:	To test if the correct data is displayed for the user logged in (Learner).				
Test Type:	Functional – Whitebox				
Test Data:	Form Data: `LearnerFullname, LearnerDateOfBirth, LearnerGender, LearnerEmail, LearnerPhoneNumber, LearnerUsername, LearnerPassword, LearnerClass`				
Expected Outcome:	The session started should provide enough information for the queries I have written to draw the correct data out for each user.				
Actual Outcome:	The correct data for the learner view comes out of the database. I have included an image of the database data and the front end view of the process for for the user.				
Pass/Fail/Skipped:	PASS				
Comments and/or Recommendations:					
Evidence:					



Test Reference:	8	Name of Tester:	Ed	Date of Test:	29/03/2022
Purpose of Test:	To test if the correct data is displayed for the user logged in (Tutor).				
Test Type:	Functional - WhiteBox				
Test Data:	Form Data: `LearnerFullname, LearnerDateOfBirth, LearnerGender, LearnerEmail, LearnerPhoneNumber, LearnerUsername, LearnerPassword, LearnerClass`				
Expected Outcome:	The session started should provide enough information for the queries I have written to draw the correct data out for each user. Should display all data listed correctly for the user.				
Actual Outcome:	It does as expected - correct data is being drawn out of the database.				
Pass/Fail/Skipped:	PASS				
Comments and/or Recommendations:					

Evidence:



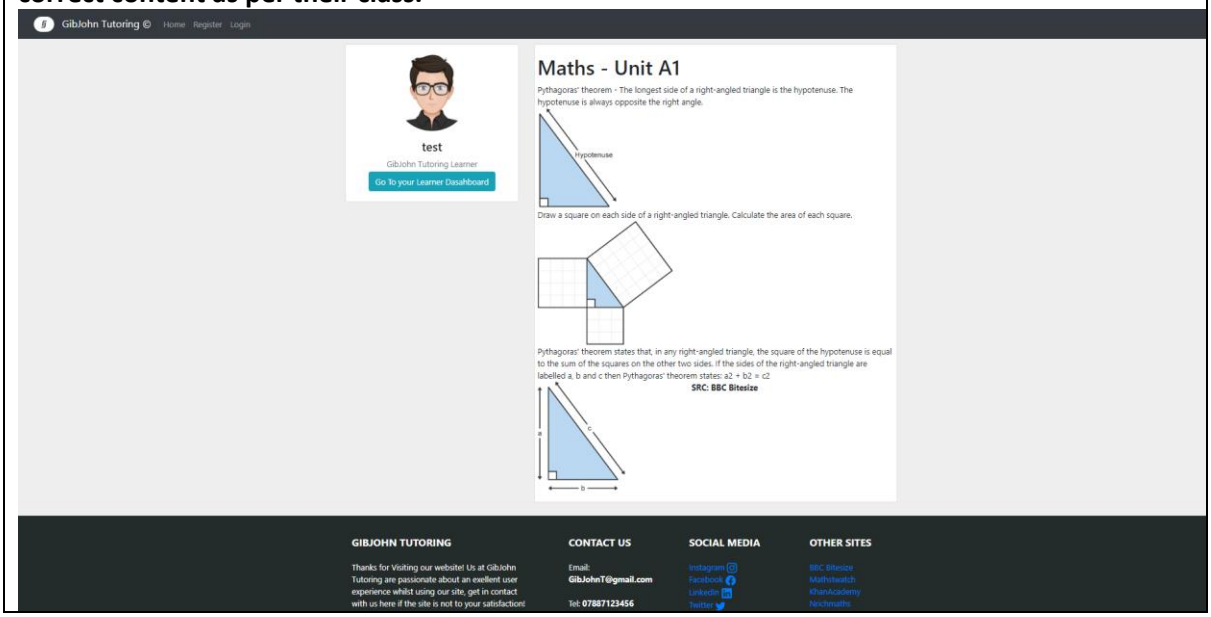


Test Reference:	9	**Name of Tester:**	Ed	**Date of Test:**	29/03/2022
Purpose of Test:	Learner Courses are shown different content depending on the class value. (Checking the logical operation).				
Test Type:	Functional – White Box				
Test Data:	Learner Class and the if statement in the maths, English and science redirect pages.				
Expected Outcome:	If a learner class is 1 then they see A1 content, IF a learner class is 2, they see A2 Content, IF a learner class is 3 they can see A3.				
Actual Outcome:	The content displayed is correct for all learners.				
Pass/Fail/Skipped:	PASS				
Comments and/or Recommendations:	If more content is added then you need to make separate redirect pages, with the same logic as the code below.				

Evidence:

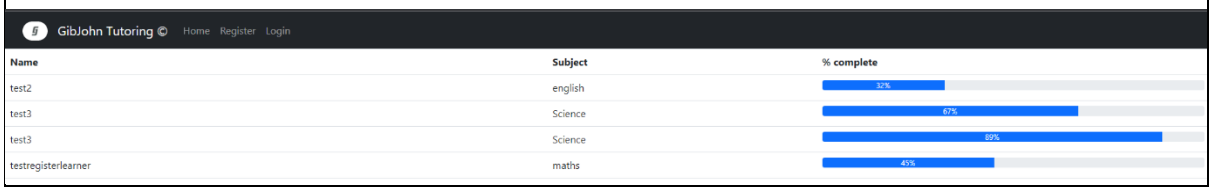
```
// this is the actual logic behind it - just if and elseif statements.
if($lc == "1"){
  header("Location:/Projects/Blade-test/test_BladeOne/BladeOne/maths1learner.php");
} else if($lc == "2") {
  header("Location:/Projects/Blade-test/test_BladeOne/BladeOne/maths2learner.php");
} else {
header("Location:/Projects/Blade-test/test_BladeOne/BladeOne/maths3learner.php");
}
```

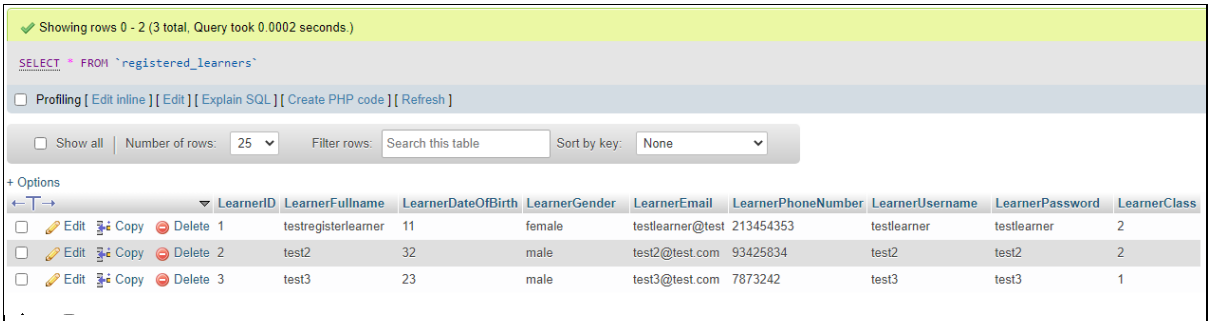
Learner class of 1 sees A1 of courses. This is the same for science and English. The learners see correct content as per their class.

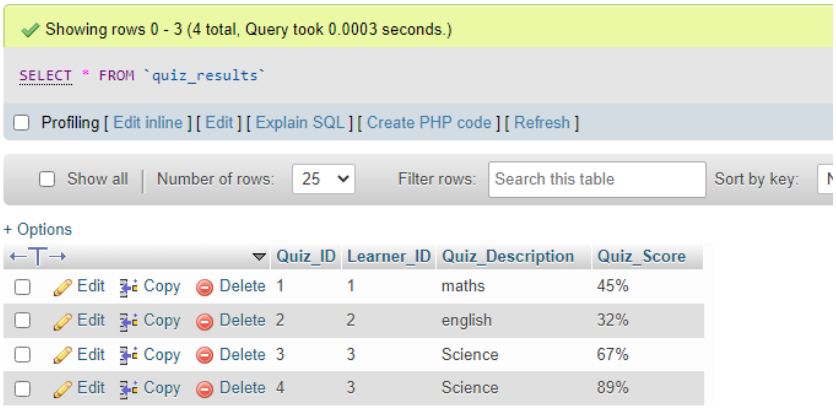


Test Reference:	10	Name of Tester:	Ed	Date of Test:	29/03/2022
Purpose of Test:	Test if the progress bar values on the tutor view of students are drawing in the correct values.				
Test Type:	Functional - Whitebox				
Test Data:	Quiz Result				
Expected Outcome:	The progress bars will use values from the database and will represent 1 student each. They are in a table, so the student and progress bar are in the same row.				
Actual Outcome:	They do. The table shows all learners and how many quizzes each one did with the progress bars.				
Pass/Fail/Skipped:	PASS				
Comments and/or Recommendations:	Could add more data and see how it copes but this is enough to get the gist of it. Also, as this is not a fully coded feature its purely developmental.				
Evidence:					

student progress. The code and images for reference is shown below.



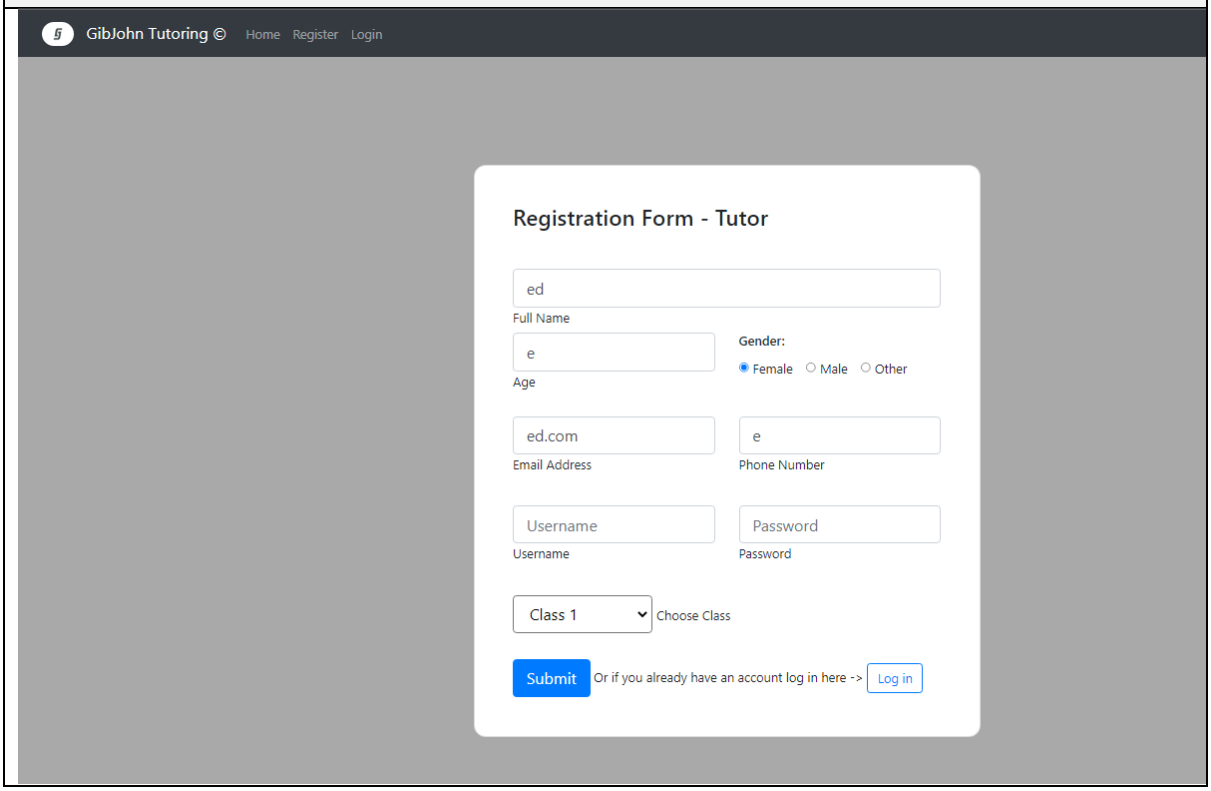


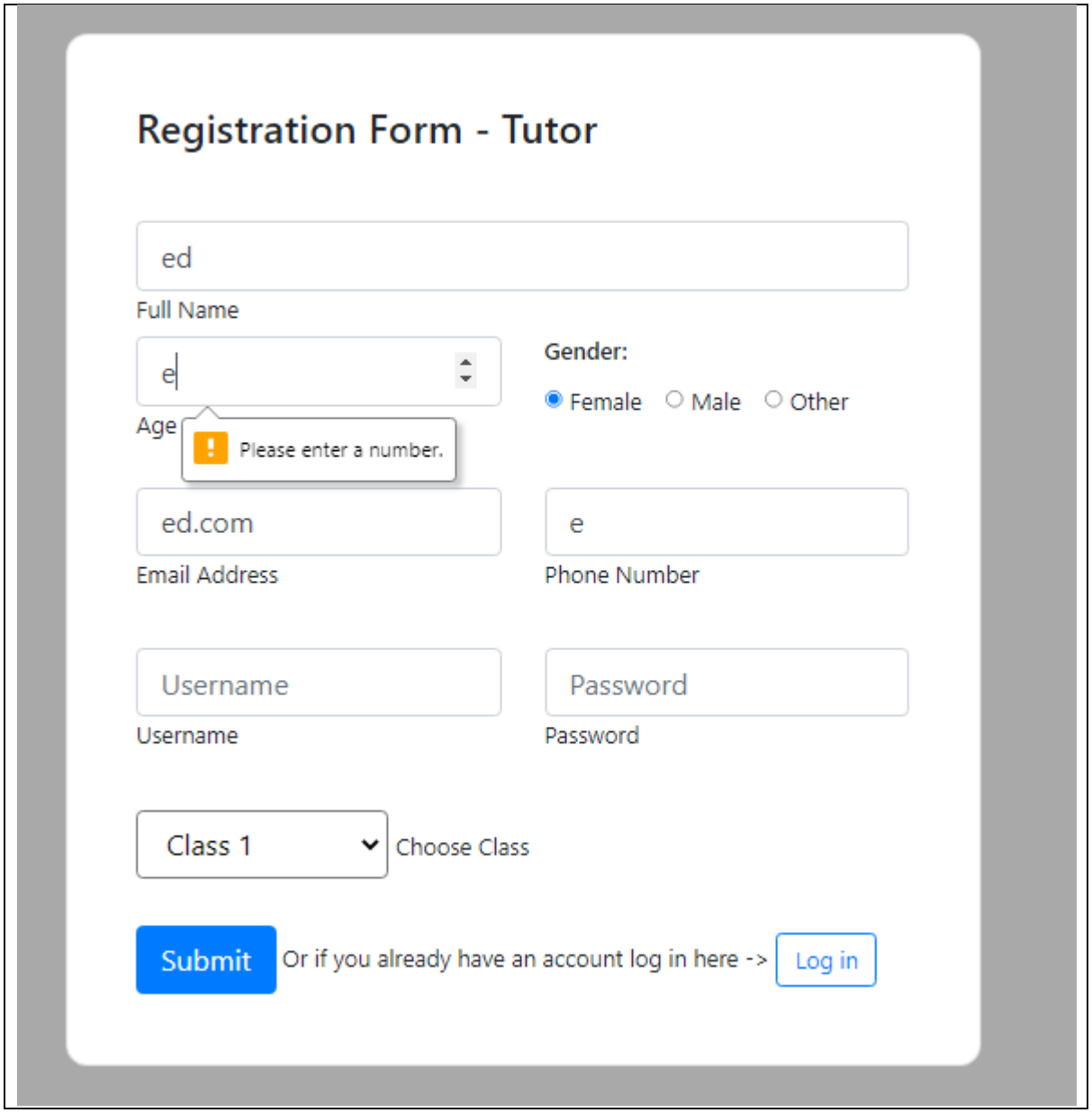


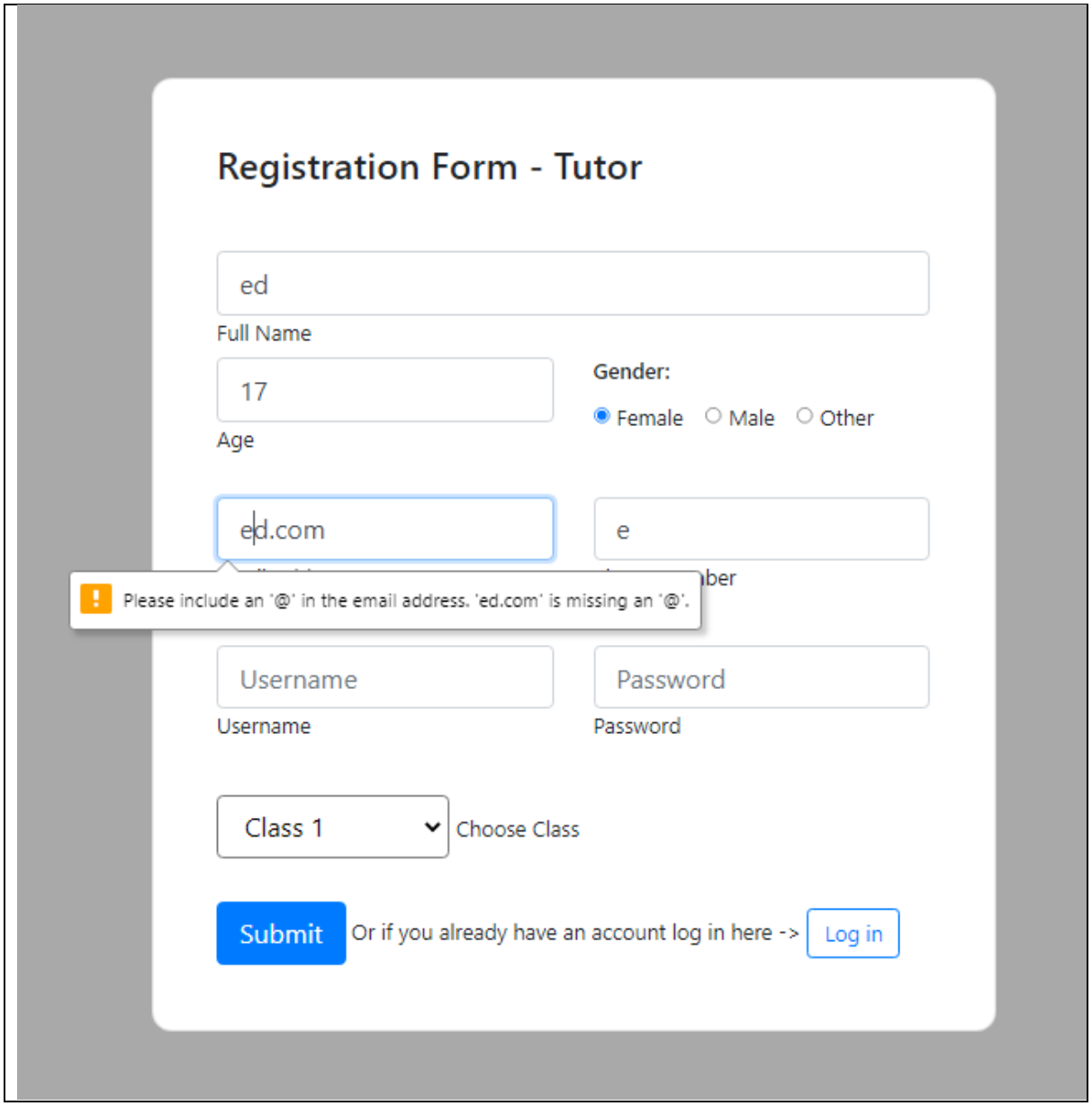

```
$sql = "

SELECT
registered_tutors.TutorID, registered_tutors.TutorFullname, registered_learners.LearnerFullname, quiz_results.Quiz_Description, quiz_results.Quiz_Scor
from
registered_tutors
left join registered_learners on registered_learners.LearnerClass
left join quiz_results on quiz_results.Learner_ID = registered_learners.LearnerID
where
registered_tutors.TutorID = 1

order by registered_learners.LearnerFullname;

";
```

The query is explained in the pages above.

Test Reference:	11	**Name of Tester:**	Ed	**Date of Test:**	02/02/2022
Purpose of Test:	Test incorrect or missing data in the register forms.				
Test Type:	Functional - Whitebox				
Test Data:	Email – ed.com (no@) Age – e				
Expected Outcome:	Should redirect to an error page.				
Actual Outcome:	Works correctly, outputs a relevant message and the program does not crash etc. It's a built in pre-programmed response – really efficient as the form will not submit until at least the age and email address are validated.				
Pass/Fail/Skipped:	PASS				
Comments and/or Recommendations:	For the next version I would program in red outlines on fields that remain empty and make it impossible for the form to submit until data is filled in.				
Evidence:					

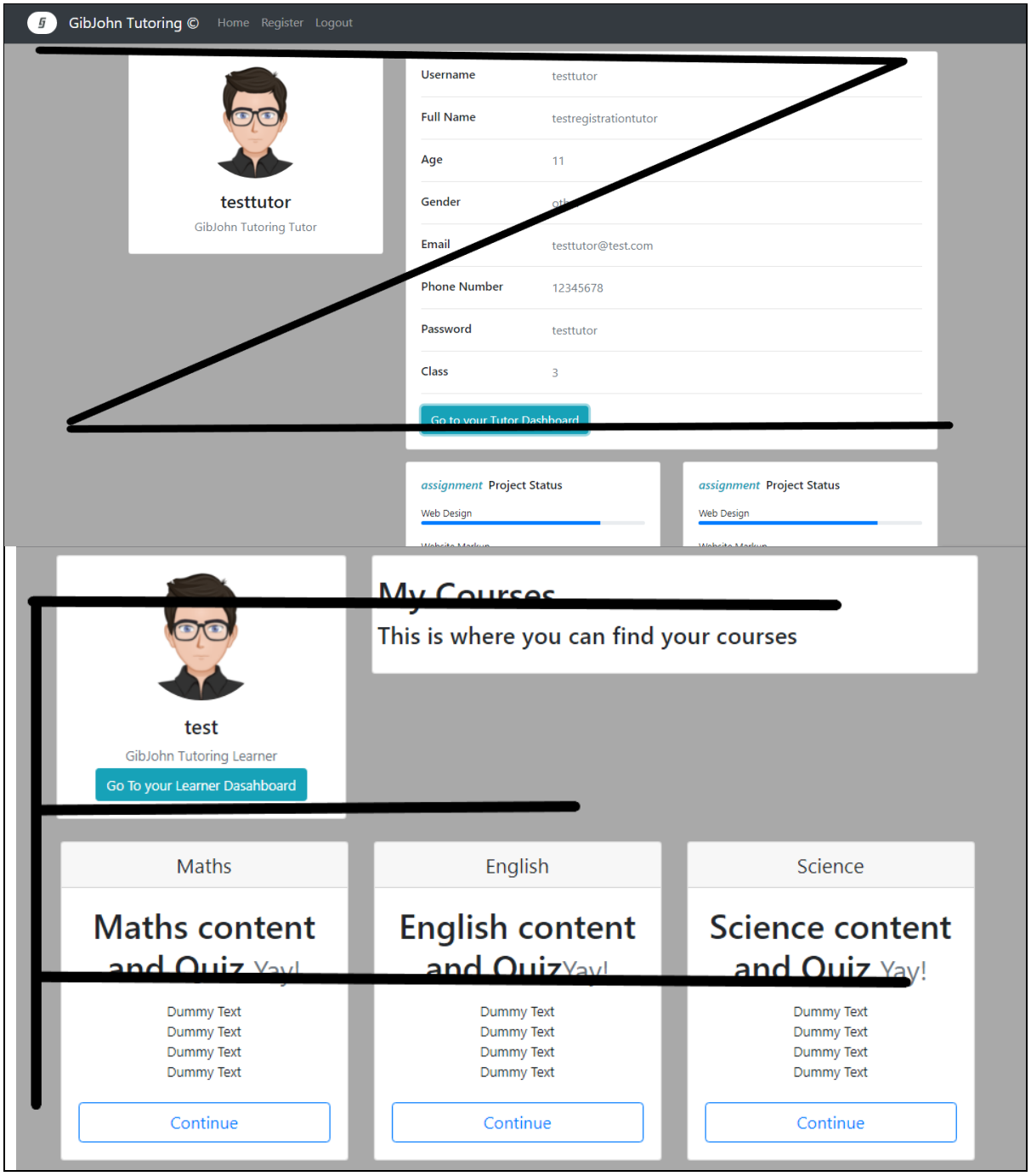



Registration Form - Tutor
ed
Full Name
e
Age
Please enter a number.
Gender:
Female
Male
Other
ed.com
Email Address
e
Phone Number
Username
Username
Password
Password
Class 1
Choose Class
Submit
Or if you already have an account log in here ->
Log in

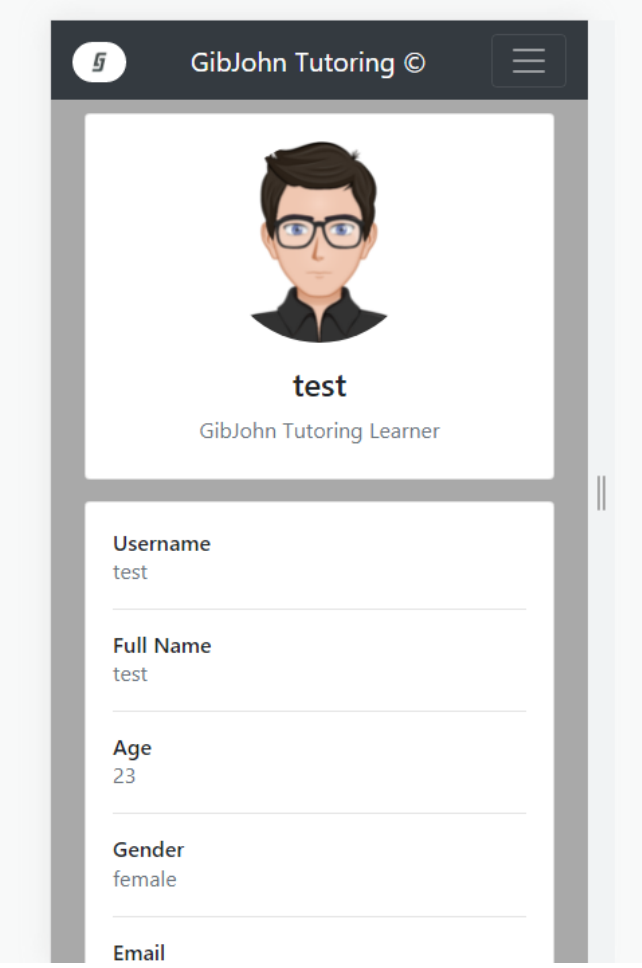

Registration Form - Tutor
ed
Full Name
17
Age
Gender:
Female
Male
Other
ed.com
e
ber
Please include an '@' in the email address. 'ed.com' is missing an '@'.
Username
Username
Password
Password
Class 1
Choose Class
Submit
Or if you already have an account log in here ->
Log in

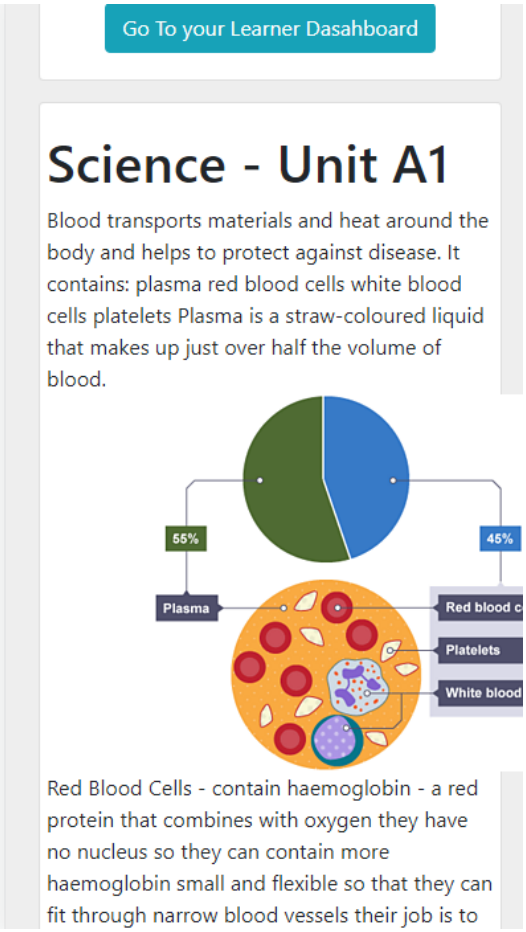
Test Reference:	12	**Name of Tester:**	Ed	**Date of Test:**	29/03/2022
Purpose of Test:	An easy-to-use program that is fast and efficient.				
Test Type:	Usability, non-functional - Blackbox				
Test Data:	N/A				
Expected Outcome:	Should be useable and very quick. /Responsive as it is not a large program that uses lots of resources.				
Actual Outcome:	Application responds very quickly and with no delay – this is probably because its runoff my own Apache web server and not open to the world wide web – it may run differently outside of my testing environment. You can test this by getting an accurate testing model of the live environment and seeing how it works, sadly I don't have access to those facilities.				
Pass/Fail/Skipped:	PASS				
Comments and/or Recommendations:	This is a usability test so for me it may be acceptable, however it may fail in someone else's opinion.				
Evidence:					
N/A					

Test Reference:	13	**Name of Tester:**	Ed	**Date of Test:**	29/03/2022
Purpose of Test:	The program should have a professional looking interface that follows good user interface and experience principles.				
Test Type:	Non-functional – Usability Black Box.				
Test Data:	N/A				
Expected Outcome:	The interface is greyscale which should be readable for most users and the user interface should keep a consistent format across all the pages. Templating will help with this.				
Actual Outcome:	The colour scheme and interface remain consistent across all webpages, with the user interface following either the Z or F pattern.				
Pass/Fail/Skipped:	PASS				
Comments and/or Recommendations:	This is a usability test so for me it may be acceptable, however it may fail in someone else's opinion. This is a non-functional requirement that will differ from person to person.				
Evidence:					

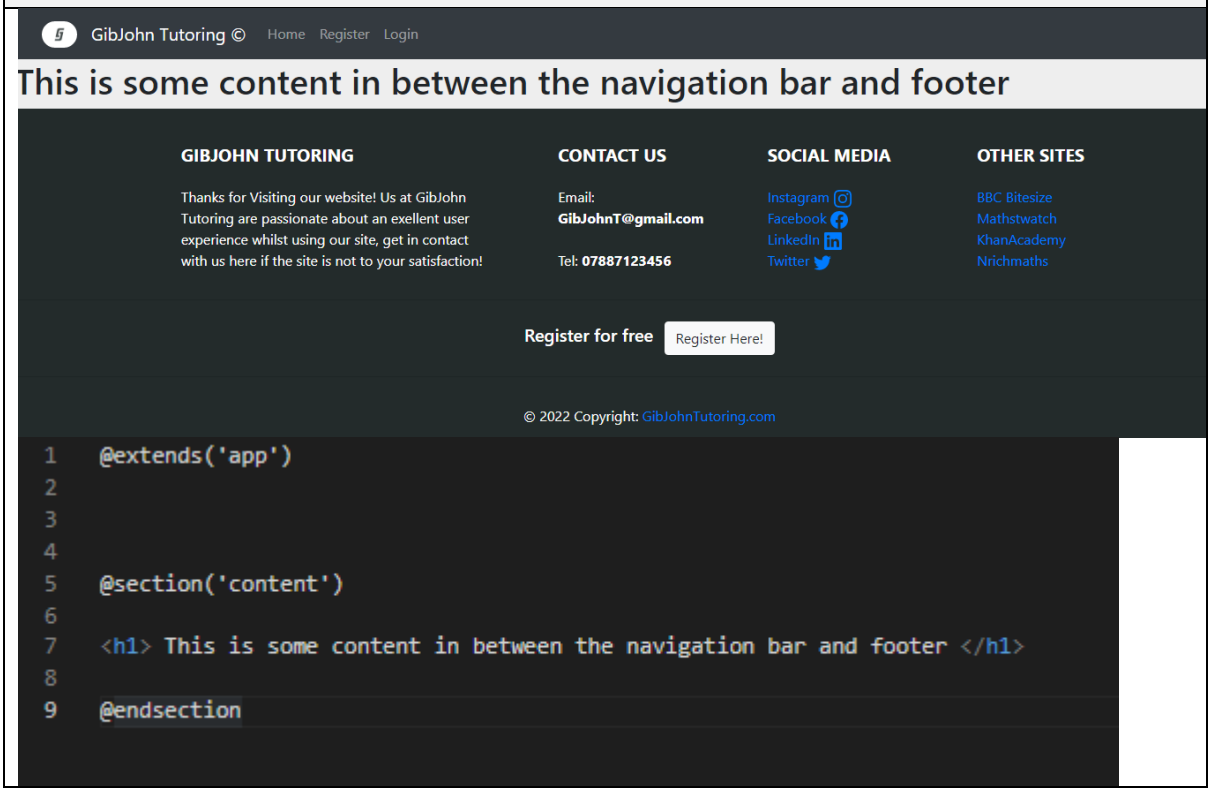

GibJohn Tutoring ©
Home
Register
Logout
testtutor
GibJohn Tutoring Tutor
Username
testtutor
Full Name
testregistrationtutor
Age
11
Gender
Email
testtutor@test.com
Phone Number
12345678
Password
testtutor
Class
3
Go to your Tutor Dashboard
assignment Project Status
Web Design
assignment Project Status
Web Design
test
GibJohn Tutoring Learner
Go To your Learner Dasahboard
My Courses
This is where you can find your courses
Maths
Maths content and Quiz Yay!
Dummy Text
Dummy Text
Dummy Text
Dummy Text
Continue
English
English content and QuizYay!
Dummy Text
Dummy Text
Dummy Text
Dummy Text
Continue
Science
Science content and Quiz Yay!
Dummy Text
Dummy Text
Dummy Text
Dummy Text
Continue

Test Reference:	14	**Name of Tester:**	Ed	**Date of Test:**	29/03/2022
Purpose of Test:	To test if the application can be used the same on mobile devices and smaller devices than the average 24-inch computer screen.				
Test Type:	Functional				
Test Data:	N/A				
Expected Outcome:	The bootstrap should automatically size adjust depending on the size. Not entirely sure of what every page will look like.				
Actual Outcome:	As Expected, the Bootstrap automatically adjusts to fit the screen size. Looks very professional.				
Pass/Fail/Skipped:	PASS				
Comments and/or Recommendations:	The images on certain content pages overlap the containers – this isn't a problem as you can scroll across to view them properly				
Evidence:					





Test Reference:	15	**Name of Tester:**	Ed	**Date of Test:**	29/03/2022
Purpose of Test:	Test if the applications templates work and extend the different relevant sections.				
Test Type:	Functional				
Test Data:	@extends('app') @section('content') @endsection				
Expected Outcome:	The templates should be filled with header footer and content sections. Non views can generate html and the echo just prints out the webpages in views.				
Actual Outcome:	The templates work – This is due to Blade which has been useful for me through this entire project. It was tough to set up, but once its working its very helpful.				
Pass/Fail/Skipped:	PASS				
Comments and/or Recommendations:					
Evidence:					




```
1  @extends('app')
2
3
4
5  @section('content')
6
7  <h1> This is some content in between the navigation bar and footer </h1>
8
9  @endsection
```

```
@section('sidebar')
<nav class="navbar navbar-expand-sm navbar-dark bg-dark">
  <div class="container-fluid">
  <a class="navbar-brand" href="#">
      <img src="gibjohn.png" alt="Avatar Logo" style="width:40px;" class="rounded-pill">
    </a>
    <a class="navbar-brand" href="javascript:void(0)">GibJohn Tutoring &copy;</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#mynavbar">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="mynavbar">
      <ul class="navbar-nav me-auto">
        <li class="nav-item">
          <a class="nav-link" href="test.php">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="registerpage.php">Register</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="loginpage.php">Login</a>
        </li>
      </ul>
    </div>
  </div>
</nav>


@show



@yield('content')



@section('footer')
```

Testing Evaluation

Fitness-for-purpose of the program

The application is not fit for purpose due to it not meeting all required requirements on the specification, also when including the fact that the application did not pass all tests which were based off the user requirements. Certain aspects of the program were not even coded/implemented, which shows that it needs some more development time to get the entire application working at a basic level with all options functional. At that point it would be tested and should pass the tests that failed for me when I tested. I have tried my best to develop the application within the 30-hour time period, however, there have always been errors and setbacks. I have explained why required aspects have unfortunately not been included. All websites have been referenced.

Recommendations with justification

Some things I picked up on were not major fails or an easy pass. The application either worked or it did not. Some could be improved upon.

Some examples:

- Test Reference 12 and 13 – These are both non-functional usability tests which mean that they are opinion based and most likely are slightly biased as I'm testing something I have developed. If another user was to test the application, they will probably have a different opinion to mine.

Change Request evaluation

I did not feel the need to carry out change requests as I am not going to be producing a second version. I have explained when and where I would have implemented missing content.

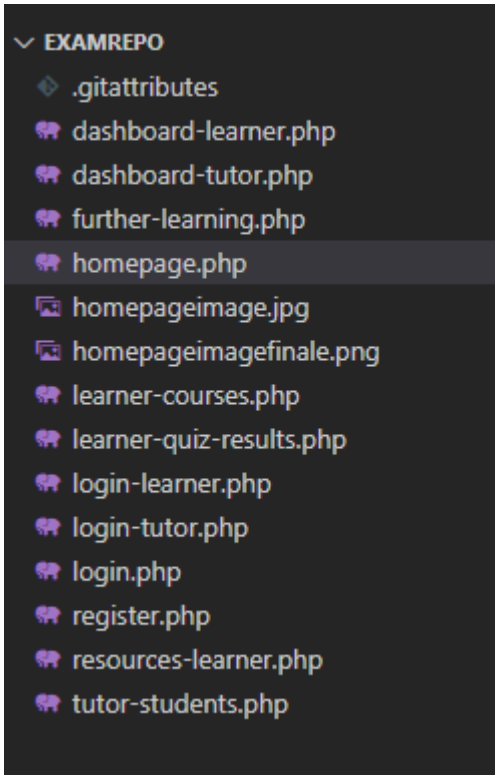
Overall testing evaluation

I think I tested everything at an acceptable level and did my best. However, I should have gone into more detail in certain parts. However, I did test the entirety of what I had produced with both white and black box testing. This as a bare minimum either makes an application fit for purpose or not. This testing phase concluded that the application overall is not fit for purpose and requires more attention in order to pass and become fit for purpose.

Development Log

Exam Session 1 – 07/03/2022

Started a GitHub repository and added all the relevant pages for the moment.



There are the beginnings of a homepage for the website with a navigation bar and some bootstrap images and formatting of the website.

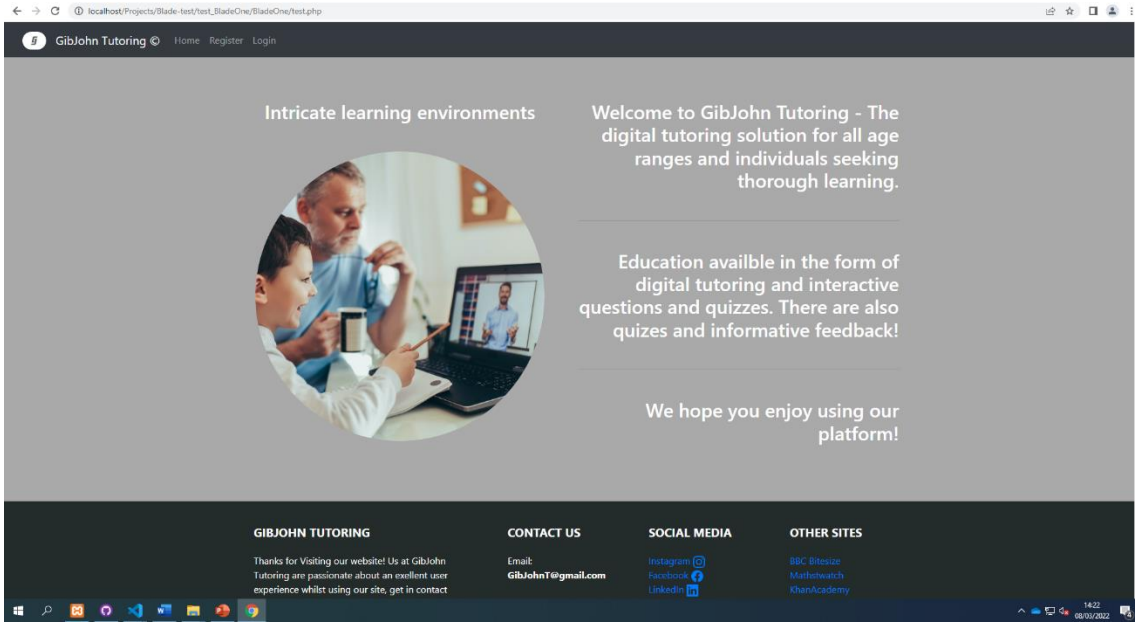
I had a go at getting blade templating to work however I hit a speed bump and decided to move on. I will aim to get templates working fully by the end of next session, and then the real coding can begin.

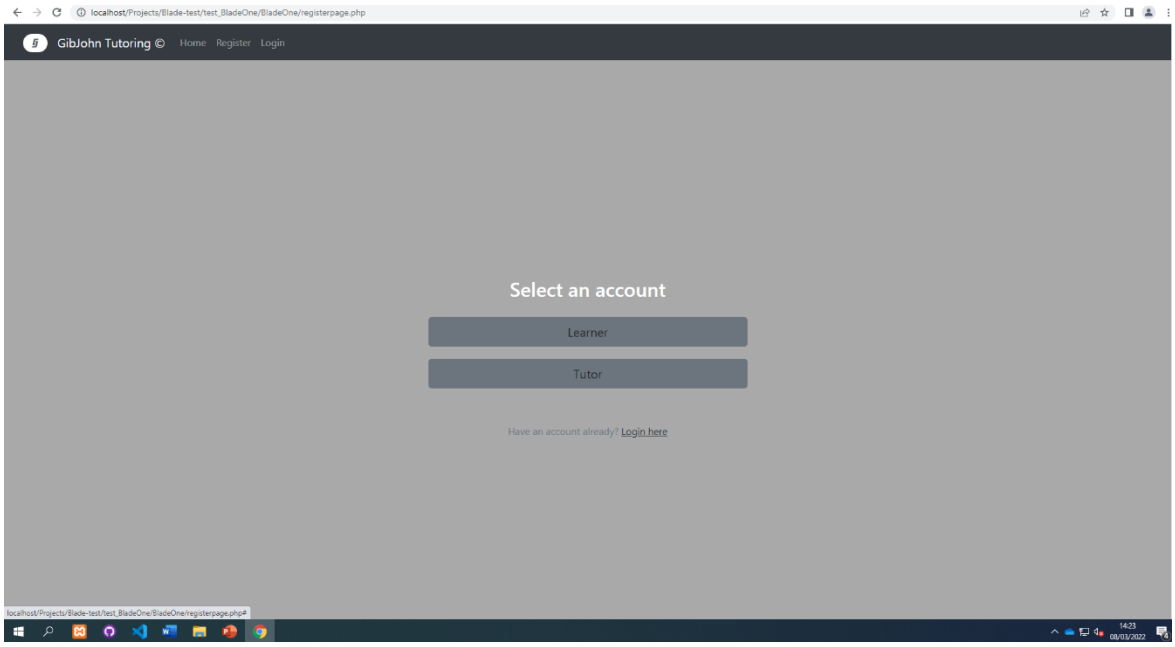
Exam session 2 - 07/03/2022

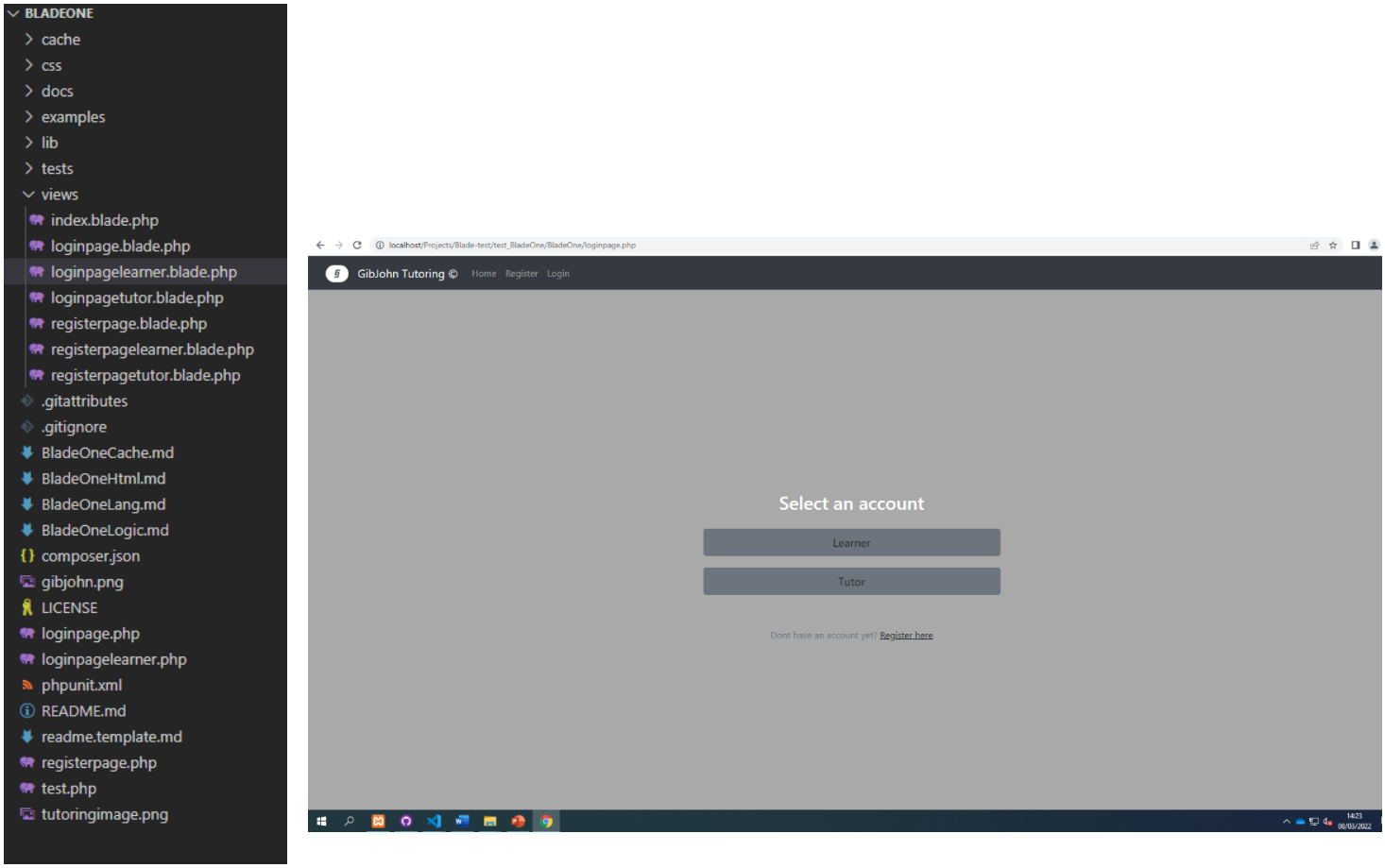
Blade Templates working now, all sorted – now I just need to copy across all my webpages. Started to research and find some good bootstrap templates and formats.

Exam Session 3 – 08/03/2022

Further use of Templating, Homepage completed and same for Register and login pages. Productive!

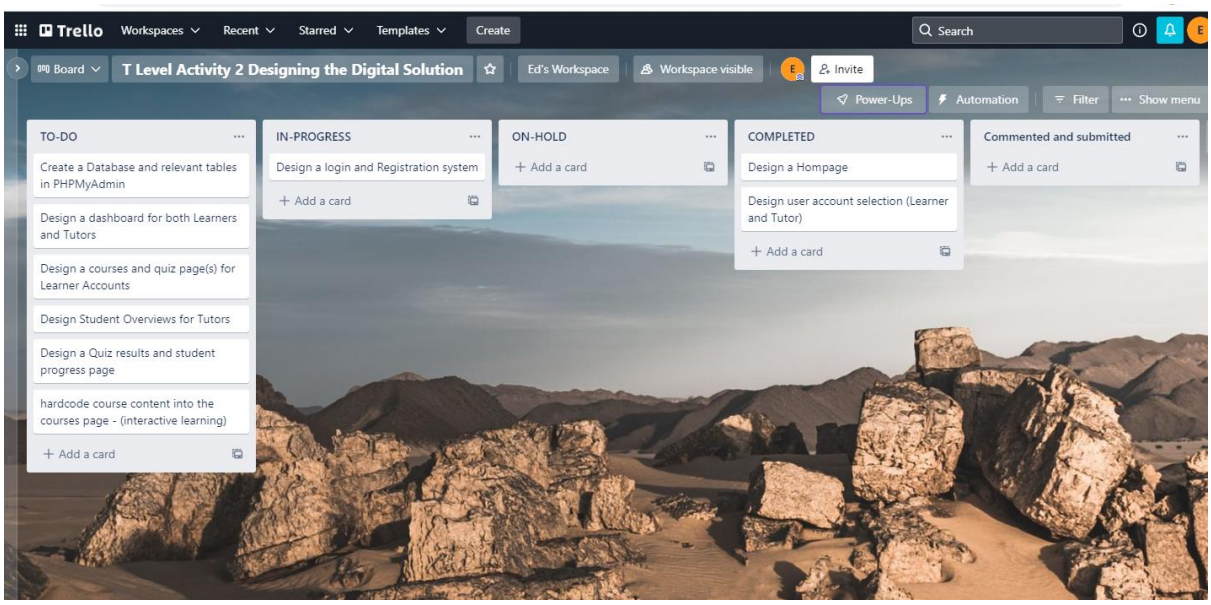


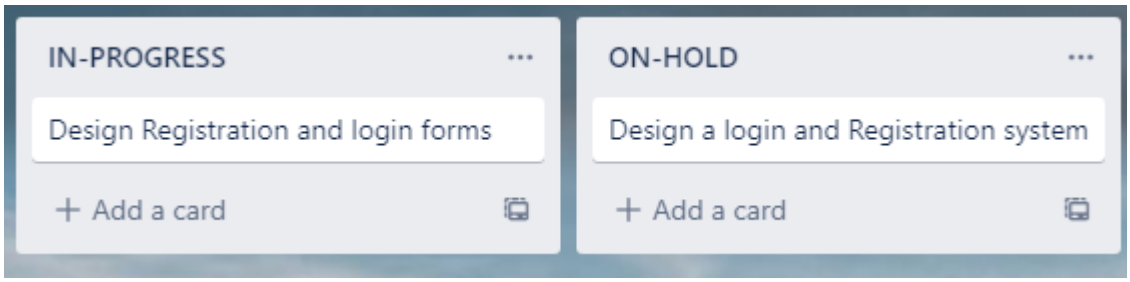



BLADEONE
cache
css
docs
examples
lib
tests
views
index.blade.php
loginpage.blade.php
loginpagelearner.blade.php
loginpagetutor.blade.php
registerpage.blade.php
registerpagelearner.blade.php
registerpagetutor.blade.php
.gitattributes
.gitignore
BladeOneCache.md
BladeOneHtml.md
BladeOneLang.md
BladeOneLogic.md
composer.json
gibjohn.png
LICENSE
loginpage.php
loginpagelearner.php
phpunit.xml
README.md
readme.template.md
registerpage.php
test.php
tutoringimage.png
localhost/Projects/Blade-test/test_BladeOne/BladeOne/loginpage.php
GibJohn Tutoring ©
Home
Register
Login
Select an account
Learner
Tutor
Dont have an account yet? Register here

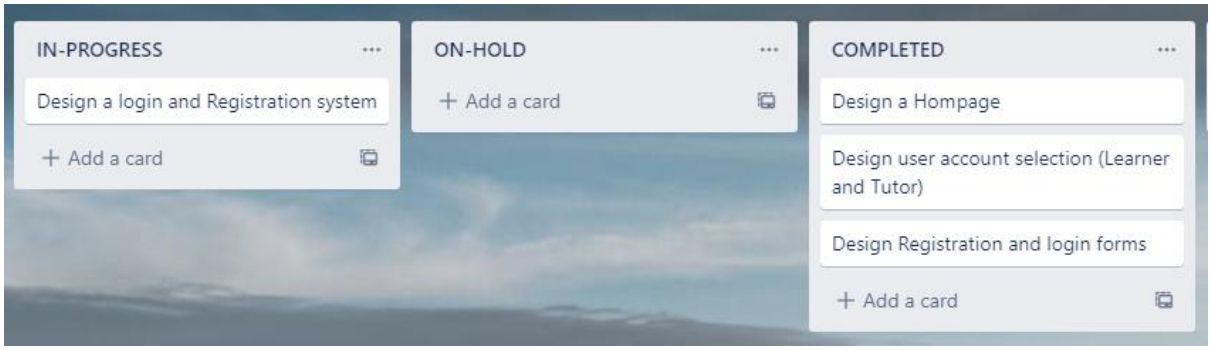
Exam Session 4 09/03/2022

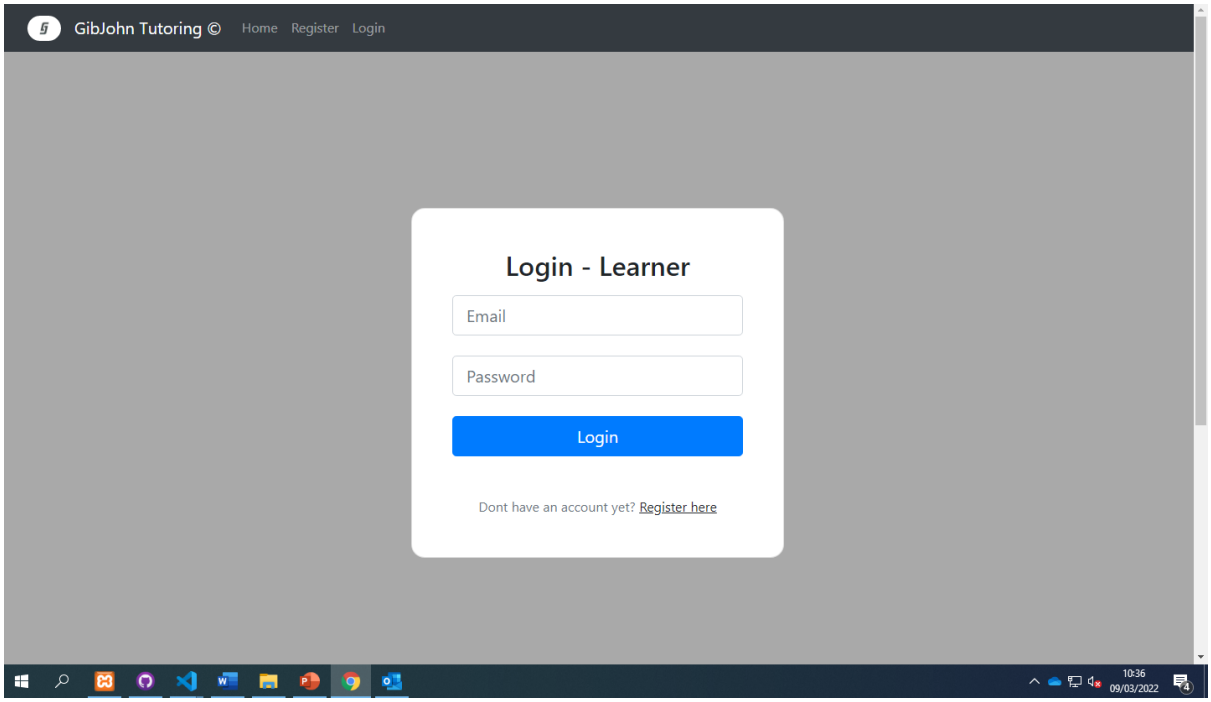
In today's session I have finished off the learner and Tutor account selection and added login form pages for both students and Tutors. I have made a start on the Application form however it is not complete just yet. I have also created a rough Trello board to track visually my progress through tasks, and when others spring to mind they will be added on the Kanban board.

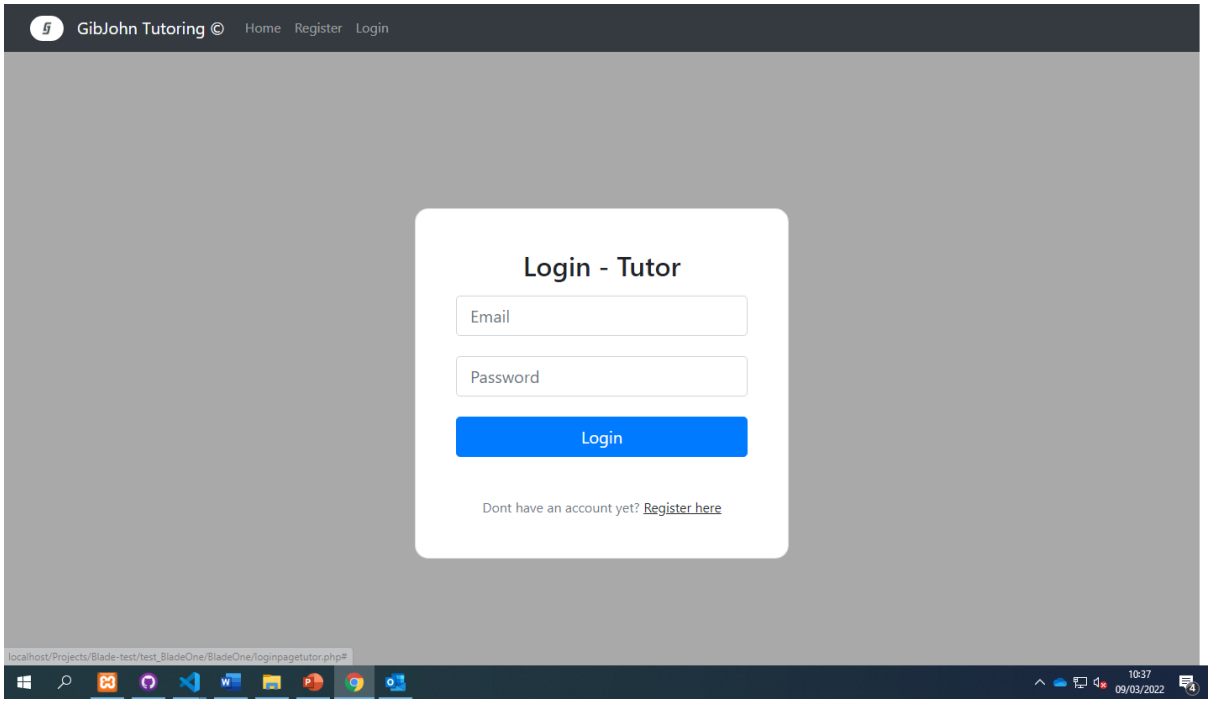


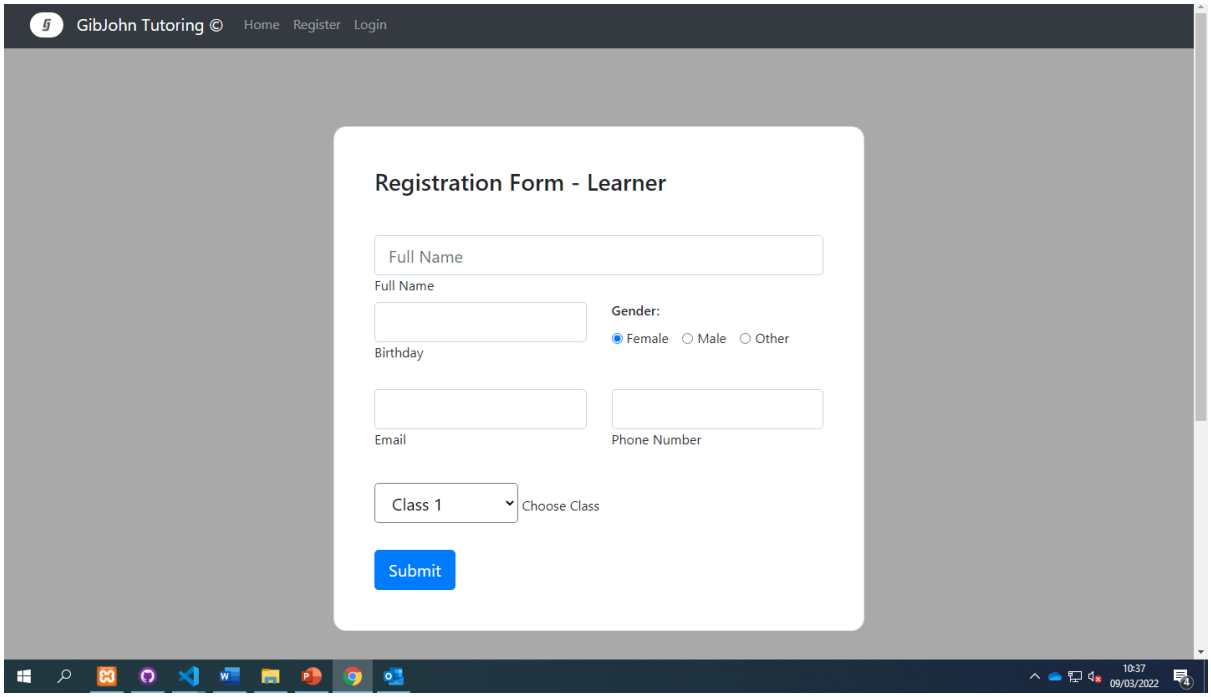


The Login and registration system is put on hold until the actual webforms have been made.



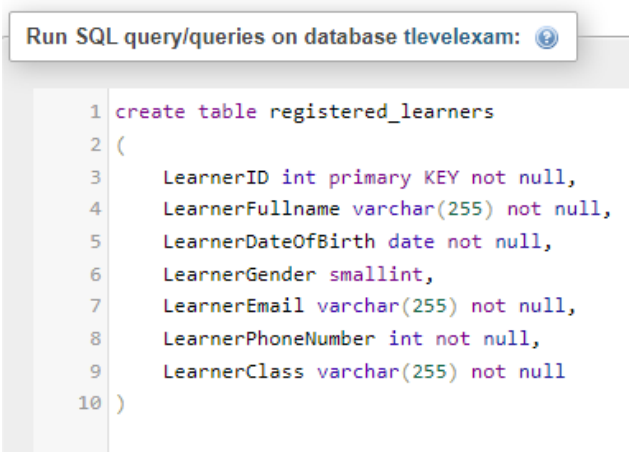

GibJohn Tutoring ©
Home
Register
Login
Login - Learner
Email
Password
Login
Dont have an account yet? Register here
10:36
09/03/2022

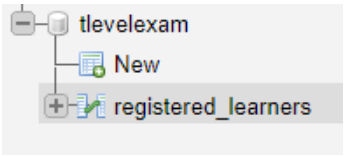

GibJohn Tutoring ©
Home
Register
Login
Login - Tutor
Email
Password
Login
Dont have an account yet? Register here
localhost/Projects/Blade-test/test_BladeOne/BladeOne/loginpagetutor.php#
10:37
09/03/2022

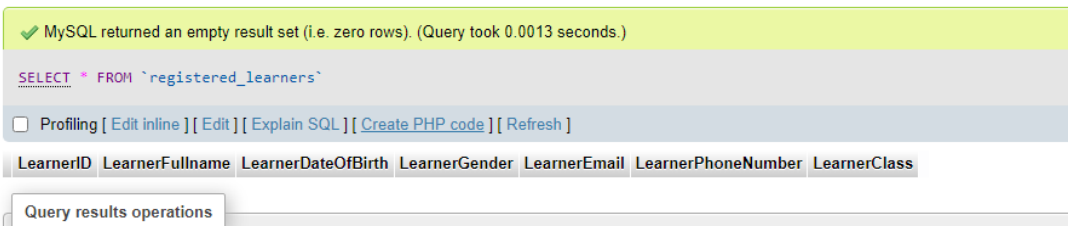


Exam Session 5 - 09/03/2022

Registration forms finished off and making a start on the Backend Database to allow users to login and register. Made a database called televelexam and added a table which stores learner register/ login data. I will shortly make one for Tutors so they can do the same.







Exam session 6 – 11/03/2022

Didn't manage to progress as far as I wanted today, I got stuck on inserting form data into the database tables – I can establish a connection; however, I have only gotten null values and a screen or errors once the form is posted. I also had to edit the date of birth form input as it was a hassle to get working so I just made it into an integer field.

Options

	LearnerID	LearnerFullname	LearnerDateOfBirth	LearnerGender	LearnerEmail	LearnerPhoneNumber	LearnerUsername	LearnerPassword	LearnerClass
Edit Copy Delete	12		0	NULL		0			
Edit Copy Delete	13		0	NULL		0			
Edit Copy Delete	14		0	NULL		0			
Edit Copy Delete	15		0	NULL		0			
Edit Copy Delete	16		0	NULL		0			
Edit Copy Delete	17		0	NULL		0			

Notice: Undefined index: fullname in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **11**

Notice: Undefined index: age in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **12**

Notice: Undefined index: gender in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **13**

Notice: Undefined index: email in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **14**

Notice: Undefined index: phonenumber in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **15**

Notice: Undefined index: username in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **16**

Notice: Undefined index: password in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **17**

Notice: Undefined index: class in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **18**
You have connected to your database!
Notice: Undefined index: fullname in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **27**

Notice: Undefined index: age in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **28**

Notice: Undefined index: gender in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **29**

Notice: Undefined index: email in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **30**

Notice: Undefined index: phonenumber in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **31**

Notice: Undefined index: username in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **32**

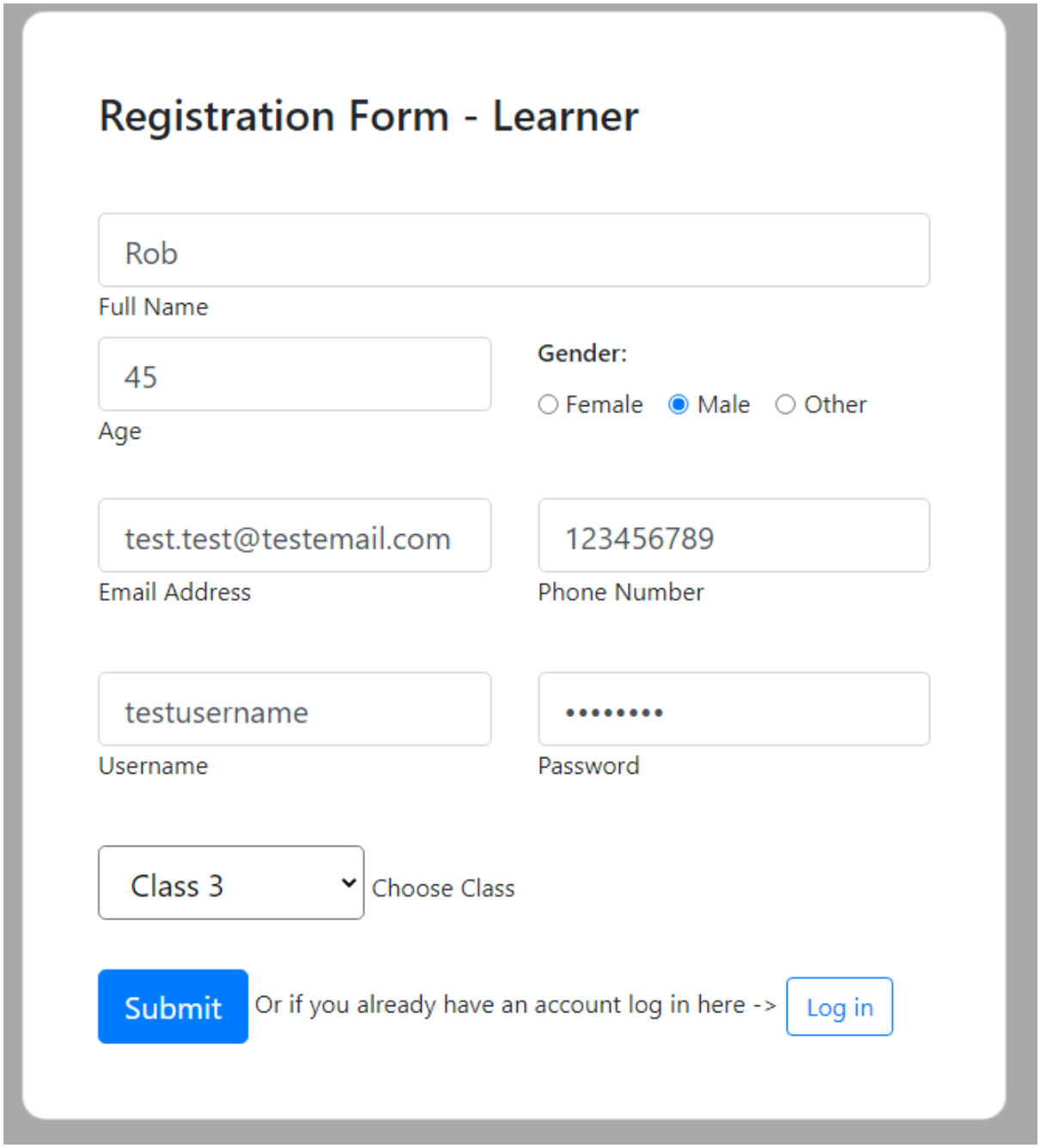
Notice: Undefined index: password in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **33**

Notice: Undefined index: class in **C:\xampp\htdocs\Projects\Blade-test\test_BladeOne\BladeOne\php\register-action.php** on line **34**
New record created successfully
USER ID IS:18

Things for next time – FIX the register form and then the login system will be a breeze. Once that is done maybe implement sessions and then start on the design of webpages for the tutor and students who want to view their courses etc.

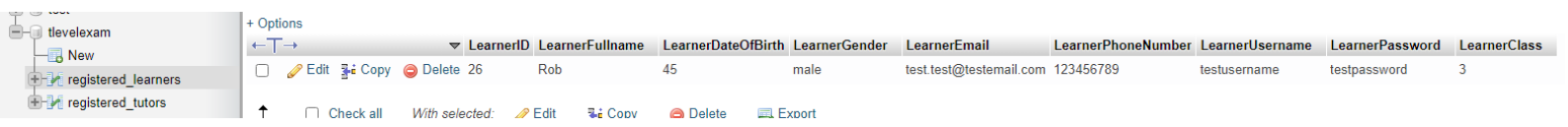
Exam Session 7 – 14/03/2022

Register Learner and Tutor accounts all working now, and a register page created for both. I have started on the login code and process so by the end of exam session 8 I will have a working register and login system.



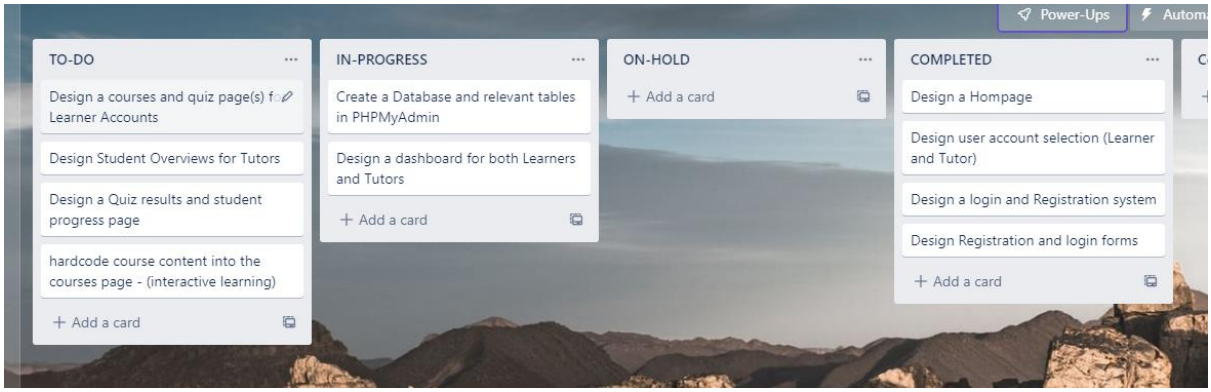
localhost/Projects/Blade-test/test_BladeOne/BladeOne/php/register-action.php

Array ([fullname] => Rob [age] => 45 [gender] => male [email] => test.test@testemail.com [phonenumber] => 123456789 [username] => testusername [password] => testpassword [class] => 3) You have connected to y
USER ID IS:26



Exam session 8 – 14/03/2022

Reference used: https://codeshack.io/secure-login-system-php-mysql/ - Very helpful website for login system. I have finished off the register and login system, I have moved onto the dashboard. I'm happy with my progress over todays 2 sessions. The logged-in screen (Dashboard) is slightly erroring as it is in a different file location, not /php/ the one I'm currently in. I'm sure it's a quick fix but from the login form it goes to nowhere currently.



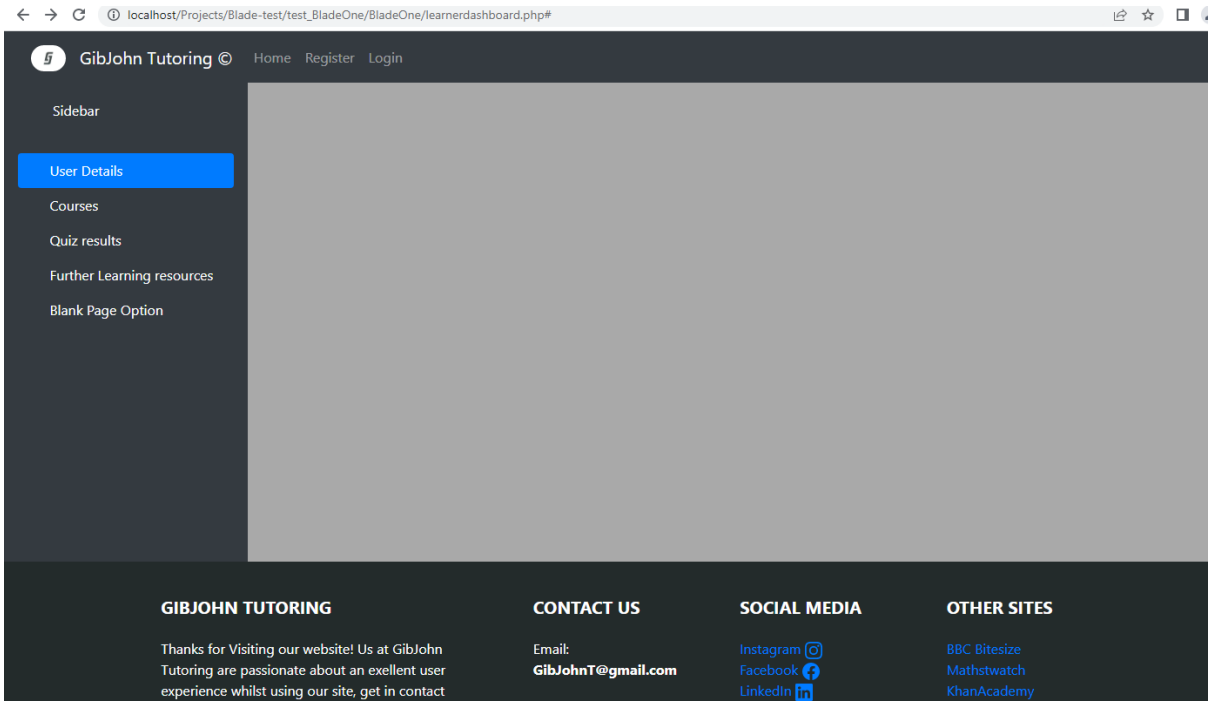
Exam Session 9 – 15/03/2022

I have made the login process a bit clearer and started the dashboard. I have made it so I draw the users details from the database via SQL queries to gain the information the user sees on their dashboards.

```
</div>
<div class="col-sm-9 text-secondary">
  <?php
  $querey = mysqli_query($conn, 'SELECT LearnerPhoneNumber FROM registered_learners WHERE LearnerID like "%'. $id . '%"');
  $row = mysqli_fetch_array($querey);
  print($row['LearnerPhoneNumber']);
  ?>
</div>
```

Exam Session 10 – 16/03/2022

Today I have managed to make a start on the learner dashboard after their initial user profile is displayed on the screen. It has 3 pages that I described in task 1 A (ii).



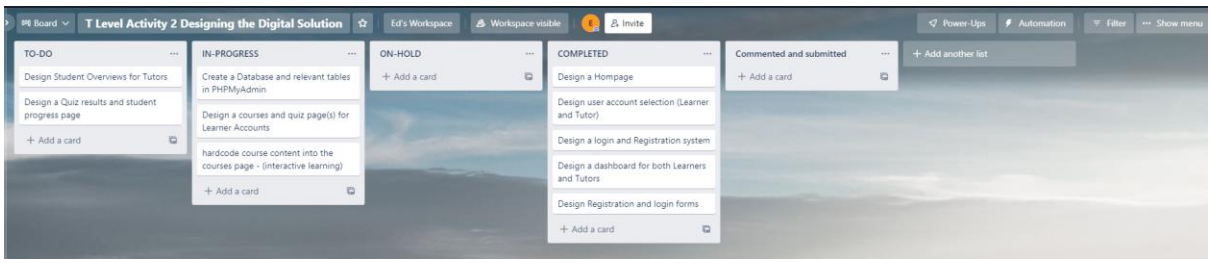
To-do for next time – finish off the dashboard for learner, figure out how to change the course content for each class (1,2,3).

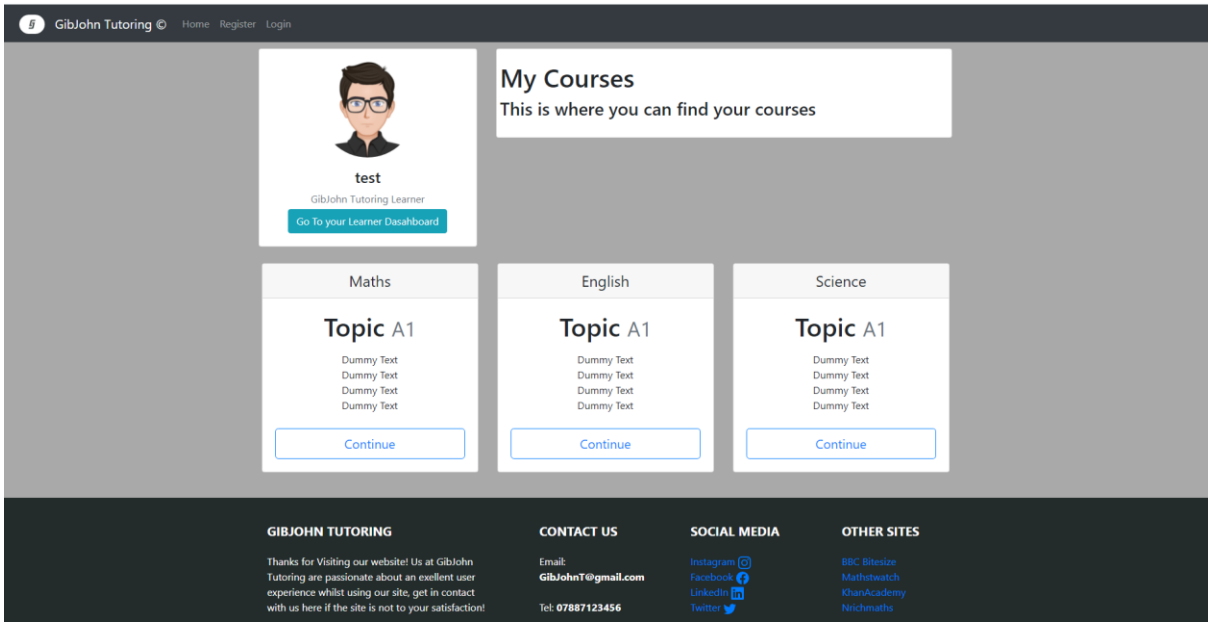
Exam session 11 – 16/03/2022

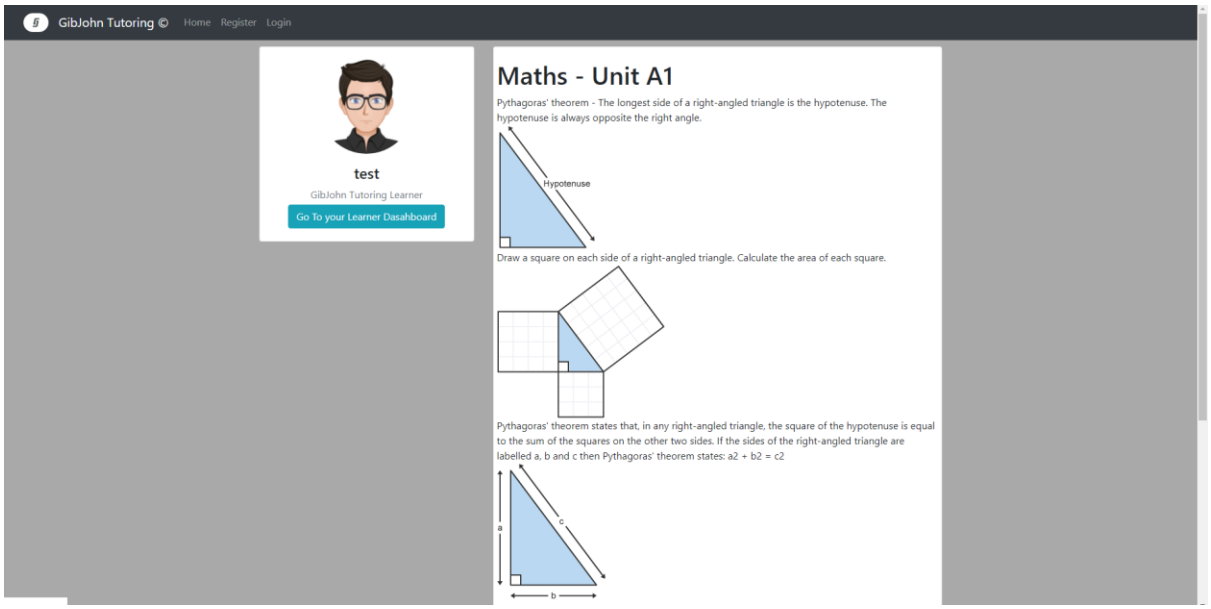
In this session I found out that the side navigation bar really messed with bootstraps positioning and made it messy and tough to figure out how to put content in the correct place. To avoid this, I made 1 central dashboard with 4 links to the relevant pages – courses, user details, further learning and quiz results.

Exam Session 12 – 18/03/2022

In this session I managed to make a learner courses page and 1 resources page for the course (hard coded). In the Next session I will mess around with the visibility of the courses for each class.







Exam session 13 – 18/03/2022

I need to get the courses corresponding with the class either1,2,3 so I can personalise the courses for each learner when the login. Next Session Aim – Try and get it to work otherwise start focusing on tutor dashboard and their students (All of 1 class) and possibly get to assigning work. After all of this I need to do quizzes and log the results in a database and then get graphical evidence of the results for both learner and tutor. (1 Week left and then I need to spend the next week Testing and Writing up what I have done).

Exam Session 14 – 21/03/2022

I have tried my best to change the page depending on the learner's class however it should not be taking this long. It should just be a simple import and a few lines of if statement. Irritated that it has taken a whole session and gotten nowhere.

```php
$conn = mysqli_connect($servername, $username, $password, $dbname)
    or die("Connection Failed");
$row = mysqli_fetch_array($querey);
$id = $row["LearnerID"];
$querey = mysqli_query($conn, 'SELECT LearnerClass FROM registered_learners WHERE LearnerID like "%'. $id . '%"');
$learnerclass = $querey;

if($learnerclass == "1"){
  header("Location:maths1learner.php");
} else if($learnerclass == "2") {
  header("Location:maths2learner.php");
}

?>
```

Exam Session 15 – 21/03/2022

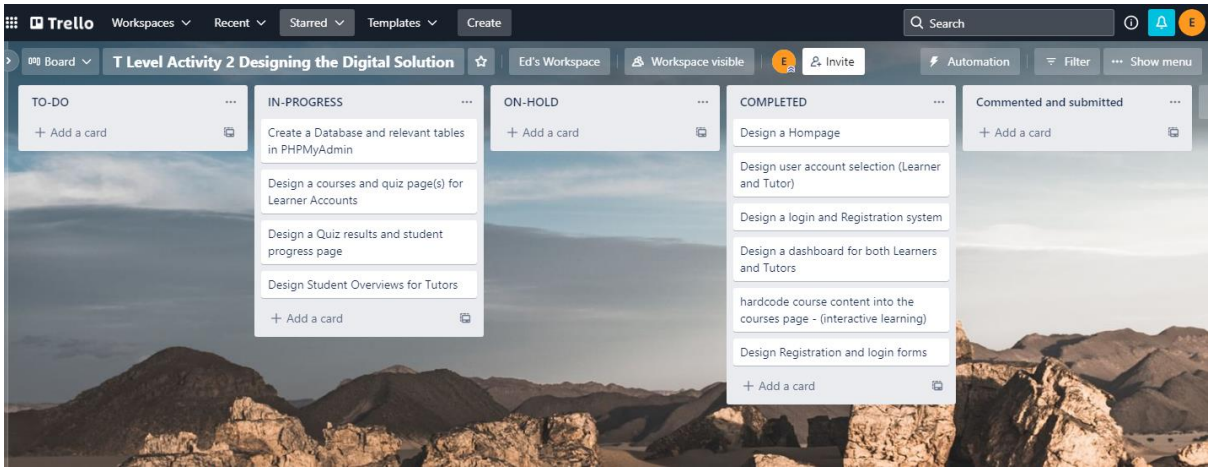
I will need to spend some time outside the exam to figure out why my simple if statement is not working. Once this is rectified however, it is a massive section done and I only must focus on the quiz and results page for the learners and the tutor's views. The tutors' views will come easily once I figure out how to display learner data, from a database, in a progress bar. However, if this is not possible, I will just output the raw results of the student's quiz. In this session I started the further learning+ resource page as I was stuck on my current problem I explained above.

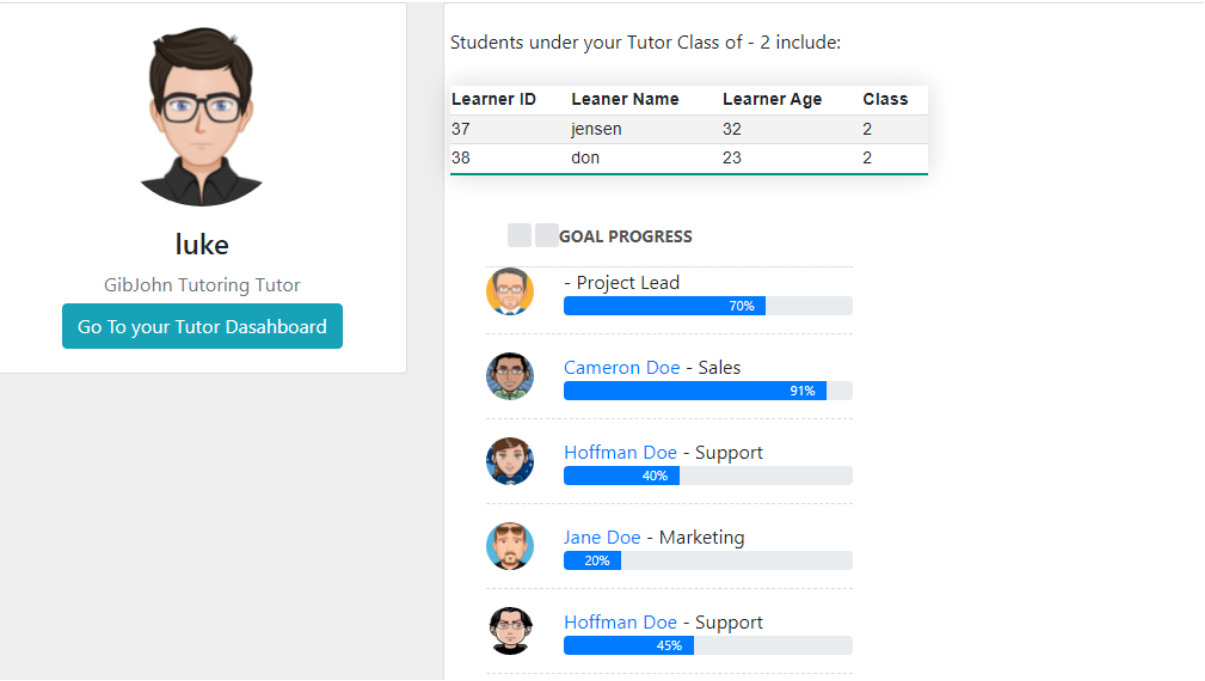
Exam Session 16 – 22/03/2022

I have managed to fix the errors I had yesterday. I've got different content for students with a different class.

Exam Session 17 – 23/03/2022

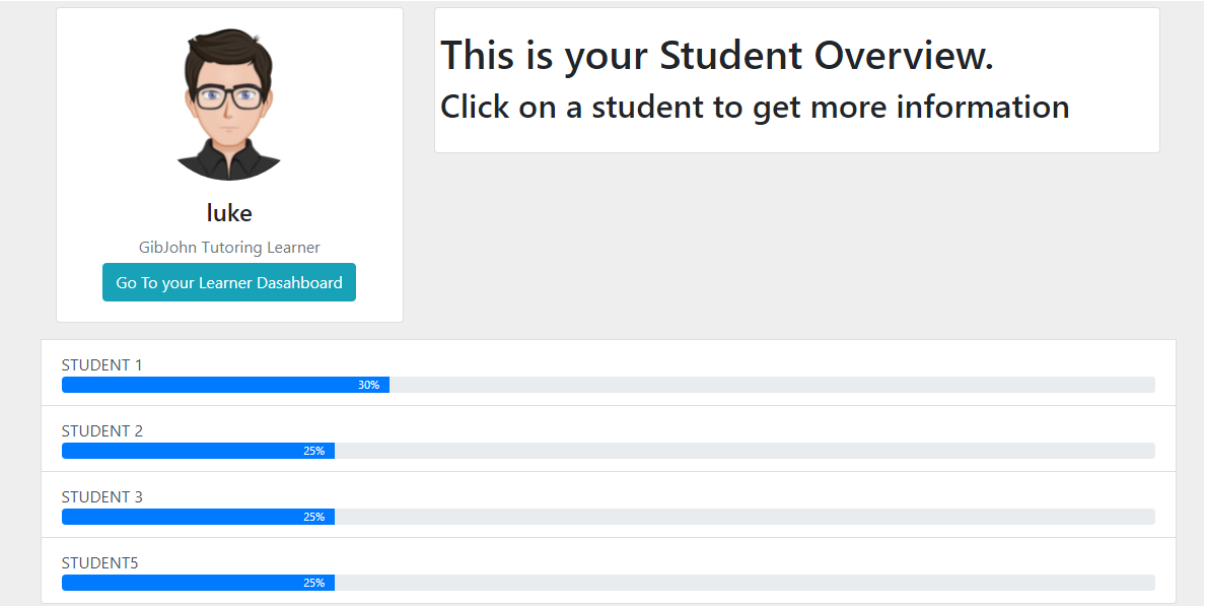
I have managed to start on the Tutor dashboard and make a tutor view with the students. I'm experimenting ways that it will work like as I need to have a progress bar, and a way the tutor can click a button to say whether a student has passed or not. This would then update a database table that would show up on the student's dashboard. After this has been completed, I will start on the quiz system for the students (learners).





Exam session 18 – 23/03/2022

I have made more of an attempt on the tutor dashboard its getting somewhere however I am still unsure on how to use database values to put into the CSS of the progress bars. This problem I will research and figure out over the next day before out next exam session on Friday. I will make the student Quiz on Friday and input the value of that to the database.



https://www.bootdey.com/snippets/view/profile-edit-data-and-skills

Last Few Exam sessions – Testing and writeup – 28/03/2022 – 30/03/2022