07/03/22

Implement Laravel’s templating solution blade for PHP,

Implement Bootstrap, CSS and JavaScript libraries

Configure XAMPP to redirect to the correct pages

Consider using Livewire

08/03/22

Livewire didn’t workout. For this solution Blade will work.

Login system, signup, signout is all working

Can’t get the content part of the page to align as needed

Todo- tidy code, create appropriate layout and comments

Graphical user interface, website

Description automatically generated

Graphical user interface, website

Description automatically generated

11/03/2022

Timeline

Description automatically generated with medium confidence

An issue with the Blade templates is proving difficult to use with a dynamic navigation bar.

Having an issue with Bootstrap’s grid system. Having consistent layouts that match the content is proving difficult.

16.03.22

A screenshot of a computer

Description automatically generated with medium confidenceTreemap chart

Description automatically generated

Switched to using Flex Boxes,

Navbar and scaling needs tweaking and fixing

<https://www.onlyxcodes.com/2018/12/multi-user-role-based-login-in-php-with-mysql-pdo.html#Database%20and%20Table>

18/03/22

Text

Description automatically generated

Getting the php to change the password in the right database isn’t working.

Code needs organising and commenting, to make sure that is understandable to someone who doesn’t understand code

Graphical user interface

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Teachers and students no longer share a dashboard. Although there is no content on either right now, the pages have different boxes intended for different content

Students and teachers now have separate login and signup pages.

4 boxes take up 25% of the space each. This is simply due to simplicity and having it easy to maintain however in the future I plan on making it more interesting design

Graphical user interface, website

Description automatically generated21/03/22

Teachers can now see students, they are able to edit their details (username, however I plan on adding email, firstname and lastname.) Teachers can’t change their ID, they can’t change or view their password.

Teachers will be able to delete students aswell.

Working on adding more student details to the database

22/03/22

Graphical user interface, text, application

Description automatically generated

Managed to get relationships working between tables in the database.

SELECT students.username AS Student\_name, courses.name AS Course, teachers.username AS Teacher\_name

FROM classes

INNER JOIN students ON classes.fk\_student=students.id

INNER JOIN courses ON classes.fk\_course=courses.id

INNER JOIN teachers ON classes.fk\_teacher=teachers.id

WHERE courses.id=2

                    /\*<td> <A class='btn btn-primary' href='edit.php?pet\_id=".$row['Student\_name']." '> Edit </a> </td>

                    <td> <A class='btn btn-danger' href='delete.php?pet\_id=".$row['']." '> Delete </a> </td>\*/

  <!-- Modal -->

  <div class="modal fade" id="exampleModalLong" tabindex="-1" role="dialog" aria-labelledby="exampleModalLongTitle" aria-hidden="true">

    <div class="modal-dialog" role="document">

      <div class="modal-content">

        <div class="modal-header">

          <h5 class="modal-title" id="exampleModalLongTitle">Modal title</h5>

          <button type="button" class="close" data-dismiss="modal" aria-label="Close">

            <span aria-hidden="true">&times;</span>

          </button>

        </div>

        <div class="modal-body">

          ...

        </div>

        <div class="modal-footer">

          <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

          <button type="button" class="btn btn-primary">Save changes</button>

        </div>

      </div>

    </div>

  </div>

Write up:

Asset Log:

|  |  |  |
| --- | --- | --- |
| Name | Purpose | Justification |
| Bootstrap | Bootstrap is a CSS framework that is recognised and used globally | Bootstrap is an open-source framework, it is the building blocks for the front end of the website. It has allowed the website to have a responsive, good-looking design that is modern and easy to use. |
| BladeOne Templating Engine | Blade is a templating engine that is seen in Laravel. BladeOne is a standalone version of blade. | Using Blade we are able to reuse code wherever necessary. Blade allows better organisation of code. Using blade you can separate all code, into a Model, View and controller framework. |

BladeOne: During development I didn’t separate CSS, HTML, Javascript and PHP as is intended when using Blade. I used Blade’s views and templates but I didn’t separate the code. This would be something to change in the next version of the website.

A screenshot of a computer

Description automatically generated with medium confidence

Home Page: The homepage is the first page any new user will see when they visit GibJohn Tutoring’s website.

The Navigation bar will change depending on the page the user in on, however when a user is unable to access a page, it will be greyed out and the user won’t be able to interact with it.

The Footer has copyright marker, letting the user know that all material on this website isn’t to be reused or copied without permission. Alt Text is used on all images to help with readers and accessibility.

Text

Description automatically generated

This PHP script calls the view

Text

Description automatically generated

Extends the view to the template

Starts or resumes a session, sessions are a way to store data across pages. Unlike cookies they’re not stored on the user’s device.

Blade’s Sections are like variables. Data can be stored in them and then called again somewhere else. In this case, a section called ‘gibjohndesc’ is being defined and called again in the template, which Is then shown on the browser.

I used Blade’s templates because layouts and templates can be used again and again. For example: on GibJohn’s website the footer and navigation bar are the same code called on different pages. This is because a page can share a layout with another page, and all layouts share the same navigation bar and footer. Any changes made to “Header.php” would take immediate effect on all pages. Websites not using Blade or similar PHP templating engine would need to change it for every page that uses it manually. Using layouts makes development easier and quicker because Blade allows the separation and reuse of code.

I used Boot

Text

Description automatically generated