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>

Task 2: Developing the system prototype

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.

Both Bootstrap and BladeOne are open source, therefore no legal or ethical implications come into play when using them.

GibJohn tutoring is an educational tutoring company that provides their customers with:

* Face-to-face tutoring sessions
* Access to learning resources
* Support to develop understanding in different subjects

GibJohn tutoring has asked me to develop a prototype digital solution, to better help and support their customers. They want a digital solution that will:

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

Potential features include:

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

The Home Page:

A screenshot of a computer

Description automatically generated with medium confidence

Using bootstrap, I took a minimalistic approach. For the prototype I didn’t fill the homepage with information. However, in the next version accurate data and user reviews, or an about us paragraph would fill the empty space and potentially make the website more attractive to customers visiting GibJohn tutoring’s website for the first time.

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 website was designed with a minimalistic approach,

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 generatedHow it works:

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.

Text

Description automatically generated

Code for the homepage template:

The head is used on all templates across the website. It gives the website its title, and links it too Bootstrap and its stylesheet.

A screenshot of a computer

Description automatically generated with medium confidence

The navigation bar is also used in all templates across the website. The navigation bar is consistent across each page, and allows the user to navigate easily through the website.

The Navigation bar uses classes to represent to the user what page they’re viewing. Each time a page is called, a variable called “page” is set to the name of the page that is being called. In this case it’s “home” for the homepage. The navigation bar checks this variable everytime the page is refreshed by using Blade’s own syntax @if statements. If the variable is equal to “home” the class “active” is assigned. This gives the appearance that the Home button is lit up when on the home page.



Text

Description automatically generated

A screenshot of a computer screen

Description automatically generated with medium confidence

Dashboard and logout are disabled, using classes and if statements. This is because the user isn’t logged in and can’t view these pages until they do so.

Teacher & Student Login/Signup selection page:

A screenshot of a computer

Description automatically generated with medium confidence

Using JavaScript, creating a singular page for login and signup for both user groups would make it easier to navigate and increase simplicity. However, for the prototype the user is redirected to the page they chose via a button.

Graphical user interface, text, application

Description automatically generated

Each input box uses data validation techniques that not only makes it easier for customers to enter the correct information, while giving feedback on errors but also protects against certain attacks that can be used to view or manipulate sensitive data in a database.

Students also have a very similar student signup page.

Graphical user interface

Description automatically generated

When a student or teacher signs up, It uses PHP’s Password\_Hash() method. This encrypts the password when stored in the mySQL database making it much more secure and less vulnerable to attack.

Config.php

Text

Description automatically generated

This allows the website to connect to the mySQL database.

Teachers table:

Graphical user interface, text, application

Description automatically generated

The user’s password gets hashed when stored in the database

When a student or teacher signs up to the website their data is stored here.

Students table:

Graphical user interface, text, application

Description automatically generated

Signup script:

Text

Description automatically generated

Data validation, makes sure the user is using the correct information and isn’t trying to run SQL injection.

Gives feedback and lets the user know if they’ve made an error.

Redirects the user if they’re already logged in

This shows the signup script for teachers, however students use the same script on a separate page that links to the student table.

Student signup, Teacher signup, student login and teacher login all use the same template. Which gives them a consistent look throughout.

Text

Description automatically generated

Once the user has signed up, they’re redirected to the login page.

Hashing password

SQL statement inserts the username and hashed password into the database.