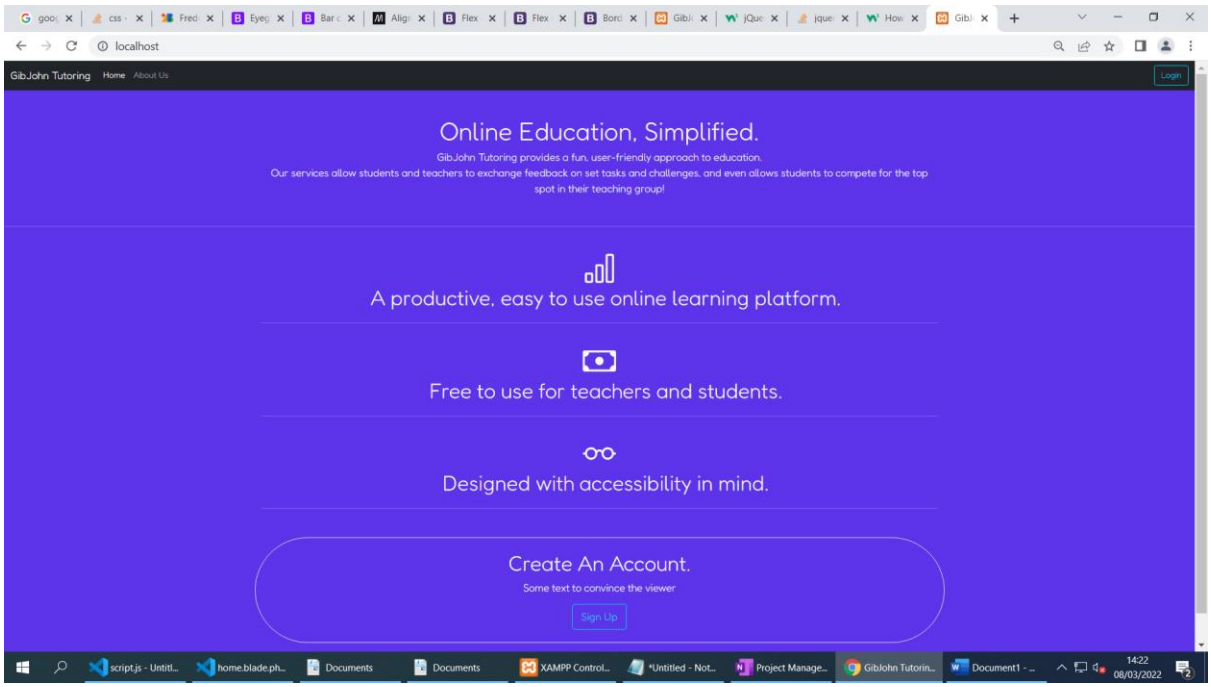
Dev Log draft

Day 1

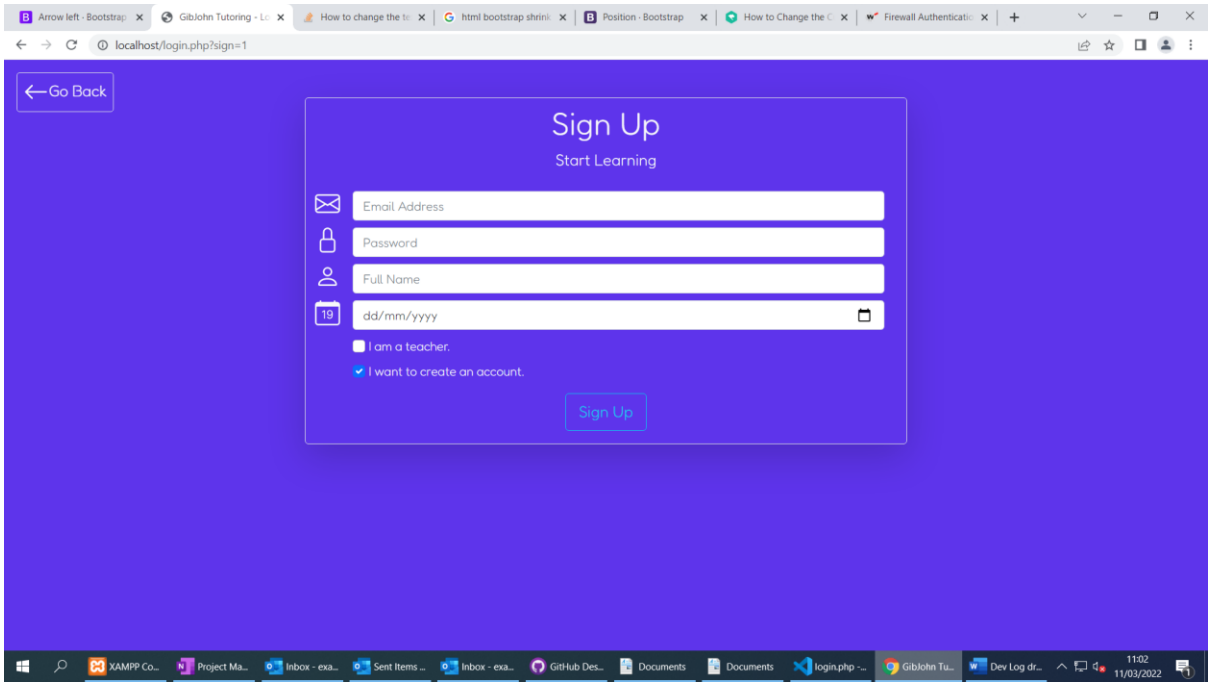
- Set up workspace
- Created a GitHub repository
- Created a base for the home page
- Created a base for the about page
- Implemented bootstrap

Day 2

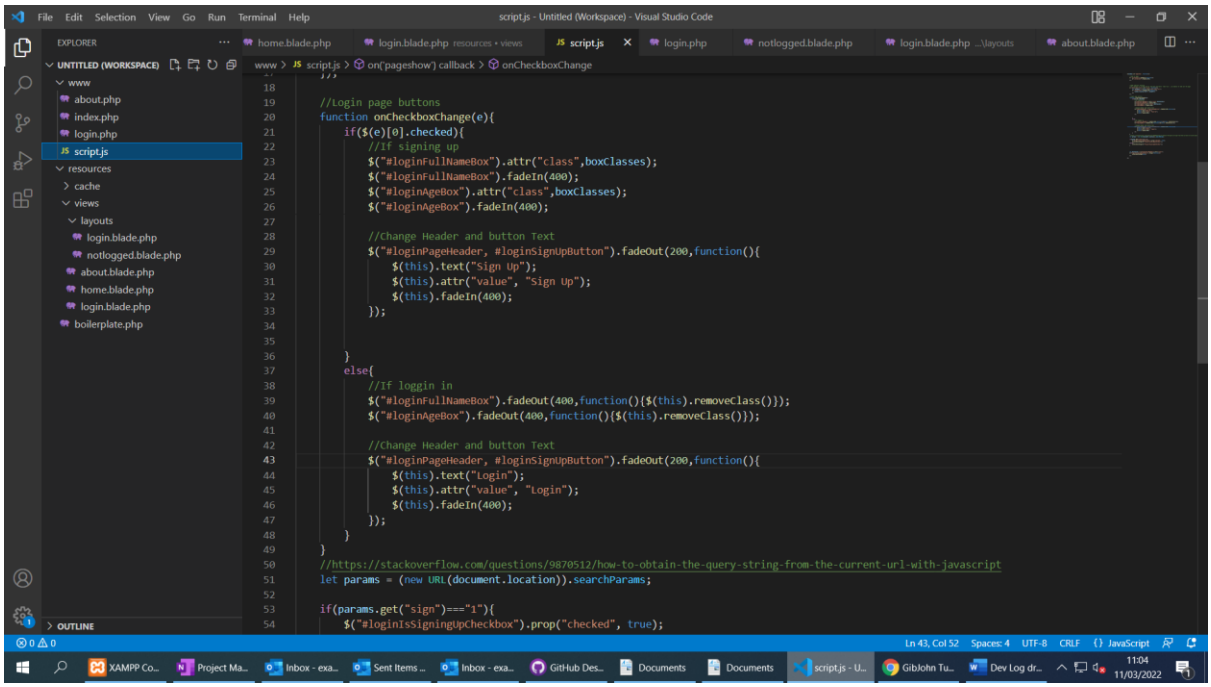
- Further work on the home page



- Although the main content of the page will not be using AJAX to save time for more important features, I have added a smooth page transition using jQuery to fade between pages.
- Changed to a different font
- Implemented flex containers to make the page usable on different device types
- Created a basic To-Do list in OneNote to manage my time properly.



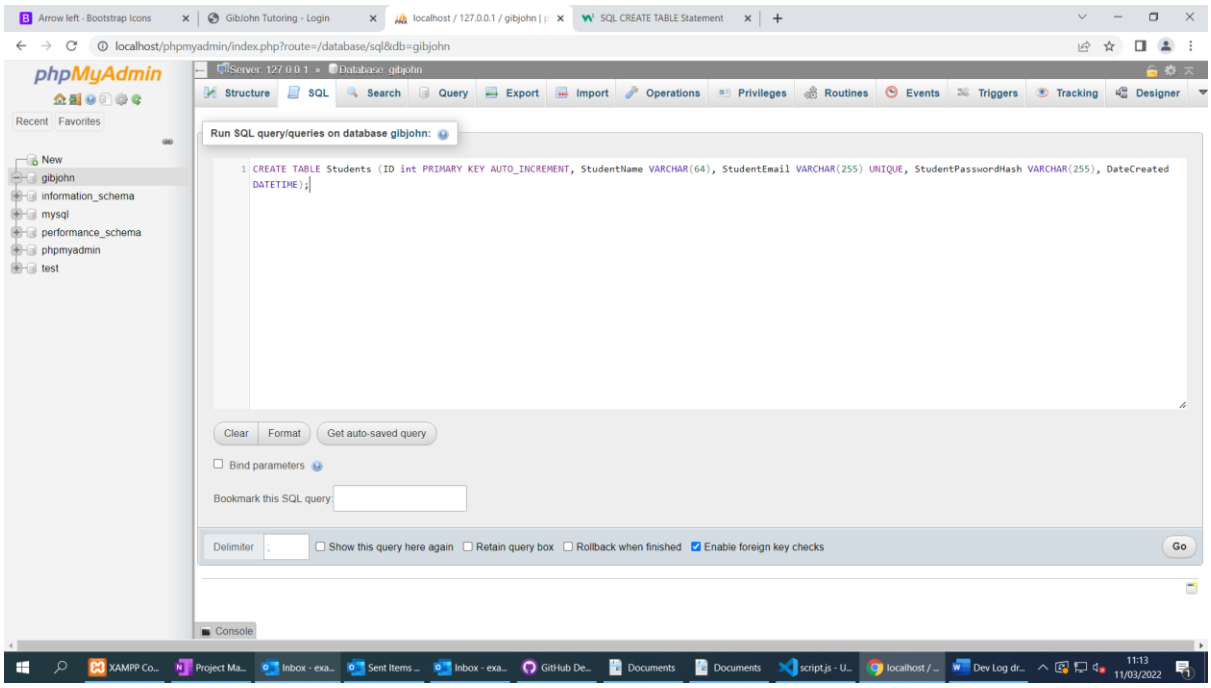
- Finished the sign up and login UI
- Made the layout dynamic using jQuery
- Created a Go Back button for better user experience

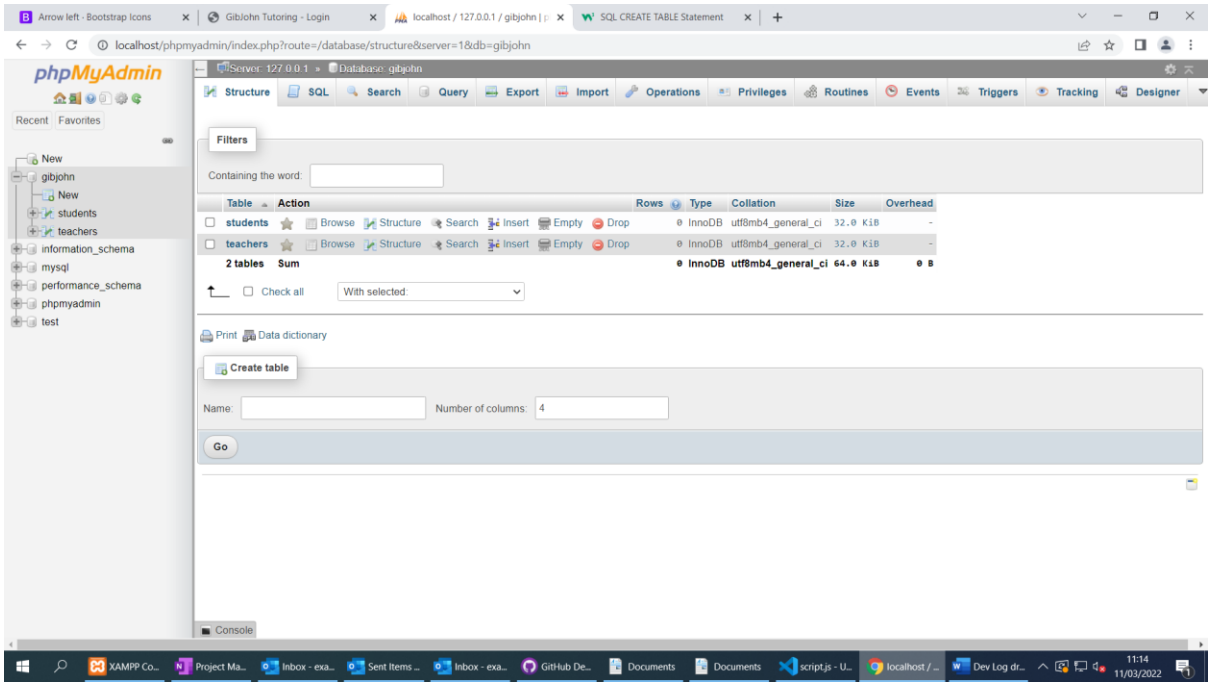


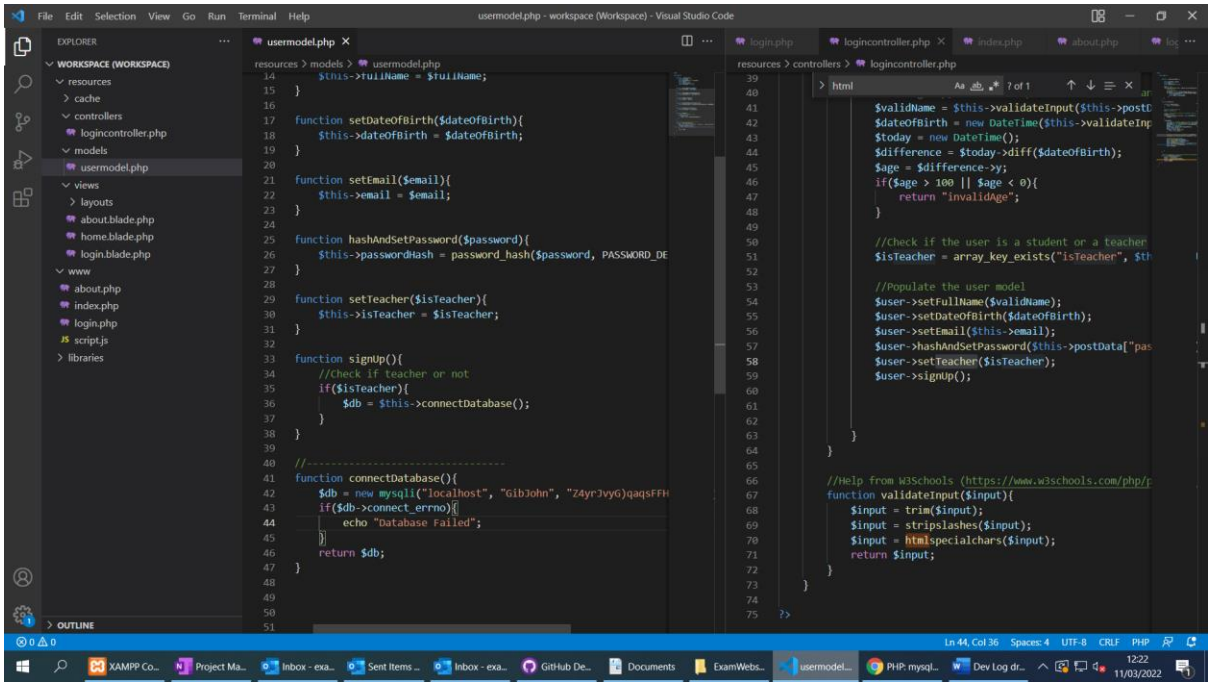
```
//Login page buttons
function onCheckboxChange(e){
    if($(e)[0].checked){
        //If signing up
        $("#loginFullNameBox").attr("class",boxClasses);
        $("#loginFullNameBox").fadeIn(400);
        $("#loginAgeBox").attr("class",boxClasses);
        $("#loginAgeBox").fadeIn(400);

        //Change Header and button Text
        $("#loginPageHeader, #loginSignUpButton").fadeOut(200,function(){
            $(this).text("Sign Up");
            $(this).attr("value", "Sign Up");
            $(this).fadeIn(400);
        });


    }
    else{
        //If loggin in
        $("#loginFullNameBox").fadeOut(400,function(){$(this).removeClass()});
        $("#loginAgeBox").fadeOut(400,function(){$(this).removeClass()});

        //Change Header and button Text
        $("#loginPageHeader, #loginSignUpButton").fadeOut(200,function(){
            $(this).text("Login");
            $(this).attr("value", "Login");
            $(this).fadeIn(400);
        });
    }
}
//https://stackoverflow.com/questions/9870512/how-to-obtain-the-query-string-from-the-current-url-with-javascript
let params = (new URL(document.location)).searchParams;

if(params.get("sign")==="1"){
    $("#loginIsSigningUpCheckbox").prop("checked", true);
```

- Some of the JavaScript handling the login screen

Day 3

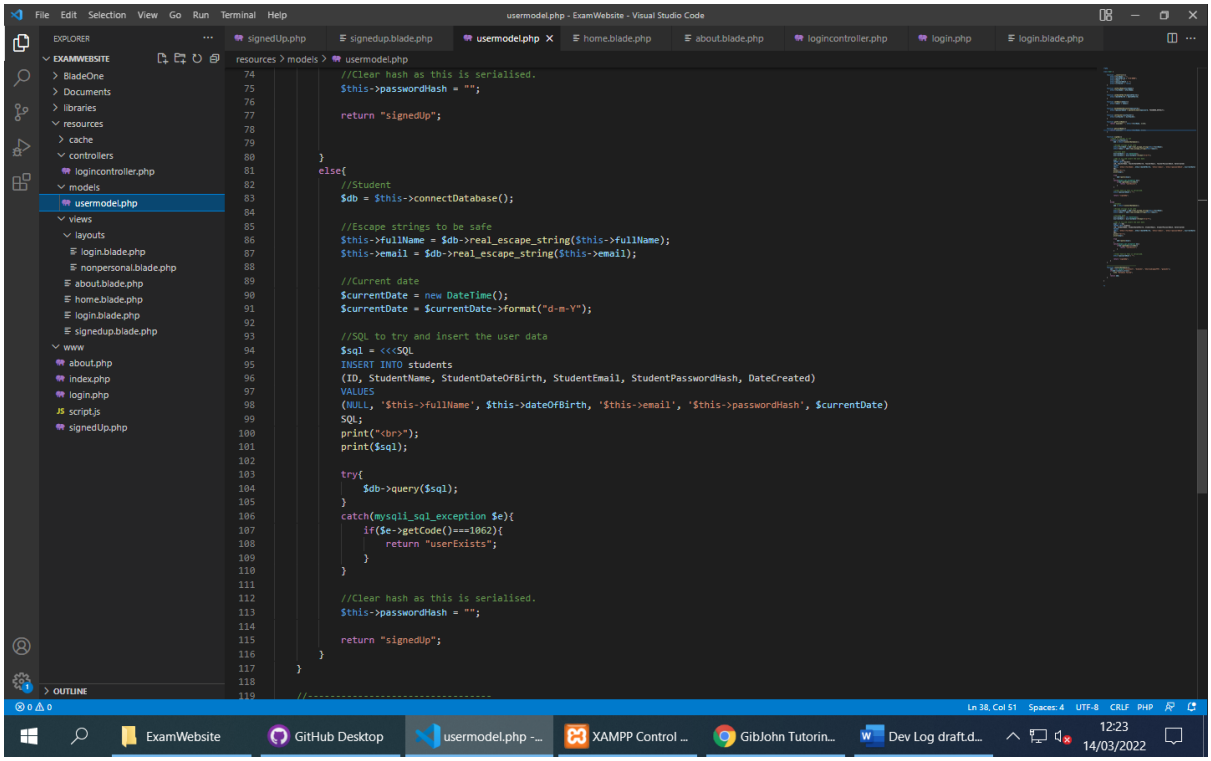

Arrow left - Bootstrap Icons
GibJohn Tutoring - Login
localhost / 127.0.0.1 / gibjohn
SQL CREATE TABLE Statement
localhost/phpmyadmin/index.php?route=/database/sql&db=gibjohn
phpMyAdmin
Recent
Favorites
New
gibjohn
information_schema
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: gibjohn
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Triggers
Tracking
Designer
Run SQL query/queries on database gibjohn:
CREATE TABLE Students (ID int PRIMARY KEY AUTO_INCREMENT, StudentName VARCHAR(64), StudentEmail VARCHAR(255) UNIQUE, StudentPasswordHash VARCHAR(255), DateCreated DATETIME);
Clear
Format
Get auto-saved query
Bind parameters
Bookmark this SQL query:
Delimiter
Show this query here again
Retain query box
Rollback when finished
Enable foreign key checks
Go
Console
11:13
11/03/2022

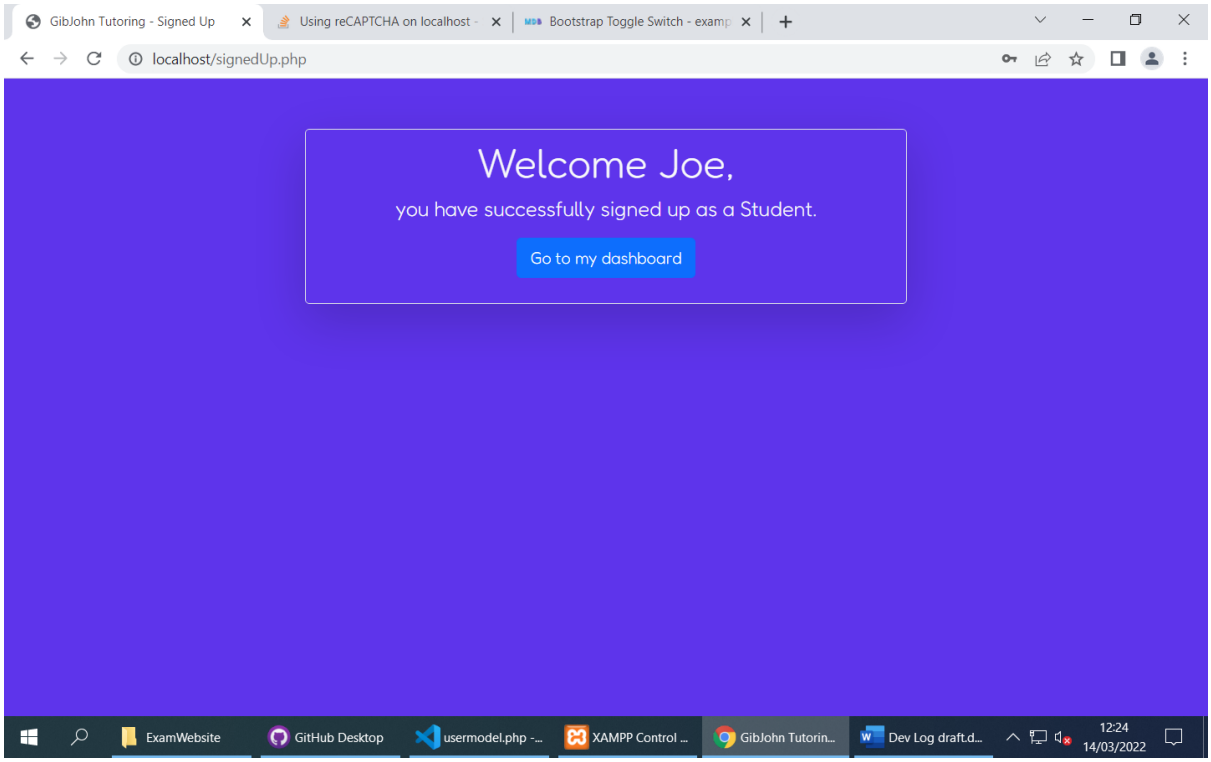

Arrow left - Bootstrap Icons
GibJohn Tutoring - Login
localhost / 127.0.0.1 / gibjohn
SQL CREATE TABLE Statement
localhost/phpmyadmin/index.php?route=/database/structure&server=1&db=gibjohn
phpMyAdmin
Recent
Favorites
New
gibjohn
New
students
teachers
information_schema
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: gibjohn
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Triggers
Tracking
Designer
Filters
Containing the word:
Table
Action
Rows
Type
Collation
Size
Overhead
students
Browse
Structure
Search
Insert
Empty
Drop
0
InnoDB
utf8mb4_general_ci
32.0 KiB
teachers
Browse
Structure
Search
Insert
Empty
Drop
0
InnoDB
utf8mb4_general_ci
32.0 KiB
2 tables
Sum
0
InnoDB
utf8mb4_general_ci
64.0 KiB
0 B
Check all
With selected:
Print
Data dictionary
Create table
Name:
Number of columns:
4
Go
Console
11:14
11/03/2022



- Started working on the login system for students and teachers using Model View Controller.
- Created a database with a Students and Teachers table
- Login page sends POST data to PHP script
- Script calls Login Controller
- Login controller validates inputs and handles the login or sign-up procedure using the User Model
- Made sure code is well commented with credit if I used something to help me.

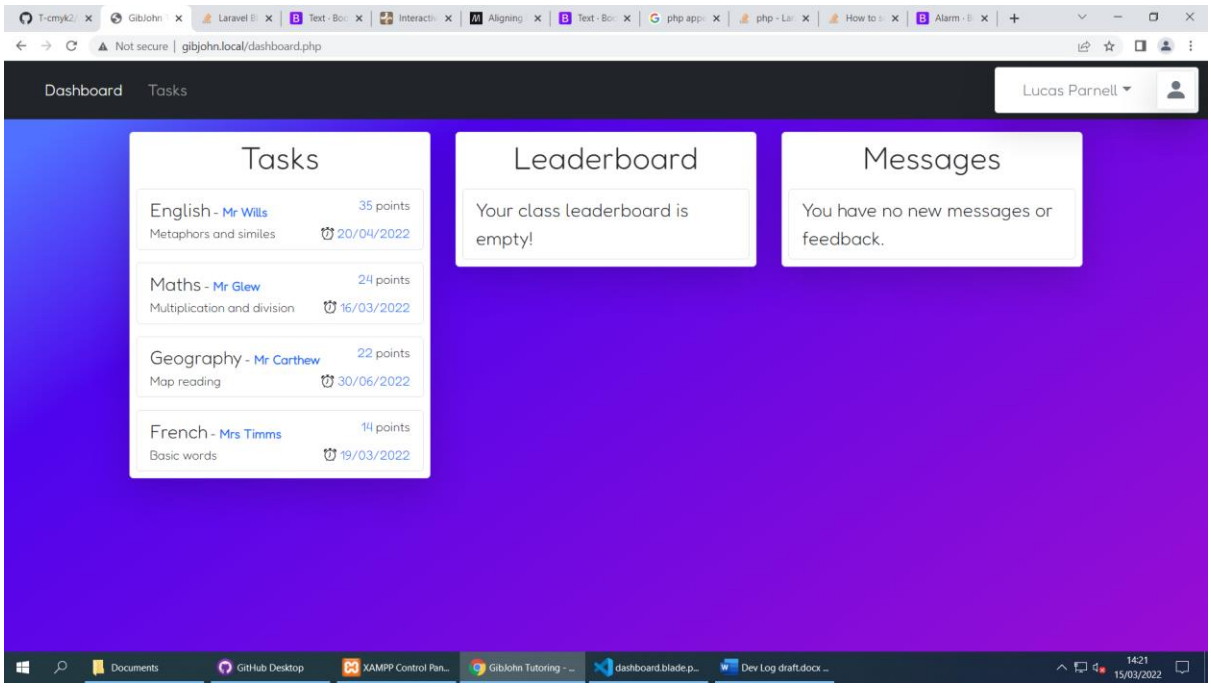
Day 4



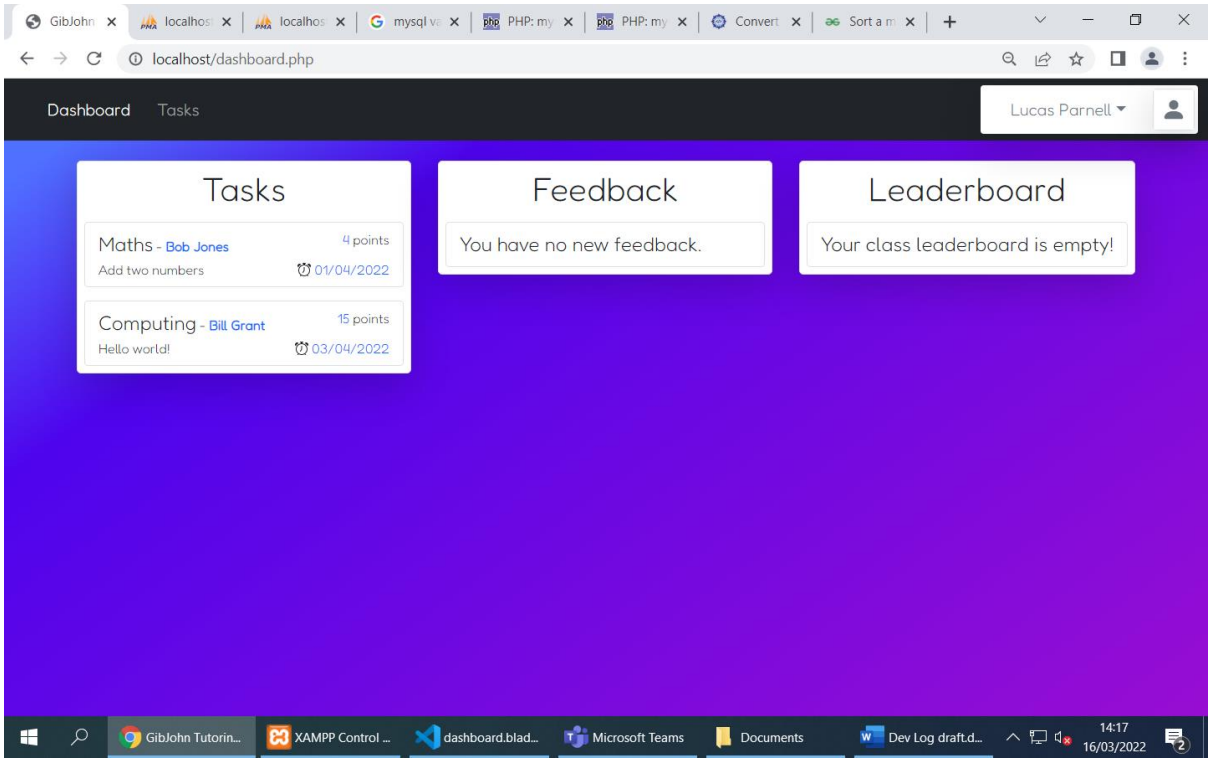


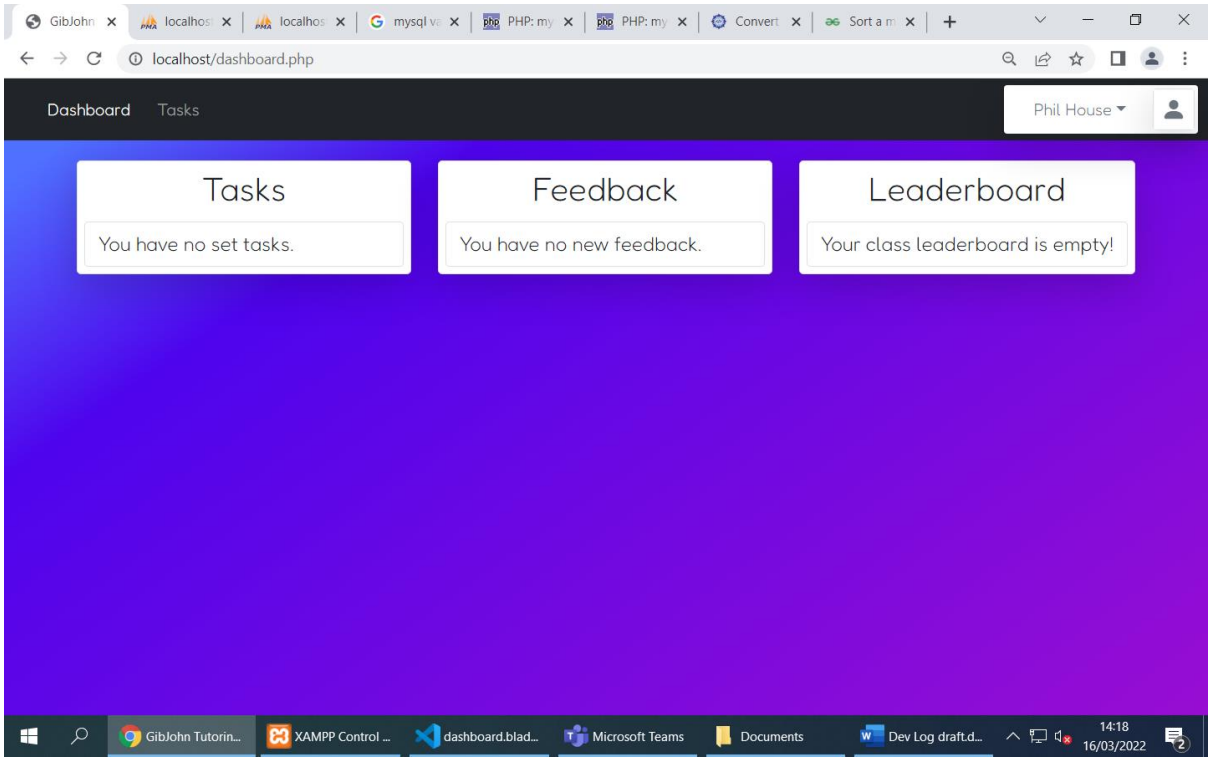
- Sign-up is now functional
- User model gets serialised and stored in the session (Sensitive data is removed)
- Some validation that checks age and duplicate users as well as sanitising the inputs (Removing illegal characters, Escaping the rest).
- Will finish login next and then begin creating a personalised layout and dashboard.

Day 4

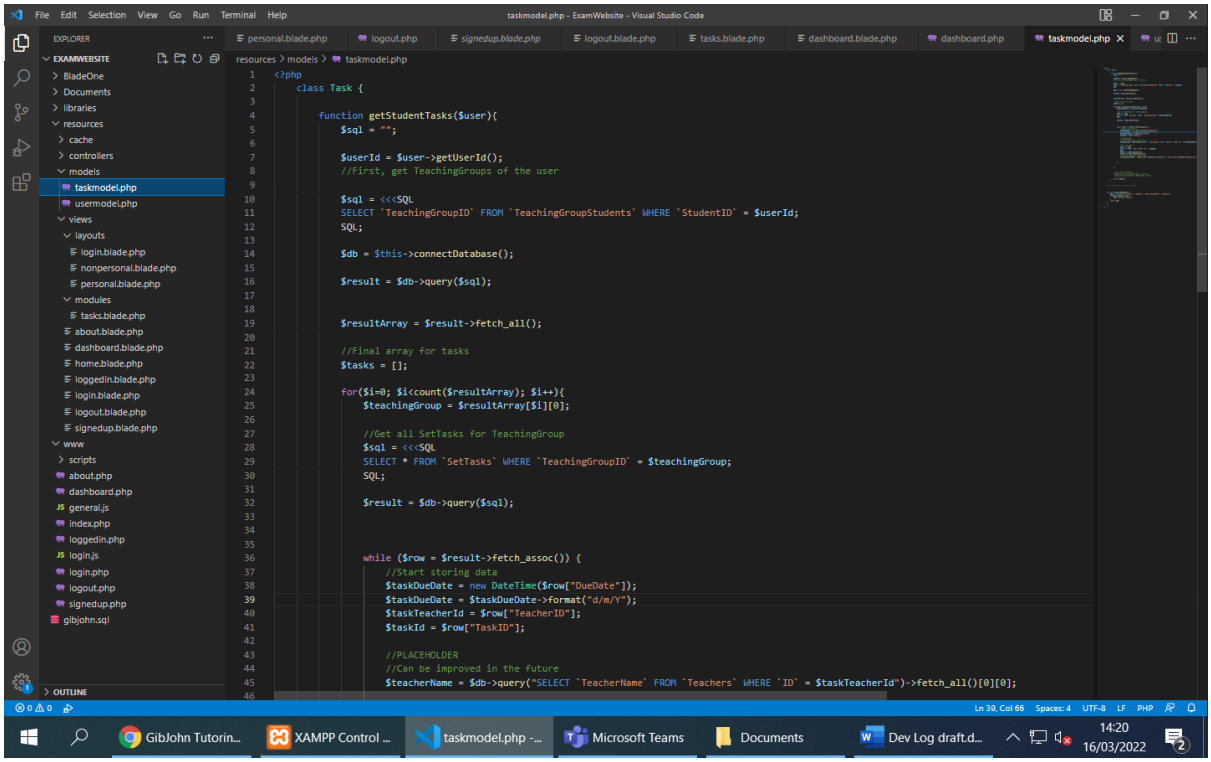


- Login system is mostly finished, just need to add client-side validation
- Worked on the personalised student dashboard, Templates can be used to easily add Tasks to the interface.
- Started working on a profile drop down interface that displays some information about the user as well as a log out button

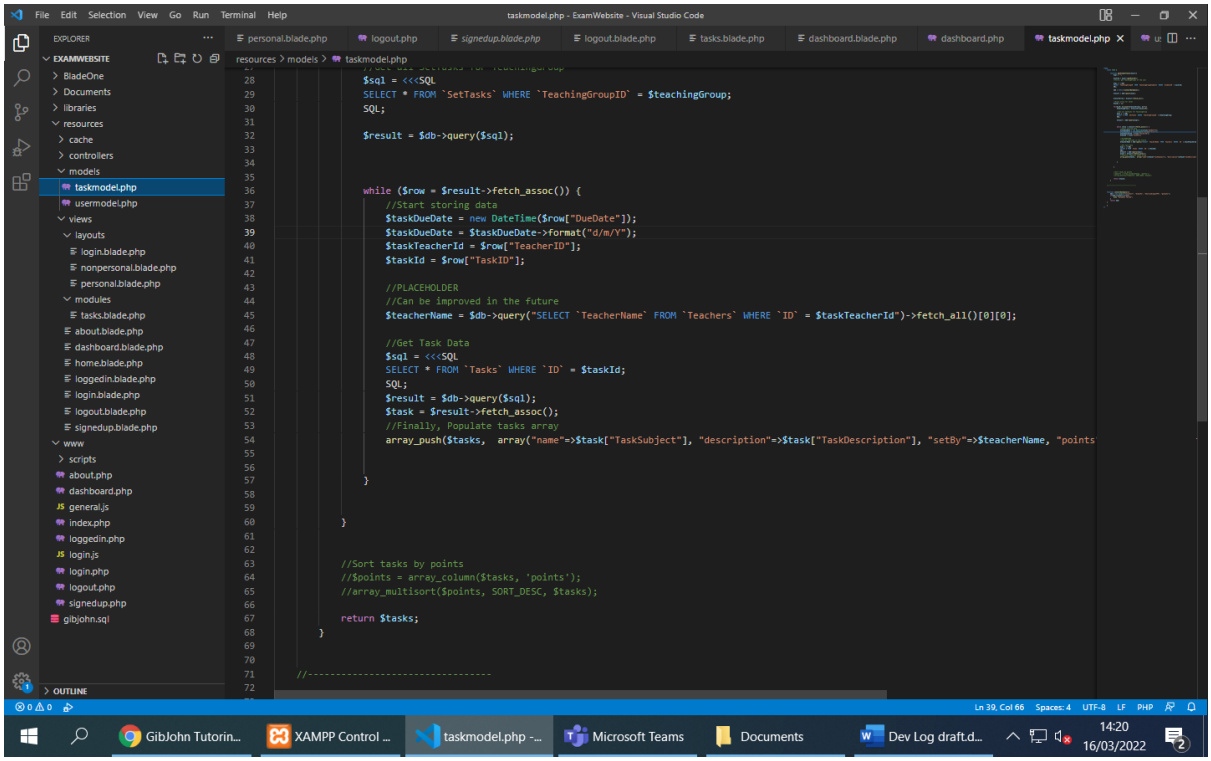




(Different users have different set tasks)

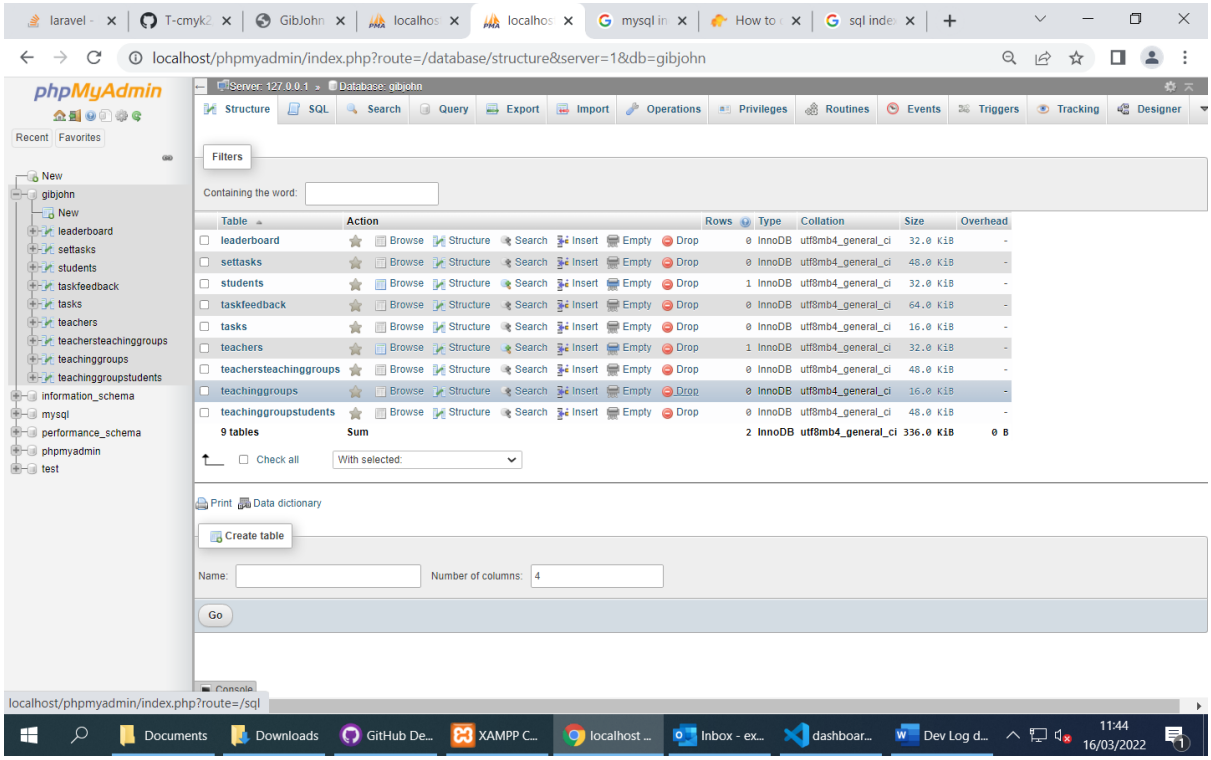


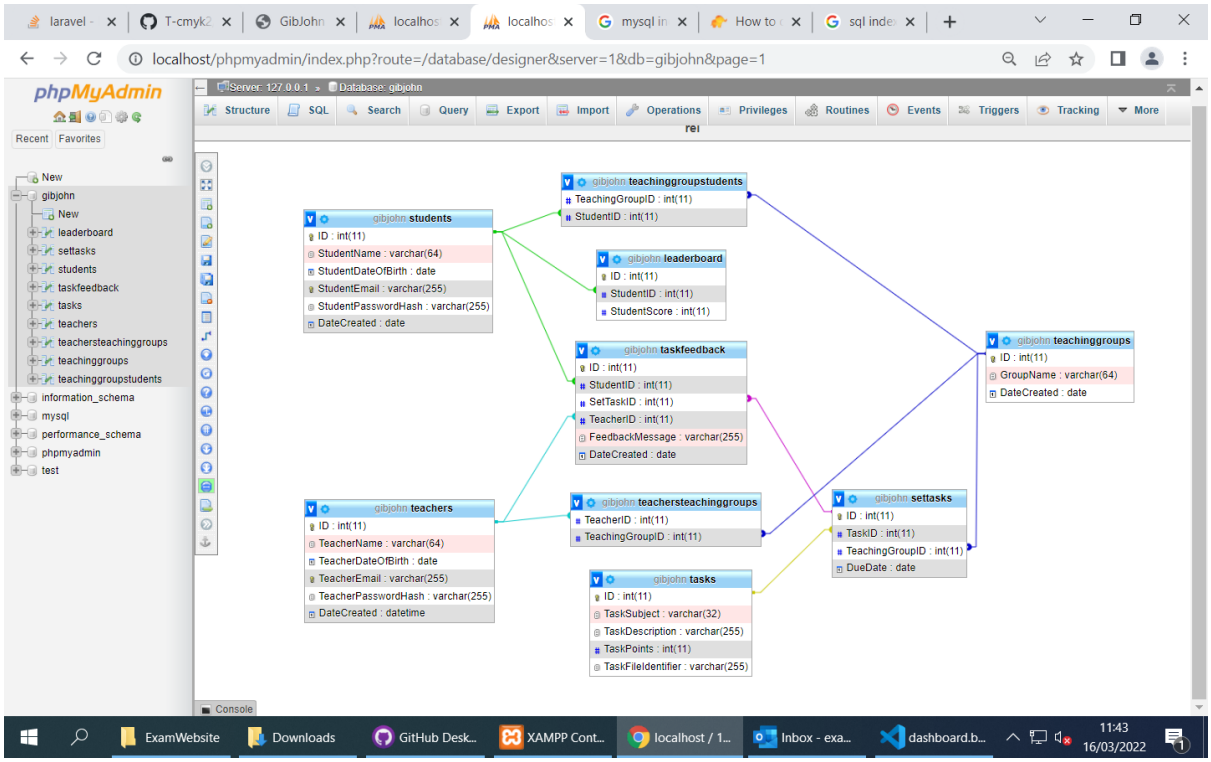
Some of the prototype code for getting a list of tasks for a given student



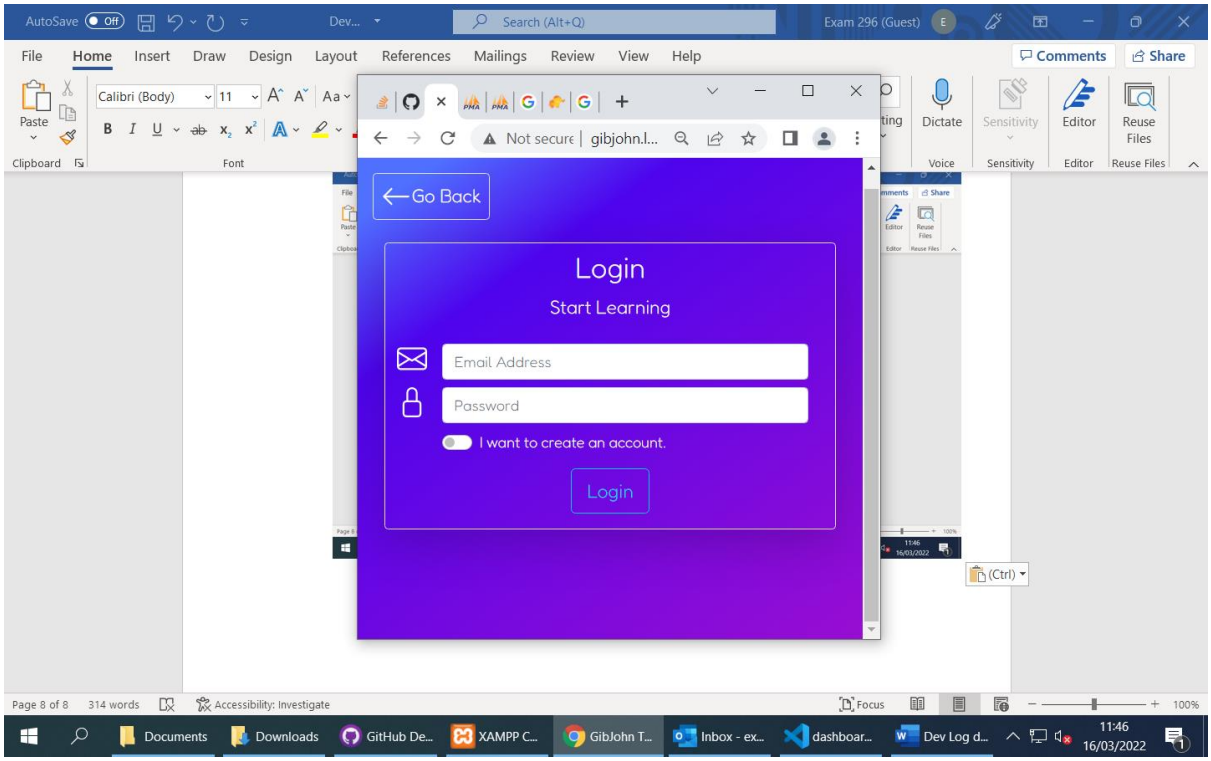
- Programmed the task model to pull a student's tasks from the database by first finding their teaching group and then getting their set tasks. (Set tasks are different from tasks)
- Fixed a few issues with dates not being stored correctly
- Added some entries into the database
- Next, I will program some of the feedback model and start on the student tasks page.

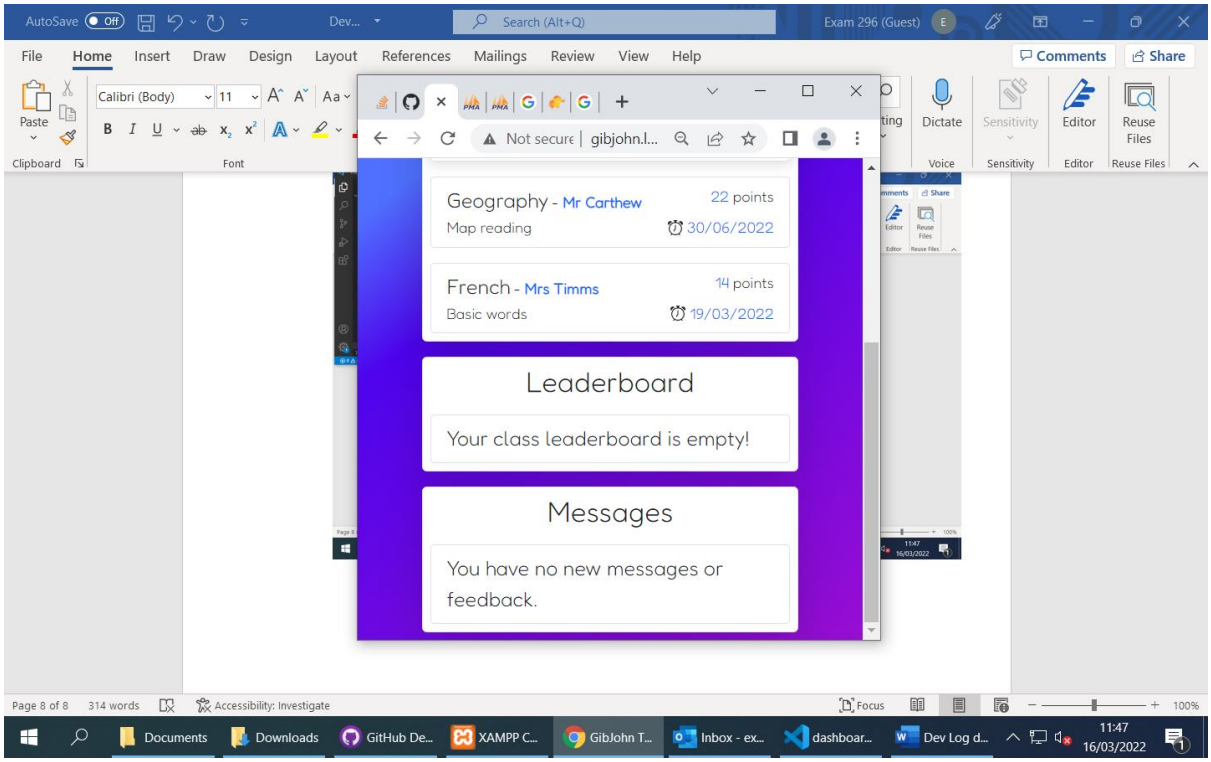
Day 5

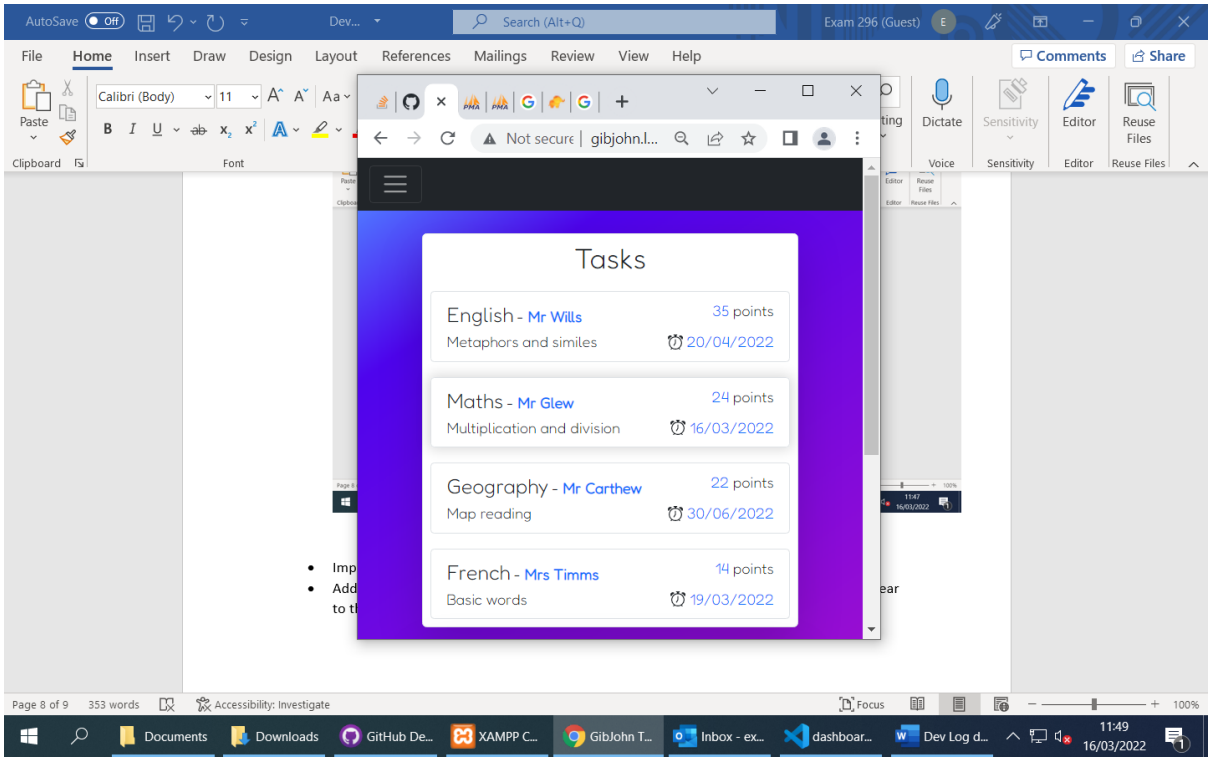




- Implemented the database from the design documentation
- Created all relationships after setting indexes
- Made the database design similar to my original documentation

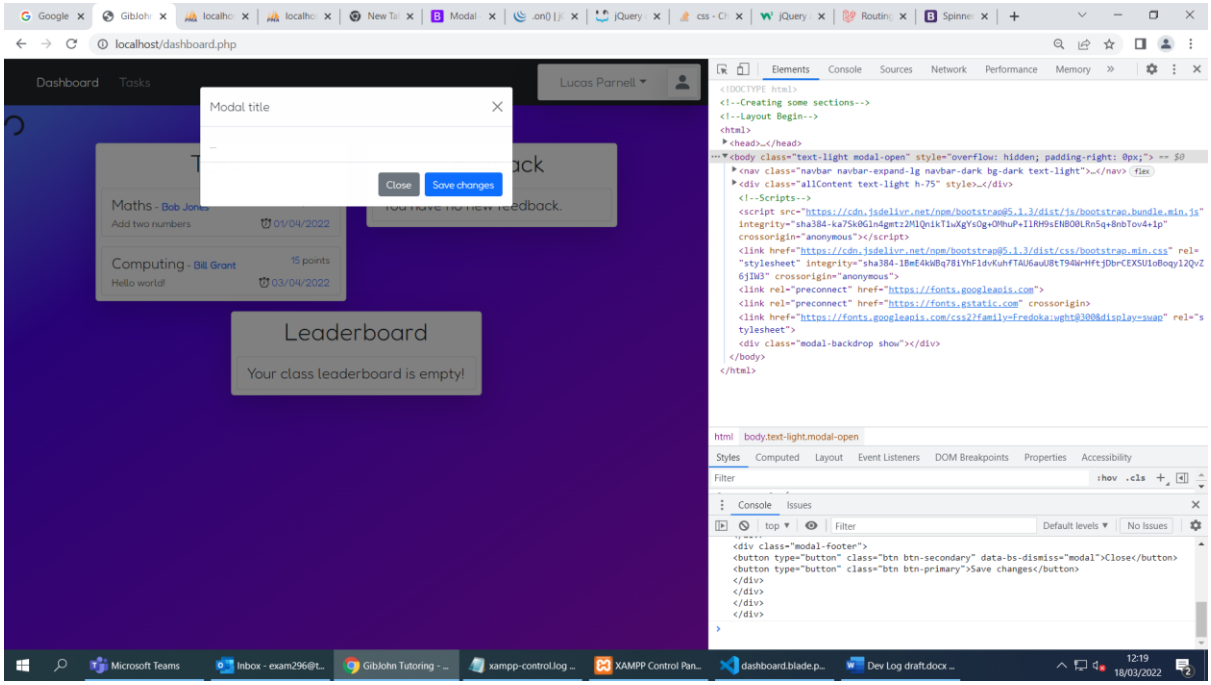

Not secure | gibjohn.l...
Go Back
Login
Start Learning
Email Address
Password
I want to create an account.
Login

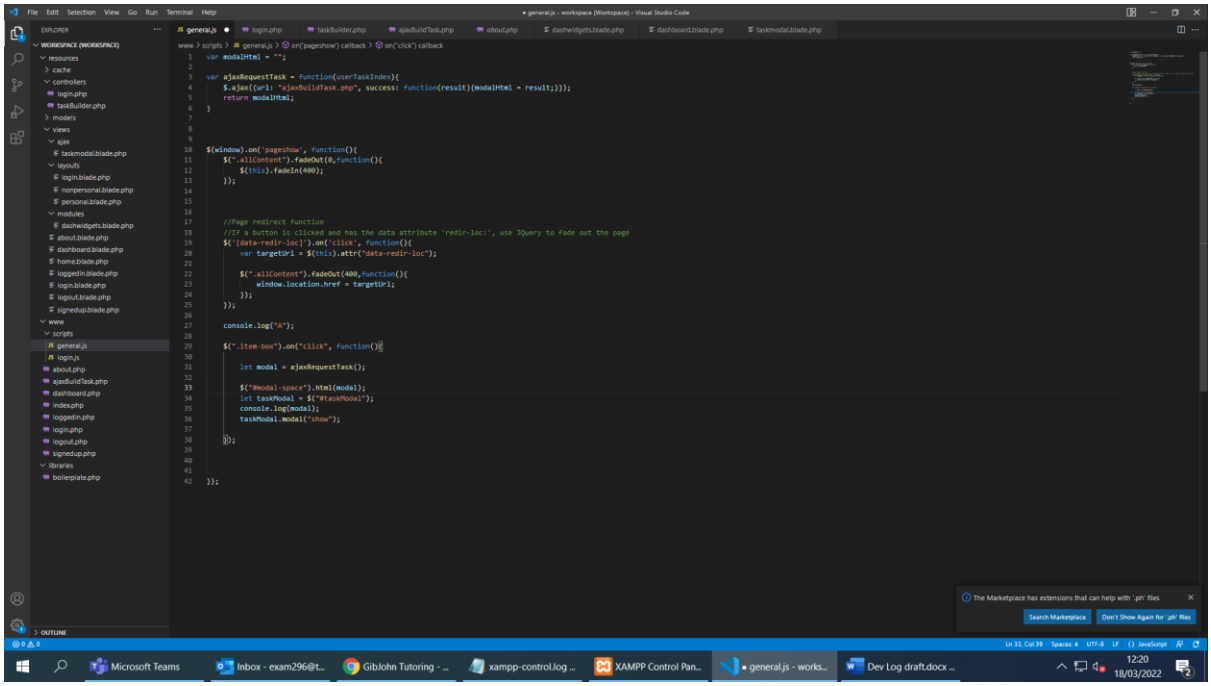

Not secure | gibjohn.l...
Geography - Mr Carthew
Map reading
22 points
30/06/2022
French - Mrs Timms
Basic words
14 points
19/03/2022
Leaderboard
Your class leaderboard is empty!
Messages
You have no new messages or feedback.



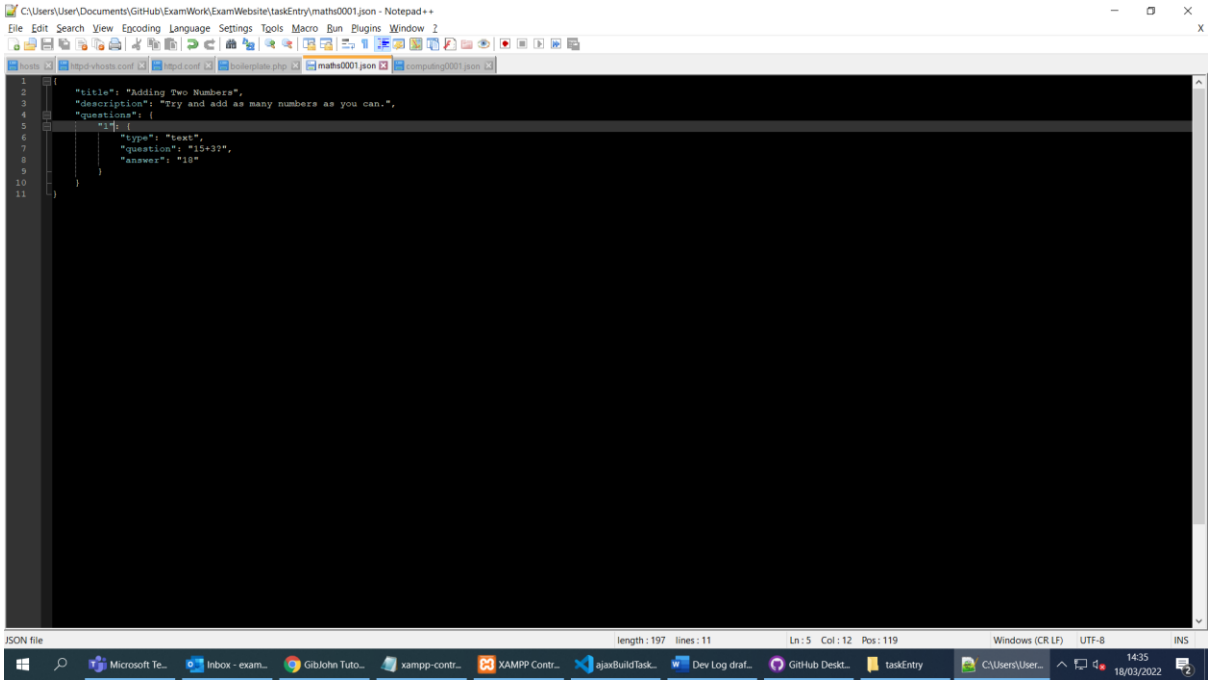
- Improved scaling of the flexboxes, website is now usable on mobile devices
- Added a shadow effect to the task items when the mouse is over them, this makes it clear to the user that the item is being selected.
- Next will start on AJAX task system

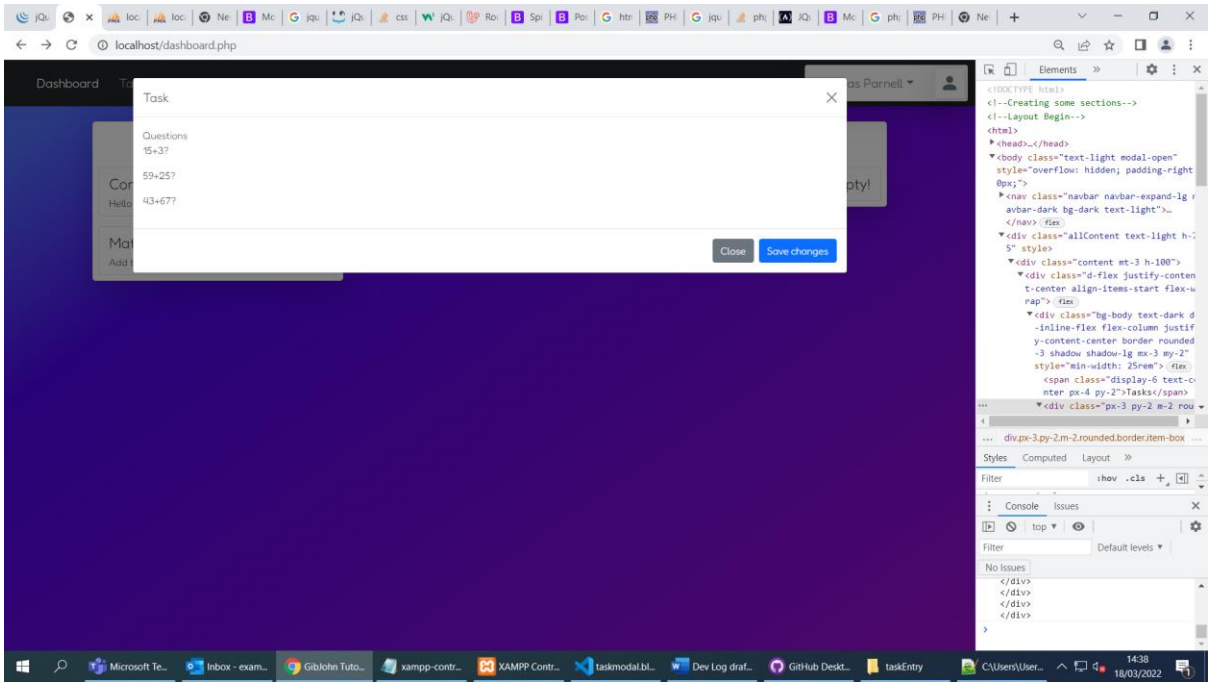
Day 6





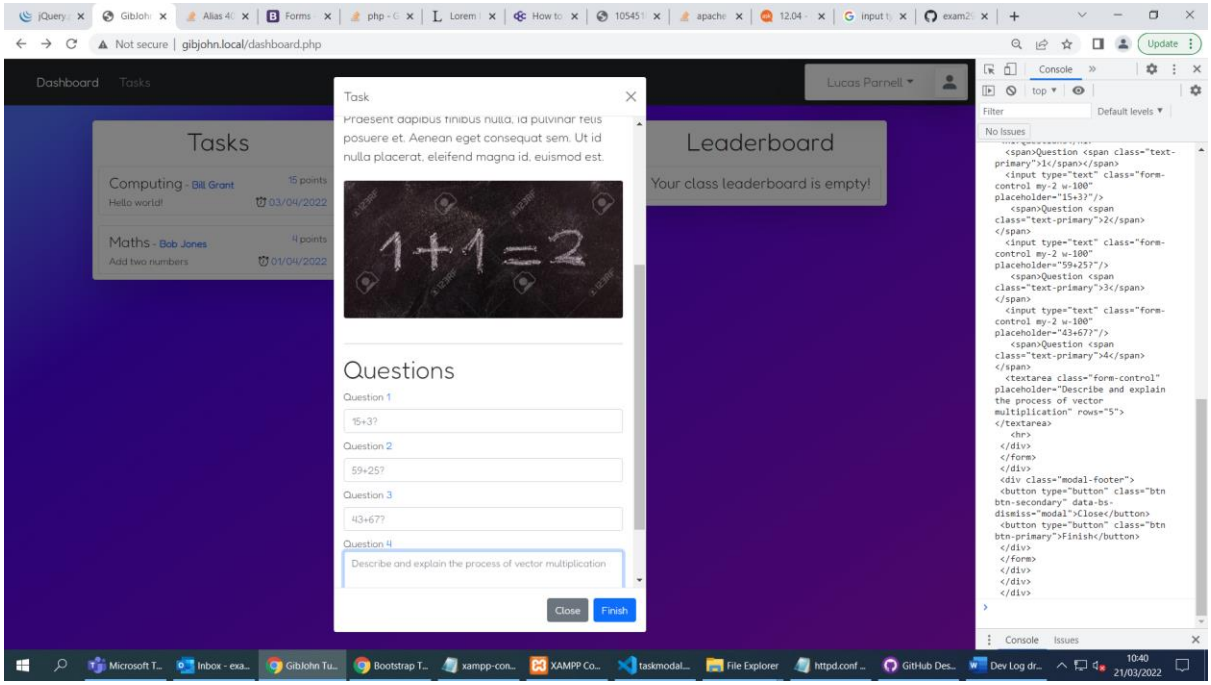
- Started on AJAX code to build and load task content using JSON files on the server and templates for the HTML, JSON files will contain information and the actual content of the task, Files are referenced in the database entries for each task under the column “taskFileIdentifier”
- Set up basic AJAX request, client can request a modal from the server, it then places it into an empty div and renders it using $(“[modalSelector]”).modal(“show);
- Added a loading icon while the content is being created as depending on the load of the website it may take some time.
- Server can now read from JSON file and push the data into arrays, these can be sent to the template which can be processed and returned to the client.

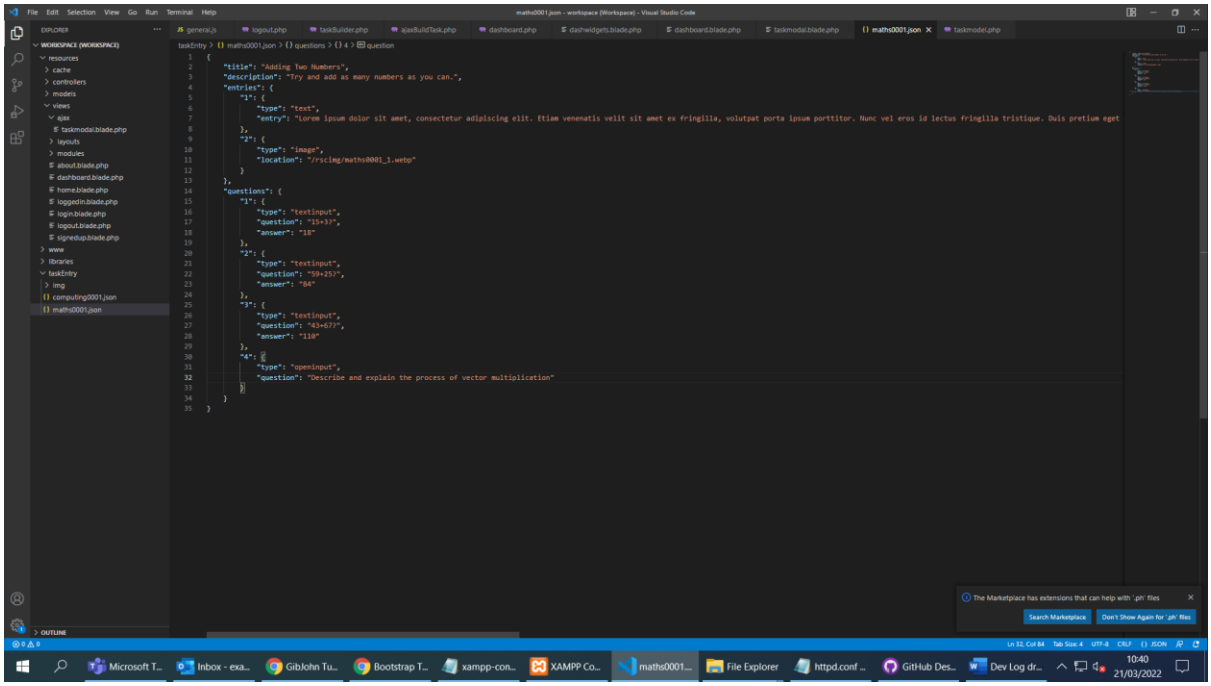




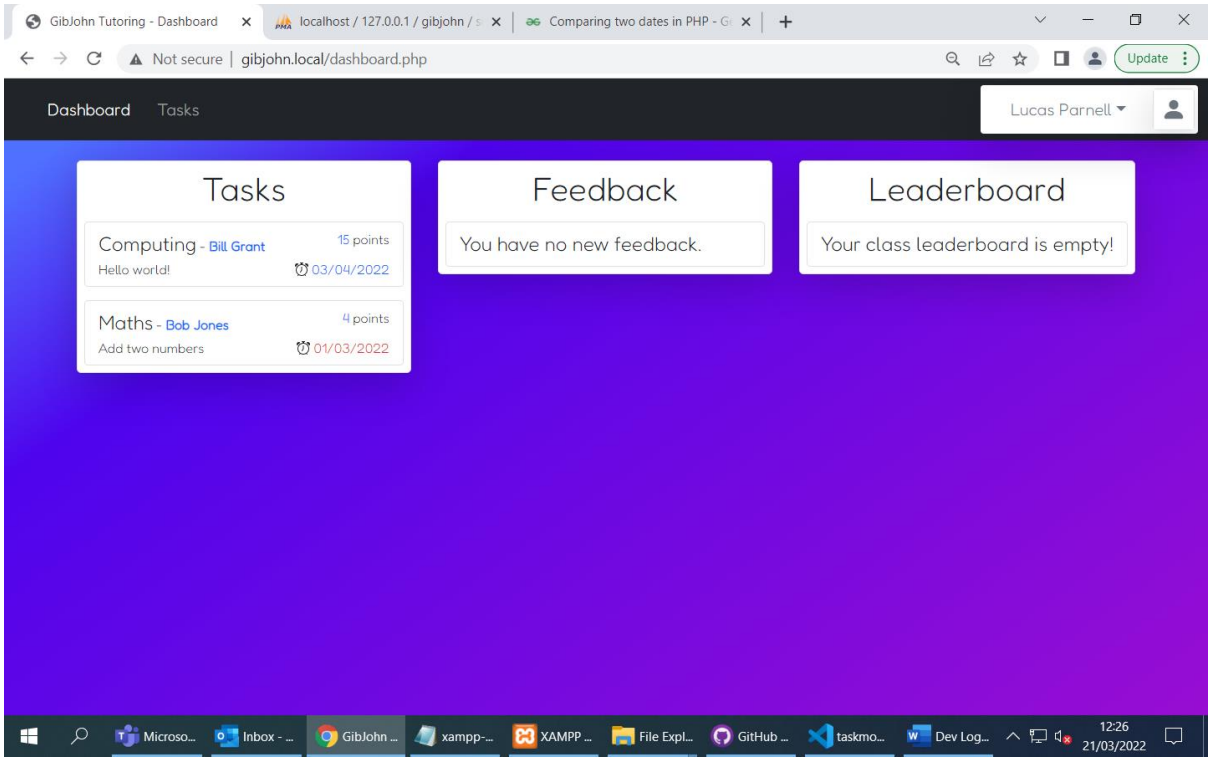
- Questions passed to template

Day 7



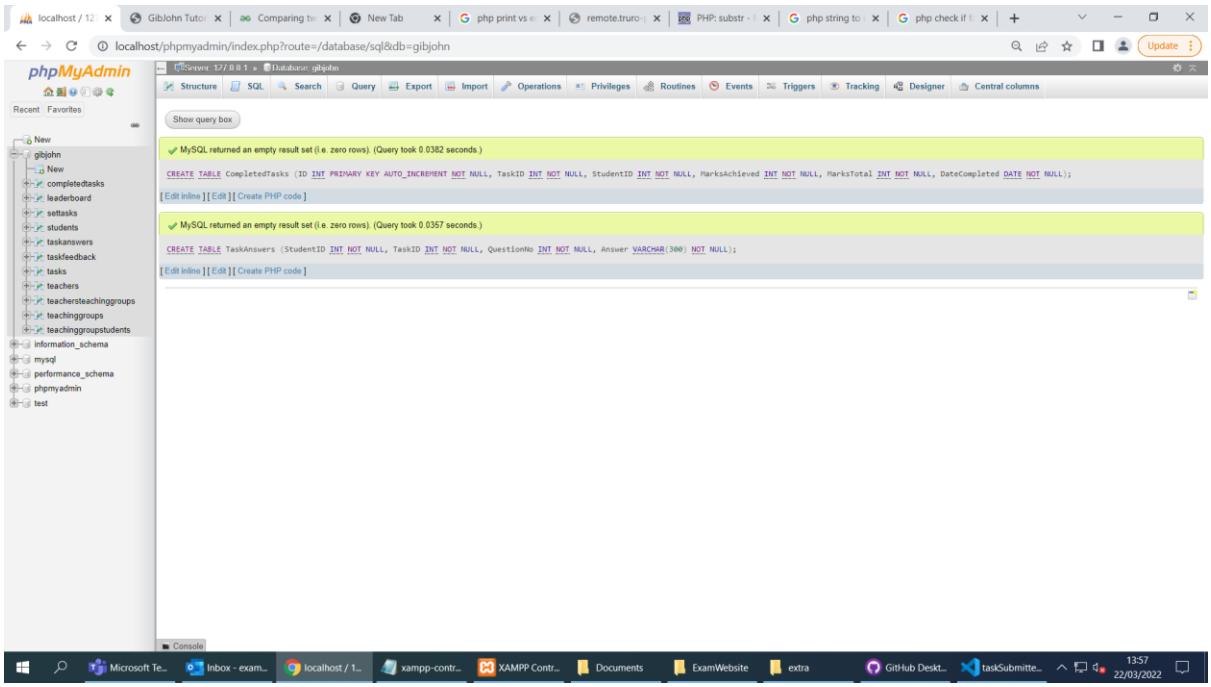


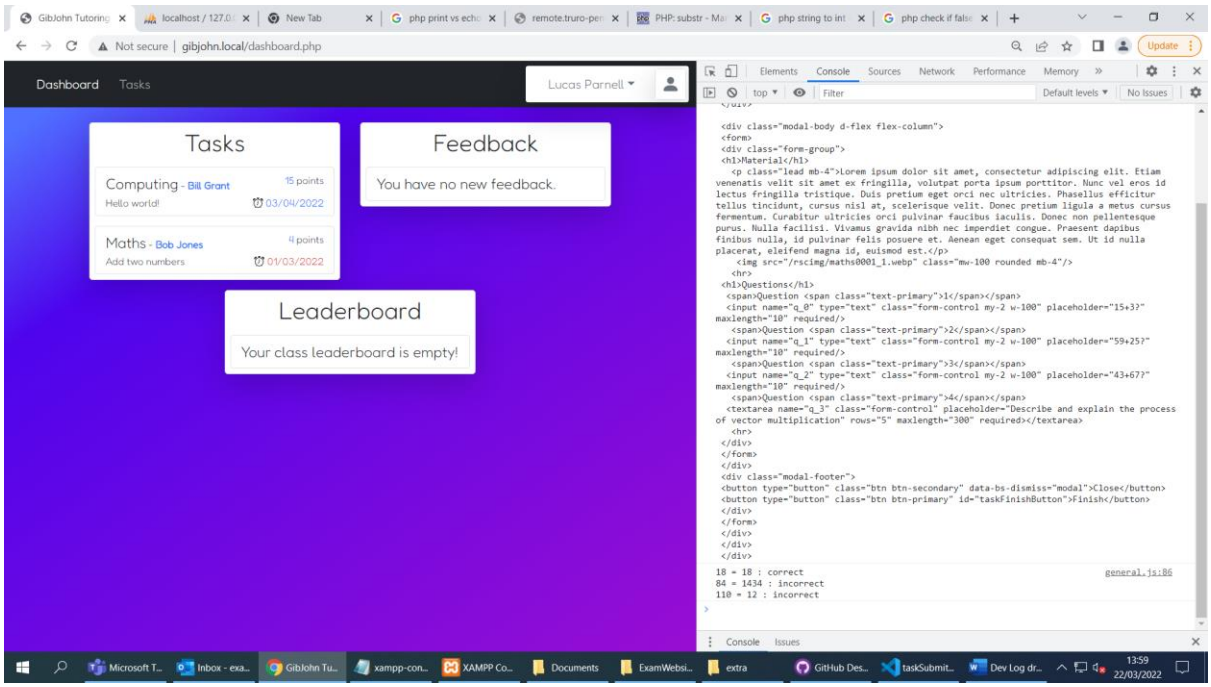
- Almost finished task feature
- An alias set in the vhosts file is used to access task images without introducing security issues
- Can easily add questions, descriptions, textboxes, images and textareas in the JSON file
- Blade takes these variables and sorts them into a presentable template which is sent to the client.



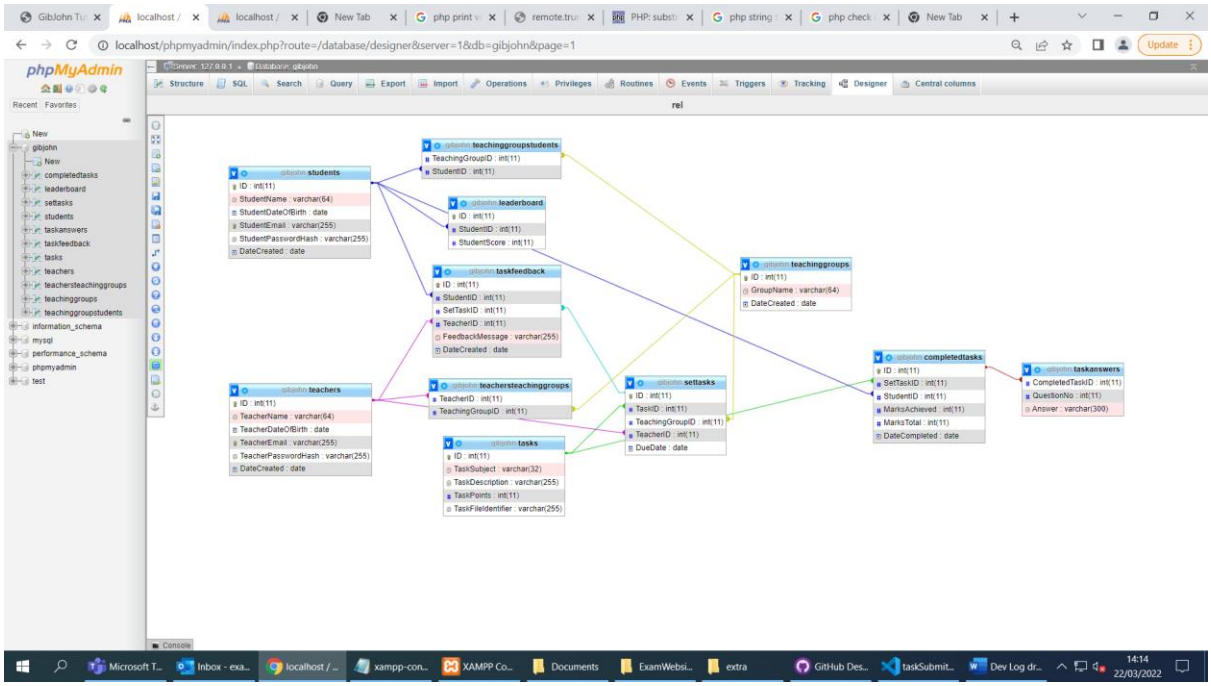
- Added an indicator for whether a task is overdue
- Started working on task submit process on the server-side
- Started work on some of the teacher side features

Day 8

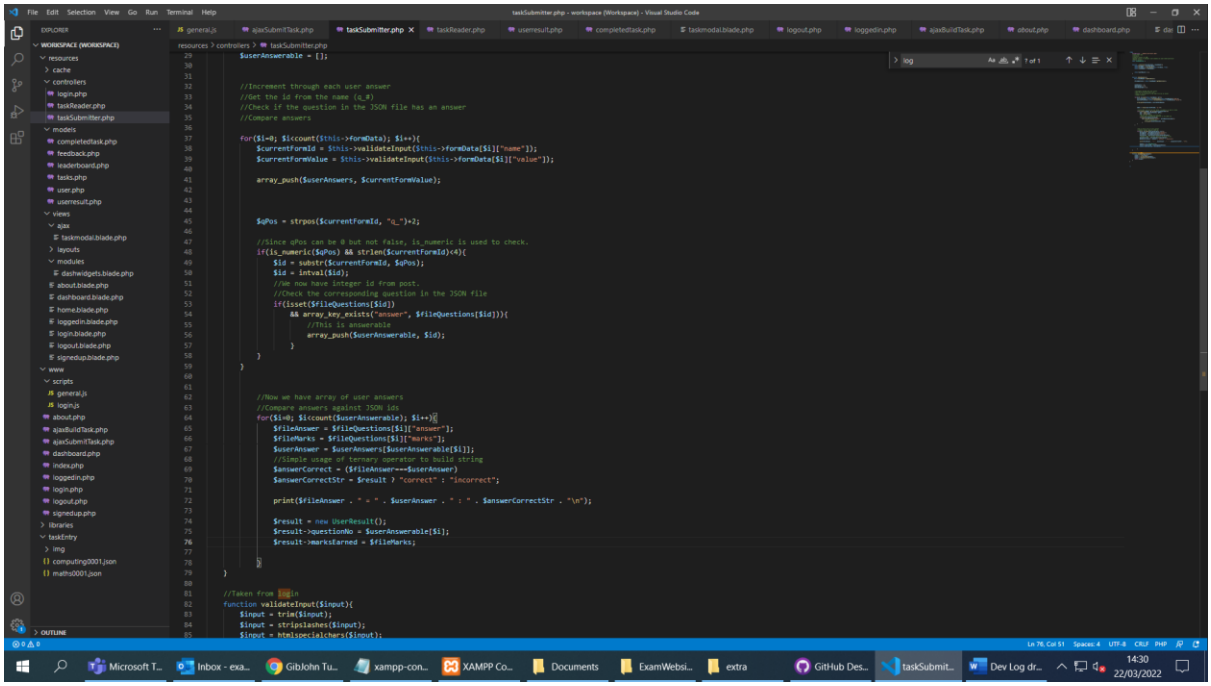




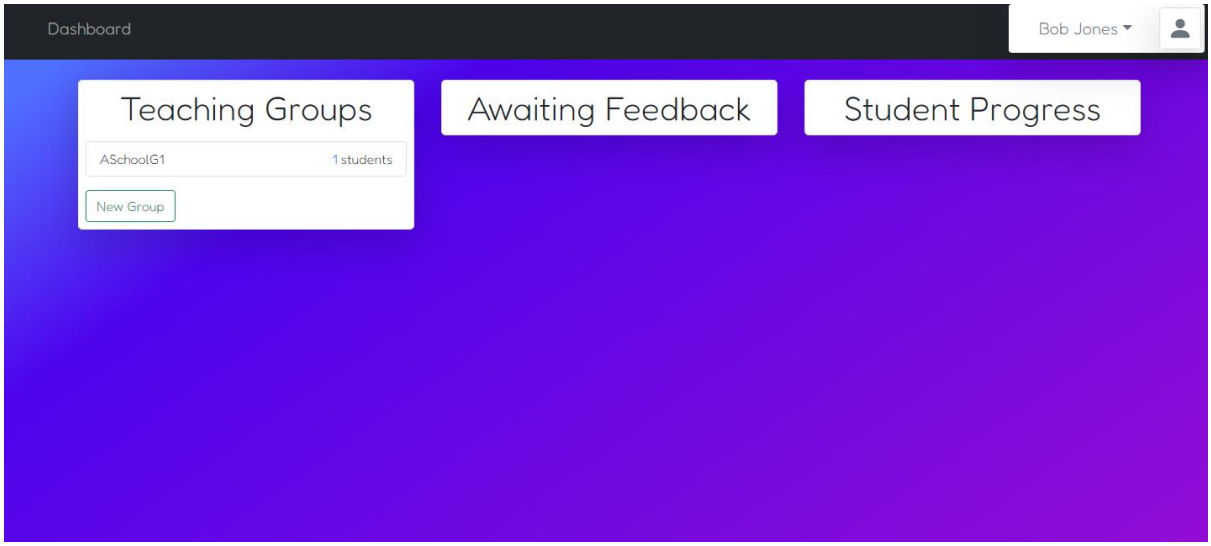
- Answers are shown in the console after being returned by the AJAX PHP script



- Student side is functional, task answers are stored in the database
- Points are stored in leader board and displayed
- Since the database is relational, if a completed task is deleted, all of its associated values in other tables are too

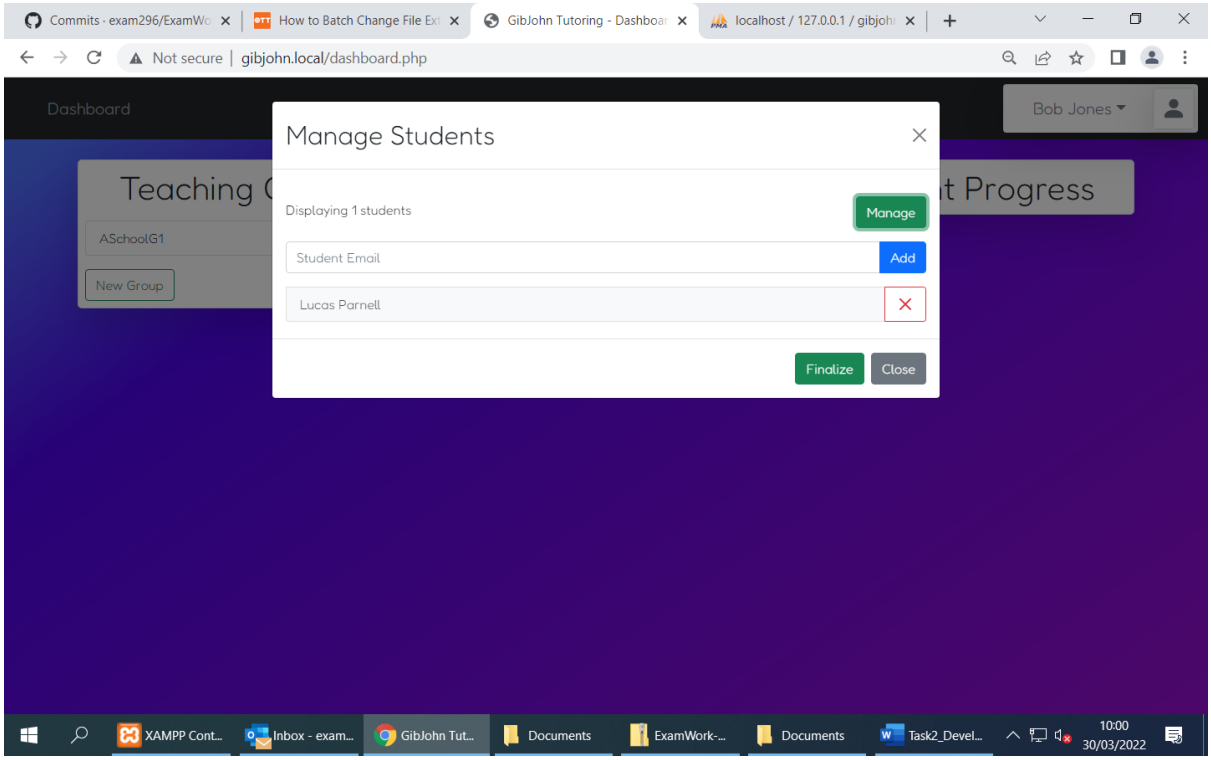


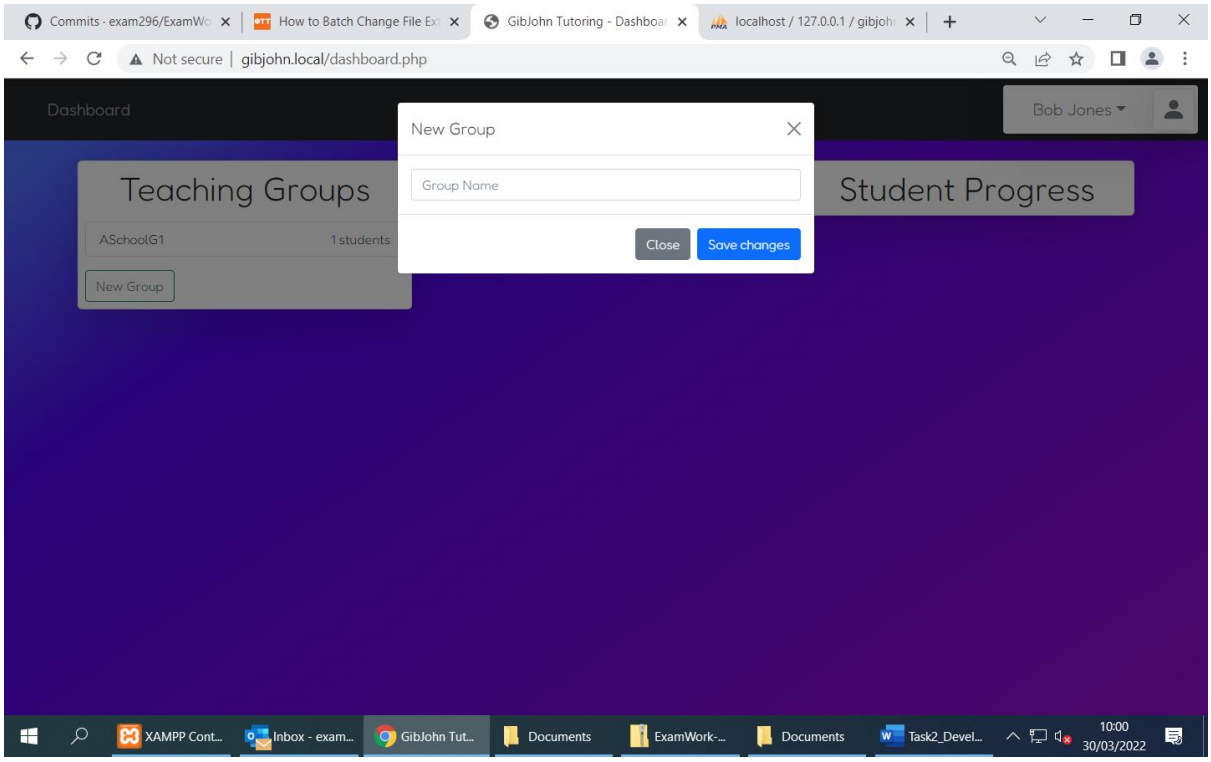
- Code to check user answers, open-ended answers will be added to task answers and teachers will be able to give feedback and award marks for them.



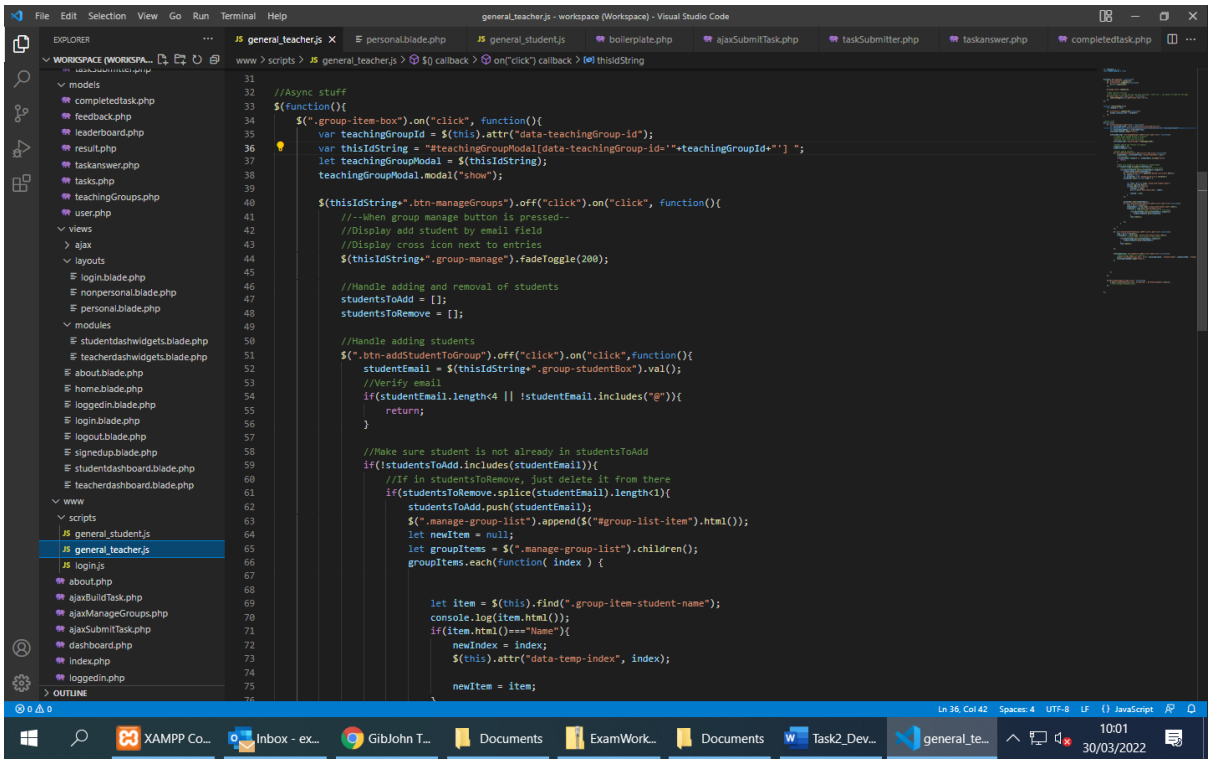
- Created layout for teaching groups on the teacher side.
- Will finish this next.

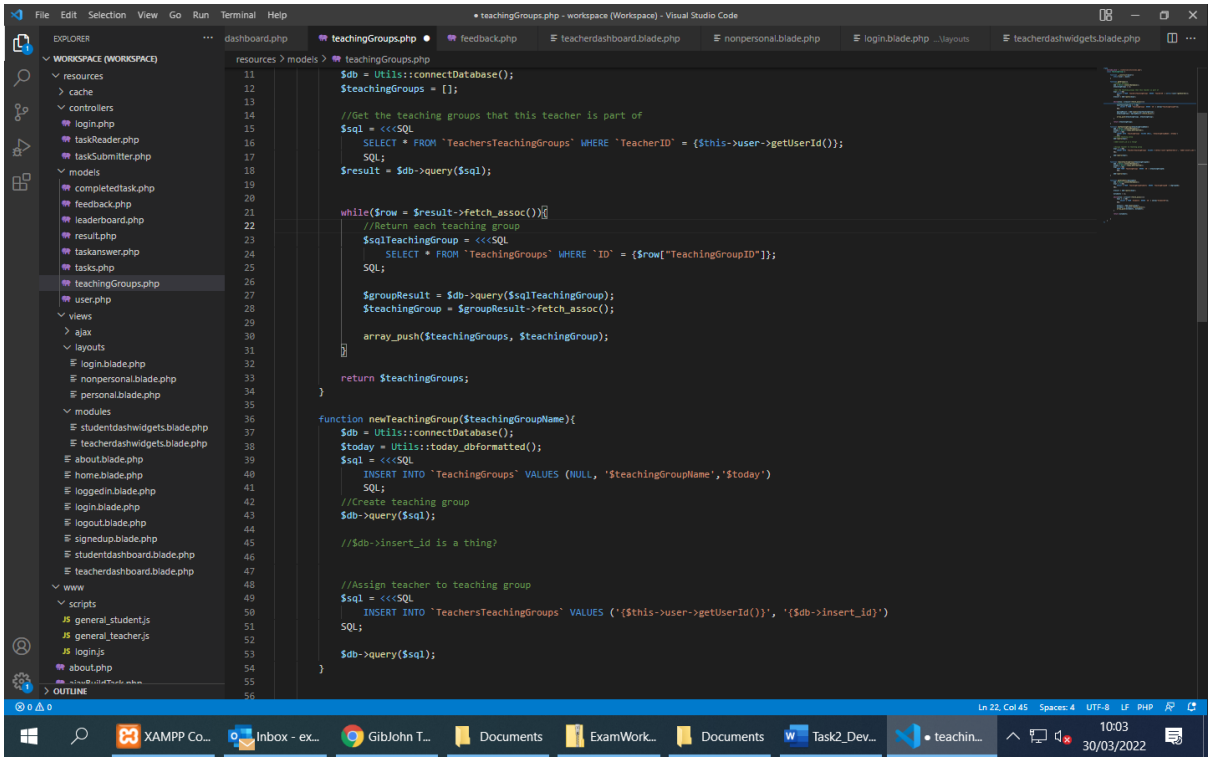
Day 9





- Teachers can now create new teaching groups
- Teachers can now assign and remove students to groups given their email address
- Separated the student script file and the teacher script file
- Started writing ajaxManageGroups.php to handle teaching group changes





Some of the TeachingGroup class