Dev Log draft

Day 1

* Set up workspace
* Created a base for the home page
* Created a base for the about page
* Implemented bootstrap

Day 2

* Further worked on the home page

A screenshot of a computer

Description automatically generated

* 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, will work on more tomorrow.

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

Day 3

Graphical user interface, text, application, email

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

* 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

Day 4

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

* Sign-up is now functional
* User model gets serialised and stored in the session (Sensitive data is removed)
* Some validation for age and duplicate users (and input sanitation)
* Will work on login next and then begin creating a personalised layout and dashboard.

Day 4

A screenshot of a computer

Description automatically generated

* 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

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Programmed the task model to pull a students tasks from the database by first finding their teaching group and then getting their set tasks. (The two users have different task lists)
* Fixed a few issues with dates
* Added some entries into the database
* Next, I will program some of the feedback model and start on the student tasks page.

Day 5

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Implemented the database from the design documentation
* Created all relationships after setting indexes
* Made the database design similar to my documentation to make things easier

Graphical user interface, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

* 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 time will start on AJAX task system

Day 6

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

* 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.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Questions passed to template

Day 7

Graphical user interface, application

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

* Almost finished task feature
* An Apache alias 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.

A screenshot of a computer

Description automatically generated

* Added an indicator for whether a task is overdue
* Started working on task submit process on the server-side