Challenge Week

# Project overview

My project is to create an Online Teaching Platform. At its most base form, it will allow teachers/tutors to be able to manage their teaching materials and classes from one central place, keeping each class separate from each other. I also intend on allowing students to be signed up to the service, assigned to classes (and groups within those classes) and interact with specific resources that they have permissions to do so with. It would also be ideal for teachers to be able to create assignments for the students to be able to access resources for and to make submissions for the teachers to review and assign marks for.

Finally, allowing communication channels for students to interact directly with their teachers and in group forums would be a great feature to include. This is more of a stretch goal than a requirement, as this is not a simple feature and is distinct enough that the rest of the application can be developed without having it included in the first full release at the end of this project.

# Similar Product Research

There are many platforms that provide an online teaching experience. These include services such as Udemy, SkillShare and WordPress sites using the LearnDash plugin. Each of these implements the experience differently, with access to courses and adding courses being achieved in different ways. Moodle is also another teaching platform, and is the most similar platform to what I intend on producing.

## Udemy

Udemy is a free to access learning site that puts a paywall over all the courses. Creators can sign up to the site and create a course very quickly. Learners can sign up to courses (paying whatever charge is applied) and have unlimited access to them. They can access these from their own "Courses" dashboard.

Before putting together their first course, the creator must go through a short survey, which is likely to be used for internal statistic within the Udemy Business than for anything functional. It also serves top allow them to suggest some of their own resources to get the creator going based on their level of experiences in teaching, video creation and gaining an audience. Once this survey has been completed, the creator gains access to an instructor dashboard. Here they can see what courses they have, their status etc. It is also here that they start the creation of a new course (via an action button), and receive learning resource suggestions from Udemy.

The process of making a new Course is very simple. The user is led through a multi-step form that focuses on different aspects of the general makeup of the course, and provides a good UX to do so. Once that form has been filled in, there is a full CMS provided for the creator to put together the working components of their course, such as Learning Outcomes, Requirements etc. It also provides a space for each component in the course to be added. Once the course has been fleshed out with content, it is then submitted for an internal review by Udemy.

The way that the application UI and UX is designed is something that is going to inspire my own designs and application flow quite significantly. As the application that I am going to be making is not a SASS product, and is going to be a working example of what can be provided to a teaching institution (such as the University of Essex), the separate parts of the app will only be accessible to those with related roles (Teacher/Student/Admin).

My research for this service has been done solely through interacting with the site myself. This was possible as all aspects of the creation service are free, and I own a few courses myself.

## SkillShare

SkillShare offers a different business model to generate income from courses. Where Udemy uses a Course as a Product model, SkillShare provides a Subscription service to access all content that they provide. Teachers are then paid out based on their course metrics. To summarise, they are paid out from a set teaching fund based on their engagement, total time spent on their content (with a minimum of 75 minutes across courses a month) (1).

The student experience of SkillShare is very similar to that of Udemy, providing a simple and clean UI/UX for them to interact with the service and classes. The Teacher experience is somewhat different in terms of signing up, but the actual course creation is very similar. (2) (3)

I personally prefer the Udemy experience completely, as it provides a fresher UI and has a simpler UX for all users to interact with. The other aspects (such as paying for the service or course creation restrictions) are not related to the product that I intend on outputting, so I will not provide my opinion on those.

## LearnDash WP Plugin

LearnDash is a very different product to the previously mentioned services. First of all, it is a plugin to be used on a WordPress site. WordPress is a technology/Framework that has evolved over the years from a simple Blog CMS to something a lot more complex. The use of plugins such as LearnDash change the base behaviours of the basic CMS, and the use of themes are used to change how the content is outputted to the user.

Reviewing the LearnDash website, it outlines what can be done with the plugin. Courses can be broken down into various sections, lessons, and topics. At each of these points quizzes can be placed. (4). I personally like the way that the courses can be broken down this way. It allows the content to be broken down into small chunks to better categorise and group content, which can provide an effective UX. Whilst I do not intend on my content to be structured in the same way, having this kind of structured nesting is something that I intend on bringing into my project, as I feel like having this will make both creating content and resources simpler, but also making consuming and accessing it easier as well.

My experience of how actually using this plugin comes from a video that goes through the plugin and shows off all the features and experiences that can be used with the plugin (5). The video is a few years old, but is the current video that is used on the developer’s website for the plugin.

The layout of the content for the students is very clean, and I think that there are a lot of different aspects of it that I want to include. There are too many for me to list them all here, but a few of note is the shape of the CTA buttons, the use of accordions to collapse content groups and progress bars to show course and lesson completion.

Given the limitations of the WordPress admin, I also think that the Teacher experience is just as useful. It is more complicated than the previous two solutions, as it has a lot more options on how to display content, down to the colours used in the theme and logos. This is since it is a distributable plugin that needs to be adaptable to different business' branding needs. The actual content creation (which is the prime part that I am concerned about in this part of my review) is more flexible than in the other two solutions as well, and allows for more unique content.

Overall, I think that there is a lot that this solution provides, and will be something that I go back to over the course of my project for both sides of the app (teacher and student). Like the other two solutions though, the actual use of these services is more about providing actual lessons rather than just access to resources for lessons that are taught in person. This leads onto the fourth solution that I have researched: Moodle

## Moodle

Moodle is a service that is very similar with the project that I am going to be working on. Because of this, it will provide a lot of inspiration for features and application flow. I have used the platform extensively over the past 7 years, so am very familiar with many of the features that it offers from a student's point of view. As such I can cherry pick the things that I like about it (such as being assigned only the modules I need, the grouping of module resources etc). As well as this, I can see the things that I do not like about it, such as how the dashboard area works.

I have not experienced the teacher's side of the platform before, so did need to look at some documentation to understand what features the Teachers have access (6). As it is a large system that is highly customisable to a very tight degree, there is a lot to process. The key points that I was looking for was what could be added and how it displays and arranges those options. With that being the case, the following are the key points that I have found that I feel would be useful to include in my solution:

* Course Structure Panel - This allows the sections and folders that have been made to house the content to be moved around and visualised
* Content Creation Panel - This allows the content on the currently selected section to be modified
* Resources/Blocks - This allows the resources and content blocks to be dropped into the current content page in a list of items.

The names that I have applied to these are not what they are called in the documentation, but are suitable descriptors for their functions. These few features (and perhaps related features) are going to make their way in some form into my solution if I can manage it, as they would provide a sizeable improvement to the resulting product.

# Technologies Research

As I am going to be making a Web application, I need to consider the available tech stacks available to me, what each of them brings to the table, and which one I intend on using. The major stacks that are current (and I am familiar with enough to use) are the ME[ARV]N and L[EA]MP. These differ quite a bit, and even have different meanings in different situations. (7)

## ME[ARV]N Stack

MEAN, MERN and MEVN are all stacks that are very similar to each other, differing only in the front-end framework that they utilise. The M stands for MongoDB (or MySQL if you need a relational database), the E is for Express.js and the N is for Node.js. The A, R and V options are Angular, React and Vue respectively.

The database used is the first thing to be considered. MySQL is a relational SQL database, which is used for storing well-structured data that has heavy relationships with each other. MongoDB is the exact opposite of that, and is for storing well customised data structures that may not be related to each other.

Angular is the oldest option, and as such is the 'original' use of this stack for JS development. This has since been replaced as the most popular use of the stack as React has grown in popularity for developers over the years, and is currently the most desired front-end frameworks, with Vue and Angular following that (in that order of desirability) (8)

I have used both React and Vue in the past, and I preferred React. This is for two reasons:

* The syntax and component creation is easier for me to work with
* State is more obvious and simpler to interact with
* There is a very healthy environment of libraries and documentation to work with React.

Node is the backend JavaScript runtime environment for each of these, and is used to run ExpressJS. Express is used to handle incoming requests from the client. This includes GET requests for pages but also GET, POST and PUT requests for performing CRUD operations (Create, Read, Update, Delete).

## L[EA]MP Stack

LEMP and LAMP are PHP stacks. These are often used to serve static applications and might make use of some JS frameworks to manipulate the DOM to perform some reactive tasks. These use Linux, MariaDB/MySQL and PHP alongside either (E)nginx or Apache to be able to serve up content. This PHP can be in the form of a framework such as Laravel, or a standard PHP application with no specific framework. Linux is the operating system of the web server, which then runs both a database server and a web server. The PHP is then run on the web server.

This is a really effective stack for producing web pages that can be generated on the server (using the PHP code), with little responsive behaviour on the client. This is because the server can do all of the lifting for the client and send just the information that the page needs to load (usually just HTML and CSS, with some small amount of JS). This is not ideal for this project, as much of it will be heavily interactive.

## Other stacks of Note

There are a number of full stack frameworks such as Next and Nuxt which make use of Node and React to handle requests on the server and create reactive applications on the front end. These frameworks also tend to have some method internally to handle data requests, but leaving that actual implementation for getting that data to the developer. For example, you can use Next with Prisma and GraphQL to create a full stack JS application without having to directly interact with Node or Express, which can become quite a handful as the application grows. There are plenty of video tutorials that explain using these kinds of stacks, and may be something that I use to implement my solution. An example of such can be found here -> https://www.youtube.com/watch?v=4cpqSOQKSo8&list=PLn2e1F9Rfr6k6MwzS-p9FGK1NDBxxwLPk&index=4.

# Project Objectives

## MVP

The first thing that I need to achieve for this project is the design and implementation of a suitable data structure for storing information about users (Admin, Teachers and Students), courses, classes and modules. This is integral to the project, as everything else hinges on having access to this content. I will use a variety of tools to design the data structure, though the implementation of that will be dependent on whichever tech stack I choose.

The next thing to implement is an API to allow CRUD actions to be performed to make use of the data structure. I have the choice of making an REST or GraphQL API for handling this. REST is simpler to set up for direct interactions with some data source, but not so much for heavily related content where you might need (or want) to dig deep into the underlying relations. For this reason, I will likely create a GraphQL API to interact with the data, despite the increased difficulty to set up and implement.

Next, I need to be able to actually interact with this API and data structure in some way other than direct code interactions or API calls. To do so, I will create some interface for an Admin user. They will be able to do everything that a Student and Teacher can do, as well as being able to add/remove users. They will be able to perform interactions on all the data that is in the database that is user-generated.

I next need to implement the teacher's interface. This will allow them to perform CRUD operations on all their own content, as well as being able to manage their classes. They will also be able to enrol and remove students on their classes.

Finally for the MVP, I need to implement a UI to allow student0073 to view content that they have access to (classes for which they are enrolled). They will only (at this point) be able to view the data, though adding options to favourite and pin content is something that would be a nice addition to include.

## Extra

The first of the extra objectives is to add assignments, that would ideally be able to accept submissions from students, with teachers being able to respond to those submissions. This is quite a lot of extra work, and would be broken down into chunks that can be implemented separately. First, Teachers would need to be able to perform CRUD operations on the assignments, then students would need to be able to view that content. Next, student's need to be able to submit their work, and teachers to be able to view and download those submissions. Finally, teachers need to be able to provide feedback and notes on those assignment submissions.

A second feature that I would like to implement after the MVP is for teachers and students to be able to communicate through the application. This would enable Q&A forums and direct communication between Teachers and their Students. This is something that I foresee being the most difficult to implement, so will be the last thing that I attempt to add to the application.

# References

1. **SkillShare.** Earn From Your Teaching. *SkillShare.* [Online] 9 October 2023. [Cited: 9 October 2023.] https://help.skillshare.com/hc/en-us/articles/4415798406285-Earn-From-Your-Teaching.

2. **Jimmy Tries World, YouTube.** I tried THREE WEEKS of Skillshare Classes: Skillshare Review. [Online] 18 September 2020. [Cited: 9 October 2023.] https://www.youtube.com/watch?v=uPapW5o8nvk.

3. **Heilo, YouTube.** How To Upload A Class On Skillshare As A Teacher (2023 - Full Tutorial). [Online] 13 March 2023. [Cited: 9 October 2023.] https://www.youtube.com/watch?v=a2ja4EfzCQ4.

4. **LearnDash (Justin Ferriman).** LeardDash. [Online] 28 May 2019. [Cited: 9 October 2023.] https://www.learndash.com/best-wordpress-lms-plugin/.

5. **LearnDash.** LearnDash 3.0 - The New Standard in Online Courses. [Online] 28 May 2019. [Cited: 9 October 2023.] https://www.youtube.com/watch?v=9zkZ0qZnGR0&t=8s.

6. **Moodle.** MoodleDocs. [Online] 2023. [Cited: 9 October 2023.] https://docs.moodle.org/403/en/.

7. **ThemeSelection.** Best Web Development Stack 💥2023. [Online] 23 August 2023. [Cited: 9 October 2023.] https://dev.to/theme\_selection/best-web-development-stack-2jpe.

8. **Stack Overflow.** Stack Overflow Developer Survey 2023. [Online] 2023. [Cited: 9 October 2023.] https://survey.stackoverflow.co/2023/#section-admired-and-desired-web-frameworks-and-technologies.