University of Essex

School of Computer Science and Electrical Engineering

Capstone Project Dissertation

GoLearn — Learning Resource Management System

Bradley Beasley

Supervisor: **Dr Alexandoros Voudoris**

April 15, 2024
Colchester

Abstract

There are plenty of services and programs out there for allowing Teachers to upload resources and for their learners to access them. Most of these are not purpose designed for places of learning, such as school, colleges and universities. Those that exist for this specific use-case tend to be over-engineered and can be overly complicated (Moodle is a good example of this).
The purpose of GoLearn is to provide a simple LRM experience for teachers and students to interact with the same resources without a large amount of overhead in setting up the system: it should work out of the box.
This report contains a look into methods of implementation for web applications in general, plus the choices in technology and design that were made for GoLearn.

Contents

CHAPTER 1

Introduction

Chapter 2

Background

2.1 Similar Services

There are various LRMS's available in the market, with some of the more popular one being:

Udemy is a free to access learning site that puts a paywall over all of 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.

Skillshare is a similar service to Udemy, but is subscription based rather than pay per course. This means that learners can access all of the courses on the site for a monthly fee. Creators can create courses and upload them to the site, with the same access restrictions as Udemy.

LearnDash[1] is a WordPress plugin that allows you to create and sell courses on your WordPress site. It is a very popular plugin, with over 50,000 active installs and a 4.5-star rating on the WordPress plugin repository. It is a very powerful plugin, with a lot of features, but it is also very complex and can be difficult to use.
Being a plugin for WP, it is very easy to customise how the plugin looks by modifying the theme on the site to best suit the business / individual's needs. The plugin allows teachers to create courses and students to access them, with a lot of customisation options for the courses.

Moodle[2] is the closest comparison that is currently available to GoLearn. It is a PHP based LMS that is very powerful and feature-rich. It allows learning institutions to design their own courses/modules and add specific 'student' users to them. This is a very similar model to what GoLearn is aiming to achieve. The main difference is that Moodle is very complex and can be difficult to use. It is also very resource-intensive, requiring a lot of server resources to run effectively.

2.2 Tech Stacks

A ‘Tech Stack’ is the term used to describe the combination of programming languages, frameworks, libraries, and tools that are used to build a software application. The choice of a tech stack is crucial as it can affect the performance, scalability, and maintainability of the application. The following are some of the popular tech stacks used in web development:

LAMP/LEMP Stacks This stack consists of 4 major components: Linux, Apache/Nginx, MySQL and PHP. Linux is the OS that is used in the stack, with Apache or (E)nginx as the web server, MySQL as the database and PHP as the server-side language. This stack is very popular and is used by many web developers. It is used by various Frameworks (such as Laravel), as well as in vanilla PHP applications (no framework used).

PHP is the programming language of choice for this kind of stack for websites and applications that need to render the pages dynamically, but don’t need a lot of reactivity to user input on the page. For for example, a blog or a news site would be a good use case for this stack, as they largely need to retrieve data from a db when a page is loaded, and can use caching strategies easily to only make a DB request for cached pages after a certain period after the cached content was stored.

On the other hand, applications that have a lot of user interaction to perform CRUD operations would be harder to write within this stack, as JS would have to be loaded and ran to submit the changed data. Other stacks (as mentioned below) instead rely less on page loads to perform actions, and instead are designed with responsive and reactive implementations in mind.

MEAN/MERN/MEVN Stacks The ‘M’ refers to the database implementation used (such as mySql, MongoDB). . . , the ‘E’ refers to ExpressJS, which is a NodeJS framework used for all forms of JS applications, and is the layer between the templating framework and NodeJS to serve an application. The ‘V’, ‘A’ and ‘R’ refer to the templating framework used to create the layouts used, and reference ‘VueJS’, ‘AngularJS’ and ‘ReactJS’ respectively. The ‘N’ refers to ‘NodeJS’, which is a server-side JS runtime environment that is used to run JS code on the server.

This stack is very popular for creating SPAs (Single Page Applications) and PWAs (Progressive Web Applications). These are applications that are designed to be very reactive to user input, and can update the page without needing to reload the page. This is done by using JS to update the page, rather than relying on the server to render the page and send it to the client. This is achieved by requesting information from the server about what to load and how to load it, and then using JS to render the page based on the information received. This is a very powerful stack, but can be difficult to learn and use effectively.

Full Stack Frameworks A full stack framework is a framework that is designed to be used for both the front-end and back-end of an application. This means that the framework can be used to create the server-side code, as well as the client-side code. This is very useful as it means that development teams can rely on a single codebase for the entire application, rather than having to use multiple codebases. This can make development faster and easier. As well as this, it allows type safety and code completion to be used across the entire application easier than in separate front- and back-end applications.

Some examples of full stack frameworks are NextJS and NuxtJS. These frameworks are designed to be used for both the front-end and back-end of an application. They are very powerful and can be used to create very complex applications. They each use a different templating framework (NextJS uses ReactJS, and NuxtJS uses VueJS). The benefit of these frameworks is that they allow for server-side rendering as well as client-side rendering, which can be very useful for SEO and performance reasons.

One drawback of these frameworks is that there can be a steep learning curve to get started with them. As well as this, the codebase can get very complex and the boundaries between client and server code can get blurred, which can make it difficult to debug and maintain the codebase if it is not well organised.

2.3 Databases and ORMs

A database is a collection of data that is stored in some structured format. Databases are used to store data that can be accessed and manipulated by software applications. There are many different types of databases, each with its own strengths and weaknesses, but all fall into one of two categories: relational or non-relational.

Relational Databases store data in tables, with each table containing rows and columns. The columns represent the attributes of the data, while the rows represent individual records. Relational databases use SQL (Structured Query Language) to query and manipulate the data. Some popular relational databases include MySQL, PostgreSQL, and SQLite.

This category of database is called relational because it allows for relationships to be defined between the columns in different tables. For example, a database for a school might have a 'students' table and a 'courses' table. The 'students' table might have a column that references the 'courses' table, which would allow for a relationship to be defined between the two tables. This is called a foreign key, and can be used to enforce that a student can only be enrolled in a course that exists in the 'courses' table.

Non-Relational Databases store data in a more flexible format, such as key-value pairs, documents, or graphs. Non-relational databases are often used when the data is unstructured or when the data model is likely to change frequently. Some popular non-relational databases include MongoDB, Cassandra, and Redis.

This category of database is called non-relational because it does not enforce relationships between tables. This can make it easier to work with the data, as the data can be stored in a more flexible format.

An ORM is a layer between the application and the database that allows the application to interact with the database using an object-oriented interface. These are used to simplify the process of interacting with the database, and to make it easier to work with the data in the database. They supply an API for developers to be able to perform CRUD operations on the database without having to construct many (if any) queries or statements to do so.

Some examples of ORMs include Prisma, TypeORM, and Sequelize. These ORMs are designed to be used with different databases, and have different features and capabilities. For example, Prisma is designed to be used with PostgreSQL, MySQL, and SQLite, and has a strong focus on type safety and code generation. TypeORM is designed to be used with TypeScript, and has a strong focus on type safety and code generation. Sequelize is designed to be used with MySQL, PostgreSQL, SQLite, and MSSQL, and has a strong focus on performance and scalability.[3]

CHAPTER 3

Technical Specification

3.1 Framework

Of the stacks mentioned in Section 2.2, the NextJS framework was chosen for the GoLearn project. This was chosen for the following reasons:

ReactJS[4] ReactJS is a very powerful and popular front-end framework that is used to create SPAs and PWAs. It is very flexible and can be used to create very complex applications. It is also very easy to learn and use, which makes it a good choice for the GoLearn project.

As well as this, it was a framework that the author had used before, meaning that he could get started with the project quickly and easily and wouldn't have to spend a lot of time learning to use a new framework or architect his own solution from scratch. This proved to save a lot of time in the development process, as the author was able to get started on the project and make progress quickly.

Server Actions[5] As of NextJS 14, the framework included a feature called 'Server Actions'. This allowed functions to be shared between server- and client-side code. Whenever a server action was called, a request is sent to the server as a Fetch/XHR request from the client, the server would handle the request and return the result to the client. This would prove to be a very useful feature, as it allows for CRUD operations to be ran

3.2 Database and ORM

The ORM chosen for the GoLearn project was Prisma. This was chosen for the following reasons:

Type Safety[6] Prisma is designed to be used with TypeScript, and has a strong focus on type safety and code generation. This means that the author could be confident that the code he was writing was correct and that the database schema was being enforced correctly. This was very important for the GoLearn project, as it was a project that was being developed by a single developer, and there was no QA team to review the code.

Code Generation[7] Prisma uses code generation to generate the database schema and the client-side code that is used to interact with the database. This means that the author could be confident that the code he was writing was correct and that the database schema was being enforced correctly. This was very important for the GoLearn project, as it was a project that was being developed by a single developer, and there was no QA team to review the code.

Prisma can be used with many different databases, including PostgreSQL, MySQL, and SQLite. One of the factors that was considered was the ease of creating and maintaining the database. Originally the author had considered using a service called 'Supabase', which is a service that provides a PostgreSQL database, and is a service that Prisma has a lot of support for.

Later in the project when the application hosting was being considered, it was decided that the author would the application on Vercel, which is a service that provides hosting for NextJS applications. Vercel has a feature called 'Vercel for Databases', which allows you to create a PostgreSQL database that is hosted on Vercel. It was a relatively simple process to migrate the database from Supabase to Vercel, and the author was able to do this without too much difficulty. The Vercel database is also a PostgreSQL database, which meant that the author could continue to use Prisma with the database without any issues.

CHAPTER 4

Implementation

4.1 Authentication

4.2 User Management

4.3 CRUD Operations

4.4 Data Loading

4.5 Known Issues

CHAPTER 5

Testing

5.1 Unit Testing

5.2 User Testing

CHAPTER 6

Project Planning

CHAPTER 7

Conclusions

Bibliography

[1] J. Ferrimanm, “Learndash 3.0.” https://www.learndash.com/best-wordpress-lms-plugin/, 5 2019.

[2] Moodle, “Moodle.” https://moodle.org/, 1 2024.

[3] M. Wanyoike, “9 best javascript and typescript orms for 2024.” https://www.sitepoint.com/javascript-typescript-orms/, 3 2023.

[4] D. Abramov and R. Nabors, “Introducing react.dev.” https://react.dev/blog/2023/03/16/introducing-react-dev, 4 2024.

[5] Next.js, “Server actions and mutations.” https://nextjs.org/docs/app/building-your-application/data-fetching/server-actions-and-mutations, 4 2024.

[6] Prisma, “Type safety.” https://www.prisma.io/docs/orm/prisma-client/type-safety, 4 2024.

[7] Prisma, “Generating prisma client.” https://www.prisma.io/docs/orm/prisma-client/setup-and-configuration/generating-prisma-client, 4 2024.