
Learn on Your Own

Gary Mc Hugh

Sean Mc Grath

B.Sc.(Hons) in Software Development

APRIL 16, 2017

Final Year Project

Advised by: Gerard Harrison

Department of Computer Science and Applied Physics
Galway-Mayo Institute of Technology (GMIT)



Contents

1	Introduction	7
1.1	Chapter Summaries	8
1.2	GitHub Links	10
2	Context	11
2.1	Technologies	11
2.1.1	Document Database	11
2.1.2	Java Framework	12
2.1.3	Web application framework	12
2.2	Research of key technologies	13
2.3	Features	13
3	Methodology	15
3.1	Development approach	15
3.1.1	Scrum	15
3.1.2	Extreme Programming	16
3.1.3	Kanban	16
3.1.4	Rapid Application Development	16
3.2	Dummy Database	17
3.3	Architecture Choice	18
3.3.1	MongoDB	18
3.3.2	Spring Boot	18
3.3.3	AngularJS	19
3.3.4	Technology popularity	19
3.4	Meetings	21
3.5	Source Control	21
3.5.1	GitHub	21
4	Technology Review	23
4.1	Back-End	23
4.1.1	MongoDB	23

<i>CONTENTS</i>	3
-----------------	---

4.2 Middleware	24
4.2.1 Spring Framework	24
4.3 Front-End	25
4.3.1 HTML5/CSS	25
4.3.2 JavaScript	26
4.3.3 Bootstrap	26
4.3.4 AngularJS	27
4.4 JavaScript Tools	28
4.4.1 Bower	28
4.4.2 NPM	28
4.5 Test Server	29
4.5.1 Apache Tomcat	29
4.6 Hosting	29
4.6.1 Amazon Web Service	29
4.7 Slack	30
4.8 GitHub	31
4.9 Technology Alternatives	33
4.9.1 MySQL	33
4.9.2 Ember	33
4.9.3 Guice	34
4.9.4 Heroku	35
5 System Design	36
5.1 Database	37
5.1.1 Purpose	37
5.1.2 Functions	37
5.1.3 Database Collections	40
5.2 Web Application Functionality	43
5.2.1 Authentication	43
5.2.2 Home Page	44
5.2.3 Courses Page	46
5.2.4 Individual Courses	48
5.2.5 Lecturer Page	50
5.3 Mobile compatibility	51
5.4 Code Samples	56
6 System Evaluation	61
6.1 Chosen Technologies	61
6.1.1 MongoDB Evaluation	61
6.1.2 Spring Framework Evaluation	62
6.1.3 AngularJS Evaluation	62

<i>CONTENTS</i>	4
6.2 Robustness	62
6.3 Performance	63
6.4 Limitations	63
7 Conclusion	64

About this project

Abstract The purpose of this project was to create a web application containing video tutorials on various programming languages. This web application was designed to help people like students who are new to programming. We wanted to develop a platform where students who are struggling with programming can come and watch tutorials on a variety of subjects to help improve their skills. The users can also ask questions to the lecturers on the course page in a comment section.

The need for an application like this comes from our experience throughout our time in college. We noticed that almost all websites online that provide programming tutorials have a cost attached to watch the tutorials. As students, we feel there is a lack of free resources to help people who are new to programming. We aim to solve this problem by creating a free to use video tutorials web application.

Our system will allow Users to register and log in to the application, from there they will see a variety of courses in various programming languages such as C#, Java and Python. The user will be able to go to that course and watch a set of tutorials on that subject. They can leave comments on the course page to ask questions to the lecturer or give their feedback.

This project involved research, learning and understanding many new frameworks such as Spring, MongoDB and AngularJS. We also had to understand the architecture of making an application with these technologies.

We plan to create an intuitive, easy to use and thoughtfully designed web application to achieve this.

Authors This project was created by two fourth year software development students in GMIT: Gary Mc Hugh & Sean Mc Grath, as part of our BSc in Computing in Software Development honours degree.

Sean was responsible for the user interface which was developed using AngularJS and handling data to and from the front-end.

Gary was responsible for the creation of the MongoDB database and Java code to store and retrieve the data to and from the database, as well as exposing the Java code as an API endpoint to be used by AngularJS.

Our shared time was spent designing the user interface for the web application and defining the functionality of our application.

Acknowledgements We would like to thank our supervisor Gerard Harrison for taking time out of his schedule to meet us each week. His advice and guidance throughout the development of this project was essential to the completion of this project before the set deadline.

Chapter 1

Introduction

We decided to create a free to use online learning platform containing programming tutorials on various popular subjects. We done this in the form of a web application, which is designed to be used with any web browser. The user interacts with the intuitive user interface, the user's requests are handled by AngularJS which uses the functions created in the Java code. All the information going to the front-end comes from the Mongo Database.

The reason for creating this application was due to the lack of free online learning platforms available to students. Throughout our time in college we noticed that most online learning platforms like Udemy and Lynda make Users pay to watch the tutorials. Our web application aims to rectify this by providing a free, easy to use learning platform that contains useful programming tutorials on a variety of subjects.

The goal of our project is to provide a free service that students can go to, watch some useful tutorials and ask any questions they may have based on the tutorials or based on what they are struggling with.

To develop this system, we had to connect various platforms together and have them pass data between eachother. We had to connect the Spring Framework (which uses Java) to the Mongo Database, we also had to connect AngularJS to Java code to pass the data between them. We done this by creating API end points using Spring that the AngularJS used to get access to the data. Spring was connected to the Mongo Database using the Mongo libraries that are available when using the Spring Framework.

The pages contained within the application include a registration page which allows a new user to enter their details to register with the application, a login page that allows a previously registered user to login. A courses

page which contains a graphical representation of all the available courses. A lecturer page which contains all the lecturers that are responsible for a course. The user can select a lecturer to view their profile or select a course to be brought to the tutorials.

The web application was created using AngularJS, HTML (Hyper Text Markup Language), CSS (cascading Style Sheets), JavaScript and Bootstrap. Bootstrap is a HTML, JavaScript and CSS framework that can be used to quickly create professional looking web applications that are compatible with mobile devices.

We set out to create an easy to use, intuitively designed and well thought out web application using modern technologies in a modern architecture.

We used an agile development methodology by setting out milestones for certain components to be finished by and incrementally building the application. We attended weekly meetings with our supervisor throughout the development of this project.

1.1 Chapter Summaries

Below is a summary of all the chapters in this report:

Introduction

In this section, we summarised the project idea and discussed the basic flow of data in the application. We then gave our reasoning for creating the application and outlined some of the features of the application. Finally, we discussed the technologies and methodologies that would be used to create the web application.

Context

Here we propose the aims and objectives that we have carefully set out for this project. We then review the technologies that we decided to use, comparing them to a valid alternative. We also discuss the research that was done for each technology highlighting the learning curve of each. We then finish by summarising the features of the application.

Methodology

The Methodology illustrates how we started out the project through our research and the forms of Agile that we used throughout the development of the application. We briefly discuss the technologies that were used to create the application and our approach to weekly meetings. Lastly, we consider the Source Control used and discuss the features we utilised within GitHub.

Technology Review

Here, we analyse all the technologies that were used while developing the application. We also explain our reasoning for using these technologies.

System Design

In this section, we provide a detailed overview of the application, discussing the various features of the application, including screenshots and code samples where relevant. We also show how we have configured the MongoDB database.

System Evaluation

The system evaluation contains a brief summary of how we chose our technologies followed by our evaluation of the main technologies that we used. This is followed by an evaluation of the robustness and performance of the technologies used as well as the application itself. Finally, we discuss the issues and limitations we faced while developing this application.

Conclusion

Here, we provide a broad overview of our development, thought processes, meetings, technologies used, issues encountered and our overview of the project as a whole. Finally, we will review our overall experience and discuss what we would have done differently.

1.2 GitHub Links

GitHub Repository

Below is a clickable link to our GitHub repository:

<https://github.com/seaniemc/LearnOnYourOwn>

GitHub Issues

Below is a clickable link to our GitHub Issues:

<https://github.com/seaniemc/LearnOnYourOwn/issues>

GitHub Kanban boards

Below is a clickable link to our GitHub Kanban boards:

<https://github.com/seaniemc/LearnOnYourOwn/issues#boards?repo=79832001>

GitHub ReadMe

Below is a clickable link to our GitHub ReadMe:

<https://github.com/seaniemc/LearnOnYourOwn/blob/master/README.md>

Project Report

Below is a clickable link to our Project Report:

<https://github.com/seaniemc/LearnOnYourOwn/blob/master/project.pdf>

Chapter 2

Context

The aim of this project was to create a free to use online learning platform for students that contains video tutorials on various programming languages.

Our Objectives were to:

- Improve our knowledge of the Java Programming language.
- Learn how document databases such as MongoDB work.
- Learn how to use the AngularJS Framework.
- Learn how to use the Spring Framework.
- Create intuitive user interfaces for the application.
- Create an easy to use and robust web application that accomplishes our aim.

2.1 Technologies

Below we will discuss the key technologies used in our project, our reasoning for choosing them and the research undertaken to understand these technologies.

2.1.1 Document Database

MongoDB was developed by MongoDB Inc. and published under the GNU Affero General Public License, its initial release was in 2009. [1] MongoDB stores data in flexible JSON like documents, meaning data can vary from document to document. MongoDB is the most popular document database available today, being used by the likes of Facebook, Buzzfeed, EA and eBay.

Having done some research on document databases such as CouchDB and Microsoft's DocumentDB and comparing them with MongoDB, we decided to use MongoDB. The primary reason for this was that we would not have changing data or be creating a mobile application to go with the web application, which would be a good reason to use CouchDB. [2] Its popularity also suggests that it is robust and has good reliability.

2.1.2 Java Framework

The Spring Framework was developed by Pivotal Software and was released under the Apache 2.0 license, its initial release was in 2002. Spring provides a configuration model for modern Java based enterprise applications. The Spring frameworks features include Dependency Injection, Data Access and RESTful web service frameworks.[3] The Spring Framework is the most popular application framework when using the Java programming language. Spring is used by some large companies such as Amazon, ESPN, and VISA. We decided to research and compare both Spring and Java EE [4] as they were the two application frameworks we had heard the most about. We quickly realised that Spring would abstract away a lot of the complex XML configuration from us if we used Spring Boot, leaving us to focus on the applications features. This was a major factor in our decision to use the Spring Framework.



2.1.3 Web application framework

AngularJS was developed by Google, its initial release was in 2010, making it a relatively new framework. AngularJS is a JavaScript based front-end web application framework. [5] It is open source and is one of the largest projects available on GitHub. AngularJS has been used as part of the MEAN (MongoDB, Express.js, AngularJS and Node.js) stack and in the Ionic Framework. It is one of the most popular JavaScript frameworks available today and as it was made by Google, it is probably not going anywhere any time soon. Having looked at its popularity and compared it to other technologies such as Ember.js, it was an easy to decision to use AngularJS for this project due to its large online community and online resources available.

2.2 Research of key technologies

As we were using this project as an opportunity to learn new and modern technologies, there was a steep learning curve when learning those new technologies. A lot of our initial time during this project was spent researching our chosen technologies and completing online tutorials to improve our skills in them. This helped us to get a more in-depth understanding of the technologies and explore the capabilities and limitations of each. We found MongoDB easy to use as there is no Primary Key or Foreign Key constraints and we could store different values in different fields. This meant that we could be dynamic in the way that we stored our data.



We found that the Spring Framework had the steepest learning curve due to its annotations and XML configurations. However, as our understanding of Spring increased, so did our appreciation for it. Spring abstracts a lot of the configuration away from you and has some great features such as Spring security, which provide you a fast and easy way to get your project up and running allowing you to focus on the features of your application.

We found that AngularJS also had a steep learning curve due to its syntax. Having completed tutorials and reading the comprehensive AngularJS documentation, [6] we improved our capabilities in using this technology. The large online community for AngularJS meant that there was plenty of information available to us, allowing us to develop our understanding of the framework and how we could use it in our project.

2.3 Features

The features of this application are:

- Courses page containing a list of all the available courses in the application.
- Individual course pages containing video tutorials, a comment section and the lecturer's details for that course.
- Lecturer page that displays the details of all of the lecturer's.
- Edit page for Courses, which lets you create, update and delete any courses details.

- Edit page for Lecturer that allows you to create, update and delete any lecturer's information.
- Sign up page which allows a user to sign up and stores the user's details in the database.
- Login page that allows the user to sign in using their details.
- The application also allows the user to sign in using their google account through the Google API.

Chapter 3

Methodology

After planning out our project extensively, we quickly realised that we had to split up the workload for the project. This was in order to complete the project before the given deadline. As Gary had completed his previous third year project using Java to get access to a large database, we decided that he would focus on this side of the development. Since Sean had experience with AngularJS, we decided that he would take on the front-end development handling the data from the API endpoints.

3.1 Development approach

In this project, we took an agile approach to our development. As is done in industry, we chose to use a couple of different flavours of agile throughout the development rather than strictly adhering to one methodology. This allowed us to create the optimum methodology that suited us and our project, as well as allowing us to experiment with using various methodologies.

3.1.1 Scrum

One of the methodologies that we selected to use was Scrum. We chose the Scrum methodology due to its high visibility to progress, meaning we could quickly see how far along we were in the project. At the end of each sprint we demonstrated our work to each other and evaluated it as part of our use of the Scrum methodology.

3.1.2 Extreme Programming

We also used a small amount of the Extreme Programming methodology. We done this through pair programming. We found this approach particularly effective as we could spot each other's mistakes and suggest improvements. It was also useful when diagnosing the cause of errors during the development of the application.

3.1.3 Kanban

We took some inspiration from the Kanban methodology when using GitHub for our source control. GitHub has a feature that lets you use Kanban boards to improve productivity.

Here is an example of what the Kanban board looks like [7]:

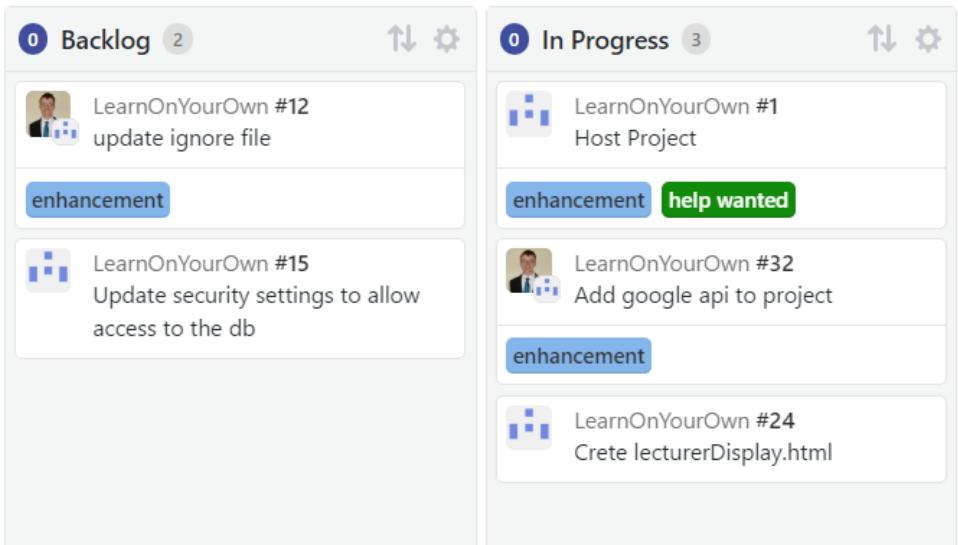


Figure 3.1: Kanban Board

3.1.4 Rapid Application Development

Finally, we used the Rapid Application Development (RAD) model in order to incrementally build the application. This allowed us to work in parallel during a sprint and assemble our work at the end of that sprint to make a working prototype. We found that regular meetings were essential to tweaking design and fixing bugs that we discovered along the way. The GitHub Issues feature helped us to keep track of what we had to work on, paired with the Kanban boards which increased our productivity.

3.2 Dummy Database

Since our web application relied on the Mongo Database containing data to test the code that performed the CRUD (Create, Read, Update, Delete) operations on the database, we needed to come up with a solution to temporarily perform these operations. The final dataset would not be added to the database for some time. The solution we came up with was to add a dummy dataset to the database that we could use to test the application.

Below is an example of some test data that was stored in the database:

```

1  //insert statement for MongoDB
2  db.employees.insert([
3  {
4      "EmpNo": "2",
5      "FirstName": "Brian",
6      "LastName": "Hall",
7      "Age": "27",
8      "Gender": "Male",
9      "Skill": "Javascript",
10     "Phone": "408-1298367",
11     "Email": "Brian.Hall@gmail.com",
12     "Salary": "60000"
13 },
14 {
15     "EmpNo": "3",
16     "FirstName": "Chris",
17     "LastName": "White",
18     "Age": "40",
19     "Gender": "Male",
20     "Skill": "Python",
21     "Phone": "408-4444567",
22     "Email": "Chris.White@gmail.com",
23     "Salary": "100000"
24 },
25

```

Figure 3.2: Insert Statement

This dataset allowed us to test our ability to add data to the database as well as return that data to the application via AngularJS. Once the implementation of the actual dataset was completed we could easily switch between this mock dataset and our actual dataset for the application.

3.3 Architecture Choice

We wanted to use this project as an opportunity to research and apply modern technologies that are commonly used in industry. We also wanted to use these technologies in an architecture that would be substantial for a Level 8 degree while creating a robust and well-designed system. We wanted to ensure that we used newer technologies that had the stability to be applied to our project. Our goal was to use technologies that were both new and that we believed would be used for years to come. This would aid us with our employability having used these technologies.

Here is a list of the main technologies that we used:

- MongoDB
- Spring Boot Framework
- AngularJS

3.3.1 MongoDB

MongoDB is an open source, cross platform, document orientated database program. It is a NoSQL database that uses a document storage and data format called BSON, which is a Binary representation of JSON like documents. MongoDB was developed in 2007 and made open source in 2009, [8] making it relatively new in comparison to MySQL and Neo4j.

3.3.2 Spring Boot

Spring Boot is a form of the Spring framework that reduces the configuration the developer has to do. This allows the developer to quickly create and deploy production grade, Spring based applications. It embeds Apache Tomcat within the application so there is no need to deploy WAR files. Spring Boot provided an excellent platform for us to get our project up and running quickly, allowing us to focus on the creation of a well-designed and robust web application.

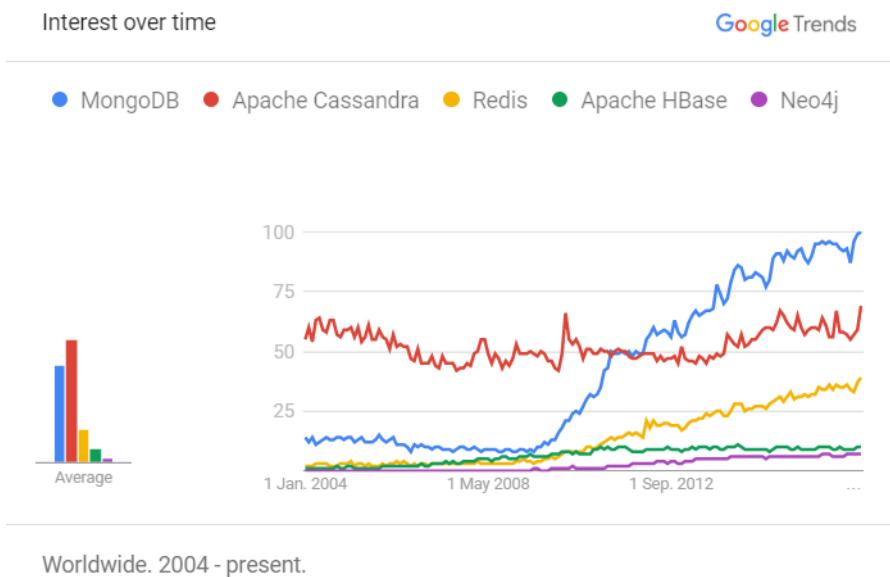
3.3.3 AngularJS

AngularJS is a JavaScript based, open source front-end web application framework that is owned by Google. AngularJS allows us to write client side applications in a clean MVC (Model, View, Controller) way. Overall, AngularJS is a framework to build large scale and high performance web application while keeping them as easy-to-maintain.

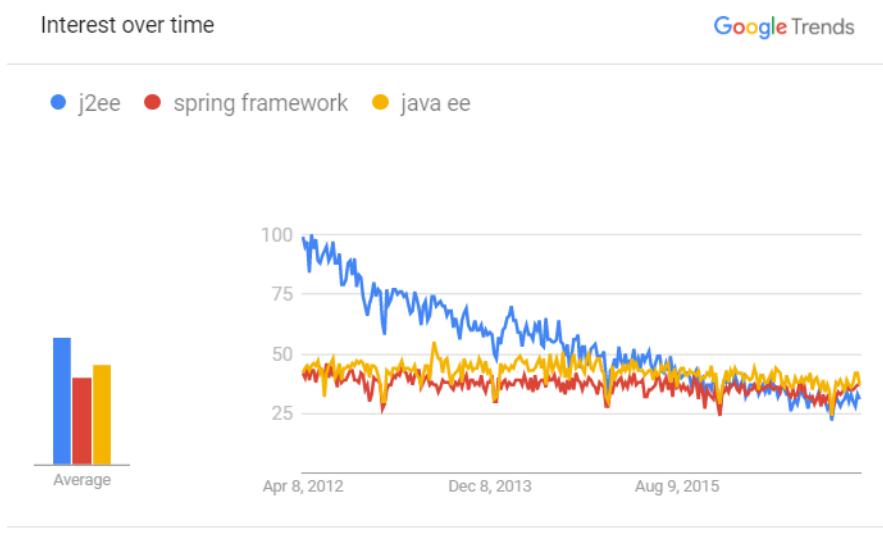
3.3.4 Technology popularity

Here are some graphs to show the above technologies popularity in comparison to their competitors:

- MongoDB [9]:

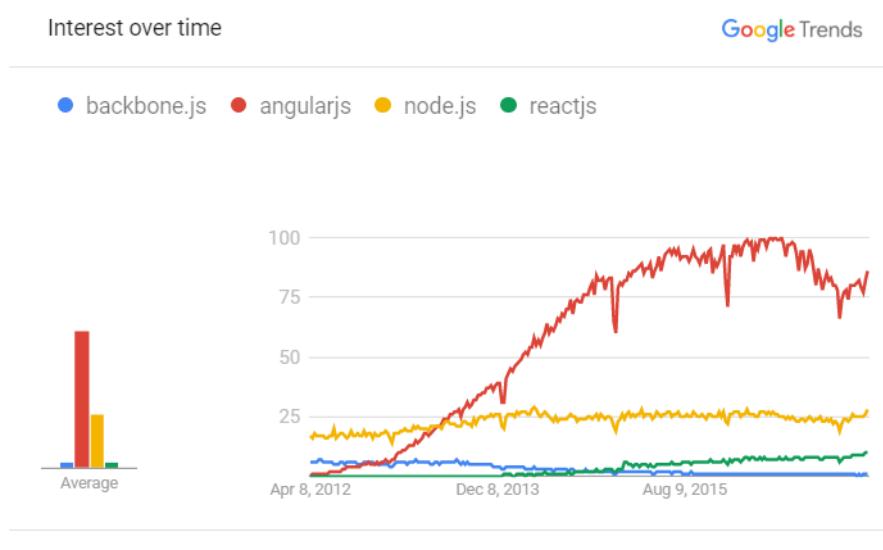


- Spring Framework [10]:



Worldwide. Past 5 years.

- AngularJS [11]:



Worldwide. Past 5 years.

We were not familiar with any of these technologies coming into this project, this meant that there was a big learning curve that tested us throughout the development of the project. We learned many things throughout this project including research and planning, documenting throughout development, using GitHub correctly and its features, meeting deadlines and the time it takes to learn and implement new technologies.

3.4 Meetings

Throughout the development of this project we attended weekly meetings with our supervisor. The purpose of these meetings was to ensure the development was going smoothly and that the software would be completed on time. We were also given key guidance in relation to the standards that we were expected to meet for this project and the key elements that the project should contain. As the development came to an end we met every two weeks due to our workload. We kept in constant communication between meetings by meeting in person and messaging each other on a mobile application called Slack.

3.5 Source Control

We utilised source control throughout the development of this project, this source control came in the form of GitHub. We used GitHub Issues, GitHub Kanban boards and various branches throughout our development.

GitHub is a web-based Git or version control repository and Internet hosting service. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project. [12]

3.5.1 GitHub

We chose to use GitHub as our source control for this project as we have used it throughout our time in college. We know that it is reliable and used worldwide. GitHub is used to host some of the largest projects, such as AngularJS [13] and Docker[14], this in our opinion proves that it is a reliable place to host our project. We were also drawn to use GitHub due to its issue tracker which is integrated with ZenHub's Kanban boards, GitHub's wiki's,

GitHub's branching feature as well as other features.

We used GitHub issues to keep track of any problems that we ran into during this project, this helped us to organise our work and keep track of what had to be done.[15] GitHub issues encourages you to break up a large problem into smaller and more solvable parts. This allowed us to step by step solve the more complicated problems in this project.

We used GitHub's Kanban boards which are integrated with GitHub Issues through ZenHub. Kanban boards helped us to organise our project into parts that have been completed, parts that we were currently working on and parts that would be done in future.[7] Kanban boards are proven to reduce idle time between iterations which increases productivity. We feel like GitHub's Kanban boards allowed us to maximize our productivity by allowing us to clearly see what outstanding issues we had.

Finally, we used Branches throughout our development process, we opened a new branch for each feature that we created. This allowed us to keep a working version of the project in master at all times. When a feature had been finished and it was working correctly, we would merge that branch into the master giving us a new version of the project. We found this branching strategy particularly effective as we could both work on separate features at the same time and add them both to master without breaking the build.

Chapter 4

Technology Review

In this chapter, we will discuss all the technologies that were used throughout the development of this project.

4.1 Back-End

4.1.1 MongoDB

We chose to use MongoDB to store the data from our application for this project. MongoDB is the leading NoSQL database management system. MongoDB is an open source, cross platform, document orientated database program. It uses Collections and Documents. A collection is a group of MongoDB documents. It is the equivalent of table in a relational database. Collections do not have a schema and all documents stored in each collection should be related. A Document is a set of key-value pairs.[16] Documents have a dynamic schema meaning documents in the same collection do not need to have the same set of fields. A document is the equivalent of a row in a relational database.



We chose to use MongoDB for the following reasons. The structure of each object in a document is very clear making it easy to see what data is stored. MongoDB is highly scalable, meaning it can handle a lot of concurrent users and further expansion of our application will be possible. It is schema less allowing us to have freedom in the way we store our data.[17] There are no complex joins like there are in MySQL, meaning we can query data from different collections easily. Finally, it has fast data access due to its use of internal memory, meaning we can quickly retrieve information for the user.

4.2 Middleware

4.2.1 Spring Framework

We decided to use the Java programming language as it has various frameworks that are available to use when creating a 3-tier web application. These frameworks include JEE, J2EE and Spring. We chose to use Java as we are very familiar with it from using it throughout our time in college.

We chose to use the Spring Framework to handle our database connections and operations. Spring is an open source Java platform that provides an infrastructure to develop robust Java applications easily and rapidly. The Spring Framework is the most popular application development framework for enterprise Java.[18] This was a big factor in our decision to use this framework, its extensive use means that it is tried and tested and maintaining its popularity due to its robustness. We chose to use a subset of Spring called Spring Boot.



Spring Boot is Spring's convention over configuration solution to creating standalone, production grade applications. It abstracts a lot of the Spring configuration from the developer allowing them to focus on features instead of writing lines of XML.

We chose to use Spring Boot for the following reasons:
Dependency management, versions of commonly used libraries are pre-selected and grouped in different starter POMs that you can include in your project. When you pick one, you're selecting dozens of dependencies that would otherwise have to be selected yourself individually.
Auto configuration, you do not have to manually configure dispatcher servlets, static resource mappings etc. They are all done for you.[19] Easy configuration, bean properties can be configured in the application properties file meaning you don't have to do it in Java or in the XML configuration file. Apache tomcat, tomcat is built into Spring meaning that you do not have to deploy any WAR files or run a separate server to see the application.



4.3 Front-End

4.3.1 HTML5/CSS

HTML5 (Hypertext Markup Language) combined with CSS (Cascading Style Sheets) was the obvious choice for the user interface of our web application. HTML is the standard markup language for creating web pages and applications, HTML along with CSS and JavaScript form the basis for almost all web pages on the internet. Having used HTML in previous projects we just had to brush up our knowledge by using online tutorials from w3schools.[20] This gave us the skills that we needed to add AngularJS on top of HTML.



CSS allows you to easily change the layout of the HTML from a single place. CSS is used to describe the presentation of a document written in markup language. It also allows you to separate the HTML document and the CSS code into individual files, making it easy to focus on a single item without getting lost. Overall, we found the HTML and CSS to be one of the more simpler parts of the project.

Below is an example of the form used to create comments in this project:

```
<!-- FORM TO CREATE COMMENTS -->
<div id="todo-form" class="row">
  <div class="col-sm-6 col-sm-offset-3 text-center" style="padding:0;">
    <!-- FORM TO WRITE COMMENT TO DATABASE-->
    <form name="todoForm" class="todoForm" ng-submit="createTodo()">
      <div class="form-group">
        <!-- BIND THIS VALUE TO formData.title IN ANGULAR -->
        <div class="input-group">
          <input type="text"
            class="form-control input-text"
            placeholder="I need to..."'
            ng-model="formData.title"
            autofocus required>
        <!-- createToDo() WILL CREATE NEW COMMENTS -->
        <div class="input-group-addon" style="background-color:#fff;">
          <button type="submit" class="create-todo">Add</button>
        </div>
      </div>
    </form>
  </div>
</div>
```

4.3.2 JavaScript

Having completed tutorials on HTML and CSS, we began to look into JavaScript. It is important when learning AngularJS to have a working knowledge of JavaScript as it makes it easier to write the AngularJS Controllers and Services. We completed some online tutorials using JavaScript in order to improve our knowledge of this language.[21] JavaScript proved to be tricky to learn in the beginning as it is untyped and doesn't show compiler errors, but became useful once we understood it. It has gained a lot of popularity in the past few years making it a good technology for us to learn. It is supported by all modern web browsers without needing any plugins, making it a good choice for web applications.



4.3.3 Bootstrap

Bootstrap is a popular free and open source front-end framework for designing web applications and mobile applications.[22] It was created to work with HTML, CSS and JavaScript. *Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.*[23] We decided to use Bootstrap as it vastly improves the look and feel of any HTML website with its professional looking HTML components (buttons, forms, navigation). The big advantage for us with using Bootstrap is that it makes your web application mobile friendly. This means that users will be able to view the application on their mobile device and interact with it easily. Bootstrap is also compatible with all modern browsers, meaning you will get a consistent look and feel no matter what web browser is being used.



Bootstrap

4.3.4 AngularJS

AngularJS is a very powerful JavaScript Framework that is owned by Google. Over the years there has been many different JavaScript Frameworks. However, none of them have ever seemed to stick. They tend to gain and lose popularity pretty quickly. AngularJS is one of the few JavaScript frameworks that have stuck, fighting off Facebooks React and Nodejs. Regular JavaScript has a distinct lack of structure leading to the creation of many frameworks. AngularJS is the solution to this problem. Its increasing popularity along with its easy to understand structure led us to use this technology.



ANGULARJS

We chose AngularJS due to its popularity, having completed some online code academy tutorials on AngularJS [24] we began to gain a comprehensive understanding of it. This included services, controllers, views and routing. AngularJS is used in Single Page applications using a Model View Controller (MVC) model.[25] Each view gets its own controller, this view is then dynamically injected into the primary index.html view. A service is also attached to each view. AngularJS makes use of Two-way binding allowing fast access to data.

Here is the logical structure of the AngularJS components:

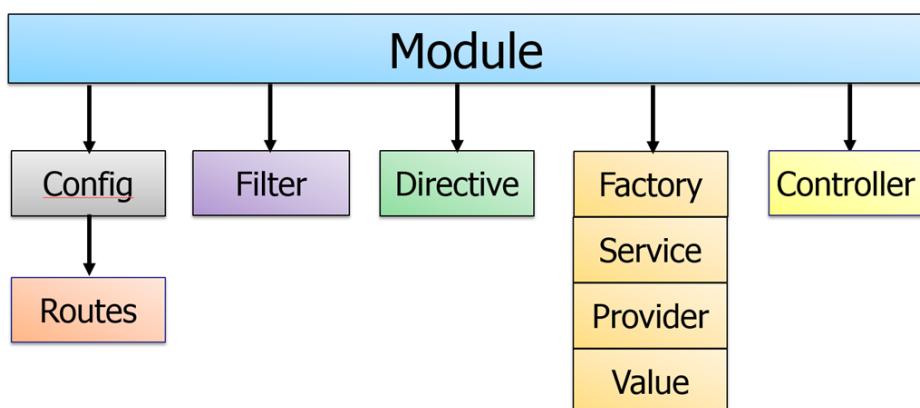


Figure 4.1: AngularJS Structure

4.4 JavaScript Tools

4.4.1 Bower

Bower [26] is an open source package management tool that is used for client side web development. Bower helps you install the correct versions of the packages you need. We found bower useful as it saved us time during the beginning of our project. We did not have to go look for installation links as Bower manages this for you. Bower loads items via dependency injection in the app.js file.

sample of bower.json from our project:

```
{
  "name": "todoapp",
  "dependencies": {
    "angular": "^1.5.2",
    "bootstrap-css-only": "^3.3.6"
  }
}
```



Figure 4.2: Bower Sample

4.4.2 NPM

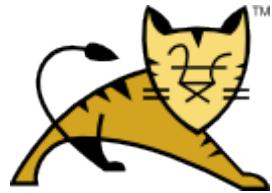
NPM [27] is similar to bower and is the main package manager for JavaScript Objects. It is included by default when using Node.js. NPM is often used with Bower in projects. We let Bower take the lead on installations and used NPM for small parts along the way.



4.5 Test Server

4.5.1 Apache Tomcat

Apache Tomcat [28] is an open source Java Servlet container, it provides a pure Java HTTP web server. Throughout the project, we used the Tomcat server to test the functionality of our application. Tomcat is built into the Spring Boot Framework however, when testing out potential features such as the google API login they were deployed on a standalone Tomcat server. This was to ensure that the API worked and was bug free before implementing it into our application. Using Tomcat as a test server proved effective, the newer versions of Tomcat (9 and above) abstract a lot of the initial configuration away from the user, which is a lot cleaner than previous versions that I have used.



4.6 Hosting

4.6.1 Amazon Web Service

We used Amazon Web Services [29] to host our project, having struggled with Heroku's complex way of getting a project hosted, we found Amazon Web Services more user friendly. Amazon Web Services offers you cloud based web hosting that is scalable and low cost. Although it does not offer Git tracking like Heroku does it is still a great way to host your applications for others to see. Amazon Web Services is used by many large companies such as Vodafone, Intel and Adobe [30], this proves its reliability and robustness. The scalability coupled with the simpler way of hosting the project compared with Heroku is the main reason we chose Amazon Web Service hosting over Heroku.



4.7 Slack

Slack [31] is a cloud based team collaboration tool, it contains chat rooms that allow teams to instant message and share files with eachother. The big advantage with Slack for us was that it could be integrated with GitHub, this meant that we would get notifications whenever a new commit was made or a new issue was created. This allowed us to solve issues as they happened and view the latest builds of our project. This increased our productivity as we could receive instant feedback from eachother, make suggestions on new builds and solve issues rapidly. Slack is a great tool for any group project using GitHub as it gives the team a way to communicate through instant messaging and constantly stay up to date with the GitHub repository and any issues that team members have run into. This is a tool that we will definitely be using in future projects.



Below is an example of some GitHub notifications we received using Slack:

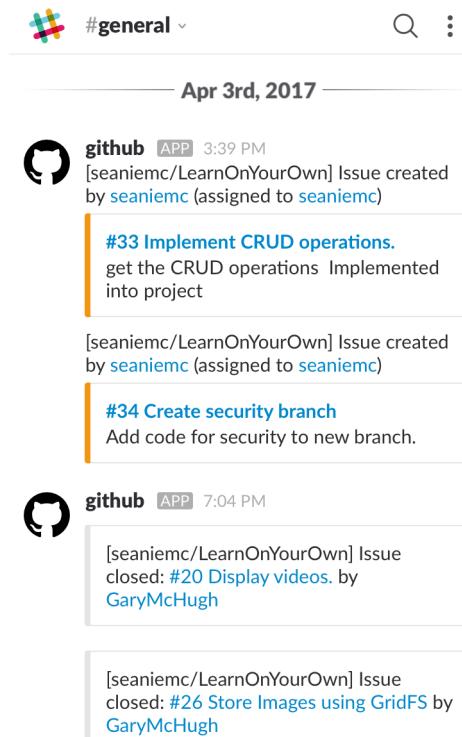


Figure 4.3: Slack Screenshot

4.8 GitHub

We used GitHub [32] throughout the development of our project as a way of using source control, keeping track of issues as well as creating different versions of the project. We found GitHub to be a very useful tool as it allowed us to create different branches for new features. This meant that we could keep the master branch as a working version of the project and create new branches every time we wanted to add a new feature. The working feature would then be merged into the master branch giving us a new version of the project. This allowed us to work on different features in parallel by using different branches.



We also made use of GitHub's issue tracker [15] to help plan out the next steps in the development and highlight any bugs/issues we found. GitHub issues allows you to break up large problems into smaller and more solvable parts.

<input type="checkbox"/>	! Create security branch	enhancement
	#34 opened 5 days ago by seaniemc	New Issues
<input type="checkbox"/>	! Implement CRUD operations.	enhancement
	#33 opened 5 days ago by seaniemc	New Issues
<input type="checkbox"/>	! Add google api to project	enhancement
	#32 opened 18 days ago by GaryMcHugh	In Progress
<input type="checkbox"/>	! Write to the db using google api	enhancement
	#29 opened 18 days ago by GaryMcHugh	Back-end / DB
<input type="checkbox"/>	! Display Image using GridFS	enhancement help wanted 8
	#27 opened 26 days ago by GaryMcHugh	Icebox

Figure 4.4: GitHub Issues

We also made use of GitHub's Kanban boards [7] which are integrated with GitHub Issues. Kanban boards are proven to reduce idle time between iterations which increases productivity. We feel like Kanban boards allowed us to maximize our time as we could quickly see what needed to be done. Kanban boards allow you to visually organise your issues into sections to step by step

build your application.

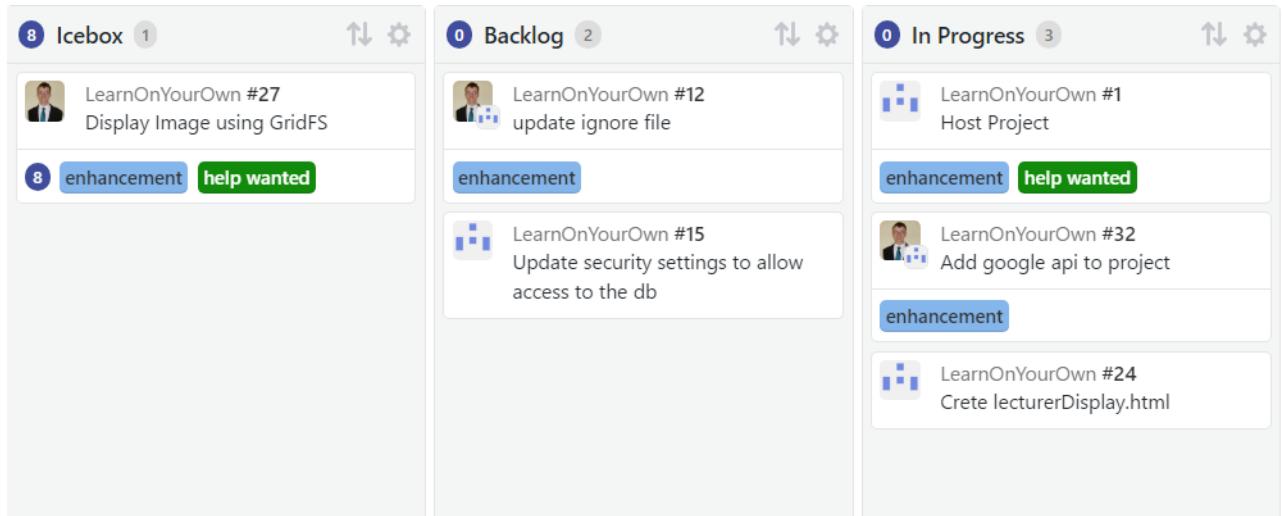


Figure 4.5: GitHub Kanban board

GitHub is used to host some of the largest projects on the web such as AngularJS and Docker. This proves that it is a reliable source control platform. It offers a range of extra features such as wiki for documentation, Issues and Kanban boards. GitHub is a great platform for software projects and creates an easy way for teams to collaborate on a project. GitHub is a vital tool that we would recommend others to use and will ourselves continue to use in the future.

4.9 Technology Alternatives

4.9.1 MySQL

MySQL is an open source relational database management system (rdms) which is the most popular database management system today. It uses a table format to store data, the data is stored in rows and columns.[33] The tables are linked through Primary and Foreign Keys. MySQL's reliability has been proven through the years with its extensive use, it offers complete ACID (Atomicity, Consistency, Isolation, Durability) transactional support. We were familiar with MySQL having completed modules in it during previous years. Gary had also used MySQL in his previous third year project. As we would be storing large files such as Videos we decided not to use this type of database. We also wanted to use this project as an opportunity to learn how a document database works, this would in turn increase our employability.



4.9.2 Ember

Ember.js [34] is an open source JavaScript framework based on the MVVM (Model, View, View Model) which allows users to create single web page applications. Ember.js would be a valid alternative to AngularJS for our project. It is used by large companies such as Microsoft, TED and LinkedIn.[35] This shows that it has proven reliability. We chose AngularJS over Ember due to AngularJS's increasing popularity over the past few years.



Below is an example of what some Ember.js code looks like:

```
App.ArtistsRoute = Ember.Route.extend({
  model: function() {
    var artistObjects = [];
    Ember.$.getJSON('http://localhost:9393/artists', function(artists) {
      artists.forEach(function(data) {
        artistObjects.pushObject(App.Artist.createRecord(data));
      });
    });
    return artistObjects;
  }
});
```

Figure 4.6: Ember.js code sample

4.9.3 Guice

Guice [36] is an open source software framework for the Java platform. It was released by Google in 2008 and has had newer releases since then, the latest coming in 2016. It provides dependency injection and annotations to configure Java objects just like Spring does. Guice works and looks a lot like the Spring framework. Guice has never gained the popularity that Spring has but remains as a good alternative to the Spring framework, although its documentation could use some work.

here is what some Guice code looks like:

```
@Singleton
public class EmailService implements MessageService {

  public boolean sendMessage(String msg, String recipient) {
    System.out.println("Email Message sent to "+recipient+" with message="+msg);
    return true;
  }
}
```

Figure 4.7: Guice code sample

4.9.4 Heroku

Heroku [37] is a cloud Platform as a service (paas) that can be used as a web application deployment tool. It supports many programming languages such as Java, Python, Scala and many more. Heroku lets you host five applications where you get one dyno for each application. Heroku is free when using the sandbox version, the biggest advantage of Heroku is that it can track and deploy your latest build of a project on GitHub automatically. Heroku is an excellent alternative to Amazon Web Services however, its major downfall is its complexity when trying to get your application on it and error free. Having tried to host our application multiple times on this platform we decided that we would opt to use Amazon Web Services instead.



Chapter 5

System Design

In this section we will discuss in detail the design of our systems architecture. Below is a diagram of the system design:

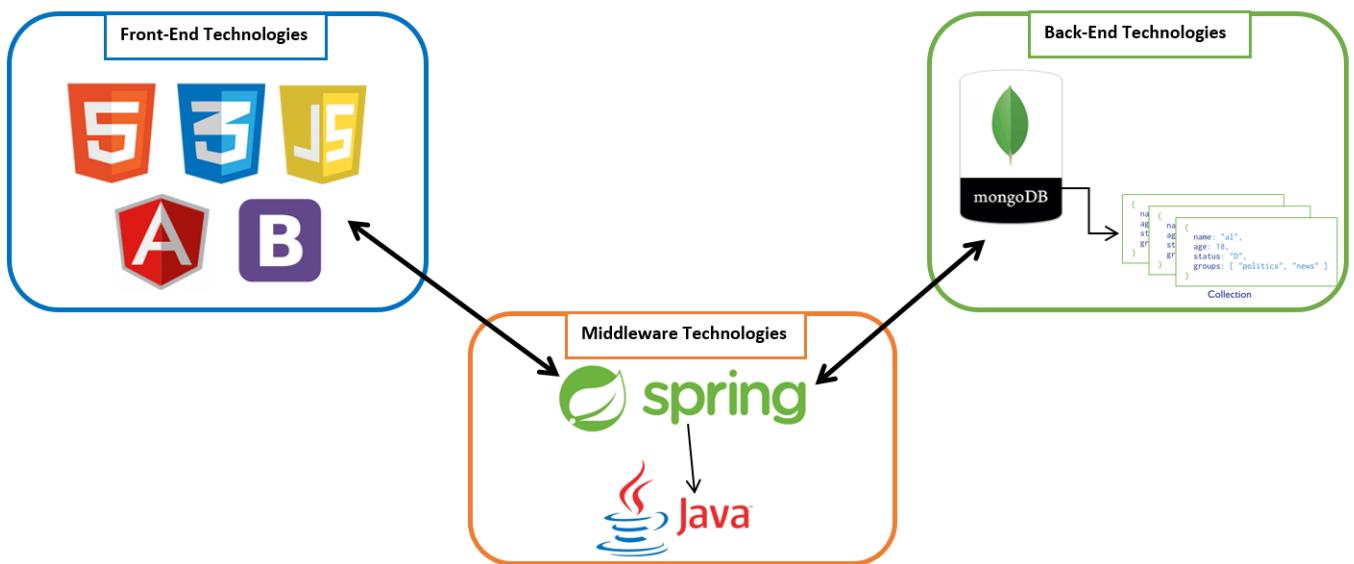


Figure 5.1: System Design

5.1 Database

5.1.1 Purpose

In this section, we will describe how the database supports the system with details about the documents and collections that are stored in the MongoDB. We also describe the functional and non-functional usage of the collections. The database is broken into different collections each collection holds data objects used to populate the view, or store data such as Course details, Lecturer details and User details for the system. In this document, we will also discuss how the database is integrated with the Web Application. Different entities can be represented in different collections as we will highlight later in the design document.

Below we will state the definitions which are used to describe the different stakeholders of the application and their purpose. These definitions will be used throughout the document.

- **Users:** A User is a person who will log in and use the services provided by the system.
- **Lecturer:** A Lecturer is a person who creates and provides the content for the user's. Lecturers have admin privileges, meaning they can Create, Update and Delete content.
- **Course:** this is the product which will be used by User's to improve their programming skills.
- **Todo:** Represents the comment section which allows User's to ask questions and give feedback to the Lecturer.

5.1.2 Functions

MongoDB provides many built-in functions which allow for very quick data retrieval times. Find by ID is one of the fastest ways of retrieving data from the database as the ObjectId that MongoDB provides is indexed. MongoDB uses Binary-Tree indexes. This means that searching for a particular value in a Binary-Tree has $O(\log n)$ time complexity on the average and in the worst case, which allows for very fast retrieval of data.

Here are a list of the functions/methods in our application and their purpose:

- **getAllCourses:** This procedure makes use of the in-built 'findAll' method of MongoRepository to return all of the data in the courses collection. It does not take any method parameters as it does not need any when returning all of the data.
- **createCourse:** This method is used to create a new course, it uses MongoRepository's 'save' method. This method is passed a Course object which contains values for all the fields in the course collection. The save method adds a new course to the collection using the course object.
- **getCourseById:** This method is used to return a single course to the user when they select a course from the courses page. This method uses the 'findOne' method from MongoRepository. It takes in a course id and returns the data for the course matching that id.
- **updateCourse:** This method uses a combination of MongoRepository's findOne and save methods. It takes a Course object and id as its parameters. It uses the findOne method with the course id to return the course you want to update. It then lets you change those fields on the front-end by overwriting the values it currently stores for that course. It does this by using the save method.
- **deleteCourse:** The delete course takes in a course id as its parameter. It then uses the delete method from MongoRepository to remove the details of that course from the database.
- **getAllLecturers:** This method is used to return all of the data in the lecturer's collection, it makes use of the in-built 'findAll' method of MongoRepository. This method does not take any method parameters as it simply returns all of the data stored in the lecturer's collection.
- **createLecturer:** This procedure creates a new lecturer in the database. It makes use of the 'save' method from MongoRepository. It takes a Lecturer object as its method parameter, this object contains values for all the fields that are in the lecturer's collection. A new lecturer is stored in the database by the save method using the lecturer object.
- **getLecturerById:** This procedure returns a single lecturer to the user, this occurs when the user selects a lecturer from the lecturer's page. The 'findOne' method is called to display the details of that lecturer in the lecturer's profile, It takes a lecturer id as its method parameter and returns the data for the lecturer that matches that id.

- **updateLecturer:** This method allows you to update the lecturer's details that the admin has selected. It uses both the findOne and save method of MongoRepository. It takes a Lecturer object and lecturer id as its parameters. The id is used with the findOne method to return the data for the lecturer that you want to update. You can then update this information on the front-end. The save method is then called with the new Lecturer object, this overwrites the data stored for this lecturer updating his/her details.
- **deleteLecturer:** The delete lecturer method makes use of the delete method of MongoRepository, it takes in a lecturer id as its parameter, it then deletes the details that are stored of that lecturer.
- **getAllUsers:** This method works in the same way as getAllLecturers but using the user collection.
- **createUser:** This procedure is called when a new user registers their details to the application. It works in the same was as createCourse does, using a User object.
- **getUserById:** This method works in the same way as the previous methods that retrieve data using the id field.
- **updateUser:** This methods functionality is the same as updateLecturer.
- **deleteUser:** This method is used to delete a user's details from the application. It works just like the deleteLecturer method.
- **getAllTodos:** A Todo relates to a comment in the courses page. This method works in the same way as getAllLecturers does.
- **createTodo:** This procedure is called when a user adds a comment to a course page. It works in the same was as createCourse does, only using a Todo object.
- **getTodoById:** This method works in the same way as the previous methods that retrieve data using the id field.
- **updateTodo:** This methods functionality is the same as updateLecturer method.
- **deleteTodo:** This method is used to delete a comment from the application. It works just like the deleteLecturer method.

5.1.3 Database Collections

Course's collection:

Field	Description
id	Course ObjectId given by Mongo
name	the name given to the course
lecturer	the name of the lecturer for the course
picture	thumbnail for the course
video	array of all the video tutorials for the course
description	a brief description of the course
enrolled	an array of all students taking the course

Table 5.1: Course's Collection.

Todo's collection:

Field	Description
id	Comment ObjectId given by Mongo
title	The content of the comment
createdAt	stores the date the comment was created

Table 5.2: Todo's Collection.

Lecturer's collection:

Field	Description
id	Lecturer ObjectId given by Mongo
name	the name of the lecturer
picture	a professional picture of the lecturer
courses	an array of all the courses they lecture
description	a brief description of the lecturer's achievements

Table 5.3: Lecturer's Collection.

User's collection:

Field	Description
id	User ObjectId given by Mongo
email	email address of the user
username	the username the of the user
password	user's password to login
role	describes the level of user

Table 5.4: User's Collection.

photo.files's collection:

Field	Description
id	File ObjectId given by Mongo
filename	the name of the file being stored
chunkSize	the size of the current file chunk stored as a long
uploadDate	exact date and time the file was stored on
length	the size of all the chunks combined
contentType	describes the type of content stored
md5	used for security of data

Table 5.5: photo.file's Collection.

photo.chunks's collection:

Field	Description
id	The ObjectId given by Mongo
n	the number relating to the chunk for ordering
data	contains all the file data in bytes

Table 5.6: photo.chunk's Collection.

5.2 Web Application Functionality

In this section, we will discuss the system's functionality and the amount of data it is storing. Without having to change the database's architecture this system can be extended to accommodate for more users and more courses. As shown above we are using separate collections for Lecturers, Users and courses. We are also using GridFS [38] to store images and videos in our database.

5.2.1 Authentication

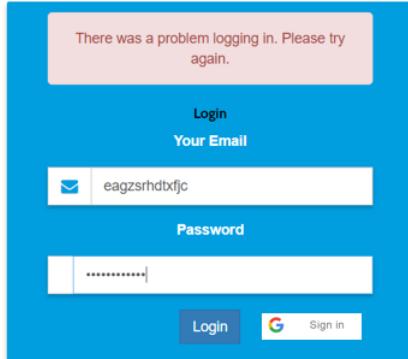
Sign Up Page

The Sign-Up page allows a new user to register themselves with our application. The user is asked to enter their details into the form, this form contains validation, meaning any incorrect details such as an invalid email address format will be recognised. The user's details are stored in the User collection within the database. The sign-up page is shown in Figure 5.2

Figure 5.2: Sign Up Page

Login Page

Once a user has registered with the application they will be able to login into the application using their created credentials. There is validation on this form so that if incorrect details are entered the user will not be able to login to the application. The application also features a Google login using a Google API, this allows the user to login using their Gmail account. This feature is shown in Figure 5.3.



The screenshot shows a login page with a teal header bar containing navigation links: 'Learn on Your Own', 'Courses', 'Lecturers', 'Contact', 'Sign Up', and 'Login'. The main area has a light blue background. At the top, there is a pink error message box with the text: 'There was a problem logging in. Please try again.' Below this, the word 'Login' is centered above two input fields. The first field is labeled 'Your Email' and contains the placeholder text 'eagzsrhdtxfjc'. The second field is labeled 'Password' and contains a series of dots representing the password. At the bottom of the form are two buttons: a dark blue 'Login' button and a white 'Sign in' button with a Google logo.

Figure 5.3: Login Page

5.2.2 Home Page

The home page should encapsulate the entire application allowing the user to access all of the applications functionality from one place. This means that it should contain ways of navigating to the common functionality of the application. The home page contains an overview of the entire application. The user is greeted with a nicely designed and professional looking home page that contains an image with the application and a little bit about the application beneath it. Which can be seen in Figure 5.4.



About Us

Welcome to "Learn on your own"! This is a repo for knowledge on coding in various languages and using various platforms. Please browse and feel free to send feedback using the Contact form.

Browse by course or lecturer. We have bios for all our well trained industry experts on the lecturers page.

[Contact Us >>](#)



Figure 5.4: Home Page

This is followed by a list of some of the lecturers for the application and the newest courses on the application. This allows the user to quickly navigate to the newest courses or to view a lecturer's profile. There is also a brief description of the lecturer and course under the images for each. This can be seen in Figure 5.5 below.



Figure 5.5: More of the Home Page

5.2.3 Courses Page

Course of the Month

At the top of the courses page is our selected course of the month, the course of the month is our way of promoting new technologies. We select this course based on the newest and most trending technologies that we think will be useful to know. Angular2 is Google's new release and upgrade of AngularJS, due to the recent success of AngularJS we decided that Angular2 would be a good technology for our course of the month. This month's course of the month is shown in Figure 5.6.

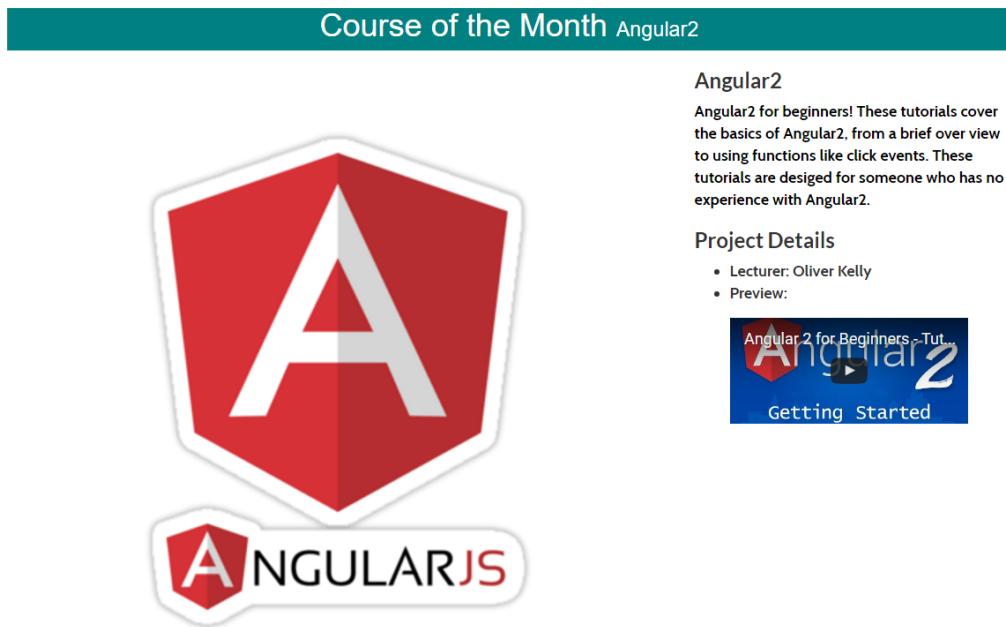


Figure 5.6: Course of the Month

List of Courses

The course page also contains a list of all courses that are available to the users. This provides an easy way for the users of the application to select the course which they need to improve their skills in. The administrators of the application can add courses to the application while it is online. This change will then be reflected to the end user. The list of all the available courses can be seen in Figure 5.7.



Figure 5.7: Available Courses

5.2.4 Individual Courses

The user has the ability to select each course individually, they will then be brought to this course page where the details of that course will be displayed. The details include the course name and picture, the lecturer giving the course, a comment section to ask questions and give feedback and finally the video tutorials for that course. The course page can be seen in Figure 5.8 and 5.9 below.

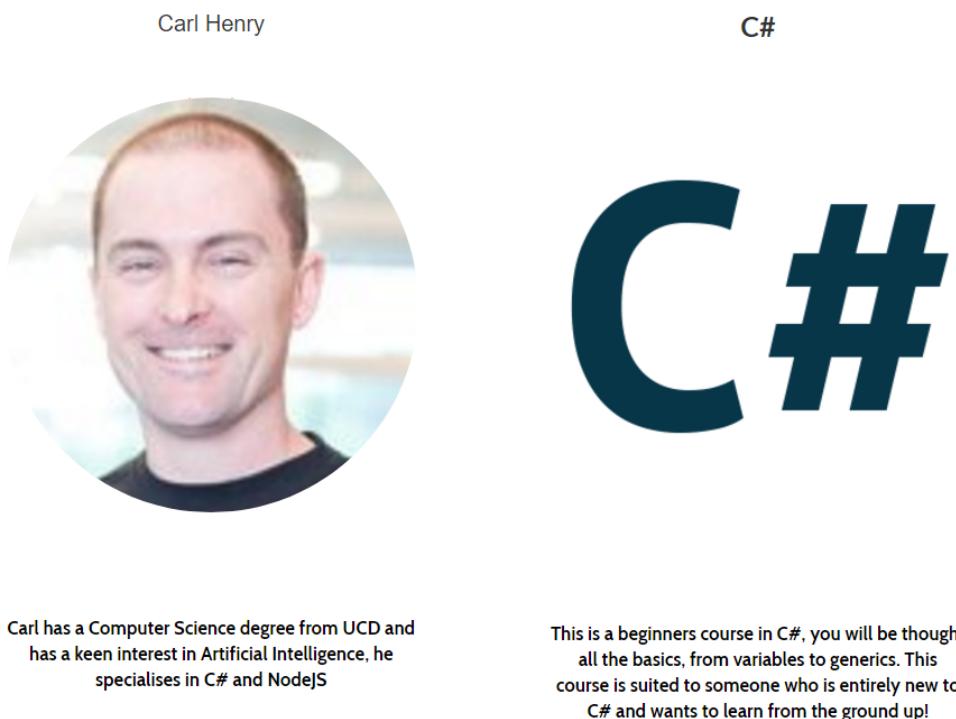


Figure 5.8: C# Course page

A screenshot of a course page for "C# BASICS". On the left, there is a comment section with a heading "Please leave a comment below" and a "Comments" button. Two comments are listed: one from a user asking about tutorial 4, and another asking about errors in the input textbox. On the right, there are two course thumbnails. The first thumbnail is titled "1. How to program in C# - BASICS - Beginner Tutorial" and shows the "C#" logo and a play button icon. The second thumbnail is titled "2. How to program in C# - Beginner Tutorial" and also shows the "C#" logo and a play button icon. Both thumbnails feature a purple hexagonal icon with a white "X" shape.

Figure 5.9: C# Course page continued

5.2.5 Lecturer Page

The lecturers page contains a nicely formatted list of all the lecturers that give courses on various subjects. This allows the users to easily see all the lecturers who give courses on the application. The user can select any lecturer and view their profile, their profile includes the course they lecture with a description of that course as well as the lecturers picture and a description of their academic achievements. The list of lecturers is shown in Figure 5.10.

Our Lecturers Our Lecturers are highly trained and skilled with excellent industry experience.

James Groden  <small>James has a masters in Neural Networks and he has a keen interest in Computer Science and specialises in Java and MySQL</small>	Carl Henry  <small>Carl has a Computer Science degree from UCD and has a keen interest in Artificial Intelligence, he specialises in C# and NodeJS</small>	Ciara Smith  <small>Ciara has a masters in Software Development and has worked in industry for IBM and Facebook, she specialises in JavaScript and MongoDB</small>	Oliver Kelly  <small>Oliver has a degree in Software Development and is our youngest Lecturer, he specialises in Angular2</small>
---	---	--	--

Figure 5.10: Lecturers Page

Admin Privileges

Admins have the ability to create, delete and update lecturer's information. This is done through Modal's in the application. The admin simply selects the button according to the action they want to perform, they will then be prompted to make the changes that they want. Those changes will then be added to the application for the Users to see. An example of the Create lecturer is shown below in Figure 5.11.

Create Lecturer

Name	<input type="text"/>
Picture Url	<input type="text"/>
Description	<input type="text"/>
Courses	<input type="text"/>

Save changes **Close**

Figure 5.11: Create Lecturer Modal

The update lecturer and delete lecturer Modals work in the same way as the create lecturer does, so I won't go into detail with them.

5.3 Mobile compatibility

Home Page

Bootstrap is compatible with mobile devices, meaning that mobile users can use any application that made using bootstrap. This was one of the main reasons that we chose to use bootstrap, it has made our application compatible with mobile devices. A mobile device has been simulated by reducing the dimensions of an internet browser. This shows what the application would look like on a mobile device. In Figure 5.12 below, you can see what the home page looks like on a mobile device. All the contents of the page have been resized to accommodate for the smaller screen size. This means that the end user gets the same experience whether they use a computer or a mobile device.

Below is the functionality of the Home Page on a mobile device:

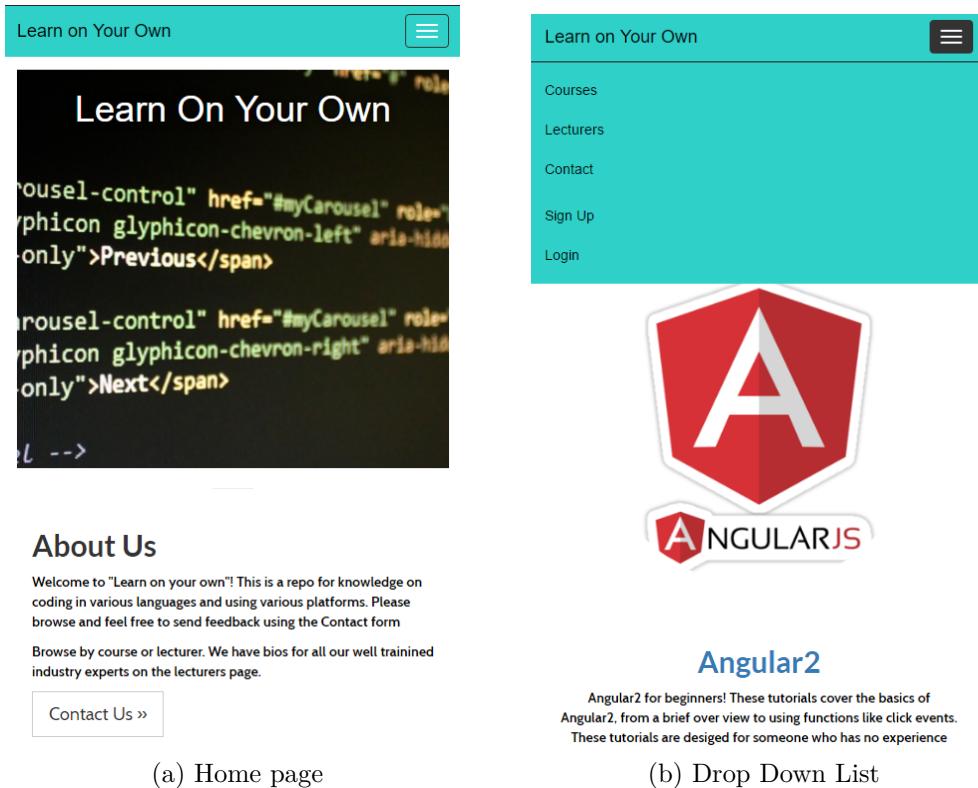


Figure 5.12: Home Screen

The navigation bar also changes to a drop-down list to allow the user to easily navigate through the application. The application keeps a consistent look and feel between the web application and mobile application, which is key to the user experience.

Courses Page

The Courses page also adapts to the mobile device while retaining the functionality of the application. The courses are displayed as they were in the web application but scaled down. The course of the month also scales appropriately.

Below are some samples of the courses page:



Figure 5.13: Courses Screen

Individual Course Page

Individual course pages also scale to mobile devices. The logo of the course scales down as well as the text. The list of videos that each course provides also scales down to the size of the window/screen. This allows the user to easily navigate their way through the course pages, they can also play the videos on the page and they will fit to their device.

Below in Figure 5.14 you can see the individual courses page has scaled.

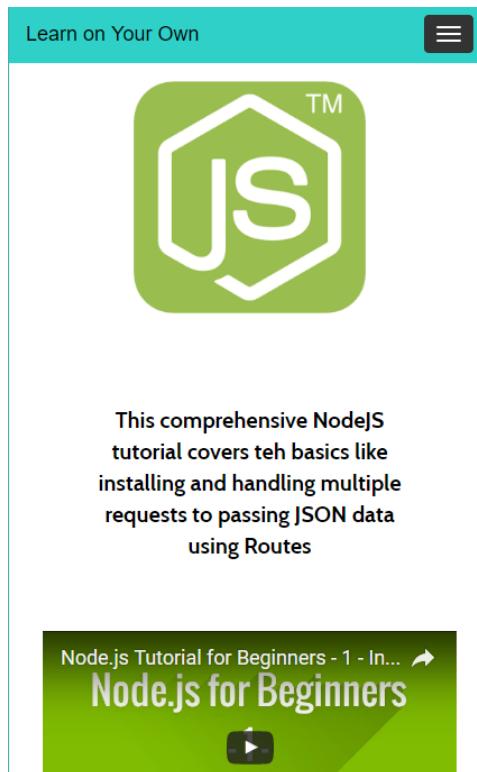


Figure 5.14: Individual Courses

Lecturer Page

The lecturers page allows the users to view all the lecturers that are responsible for creating tutorials for the application. These lecturers appear nicely in a list for mobile users as well as computer users. The mobile device view can be seen in part a of Figure 5.15. The user can then select an individual lecturer profile, administrators then have the option to create, update and delete the lecturers. The lecturer profile can be seen in part b of Figure 5.15 along with the buttons for administrators to perform the CRUD (create, read, update, delete) operations.

Learn on Your Own

Peter Grady

A portrait photograph of Peter Grady. He is a middle-aged man with a shaved head and a well-groomed beard and mustache. He is wearing a dark blue, possibly navy, zip-up hoodie. The background is a plain, light-colored wall.

James has a degree in Software Development and has a keen interest in Robotics, he has worked for Yahoo and Twitter, he specialises in C#

Michael Dever

A portrait photograph of Michael Dever. He is a man with dark brown hair, styled back, and appears to be wearing a light-colored, possibly white or cream, button-down shirt. The background is a soft-focus indoor setting.

Learn on Your Own

[Edit lecturer](#)

[Create lecturer](#)

[Delete lecturer](#)

Christopher Smith

Christopher has a degree in Computer Science as well as psychology and has a keen interest in Artificial Intelligence, he specialises in Ruby and C#

Figure 5.15: Lecturer Screen

Overall, we were pleased with the aesthetics for the application, we feel as though we created a professional and a consistent looking web application which encompasses our aims and objectives.

5.4 Code Samples

In this section, we will include some of the more interesting code snippets from our project and state what they are doing.

Below is the Course Controller which contains the Java code that is used to perform the CRUD (create, read, update, delete) operations on the database. An API endpoint is made so that AngularJS can get access to these methods.

```
//return all courses
@RequestMapping(method=RequestMethod.GET)
public List<Courses> getAllCourses() {
    return coursesRepository.findAll();
}

//create a course
@RequestMapping(method=RequestMethod.POST)
public Courses createCourse(@Valid @RequestBody Courses courses) {
    return coursesRepository.save(courses);
}

//update the details of a course
@RequestMapping(value="{id}", method=RequestMethod.PUT)
public ResponseEntity<Courses> updateCourse(@Valid @RequestBody Courses courses, @PathVariable("id") String id) {
    Courses courseData = coursesRepository.findOne(id);
    if(courseData == null) {
        return new ResponseEntity<Courses>(HttpStatus.NOT_FOUND);
    }
    courseData.setName(courses.getName());
    courseData.setLecturer(courses.getLecturer());
    courseData.setPicture(courses.getPicture());
    courseData.setVideo(courses.getVideo());
    courseData.setDescription(courses.getDescription());
    courseData.setEnroled(courses.getEnroled());

    Courses updateCourse = coursesRepository.save(courseData);
    return new ResponseEntity<Courses>(updateCourse, HttpStatus.OK);
}

//delete a course by its id
@RequestMapping(value="{id}", method=RequestMethod.DELETE)
public void deleteCourse(@PathVariable("id") String id) {
    coursesRepository.delete(id);
}
```

Figure 5.16: Course Controller

This Java code is used to define the collection name and fields that will be stored in the MongoDB database. Objects are created out of these variables using the setters while information is retrieved using the getters for methods like update. This java class also contains constructors. Code similar to this has been created for all the collections that we have used in this project.

```
//define the collection name
@Document(collection = "courses")
public class Courses {
    //define the variables in the database
    @Id
    private String id;
    private String name;
    private String lecturer;
    private String picture;
    private String video;
    private String description;
    private String enroled;

    //default constructor
    public Courses(){}
}

//constructor using fields
public Courses(String name, String lecturer, String picture, String video, String description, String enroled) {
    this.name = name;
    this.lecturer = lecturer;
    this.picture = picture;
    this.video = video;
    this.description = description;
    this.enroled = enroled;
}

//getters and setters
public String getId() {
    return id;
}
public void setId(String id) {
    this.id = id;
}
public String getName() {
    return name;
}

```

Figure 5.17: Courses Java Code

```

todoApp.controller('CoursesDetailsCtrl', ['$coursesFac', '$lecturerFac', '$scope', '$sce', '$stateParams', function CoursesCtrl(coursesFac,
    lecturerFac) {
    var id = $stateParams.id;
    console.log(id);

    //getCourse() method calls the coursesFac factory/service which calls the api endpoint and returns
    //the course information based on the course id.
    $scope.getCourse = function(id){
        coursesFac.getCourseById(id)
            .then(function (response) {
                // var items;
                response.data.video = JSON.parse(response.data.video);
                console.log(response.data.video);
                $scope.courses = response.data;
            }, function (error) {
                $scope.status = 'Unable to load course data: ' + error.message;
                console.log($scope.status);
            });
    };

    //passes the state parameter in to the method. The parameter is the course Id
    $scope.getCourse(id);

    $scope.getLecturer = function(){
        lecturerFac.getLecturer()
            .then(function (response) {
                var courses;
                //response.data.courses = JSON.parse(response.data.courses);
                // courses = response.data.courses;
                console.log(courses);
                $scope.lecturers = response.data;
            }, function (error) {
                $scope.status = 'Unable to load lecturer data: ' + error.message;
                console.log($scope.status);
            });
    };
    $scope.getLecturer();
}]
);

```

Figure 5.18: Angular Course Controller

This JavaScript code is an example of the use of AngularJS in our application. This code is responsible for accessing the defined methods of the Course Controller in Java code. Here we specify what method from the Java code that we want to use, we then parse the returned data to be in the form that we want to display it in.

```
todoApp.factory('coursesFac', ['$http', function($http) {  
  
    var urlBase = '/api/courses';  
    var coursesFac = {};  
  
    coursesFac.getCourses = function () {  
        return $http.get(urlBase);  
    };  
  
    coursesFac.getCourseById = function (id) {  
        return $http.get(urlBase + '/' + id);  
    };  
  
    return coursesFac;  
}]);
```

Figure 5.19: Angular Course Service

This JavaScript code is responsible for specifying the API end point to be used by the controller. This code gets a handle of the CourseController.Java's methods. From there we can then specify what methods we want to use and whether a HTTP GET, POST or any other HTTP method is needed for the use of this method.

Finally, we will include the HTML view that we created, this will conclude all the classes that is used for each collection. The HTML view contains a combination of HTML and AngularJS code that will inject the returned data into the web page. The AngularJS code is contained between curly braces and contains the collection name followed by a dot followed by the variable that you want to inject into that part of the view.

Below is the HTML code used for the Courses page:

```
<!-- Page Content -->


Figure 5.20: HTML code for the courses page



From the above code snippets you should be able to piece together the flow of data from one end of the application to the other.


```

Chapter 6

System Evaluation

6.1 Chosen Technologies

When choosing the technologies for our project we had to ensure that we struck a balance between both new and modern technologies and reliability. By choosing the newest technologies you may be limiting yourself in terms of online support through communities, features available, as well as reliability. Many new technologies can be unstable in its initial stages as the developers experiment and change the way they do things. This can be frustrating as a developer as it will lead to various errors throughout development. This led to our decision to use newer technologies that were well established. When choosing our technologies, we factored in the technologies stability, popularity and online community supporting it.

6.1.1 MongoDB Evaluation

We found MongoDB very easy to work with as it didn't have the strict constraints like Primary and Foreign Keys that MySQL imposes between its tables. From previous projects, we can confidently say that these constraints limit the developer's ability to add to the system. When using MongoDB, we found that the data was free flowing as it was easy to store and query. MongoDB's query time was very fast and its query language was easy to understand. The object structure in MongoDB was clear and easy to understand. MongoDB is also highly scalable meaning that it would handle a high volume of users well.

6.1.2 Spring Framework Evaluation

We found that the Spring Framework had a very steep learning curve when learning it from the beginning. This led to a lot of time spent researching and watching tutorials on how Spring works and how to do various things in it. However, once you get over the steep learning curve you begin to appreciate what the Spring Framework does. It abstracts a lot of the time-consuming configuration away from you and provides a great starting place for your project. The built-in features such as Spring Security are really helpful too as they save developers a lot of time. Spring is used in large-scale applications worldwide, proving that it would scale well in this application.

6.1.3 AngularJS Evaluation

We also found that AngularJS had a steep learning curve as we were not familiar with its syntax. JavaScript was also not part of our college course so we knew a very limited amount of JavaScript coming into this project. We feel like this further increased the learning curve when using AngularJS. We also found it hard to debug errors in AngularJS due to JavaScript's lack of typing. That being said, we did enjoy using AngularJS and we feel like we learned a lot from using it.



Figure 6.1: Basic Architecture

6.2 Robustness

JavaScript is naturally lacking in robustness; this lack of robustness is ingrained in the language. JavaScript's lack of typing, not having official guidelines, no implicit variable definitions and lack of structure in general

leads directly to its robustness. This leads to plenty of issues for developers as there is no error checking or auto completion. This leads to a lot of debugging as the errors aren't highlighted for you. You can get by this with console logging to see where in code the application is failing. Google Chromes developer tools are also a huge asset to developers using JavaScript. The use of TypeScript would help prevent these issues. AngularJS is a very good framework for adding structure to your JavaScript through controllers and services. The modularity of AngularJS makes it easier for the developer to identify where the error is coming from.

6.3 Performance

Overall, I think the application performs well for its purpose, there is also room for scalability in our application with concurrent users. MongoDB performs very well due to its ability to accomplish queries quickly, allowing for quick insertion and retrieval of data. As mentioned above MongoDB is scalable meaning that it could handle a large number of users as well as a large number of courses. This means that from a back-end point of view we can scale up by adding more courses and having more users easily.

The Spring Framework also performs very well in our application. The Spring Framework has been used in many large-scale applications such as Amazon and eBay. This shows that it can be scaled to account for a large number of users, although with our implementation of it, it may need some tweaking first. Finally, AngularJS performed as we needed it to for our implementation. The implementation of AngularJS we use is sufficient for our purposes.

6.4 Limitations

The biggest limitation we had during this project was through the use of Spring Boot. When using Spring Boot almost all of the XML configuration is abstracted away from the developer. This meant that we were unable to change some of the lower level XML configurations of the system. When using the Spring Core Framework, you have the ability to change these configurations. However, for new users of the Spring Framework using Spring Core straight away would further increase the already steep learning curve that Spring has. Although it was not a huge hindrance in the progression of our project it is a limitation of the application.

Chapter 7

Conclusion

In this project, we aimed to create a free to use online learning platform for students containing video tutorials on various programming languages.

Throughout the development of this application we learned new technologies while hitting our goals that we set out. We also increased our productivity by using features such as GitHub Issues and Kanban boards. We attended regular meetings to ensure that we were on track to meet our deadlines. All this led to the creation of a well-designed and easy to use application for students.

In future we would:

- Keep notes of each meeting attended and what was done in those meetings, sharing those notes with all parties.
- make use of more GitHub features such as GitHub milestones which would help us meet our goals.
- Create tests for our application as we created it, so we could test throughout the development process.
- Use the Travis Framework to automatically test each commit that is pushed to the GitHub repository.
- Use Angular2 or some form of TypeScript language instead of AngularJS as these technologies are newer.
- Use the Spring Core Framework instead of the Spring Boot Framework so we would get more control of our application.

We feel like we learned a great deal throughout the development of this application, the main components would be:

- New technologies such as MongoDB, The Spring Framework, AngularJS
- Teamwork
- Communication
- The use of various methodologies (Scrum, RAD Kanban, Extreme Programming)

All of the skills and development techniques that we learned while developing this application will help us a great deal in future during our working lives.

Our ability to research new technologies and understand how these technologies could be used together was key to the creation of this application. Throughout the development of this project we found and utilised new tools to make the development process easier.

This report contains the details and represents our journey throughout the creation of this application. We met our goals, solved any problems encountered and adapted the application when it was required.

This project was a great learning experience for both of us, allowing us to explore new technologies in an area that we were interested in. We thoroughly enjoyed the freedom and responsibility given to us to create our own large scale application. Overall, we found this project to be both challenging but extremely rewarding.

Bibliography

- [1] Mongo Official website: <https://www.mongodb.com/>
- [2] MongoDB compared with CouchDB: <http://stackoverflow.com/questions/3375494/nosql-mongodb-vs-couchdb>
- [3] Spring Framework Official Website: <http://projects.spring.io/spring-boot/>
- [4] JavaEE Official Website: <http://docs.oracle.com/javaee/7/>
- [5] AngularJS Official Website: <https://angularjs.org/>
- [6] AngularJS documentation: <https://docs.angularjs.org/api>
- [7] GitHub Kanban Boards:
<https://github.com/seaniemc/LearnOnYourOwn/issuesboards?repos=79832001>
- [8] About MongoDB: <https://www.mongodb.com/what-is-mongodb>
- [9] MonogDB popularity Graph: Click here to go to the graph
- [10] Spring Framework popularity Graph: Click here to go to the graph
- [11] AngularJS popularity Graph: Click here to go to the graph
- [12] GitHub quote: <https://en.wikipedia.org/wiki/GitHub>
- [13] AngularJS's GitHub Repository: <https://github.com/angular/angular.js>
- [14] Docker's GitHub Repository: <https://github.com/docker/docker>
- [15] Our GitHub Issues: <https://github.com/seaniemc/LearnOnYourOwn/issues>
- [16] Information on MongoDB:
https://www.tutorialspoint.com/mongodb/mongodb_overview.htm

- [17] MongoDB advantages:
<https://www.tutorialspoint.com/mongodb/mongodbadvantages.htm>
- [18] Useful information on The Spring Framework:
<https://www.tutorialspoint.com/spring/springoverview.htm>
- [19] Stack Overflow post on Spring: <http://stackoverflow.com/questions/32033058/why-use-spring-boot-rather-than-spring-boot-less>
- [20] W3schools HTML tutorial: <https://www.w3schools.com/html/>
- [21] Tutorialspoint JavaScript tutorial: <http://www.tutorialspoint.com/javascript/>
- [22] Bootstrap Official website: <http://getbootstrap.com/getting-started/>
- [23] Bootstrap quote: <http://getbootstrap.com/>
- [24] Code Academy AngularJS tutorials: <https://www.codecademy.com/learn/learn-angularjs>
- [25] Useful AngularJS information:
<https://www.tutorialspoint.com/angularjs/angularjsoverview.htm>
- [26] Bower's Official Website: <https://bower.io/>
- [27] NPM's Official website: <https://www.npmjs.com/>
- [28] Apache Tomcats Official Website: <http://tomcat.apache.org/index.html>
- [29] Amazon Web Service Official Website:
<https://aws.amazon.com/websites/>
- [30] Amazon Web Service Customers: <https://aws.amazon.com/solutions/case-studies/?nc2=hqlnylivestreamblu>
- [31] Slacks Official Website: <https://slack.com/>
- [32] GitHub's Official Website: <https://github.com/>
- [33] MySQL Official Website: <https://www.mysql.com/>
- [34] EmberJS Official Website: <https://emberjs.com/>
- [35] EmberJS Customers: <https://emberjs.com/ember-users/>
- [36] Guice GitHub Repository: <https://github.com/google/guice>

- [37] Heroku Official Website: <https://www.heroku.com/>
- [38] GridFS Official Documentation: <https://docs.mongodb.com/manual/core/gridfs/>