DUBLIN CITY UNIVERSITY
SCHOOL OF ELECTRONIC ENGINEERING

Web Resources for Self-Diagnostic Testing of Mathematics

Christian Hope Crow

April 2020

BACHELOR OF ENGINEERING

IN

ELECTRONIC AND COMPUTER ENGINEERING

MAJORING IN

DIGITAL INTERACTION

Supervised by Dr C. Brennan

Acknowledgements

I would like to thank my supervisor, Dr Conor Brennan, for guiding me through this difficult process. He supported me in many ways by having progress meetings with me, assisting me with the mathematics work within the project, and in general, helping me with my first major project. I would also like to thank Dr David Molloy for helping with some of the Web Application aspects of this module, as it was my first time working with such things. I would also like to thank Dr Robert Sadleir who was very accommodating of my extenuating circumstances that came up during the course of this project. Finally, I would like to thank my family who were nothing but supportive during this difficult time.

Declaration

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I have read and understood the Assignment Regulations set out in the module documentation. I have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

I have read and understood the DCU Academic Integrity and Plagiarism at https://www4.dcu.ie/sites/default/files/policy/1%20-%20integrity_and_plagiarism_ovpaa_v3.pdf and IEEE referencing guidelines found at https://loop.dcu.ie/mod/url/view.php?id=448779.

Name: Christian Hope Crow   Date: April 8rd 2020

Abstract

A common issue a lecturer may face, especially in the field of mathematics, is being unable to directly address common errors that students may come across. This is where Web Resources for Self-Diagnostic Testing of Mathematics comes in. The goal of the project is to create a web application that allows student to self-diagnose their mathematical ability using sets of multiple-choice mathematics questions, allowing them to address weaknesses with their lecturers before critical assessment. This dissertation discusses the method of solution of this problem, the related literature, and the process of the implementation. This results in a web application that accomplishes its goal in helping students improve their mathematical ability.

Table of Contents

Table of Figures

Chapter 1 - Introduction

In this chapter, the basis of the project will be discussed in finer detail, going through some literature as well as problems in the area, and what this project solves. In addition to this, the document structure will also be discussed.

1.1 Web Resources for Self-Diagnostic Testing of Mathematics

A common issue a lecturer may face, especially in the field of mathematics, is being unable to directly address common errors that students may come across. This problem arose directly from Dr Conor Brennan’s experience as a lecturer for Engineering Mathematics. These common errors mean that students will keep making the same errors until they are addressed. This problem remains a constant because, while education curriculum in the field of science and history progresses rapidly, the curriculum in the field of mathematics has generally stayed the same [1].

This is where Web Resources for Self-Diagnostic Testing of Mathematics comes in. The aim of this project is to create a web application that helps student diagnoses their own mathematical ability by using sets of multiple-choice questions. This, in turn, allows students to pinpoint their weaknesses in the subject and address them with their lecturer in time for assessment.

The idea of web resources for self-diagnostic testing is that outside of regular classes students can test their understanding of the course material. These tests are specifically made with common errors in mind, so that when a student takes the test, with every question they get wrong, a different common error can be addressed. This allows for the students’ issues to be addressed on an individual basis, while not taking time out of a teacher’s busy schedule. Rather than a similar test that student may take and make mistakes on, but then the errors they made are never addressed. These errors are addressed by using the informative videos that are played once the student enters an incorrect answer, which then shows that student which common error they fell into to reach that incorrect answer.

This dissertation will cover the basis of the project, the related literature, the technologies involved, and the process of crafting the web application. It will discuss the problem at hand and the solution that has been created.

1.2 The Document Structure

Chapter 1, *Introduction*, discusses the basis of the project in finer detail, going through some literature as well as problems in the area, and what this project solve. In addition to this, the document structure will be discussed.

Chapter 2, *Technical Background*, discusses the technical background for the web application, going through what technical components needed to build this specific web application, a background on each component, and a brief introduction of how each component is implemented in this application.

Chapter 3, *Design of Web Resources*, discusses the overall project- to create a web resource that will allow to students to self-diagnose their mathematical ability, and to combat common errors that they fall into. In this chapter, the idea for the project will be discussed, along with the theory and research into the concept, and the time management involved in this project.

Chapter 4, *Implementation and Testing of Web Resources*, discusses each step in the implementation process. In the context of this project, this includes the front-end, the back-end, the database, and its respective testing.

Chapter 5, *Results and Discussion*, discusses the result of the project, both from the outcome of the testing and personal thoughts. Then, there will be an overall discussion of the project which shall include changes to be made and future work.

Chapter 6, *Ethics*, discusses the ethical issues that arose in the project. The problem itself will be addressed, as well as the solution used to solve the problem.

Chapter 7, *Conclusions and Further Work*, concludes the dissertation and provides the final thoughts on the project.

This is followed by *References*, which is self-explanatory.

Chapter 2 - Technical Background

This chapter discusses the technical background for the web application, going through what technical components needed to build this specific web application, a background on each component, and a brief introduction of how each component is implemented in this application.

2.1 Web Applications and their Server

2.1.1 What is a Web Application?

Web Application Development is the process of creating applications which can run across the internet [2]. An increasing number of sites on the Web are now web applications, meaning they mimic desktop apps rather than traditional static text-images-links documents that make up much of the Web [3].

Web applications consist of two primary parts: the Client-Side also known as the front end and the Server-Side also known as the back end. The Client-Side is referred to as the front end as this is the side of the application users will see when accessing the application, while the Client-Side contains the Server and Database. Client-Side Developments is based around programming languages such as HTML, CSS, and JavaScript, since these programming languages are executed from within a user's browser and can be used to render and format web pages. While Server-Side programming languages like Java, which is used to create Java Servlets and Java Server Pages, and SQL, which is used to create and communicate with a database, run entirely within a webs server.

While a web application can be created on using Client-Side Development, Server-Side development is also important because it provides advantages such as code protection since Server-Side code is executed before the HTML code is sent to the browser, which also means the Sever-Side code is browser independent and does not depend on a user have the correct version of a browser.

2.1.2 Tomcat Web Server

Apache Tomcat is an open source Java Servlet container that is required for facilitating the use of JSPs and Servlets, discussed in Section 2.3.4 [4]. It deploys and hosts the Servlet and takes care of the routing on behalf of the user.

2.2 Programming Software

2.2.1 Eclipse IDE

Eclipse is a Java-based Integrated Development Environment (IDE). What this means is that Eclipse is a piece of software that assists in the process of programming by making many programming tools available in one programme, such as editing source code and debugging [5]. It also offers many tools for web application development, which shall be discussed in Section 2.3.4.

2.2.2 MySQL Workbench

MySQL Workbench is a graphical tool that allows the user to interact with databases using SQL. Not only this, but it allows the user to perform a wide variety of tasks including administration and database design [6]. Using MySQL Workbench, a SQL Server was created to store the information for the application in databases. This facilitated many functions within the web application without that information being manually put into the code itself.

2.3 Programming Languages

2.3.1 Hypertext Mark-up Language (HTML)

Hypertext Mark-up Language (HTML) is a standard mark-up language used in web development for describing the structure of web pages. The most recent version of HTML is HTML5. The purpose of HTML is that it is composed of elements that tell the bowser how to render and display content [7].

Any HTML Document should contain a minimum of five elements. These Elements include the document definition, the html root element, the head that contains meta information, the title of the document, and the body which contains the documents page content. An HTML Document containing the minimum elements are shown in the code excerpt below:

```
<!DOCTYPE html>
<html>
    <head>
      <title>Title</title>
         </head>
      <body>
           <!--Page Content-->
      </body>
</html>
```

The head element normally contains the title along with the documents meta data such as the character set, links to external style sheets, and external scripts such as JavaScript files.

2.3.2 Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) are used to style and specify the layout for the content of a HTML Document so that a web pages can be athletically pleasing and easy for users to use.

A CSS file is made of declarations that affect groups of tag within a HTML file. Take the following example:

```
h1 {
      color: blue;
      font-size: 34pt; }
```

The above declaration would change all Heading 1's (h1) and change the text colour to blue and the font size to be 34.

CSS also supports the styling of id elements, which styles the element with a specified id, and class elements, which selects elements with a specific class attribute. Ids and class specifications take the following formats:

```
#id {
      css declarations;
}

.class {
      css declarations;
}
```

2.3.3 JavaScript

JavaScript is a scripting language that is embedded into HTML documents and uses object-oriented based approach to programming. JavaScript is embedded into an HTML document to program a web pages behaviour [8] and make a web page interactive without depending on Server-Side scripts [2]. The script tag used within the body of an HTML document is used to surround any internal JavaScript code within the HTML document, while the script tag used within the head of an HTML document is used to call in an external JavaScript source.

2.3.4 Java

Java is an Object-Oriented programming language, that is used to create Java Servlets and Java Server Pages (JSPs) needed for Server-Side Development in web applications. Both Servlets and JSPs run entirely within a web server rather than being executed from within the browser.

Java Servlets are Java programmes that run on the application server. They are made to take requests from the web server, process it, and return the response. This is done because Servlets are capable of handling complex requests [9].

Java Server Pages are built on top of Servlets, and thus easily work together. A page is made up of HTML with embedded JSP tags [10]. However, in order to use JSPs, the pages and the Servlet must be inside of a servlet container, for example, Tomcat described in Section 2.1.2.

2.3.5 Structured Query Language (SQL)

Structured Query Language (SQL) is a programming language that is used to create and communicate with a database. According to the American National Standards Institute, it is the standard for relational database management systems [11]. SQL differs from many programming languages in that it describes the result rather than how to get the result.

Tables are the objects contained within a database. Each table is comprised of data or information that is valuable towards a specific database. Each table has a unique name and is made up of both columns and rows. Each column contains a single data type, such as string or integer, while each row contains the data for one object within a table.

2.4 Miscellaneous

2.4.1 MathJax

MathJax is a JavaScript tool that works with CSS to display mathematics equations in browser. Since it uses CSS and web fonts, this tool scales with the browser so it is both accessible and flexible [12].

MathJax uses Tex syntax to create its mathematical equations. It combines backslashes and codes to create mathematic symbols not readily available to the user, such as pi or an integration symbol. Take the following MathJax example which renders as the quadratic equation in-browser:

```
\[x = {-b \pm \sqrt{b^2-4ac} \over 2a}.\]
```

2.5 Summary

The building blocks of this project are as follows: A front-end developed with HTML, CSS, and JavaScript. A backend developed with Java and Eclipse IDE, which was also used to develop the front-end. A database built with MySQL Workbench, and is queried with the SQL language.

Chapter 3 - Design of Web Resources

The basis of this project is to create a web resource that will allow to students to self-diagnose their mathematical ability, and to combat common errors that they fall into. In this chapter, the idea for the project will be discussed, along with the theory and research into the concept, and the time management involved in this project.

3.1 The Project

3.1.1 Web Resources for Self-Diagnostics in Mathematics

The scheme of this project is based on web application development and online learning. The core building blocks of web-based systems were used to incrementally create the web application project [ce4]. The project is a stand-alone web application that has a set of web resources that will aid students in self-diagnosing of their mathematical ability, as well as identifying and correcting common errors that students may have about basic mathematics. These resources consist of a set of multiple-choice questions that are arranged by topic, so that the student will be able to focus on specific areas that students may find challenging [13].

Each incorrect multiple-choice answer corresponds to one of several common errors. So, when a student chooses an incorrect answer, the system triggers a short video to be played outlining the common error and explaining how the student can rectify this error.

3.2 Web Application Development

3.2.1 Web Application Map

Web Application Maps are diagrams that are used during the planning phase of applications to map the layout of an application, so it was very important to take advantage step while planning the application, so that once the development of the application began there would be a layout map to look back on. In this section, the layout of the application will be discussed. The application map, shown in Figure 3.1, has three different colours which are green, blue, and grey. The green nodes are features that were unable to be included in this version of the application but would be incorporated into later versions of the application, while the blue nodes are what is completed and incorporated into this version of the application. Finally, the grey nodes show where the application could be expanded by adding in future math sections and future tests from the lecturer. Now, we will step through the blue nodes in the application map to show the current layout of the application map at this time.

When the user enters the web application, they will begin on the homepage of the application. This will introduce the application and have the means for the user to navigate the application. From the home page, the user can then navigate to either an overview of the testing sections available in this application or a page of lecturer information. The lecturer information is a static webpage, that allows the user to learn more about the lecturer in charge of this application, while also giving the user the means to contact them at any given time. On the other hand, in the section overview page, the user can choose which section they would like to attempt and enter in that respective section. Next, the user must choose which test they would like to take from this section. Inside the test, the user can either choose the correct result and receive a success notification, the incorrect result and receive a incorrect notification with a video explanation of the possible common error, or the user can upload their solutions to google drive. Next, we will go over another step in the planning process, the choice of whether or not to use web application framework.

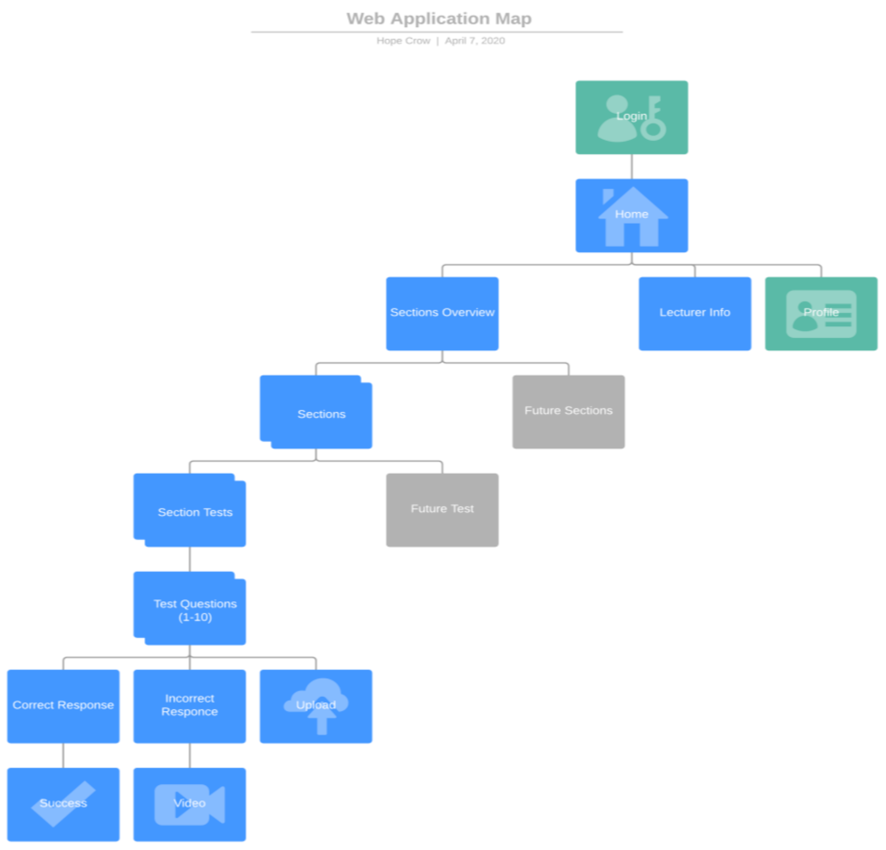


Figure 3.1. Web Application Map

3.2.2 Frameworks

Web Application Frameworks are designed to support developers in building web applications, this is done by automating many processes in web application development such as user session management, templating systems, database access, and database management. While this can save developers a significant amount of time, it means much of the developer's application, especially on the backend will be automated by the chosen framework.

While this does sound promising and would help development of the application move along quicker so that it would be easier to get into user testing and revisions. When it comes to a fourth-year project, it is more important to demonstrate programming and web development knowledge, over a developer's knowledge of being able to use a framework.

3.3 Common Errors in Mathematics

3.3.1 What are Common Errors?

A common issue a lecturer may face, especially in the field of mathematics, is being unable to directly address common errors that students may come across. This problem arose directly from Conor Brennan's experience as a lecturer for Engineering Mathematics. These common errors mean that students will keep making the same errors until they are addressed. This problem remains a constant because while education curriculum in the field of science and history progresses rapidly, the curriculum in the field of mathematics has generally stayed the same [1].

3.3.2 How do Common Errors Occur?

Students have been taught primarily based on the concept that they must be able to reach a level of fixed intelligence, rather than increase their capacity to learn [1]. This leads to students needing to be taught on an individual level. Teachers are expected to recognise common pupil errors, while understanding how they arise, and how to prevent them [1]. All errors must arise from previous occurrence [14]. Errors can be a powerful tool to diagnose learning difficulties and consequently direct remediation [15].

Several people have attempted to modify teaching methods to include specific types of pupil to pupil discussion into mathematic learning to help avoid these common errors. This draws several principles that suggest how one might design more effective mathematic lessons [14]. This is where the project for web resources of self-diagnostic testing can help.

3.3.3 How this Problem is Addressed

The idea of web resources for self-diagnostic testing is that outside of regular classes students can test their understanding of course material. These tests are specifically made with common errors in mind, so that when a student takes the test, with every question they get wrong a different common error can be addressed. This allows for the students' issues to be addressed on an individual basis, while not taking time out of a teacher's busy schedule. Rather than a similar test that student may take and make mistakes on, but then the errors they made are never addressed. These errors are addressed by using the informative videos that are played once the student enters an incorrect answer, which then shows that student which common error they fell into to reach that incorrect answer.

3.4 Gantt Charts and Time Management

3.4.1 Semester One

Work on this project began at the end of Semester 1 Week 3 on October 10th after the first project meeting with supervisor, Dr Conor Brennan. During this meeting, all the projects supervised by Dr Brennan were discussed, along with the tasks to be completed for the following week. This means that progress on the project did not commence until Week 4 of that semester. Semester 1 was spend working on learning about common errors and web application development techniques. A web application map, shown in Figure 3.1, was constructed and a layout of the web application was created.

The work in Semester 1 began with compiling a list of common errors in mathematics, this formed the basis of the tests in the application. Then, coding practice was done in both HTML, CSS, JavaScript, and SQL, as these were unfamiliar. Next, the Web Application map was developed as seen in Figure 3.1, although it was finalised at a later time. Then, the research into Web Application Development began as this was an unfamiliar area.

By the end of the Semester, a basic dynamic web application page had been created using a Tomcat Server on Eclipse [16,17]. This Application was made in two forms using Java

Servlets and JavaServer Pages [18]. Although this seemed like small steps, it was progress towards developing the full web application.

Overall, Semester 1 was spent developing ideas of what would go into the web application and starting the development of such a web application. Whereas, more of the application development was done in Semester 2 whilst a Web Application module was being taught [ce4].

3.4.1 Semester Two

Semester 2 began with the presentation. This was based on the Status Report submitted in December, and included work done over the Christmas period. Once the presentation slides and video recording were submitted, an interview with the supervisor and an examiner was arranged for further questions about the presentation.

Next, the web application map was updated, shown in Figure 3.1. This update saw the removal of student accounts as they were deemed too time consuming for the available time remaining. The use of Frameworks was also scrapped for this project, see Section 3.2.2.

Then, the front-end development on the website began. This work includes the HTML and CSS development of the home page, pages for individual sections, and lecturer information. This work also includes the visual design of the website, which was designed to be as accessible as possible to all users. Next, work began on the SQL database, creating tables for the questions and answers of each tests. This is described in Section 4.2.2.

Then, work began on crafting the multiple-choice questions using JavaScript. At first, each question had its own page which directed the user to a video if the answer was incorrect, or the next question if correct. Also, work began on adding the videos into the application to mixed results, which was eventually fixed at a later time.

Another issue was students submitting their workings for incorrect answers, as PHP was needed to upload and store the files, but this did not prove fruitful. Instead, a Google account was created so that students could upload their workings to Google Drive.

A key change was made to the web application map, which is that all questions for a single test are on one page. The box containing the question will itself load the incorrect answer video if needed, rather than redirecting the user to another page. With all key aspects of the project completed, work began on the dissertation.

3.3 Summary

In this chapter, the basis of the project was discussed, including the theory and research in that field at large. This also included the web application map associated with the project, and the possibility of the usage of frameworks within the project. Finally, the time management within the project was discussed.

Chapter 4- Implementation and Testing of Web Resources

In this chapter, each step in the implementation progress will be discussed. In the context of this project, this includes the front-end, the back-end, the database, and its respective testing.

4.1 Front-end Development / Client-Side Development

4.1.1 Creating Web Pages

This specific project began with front-end development since most elements to this web application would be static webpages, the exceptions to this are the test, login page, and registration page that are discussed later in the report. From the web application map, shown in Figure 3.1, a list of web pages that would have to be created can be taken. The first step would be to set up the homepage. The home page is a static webpage that introduces the website and allows users to navigate around the website. The content of this page would be a navigation bar, title for the application, a short description of the application, and a footer with the creator and supervisors name. An image was also added to this page to fill in empty space and make the application more inviting to users. Now, we will step through some of the HTML code that was used to create the home page, for some more information on HTML and the basic page layout, see Section 2.3.1.

To set up the homepage, the first thing that had to be done was the head tag, inside this tag was the meta data including the character set being used and a viewport to help make the application more responsive to varying sizes for browsers. The web page was then given a title, the title of each webpage should be simple but effective enough for others to easily identify what page of the application they are using. Lastly in the head tag was the links, for this page three links were used, the first being a font that is used to create a dropdown arrow for the navigation bar, secondly is the link to the CSS page used to style this web page, and finally is the link to a shortcut icon this is the icon that will be displayed on the tab of a web browser when the user is running the application. The exert of the head tag for the home page of this application is provided below:

```
<head>
      <meta charset="UTF-8">
```

```
	<meta name="viewport" content="width=device-width, initial-
scale=1">
	<title>Maths Home Page</title>
	<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
	<link rel="stylesheet" type="text/css"
href="CSS/MainStyle.css" />
	<link rel="scortcut icon" type="image/png" sizes="16x16"
href="Images/favicon-16x16.png">
</head>
```

Next is the body of the application. This includes the content of the web page that was discussed at the beginning of this section. The first part of the web page's content would be the navigation bar as this will be the very first thing the user sees at the top of the page. The goal of this navigation bar is so that it could take a user to most pages within the application, without having to go through multiple pages in the process of navigating the application. To create the navigation bar the *div* tag was used to define a division, this with the embed links would be all that is needed to create a simple navigation bar, but a second and third division had to be created in this case to create the dropdown menu that list the available sections to the user without them having to go through the overview of all the sections. The second division was made to tell the program which of the embed links would include the dropdown, while the third division oversaw holding the content for the dropdown menu itself.

```
<!-- Navigation -->
<div class="navbar">
	<a href="HomePage.html">Home</a>
	<div class="dropdown">
		<a class="dropbtn" href="Sections.html">Test Sections
			<i class="fa fa-caret-down"></i>
		</a>
		<div class="dropdown-content">
			<a
			href="MathsSections/Elementary.html">Elementary</a>
			<a href="MathsSections/PreAlgebra.html">Pre-
			Algebra</a>
			<a href="MathsSections/Geometry.html">Geometry</a>
			<a href="MathsSections/Algebra.html">Algebra</a>
			<a
			href="MathsSections/Trigonometry.html">Trigonometry
			</a>
			<a href="MathsSections/PreCalculus.html">Pre-
			Calculus</a>
			<a href="MathsSections/Calculus.html">Calculus</a>
			<a
			href="MathsSections/Statistics.html">Statistics</a>
```

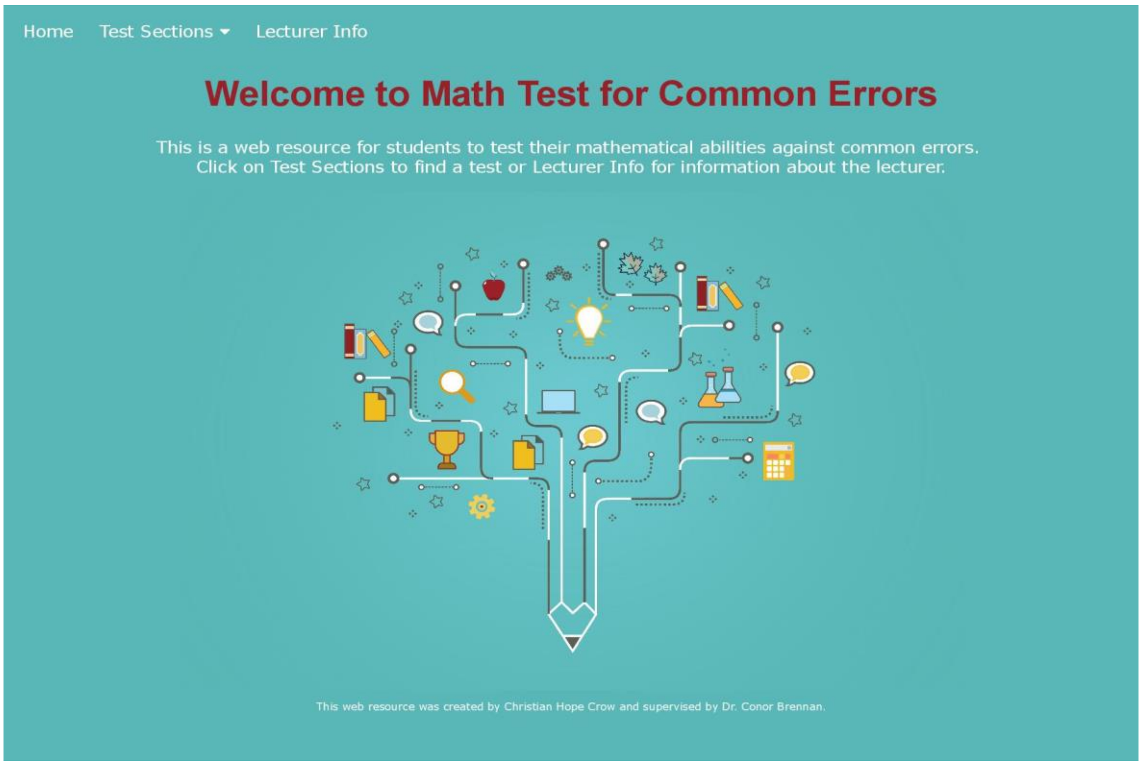
```
            </div>
        </div>
        <a href="ClassInfo.html">Lecturer Info</a>
</div>
```

Following the navigation bar is the main content of the webpage, for this simple HTML such as the *h1* or Header 1 tag is used to create a title for the webpage, with simple text for the webpage encased in the *p* or paragraph tag. To create spaces between the text, the *br* or break tag is used. Then to insert the image, the *img* tag is used. Inside this tag, an *id* is given to the image so that it can easily be style in the CSS script. Also, inside the image tag the source of the image must be identified, in this case the image is stored directly in a folder called *Images* and the title of the image is *Learning.jpg*. Then following the image, the footer of the webpage used the same paragraph tag as the header except the style command is used to decrease the font size. Using the style command is effectively having internal CSS. Below is the excerpt of code used for the main body of content in the web page:

```
<!-- header and paragraph -->
<h1>Welcome to Math Test for Common Errors</h1>
<p style="text-align: center">
        This is a web resource for students to test their
mathematical abilities against common errors.
        <br>
        Click on Test Sections to find a test or Lecturer Info for
information about the lecturer.
</p>

<!-- Welcome Image -->
<img id="homeImage" src="Images/Learning.jpg">

<!-- Footer -->
<p style="text-align: center;font-size: 10pt;">
        This web resource was created by Christian Hope Crow and
supervised by Dr. Conor Brennan.
</p>
```

The final step for these HTML web pages is to correctly style the layout and content of the webpage, this is done using CSS, which is introduced in Section 2.3.2. CSS is capable of many things such as styling fonts, tables, buttons, etc. Though CSS is used to style many elements, so the contents of the CSS file are very standard and repetitive, as it contains lots of common elements include color, background, font size, font style, width, etc. To show these elements, the CSS exerts are included below which shows a large amount of the CSS needed to style the home screen:

```
h1 {
        font-family: Arial, Verdana;
        font-style: normal;
        font-weight: bold;
        font-size: 34pt;
        text-align: center;
        color: #9A212A;
        background: #59B7B7; }


body {
        font-family: Verdana, Geneva;
        font-style: normal;
        font-size: 16pt;
        color: #FFFFFF;
        background: #59B7B7; }


#homeImage {
   display: block;
   width: 70%;
   margin-left: auto;
   margin-right: auto; }
```



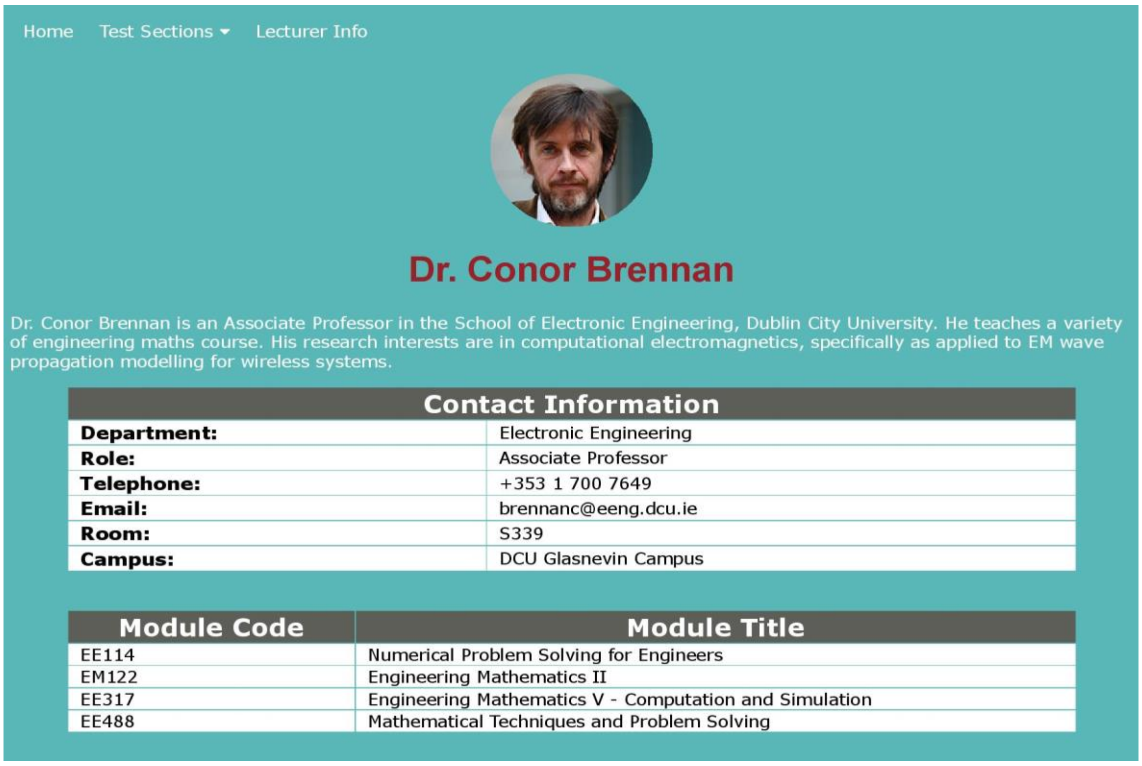
Welcome to Math Test for Common Errors

This is a web resource for students to test their mathematical abilities against common errors.
Click on Test Sections to find a test or Lecturer Info for information about the lecturer.

This web resource was created by Christian Hope Crow and supervised by Dr. Conor Brennan.

Figure 4.1. Application Home Page

The next three important web pages that were created are the Lecturer Information, Sections Overview, and finally the Section pages themselves. The development of this pages was very similar to the development of the home page, the only real difference is the content of the pages themselves, not only for the HTML but lots of the CSS used in the Home page also will be carried over with little being added, but even the CSS that will added will follow the same format of the code before it . The Lecturer Information is just a short information page about the lecturer in charge of this application and his contact information.



Dr. Conor Brennan

Dr. Conor Brennan is an Associate Professor in the School of Electronic Engineering, Dublin City University. He teaches a variety of engineering maths course. His research interests are in computational electromagnetics, specifically as applied to EM wave propagation modelling for wireless systems.

Contact Information	
Department:	Electronic Engineering
Role:	Associate Professor
Telephone:	+353 1 700 7649
Email:	brennanc@eeng.dcu.ie
Room:	S339
Campus:	DCU Glasnevin Campus

Module Code	Module Title
EE114	Numerical Problem Solving for Engineers
EM122	Engineering Mathematics II
EE317	Engineering Mathematics V - Computation and Simulation
EE488	Mathematical Techniques and Problem Solving

Figure 4.2. Lecturer Information Page

The Sections Overview is a webpage that lists all available section of mathematics on the website and gives the user a short overview of what is covered in that section. Finally, there are the section pages. Each section has its own page, while each page is relatively similar in layout the only different is the content of test available for each section. On the section page the user will be able to view what tests are available for a specific mathematics section and see a short overview of what content is covered in each individual test. Clicking on a test within a specific sections page will then take the user into the chosen test. The next subsection discusses how these tests are developed. Similar to the Lecturer Information, if time had permitted a user's information page would have been added so that once a user had logged in they would be able to view their information along with a list of the test they had completed and still need to complete, more information on this is covered in the Further Research chapter.

4.1.2 Integrating Mathematics Tests

Integrating the Mathematics Test must be done differently than creating the other webpage in this application, the reason for this is that the mathematics tests need to be interactive. This means JavaScript must be used to give the user an interactive experience. If JavaScript was not used, then each question would have to be placed on its own webpage, thus creating a very clunky experience for the user. So, for each test, there is one HTML page that creates the basic layout where the test will lay and the JavaScript file that creates the tests itself.

To begin, one must look at the HTML document, like the last section, the first thing that must be done is the head tag, the head tag for this test will be the exact same as the last head tag except for one line. This is the line that is responsible for linking the JavaScript to the HTML document. This line used the script tag to tell the HTML document that a different programming language is being incorporated into the document, this script can either include internal or external script, but since this script tag is used in the head tag rather than the body tag, the document knows that this piece of script will be external. Inside the script tag, the type of the external script must be given and the source for the external script must be referenced. The excerpt of the line is given below:

```
<script type="text/javascript" src="../../JS/AlgebraExam.js">
</script>
```

The external references to MathJax must also be included in the head tag. MathJax is a JavaScript tool that uses Tex syntax to create mathematical equations and display them in the browser, see Section 2.4.1. The excerpt for these external references is given below:

```
<script src="https://polyfill.io/v3/polyfill.min.js?features=es6">
</script>
<script id="MathJax-script" async
src="https://cdn.jsdelivr.net/npm/mathjax@3/es5/tex-mml-chtml.js">
</script>
```

Following the head is the body, similar to the head tag we only have two new developments this being the placement of *ids* needed for the JavaScript, these *ids* tell the JavaScript where to insert its data into the web page, while the class tag is used by the CSS file to format these questions.

```
<h2 style="text-align: center" id="test_status"></h2>
<div id="question1" class="QuestionFormat"></div>
```

Next is the main step of the maths test, the JavaScript file. First, all global variables for the document must be declared in this case, as the current version of this project does not have the ability to pull the questions for a database, the global variables are also were the test questions along with their responses and correct result will be placed. After the variables have been declared, the code will either contain two or three functions, depending on whether or not randomization was included in that quiz, so for this section of the report, the Calculus Test is what will be covered, since it does include the randomization function, unlike the algebra test.

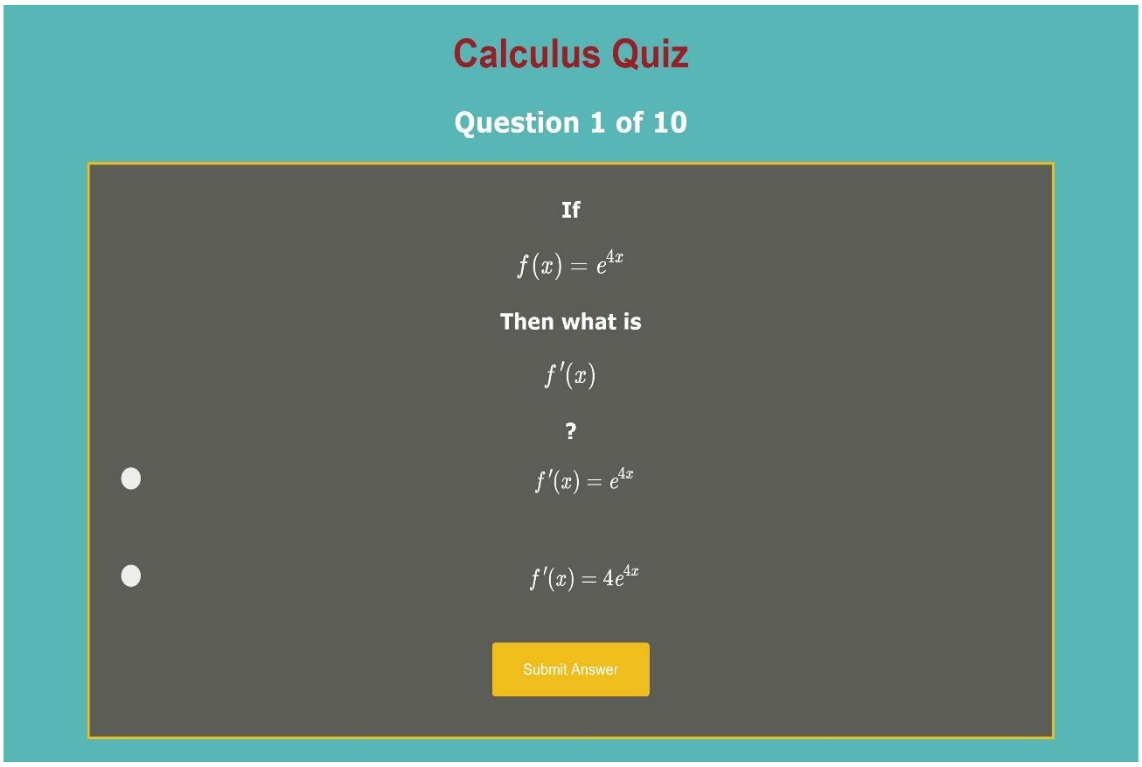


Figure 4.3. Calculus Question

The first function that is made is what allows the test to be rendered, this includes getting the element's *id* from the html file and using the *innerHTML* property to return content from the JavaScript file to an element in the HTML file. So, for example in Question 1, the question is displayed, followed by the answer choices assigned to radio buttons, and finished off with a submission button that calls the appropriate *checkAnswer* function. The excerpt of this code is shown below:

```
question1.innerHTML = "<h3>"+TestQuestions[q1][0]+"</h3>";
```

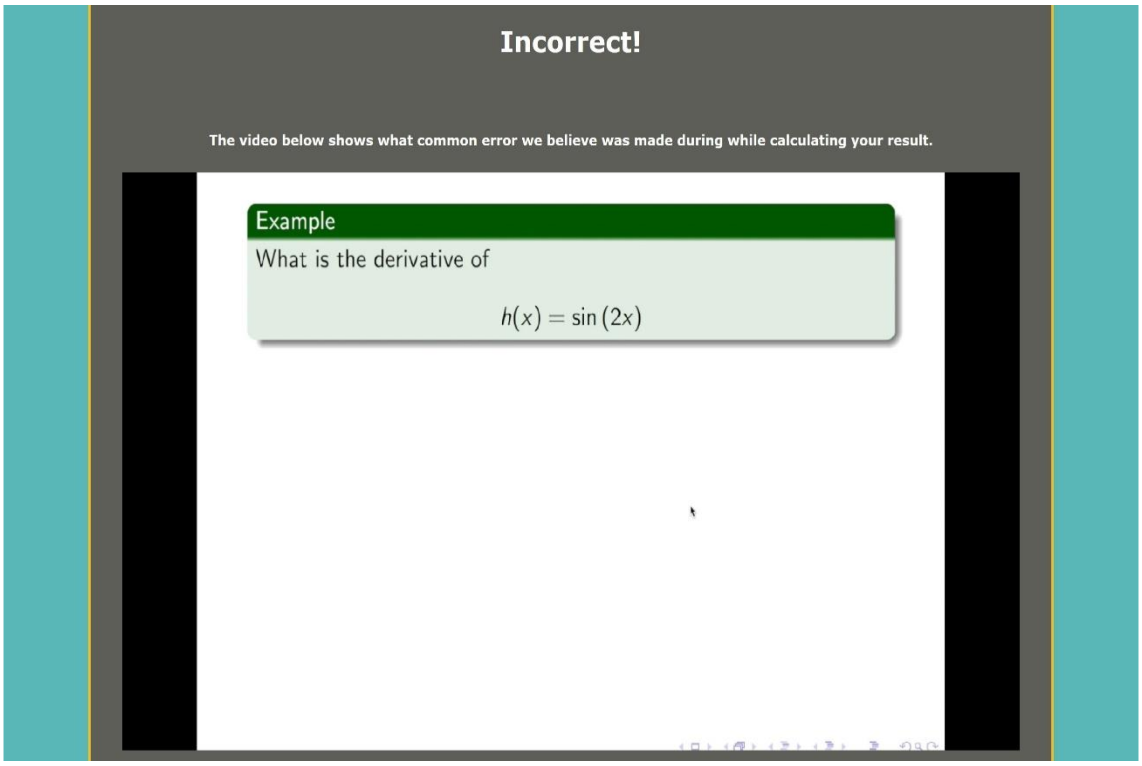
```
question1.innerHTML += "<label
class='container'>"+TestQuestions[q1][1]+"<input type='radio'
name='choices1' value='A'><span
class='checkmark'></span></label><br>";
question1.innerHTML += "<label
class='container'>"+TestQuestions[q1][2]+"<input type='radio'
name='choices1' value='B'><span
class='checkmark'></span></label><br>";
question1.innerHTML += "<button class='buttonExam' style='vertical-
align:middle' onclick='checkAnswer1()'><span>Submit
Answer</span></button>";
```

The next function is the randomization function, so for this calculus test, the randomization function goes as such: Question 1 and 2 have three possible questions that can be displayed, while Question 3 has two possible questions, and Questions 4 and 5 are standalone questions. So, altogether there are ten questions that could possibly appear on the test, but only five questions will be on the test at one time, this means that for each question that has multiple possible questions we need to randomize which question is picked. So, if the application is on Question 1 it will generate a random number from 0 to 2, while Question 2 generates a random number from 3 to 5, and finally Question 3 generates a random number from 6 or 7. Whereas the randomization function does not need to be called for Questions 4 or 5. The excerpt for the randomization function is shown below:

```
function randomization(question, num){
	if (question == 1) {
		var renum = Math.floor(Math.random() * 3);
		return renum;
	}
	else if (question == 2) {
		var renum = Math.floor((Math.random() * 3) + 3);
		return renum;
	}
	else if (question == 3) {
		var renum = Math.floor((Math.random() * 2) + 6);
		return renum;
	}
	else {
		return 0;
	}
}
```

Lastly in the JavaScript file is the *checkAnswer* function, there are five of these functions one for each question in the test, this helps to easily identify which common error video needs to be played in relation to which question is being asked. So, for *checkAnswer1*, we take in all possible answer choices from the radio button, and see which radio button has been checked, this is then set as the choice that was made. Then the choice is compared to the correct answer choice given in the question array. If they are equal, then the congratulations screen is displayed for the user, but if they are not equal, then the user is told they are incorrect and the corresponding common error video for that question is played. The *checkAnswer1* function is given below:

```
function checkAnswer1(){
  choices1 = document.getElementsByName("choices1");
  for(var i=0; i<choices1.length; i++){
    if(choices1[i].checked){choice = choices1[i].value;}
  }
  if(choice == TestQuestions[q1][3]){
      document.getElementById("question1").innerHTML = 
"<br><h2>Correct!<h2><br><p style='font-size: 
12pt'>Congratulations!<br>You choose the correct responce!</p>";
  }
  else {
      //Video will be chosen depending on incorrect result
      document.getElementById("question1").innerHTML = 
"<br><h2>Incorrect!<h2><br><p style='font-size: 12pt'>The video 
below shows what common error we believe was made during while 
calculating your result.</p>";
      document.getElementById("question1").innerHTML += "<video 
width='100%' height='auto' controls> <source 
src='../../Videos/CalQ1.mp4' type='video/mp4'></video>";
  }
}
```


Incorrect!

The video below shows what common error we believe was made during while calculating your result.

Example

What is the derivative of

$$h(x) = \sin(2x)$$

Figure 4.4. Incorrect Answer

Now that the math test have been created and integrated with the existing HTML for our web application, the next section is going to examine the Server-side of the application and talk about the development that has taken place.

4.2 Back End Development / Server-Side Development

4.2.1 Server Components

Given the use of this application, most of the work in this project went into working on the Client-Side, as this application is focused on client testing and usage, with the original intention of not saving the user's data. For the sake of complexity in this project, it was important that this application covered some full stack development. So, to add an additional level of complexity into this project the goal was to incorporate database communication, then begin working on a possible login and registration pages if time permitted. Both the database communication and login pages were able to work separately from the main interface but did not have the proper time to be embedded into the application itself. As for the registration page, it was able to be set up, but as the database was not fully integrated, therefore there was nowhere to correctly and safely store a user's registration data.

Even though these elements were not fully embedded into the application, a lot of time and hard work went into trying to get them up and running, so these are features that if more time was available hopefully would have been completed and embed into the project. These elements could have also been finished if the use of a framework were to be included into this project, but, when it came down to this choice it was more important to demonstrate programming and web development knowledge, over knowledge of using a framework.

4.2.2 Database Development

The database in this application was meant to have two purposes. The first was for storing the exam questions, their answer choices, and their solutions. The second was to store and retrieve user information related to the login and registration pages.

To solve this problem, the database *webapp* was created which holds the table *user_details* and a table for each of the sections aptly named *[section]_test*. For the integration of a database in this application the query language SQL is used, the details of this language can be viewed in Section 2.3.5. The tables were structured as follows:

Attribute Name	**Type**	**Description**	**Additional Information**
id	Integer	Unique identifier for user	Primary key
user_email	String	User's email address	Must not be null, and must be unique
user_password	String	User's password	Must not be null
first_name	String	User's first name	Must not be null
last_name	String	User's surname	Must not be null

Attribute Name	**Type**	**Description**
id	Integer	Primary key: unique identifier for question
test_id	Integer	Identifier for test within overall section
question_no	Integer	Question number within test
question	String	String containing question
answer_a	String	Answer A
answer_b	String	Answer B
answer_c	String	Answer C
correct_ans	String	Letter detailing which answer is correct

The SQL queries to create the database and its table is as follows:

```
CREATE SCHEMA `webapp`;

CREATE TABLE webapp.user_details (
  id integer AUTO_INCREMENT PRIMARY KEY,
  user_email char(255) NOT NULL UNIQUE,
  user_password char(255) NOT NULL,
  first_name char(255) NOT NULL,
  last_name char(255) NOT NULL
);

CREATE TABLE webapp.algebra_test (
  id integer AUTO_INCREMENT PRIMARY KEY,
  test_id integer,
  question_no integer,
  question longtext,
  answer_a longtext,
  answer_b longtext,
  answer_c longtext,
  correct_ans char(255)
);
```

Below is the SQL query to add a question into the table:

```
INSERT  INTO  webapp.algebra_test  (test_id,  question_no,  question,
answer_a, answer_b, answer_c, correct_ans)
VALUES (1, 1,
  "What  is  the  inverse  of  the  matrix  \\[{\\mathbf  A  }   =  \\left[
\\begin{array}{cc} 1 & 2 \\\\3 & 4 \\\\\\end{array}\\right]\\]",
  "\\[{\\mathbf   A}^{-1}     =   \\left[   \\begin{array}{cc}   1   &
\\frac{1}{2}  \\\\\\frac{1}{3}     &  \\frac{1}{4}   \\\\\\end{array}
\\right]\\]",
  "\\[{\\mathbf  A}^{-1}    =  \\left[  \\begin{array}{cc}  -2  &  1
\\\\1.5 & -0.5  \\\\\\end{array} \\right] \\]",
  "\\[{\\mathbf A}^{-1}  = \\left[ \\begin{array}{cc} 4 & -2  \\\\-
3 & 1  \\\\\\end{array} \\right] \\]",
  "B");
```

While the database is fully functioning, it is not fully integrated with the rest of the application. This is due to the setbacks related to the COVID-19 outbreak in terms of a change in work environment and the impact on other modules. This means that the questions and answers are currently “hard coded”, meaning manually entered, into the source code. It also means that there is currently no login and registration functions, although it is fully functioning. This is further outlined in Section 5.2.

4.3 Testing

4.3.1 Procedure

In order to test a web application, a different approach was needed to test this software application. A notable way of testing web applications is to automate testing by having a tool replicate a real user to show functionality.

A tool used to do this is Selenium WebDriver. It is a tool used to automate browsers for testing purposes [19]. This is done as follows. Selenium will open a browser to a specified webpage and perform a set of tasks set by the user. When run, Selenium shows the process to the user so they can spot errors. It also provides a report afterwards for more information. Recording this process allows the user to show that all functionalities are working as intended.

However, due to the COVID-19 outbreak, major changes were made to the schedule of the project. This was due to the changes in work environment and to the nature of assessment within other modules. Doing automated testing using Selenium would have been a big undertaking, as the code would have had to be refactored to accommodate the Selenium tests. Then, the Selenium tests would have to be written and integrated into the source code and then run. For these reasons, this aspect of the testing was pushed from completed work into further work.

4.3.2 User Testing

Due to the COVID-19 outbreak, User Testing was unable to go ahead as planned. Thus, this section shall describe how the User Testing would have been done hypothetically.

Before interacting with participants, two tasks would need to be completed; creating a Plain Language Statement and Informed Consent Form. A Plain Language Statement would explain in simple language what would be required of the participant during the session. An Informed Consent Form would ask the participant if they understood the Plain Language Statement and if they understood that their consent can be withdrawn if they so wish and that the session is not mandatory. This is done for ethical reasons, so the participant is aware of what the session will entail and what will become of the data they give during the session, if any.

Participants would have been presented with the web application and a set of tasks to accomplish; an example being *completing the Algebra test*. Then, upon completing these tasks, each participant would have been presented with an anonymous survey, which would ask questions about the performance of the application and how it could be improved. This information would then be taken into account for the refactoring stage of the project.

4.4 Summary

To conclude, this chapter covers the arduous task of creating the web application for this project. Beginning with the Client-Side, pages were built using HTML, CSS, and JavaScript, with some MathJax to render the equations. Next, the Server-Side was developed using server components and a SQL database.

Chapter 5 - Results and Discussion

In this chapter, the results of the project will be discussed, both from the outcome of the testing and personal thoughts. Then, there will be an overall discussion of the project which shall include changes to be made and future work.

5.1 Results

The results of this project have produced a fully functional web application which completes its main goal: to help students self-diagnose their mathematical ability.

The website consists of three main sections: the home page, the test sections, and lecturer information. The home page is self-explanatory and details what is the purpose of the website. The lecturer information gives the contact information and a background about this project's supervisor, Dr Conor Brennan. The test sections comprise of eight sections of various topics in mathematics to be addressed.

Each section comprises of a number of tests. Each test contains a set number of questions to be solved by the student. All of the questions are contained on a single page, for the convenience of the student. If a question is answered correctly, it notifies the question and directs them to the next question. However, should they answer incorrectly, a video appears which details what possible error was made in completing the question. At the end of the test, there is a button that directs the student to a Google Drive folder. Here, the students are prompted to upload their workings should they have problems with a certain question or if they have found another answer. Overall, this project is deemed a success as it completes the goal that was set out at the beginning of this project.

5.2 Discussion

5.2.1 Changes to be Made

The following are changes that would be made to the project where another week was allotted:

The first change is an aesthetic change to the website, and that is the loading time of MathJax. Currently, it takes a second or two to load and, in that time, the user can see the

MathJax script on the website. The cause of this is unknown, but it would be found and solved.

The second change is to finish the Selenium testing mentioned in Section 4.3.1. The necessary research work into Selenium is completed, so it would be a matter of writing and implementing the tests, as well as some refactoring to accommodate the tests.

The final change would be to incorporate the database into the project. Currently, the questions and answers are "hard coded", meaning manually inserted, into the JavaScript files. All of the questions and answers for each section are already located in the database, it would only be a matter of connecting the application to the database.

5.2.2 Further Work

As discussed in Section 4.3.2, the User Testing aspect of this project was moved into further work due to the COVID-19 outbreak. However, the guidelines of how to complete this aspect were outlined, which diminishes the amount of work needed to be done as research has already be done into these areas.

The second task to be completed under further work is the login and registration functions. This would once again use the database and would allow students to keep track of their progress and upload files.

5.3 Summary

Overall, this project was a success as it completed the goals set at the beginning of the project. Although, it can be improved upon by fully connecting the application to the database. It should also be further tested, which was unable to go ahead due to the COVID-19 outbreak. However, this does not impact the overall success of the project.

Chapter 6 – Ethics

In this chapter, the ethics issue that arose in the project will be discussed. The problem itself will be addressed, as well as the solution used to solve the problem.

6.1 Issues and Impact

Since this project is software-based, it will cause no environmental impact nor health and safety issues. In terms of an impact on society, there would be a greater impact. Although this project is aimed to help students self-diagnose and improve their mathematical ability, this could cause issues with students as this application could make them feel vulnerable as their lecturers would be able to view their progress. However, lecturers would be able to see their progress in their continuous assessments if they exist. It is more beneficial for students to recognise their shortcomings before they take exams or do continuous assessment. Thus, this problem solves itself and makes the impact socially relatively low.

6.2 Data Management

6.2.1 The Problem

Within the hypothetical user functions, meaning the log-in and registration pages, the user would input data to access the application. In order to register for the application, the user would have to enter the following details: full name, email, username, and password. These would then be stored in the database and would then allow the user to log into the system. The issue here is the storing of the full name and email. Now the solution to this hypothetical ethical problem will be presented.

6.2.2 The Solution

The users' data would be stored in the database that could not be accessed by other users of the website. If more time was given to this project, proper security measures would be used to encrypt this personal data, such as hashing of the passwords, etc. This is how this ethical problem would be solved.

6.3 User Testing

6.3.1 The Problem

If User Testing had gone ahead as planned, there would have been one problem to address. Since user information was not necessarily required, since a test account could have been used for this purpose, and the most pressing issue would have been the stress undergone by participants. Since the participants of the User Testing would have come a large sample size with varying backgrounds in Mathematics, this means that certain participants could have experienced distress during the testing. This problem would have been solved as such.

6.3.2 The Solution

At the beginning of the User Testing session, the participants would have been presented with a Plain Language Statement and Informed Consent Form. These forms would check that the participant understood what was required of them in the testing session and that they agreed to participate in those tasks. A significant part of the Informed Consent Form is that the participant understands that the participation is not mandatory, and that they can withdraw from the session at any time. This means that if the participant was caused distress by the exercises presented to them, that could withdraw and avoid further distress, thus, solving the ethical problem.

In addition to this, the participants would be aware of what their data was being used for. In terms of answering a short survey after performing exercises, the participants would be aware that their answers would be reviewed to further the project’s development. This eliminates the ethical issue of the anonymous survey after the exercises have been performed.

6.4 Summary

To conclude, the data management and User Testing aspects of the project would have raised ethical issues. They would have been solved by taking as little personal information about the user as possible. In addition to this, in the User Testing phase, the participants would have been aware of their ability to withdraw from the session at any time in an effort to minimise stress on the participants.

Chapter 7 - Conclusions and Further Research

To conclude this report, it should be stated that due to the COVID-19 outbreak this project was not able to be revised to a satisfactory extent, and ideally would have worked the work mentioned in Section 5.2.1, however the project was completed to a satisfactory level. Although the project could be more refined, it is a step in the correct direction for helping lectures identify and correct students' common errors.

This project also can act as a base for lecturer testing whether they are trying to identify common errors, as lots of lectures may not have the resources of time to try and implement their own testing applications. It has been stated by the supervisor of this project, Dr Brennan, that upon revision of this project, he would like to possibly implement it in the coming year, for this first-year students.

If this project were to be redone, it would be more benefitable to start with Java Server Pages to make for easier database connectivity then trying to use Java Servlets. This would also us to implement the database with a login and registration page and build off this base rather than trying to implement a database and login page, after the main parts of the project had been completed. In further web application development, the use of frameworks would be implemented, as they can build upon the basic web development skills learned in this project.

This project was also a way to develop programming ability and interest, as the project itself covered a wide range of available languages and applications in programming. This project helped to direct further career and education options, as it included many areas such as web application and database development.

Overall, this project is deemed a success due to the way it completes the goals set out initially for the project.

References

[1] A. Hansen, D. Drews, J. Dudgeon, F. Lawton, and L. Surtees, *Children's errors in mathematics*, Learning Matters, 2017.

[2] D. Molloy. EE417. Class Lecture, Topic: "Web Application Development." School of Electronic Engineering, Dublin City University, Dublin, Ireland, 2019.

[3] B. Lawson, and R. Sharp, *Introducing HTML5,* New Riders, 2011.

[4] Matthew Tyson, "What is Tomcat? The original Java servlet container", *javaworld.com*. [Online]. Available: https://www.javaworld.com/article/3510460/what-is-apache-tomcat-the-original-java-servlet-container.html [Accessed Mar. 24, 2020].

[5] Eclipse Foundation, "What is Eclipse?", *help.eclipse.org*. [Online]. Available: https://help.eclipse.org/2019-12/index.jsp?topic=%2Forg.eclipse.platform.doc.isv%2Fguide%2Fint_eclipse.htm [Accessed Mar. 24, 2020].

[6] Oracle, "MySQL Workbench", *mysql.com*. [Online]. Available: https://www.mysql.com/products/workbench/ [Accessed Mar. 24, 2020].

[7] W3Schools, "Introduction to HTML", *w3schools.com*. [Online]. Available: https://www.w3schools.com/html/html_intro.asp [Accessed Mar. 24, 2020].

[8] W3Schools, "JavaScript Tutorial", *w3schools.com*. [Online]. Available: https://www.w3schools.com/js/ [Accessed Mar. 24, 2020].

[9] Oracle, "What is a Servlet?", *docs.oracle.com*. [Online]. Available: https://docs.oracle.com/javaee/5/tutorial/doc/bnafe.html [Accessed Mar. 24, 2020].

[10] Oracle, "What is a JSP Page?", *docs.oracle.com*. [Online]. Available: https://docs.oracle.com/javaee/5/tutorial/doc/bnagy.html [Accessed Mar. 24, 2020].

[11] QuinStreet, "What is SQL?", *sqlcourse.com*. [Online]. Available: http://www.sqlcourse.com/intro.html [Accessed Mar. 24, 2020].

[12] NumFOCUS Foundation, "Beautiful math in all browsers", *mathjax.org*. [Online]. Available: https://www.mathjax.org/#about [Accessed Mar. 24, 2020].

[13] N. Roy, and A. McCallum, "Toward optimal active learning through monte carlo estimation of error reduction." ICML, Williamstown, pp. 441-448, 2001.

[14] P. Gates, *Issues in mathematics teaching*, Psychology Press, 2001.

[15] R. Borasi, "Exploring Mathematics through the Analysis of Errors." *For the Learning of Mathematics*, vol. 7, no. 3, pp. 2–8, 1987.

[16] Eclipse, "Creating an Apache Tomcat server and Web project", *Eclipse*, 2019. [Online]. Available: https://help.eclipse.org/kepler/index.jsp?topic=%2Forg.eclipse.jst.ws.axis.ui.doc.user%2Ftopics%2Fttomcatserv.html [Accessed Dec. 1, 2019].

[17] Eclipse, "Configuring the Server", *Eclipse*, 2019. [Online]. Available: https://help.eclipse.org/luna/rtopic/org.eclipse.stardust.docs.wst/html/wst-integration/configuration.html#configServer [Accessed Dec. 1, 2019].

[18] P. Kumar, "Java Web Application Tutorial for Beginners", *journaldev.com*, 2013. [Online]. Available: https://www.journaldev.com/1854/java-web-application-tutorial-for-beginners [Accessed Dec. 1, 2019].

[19] Selenium, "About Selenium", *selenium.dev*. [Online]. Available: https://www.selenium.dev/about/ [Accessed Mar. 24, 2020].