development of a web based MULTIPLE-CHOICE test engine

Charles Walker: #1504185

Supervisor: Nick Zakhleniuk, Second Assessor: Luca Citi

Computer Science

## Abstract

Multiple choice quizzes can be used in numerous ways, the most typical usage is for testing purposes where teachers give students a set of questions to examine their knowledge of a subject area. However, they can also be used for helping develop a student’s understanding of a subject, by giving feedback specific to the answer they gave. Teachers can design this feedback in a way to explain to the student where they went wrong if they answer incorrectly.

The aim of this project was to develop a web based multiple choice test engine, in a way that allows for the testing system to be used as both a testing tool, and an educational tool. By allowing teachers to create their own questions to be used within quizzes they can provide feedback with the answers that can be given to the students from the system, giving students the understanding of why they answered a question correctly or incorrectly.

Table of Contents

[1. Introduction 3](#_Toc512545664)

[1.1 List of Symbols 3](#_Toc512545665)

[1.2. Literature Review 6](#_Toc512545666)

[1.4 Professional Issues Addressed In the Project 8](#_Toc512545667)

[1.4.1 Legal Issues 8](#_Toc512545668)

[1.4.2 Ethical and Social Issues 8](#_Toc512545669)

[1.4.3 Sustainability 8](#_Toc512545670)

[1.5 Aims and Objectives 8](#_Toc512545671)

[1.6 Overview 9](#_Toc512545672)

[2. Requirements and Analysis 10](#_Toc512545673)

[2.1 Specific Requirements 10](#_Toc512545674)

[2.1.1 Functional Requirement – User Registration 10](#_Toc512545675)

[2.1.2 Functional Requirement – User Log In 10](#_Toc512545676)

[2.1.3 Functional Requirement – Join a Class 10](#_Toc512545677)

[2.1.4 Functional Requirement – Create a Question 11](#_Toc512545678)

[2.1.5 Functional Requirement – Deleting and Editing a Question 11](#_Toc512545679)

[2.1.6 Functional Requirement – Generate a Quiz 11](#_Toc512545680)

[2.1.7 Functional Requirement – Recording and Viewing Results 12](#_Toc512545681)

[2.2 Constraints and Assumptions 12](#_Toc512545682)

[2.3 User Types 12](#_Toc512545683)

[3. Development and Design of the Multiple Choice Test Engine 14](#_Toc512545684)

[3.1 Design Overview 14](#_Toc512545685)

[3.1.1 System Architecture 14](#_Toc512545686)

[3.1.2 Design Techniques 14](#_Toc512545687)

[3.3 Database Design 16](#_Toc512545688)

[3.4 User Interface Design 17](#_Toc512545689)

[3.4.1 Login 17](#_Toc512545690)

[3.4.2 Registration 18](#_Toc512545691)

[3.4.3 Confirmation page 19](#_Toc512545692)

[3.4.4 Home Pages 20](#_Toc512545693)

[3.4.5 Create Quiz Page 21](#_Toc512545694)

[3.4.6 Create Question Page 22](#_Toc512545695)

[3.4.7 Taking a Multiple Choice Quiz Page 23](#_Toc512545696)

[4.3.8 Quiz Review Page 25](#_Toc512545697)

[4.3.9 Results View Page 25](#_Toc512545698)

[4. Implementation and Testing 27](#_Toc512545699)

[4.1 Implementation 27](#_Toc512545700)

[4.1.1 Login system implementation 27](#_Toc512545701)

[4.1.2 Registering to the web app 29](#_Toc512545702)

[4.1.3 Implementation of Quiz and Question creation pages 31](#_Toc512545703)

[4.1.4 Quiz implementation 36](#_Toc512545704)

[4.1.5 Implementation of results view 38](#_Toc512545705)

[5. Conclusion 39](#_Toc512545706)

[5.1 Goals Achieved 39](#_Toc512545707)

[5.2 Further Work 40](#_Toc512545708)

[References 42](#_Toc512545709)

# 1. Introduction

## 1.1 List of Symbols

|  |  |
| --- | --- |
| Word | Definition |
| HTML | “HTML is a HyperText Markup Language file format used as the basis of a web page. HTML is a file extension used interchangeably with HTM. HTML is consists of tags surrounded by angle brackets. The HTML tags can be used to define headings, paragraphs, lists, links, quotes, and interactive forms. It can also be used to embed Javascript, and CSS” [1] |
| MySQL | “is an open source relational database management system. It is based on the structure query language (SQL), which is used for adding, removing, and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT, and UPDATE can be used with MySQL.” [2] |
| CSS | “Stands for "Cascading Style Sheet." Cascading style sheets are used to format the layout of [Web pages](https://techterms.com/definition/webpage). They can be used to define text styles, table sizes, and other aspects of Web pages” [3] |
| MCQ (Multiple Choice Test) | “Multiple choice is a form of an objective assessment in which respondents are asked to select only correct answers out of the choices from a list.” [4] |
| 1NF (First Normal Form) | “The First normal form (1NF) sets basic rules for an organized database −  Define the data items required, because they become the columns in a table.  Place the related data items in a table.  Ensure that there are no repeating groups of data.  Ensure that there is a primary key.” [5] |
| 2NF (Second Normal Form) | “The Second Normal Form states that it should meet all the rules for 1NF and there must be no partial dependences of any of the columns on the primary key” [6] |
| 3NF (Third Normal Form) | “A table is in a third normal form when the following conditions are met −  It is in second normal form.  All nonprimary fields are dependent on the primary key.” [7] |
| JDBC (Java Database Connectivity) | “Java Database Connectivity (**JDBC**) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is Java based data access technology and used for Java database connectivity. [8]” |
| SQL Injection | “SQL injection usually occurs when you ask a user for input, like their username/userid, and instead of a name/id, the user gives you an SQL statement that you will **unknowingly** run on your database.” [9] |

## 1.2. Literature Review

Written tests were first thought up by Horace Mann in 1845. The goal of these tests was to give every child the same opportunity. The idea didn’t take off at first, and oral exams continued to be used. Yet in more modern times, they have taken over and are now the key part of the education system. Many initial standardized tests were for testing mental abilities by the Army. These weren’t adapted into a format of testing within schools until later. At that time, it was used to identify more intelligent children and for entering College [10].

The US started to use a more standard testing system to test teaching and schools as well as students. Over time it has become more important for schools to be able to produce educated students with skills that can contribute economically. But this form of testing has received criticism. On whether it helps or hurts students who are disadvantaged economically and socially. Most tests however are still available to people from all backgrounds. The argument against that is that people from wealthier situations can access more resources to learn than others [10].

Testing however doesn’t just have to be about measuring a student’s knowledge. There are several advantages in using multiple-choice testing as a learning tool.

It's been identified that providing an initial test that is difficult is useful to help with learning. In that it helps a student to connect any prior knowledge. It also provides the key information for the teaching that lies ahead in a more memorable way. This is because guessing engages the mind differently than trying to memorise something that has been read. In this case it is being used as an educational tool and the outcome of the quiz itself is not important [11].

If used routinely and somewhat frequently, tests are a good way to teach students and increase the amount of information they retain from a class. If something is only learnt once it is more likely to be forgotten. So revisiting a piece of information decreases the likelihood of that happening. As well as this, repetition will improve the memory of the information learnt. Simply rereading text is a very ineffective way of achieving this, and more interactive methods such as tests are much better [11].

There are several reasons for using multiple-choice testing over testing in other formats when both being used as an educational tool and as a method of testing knowledge.

Writing a multiple-choice quiz question is difficult. Due to having to make sure that a question tests a deep level of understanding. That means that the distractors (the incorrect answers), focus on the most common misconceptions of the questions topic. This is what allows for a student to learn or to prove that they have grasped an understanding of a topic [12].

One of the advantages of multiple-choice tests is that they provide a more reliable form of assessment. There is no subjectivity when marking a multiple-choice test. There are only right or wrong answers. Whereas with a written exam they will be marked depending on the examiners interpretation of both the answers, and the suggested marking scheme as questions may not have strict answers [12].

Another benefit of multiple-choice testing, is that they able to be marked instantaneously as it’s possible to mark them automatically with software. This allows schools to use their resources on educating students as teachers spend less time marking. Even if these quizzes are not marked automatically it is far quicker for a teacher to check through a multiple-choice quiz than it is for them to read an essay. As they then have to consider it against a set marking scheme. Due to the objectivity of the answers, statistics are easy to generate and precise. If everyone in a class gets a question wrong it’s clear what needs to be retaught, this does not occur in the case of essay style questioning [12].

There are some disadvantages to multiple-choice testing. Particularly when they are conducted online. As it gives the potential for a student to search for the answer online, or use other tools to find the answer. Thus not learning anything from the test, there are ways to prevent this such as giving each question a fixed time limit. Another potential issue is that they don’t test the depth of knowledge someone has of a subject area. When compared to a test with written questions, which are better for testing the breadth of knowledge.

Cohen and Sasson have found that there is a correlating pattern between quiz grades and final exam grades. This shows that using online multiple-choice quizzes, as a tool for learning and testing is an effective way to engage and help students. As they allow for the student to use them whenever is convenient for them [13].

They also found that giving students more than one attempt at a quiz allows them to improve their grade significantly. This is mostly described as being due to confidence, as there is less fear of failure. It does not necessarily mean that they are using the tool to improve their knowledge and perform better. In their study 72% of students attempted the quiz many times and understood that by being able to take the quiz again they would be able to learn from the first attempt and improve upon that attempt the second time. By having tests online, and results being saved within the database, teachers can see the attempts at a quiz. This can help the teachers with focusing on the areas of study as well as identifying students who may be struggling [13].

When creating a test, it is also important to consider if there is a difference between online testing and paper written tests. As it is possible that not all students have the same level of understanding when using technology.

In 2001, Princeton performed some tests on 4th and 8th grade students. These were to compare students taking paper tests with those taking the test online. They found that the scores were slightly lower for those taking the online test over the paper-based test. The researchers also checked the student’s general ability to use a computer, and found that those who were more proficient generally scored better on the test [14].

Research into multiple-choice testing tends to agree that it can be a useful way of testing students. But it is improved upon by mixing the questions with written questions to ensure student knowledge. It has however been shown from these articles that they are effective in helping students to learn, and can be considered a very useful teaching tool for teachers and students.

## 1.4 Professional Issues Addressed In the Project

This subsection covers the legal issues, ethical and social issues and sustainability issues present in the project, and discusses any issues that are relevant.

### 1.4.1 Legal Issues

There are no legal issues during the development of the software, any software used was used with the appropriate licenses. All of the data contained in the database was generated by me and is completely fictional. The software was developed for educational purposes and there are no intentions to distribute the software commercially at any point.

### 1.4.2 Ethical and Social Issues

The application does not use any third party code, and no ethical issues are relevant. The system design is intended to be as user friendly as possible in order to allow for the system to be accessible to anyone.

### 1.4.3 Sustainability

The tomcat server on which the application runs is part of an open source software. As is the MySQL database system. The IntelliJ environment that was used to develop the software is not open source. Yet other software would be available to maintain the system.

## 1.5 Aims and Objectives

This original objective for this project was to develop a “Web based multiple choice test engine”, implying that a system that would allow for users to create multiple choice quizzes online. Also for those quizzes to then be accessible for other users to answer. Both the creators of the tests, and the users who take the tests should be able to have the ability to check either; the results of students who have taken their test, or the tests that they have completed.

Anyone is able to access the site and register. Using any email address, to register as a teacher an email address must be a valid academic email ending in '.ac.uk'. Once logged in teachers can create individual questions or create an entire quiz. An entire quiz consists of creating a series of questions. These are stored in the database, with each question linking to the possible answers provided. Students on the other hand have the option to take a test, either one that was predetermined and created as a whole test by a teacher. Or the students can take a random test, to quiz them on a variety of questions.

The main objective for this software however was to allow for the tests to be used as an educational tool. By allowing teachers to match answers with a comment. The comment is intended to be used to explain why the given answer is correct or incorrect. However it can also be ignored, and therefore just a simple test with no feedback can be produced. The feedback is given to a student after they have taken a quiz. At that point they receive their final mark, and a summary of each question. It determines if it was correct or incorrect, and the relevant comment depending on their given answer.

## 1.6 Overview

The remainder of this document is split into the individual stages of the development cycle. The next chapter will focus on the Analysis and Requirements, and this will elaborate on the aims and objectives that have been outlined above. It will also provide a breakdown of these aims and how they were achieved, as well as any other methods that were considered but ultimately not used.

Chapter 3 of this document will cover the design of the system. It will outline the technique chosen for designing the system, and explain how the analysis was used to make design decisions when choosing how to implement the requirements into the system.

Implementation and Testing will be covered in Chapter 4. This will show any key code features, and how each of the requirements were implemented. As well as how it was tested to show it functioning correctly. It will also discuss how effective the implementation of each requirement is.

The final chapter, will feature the conclusions. This will describe how well the original objectives of the project have been met, and how these were achieved. As well as discussing new areas that could be investigated or added to the project. As well as parts of the current work that are not completed fully due to time constraints or other problems.

After the final chapter a list of references can be found, as well as any appendices or figures mentioned in the chapters that follow.

# 2. Requirements and Analysis

The following section will cover in more detail the objectives of the multiple-choice quiz engine system. Breaking the system down into smaller steps, and analysing the possible ways these were considered. It will also consider the constraints on the system design and requirement gathering, and discuss the user types that will be present in the work.

## 2.1 Specific Requirements

This, the specific requirements section, is a detailed breakdown of each key functionality that was necessary for the multiple-choice quiz engine to function correctly. To achieve the main project goals. There is also initial discussion about possible design and implementation ideas.

### 2.1.1 Functional Requirement – User Registration

The purpose of this requirement was to allow for a user to register with the website, by providing an email and password. This was an important first step as it allows for users to have individuality and provides with a way to identify uniquely within the database. For teachers, this means that they are recorded as the creator of a quiz, and as such it could be assigned to their students. For students, it means that they would be able to take a test created by a teacher, as well as being able to see their own personal results.

To ensure that registration was as simple as possible and make the site usable for students of all ages. There is limited information required, just a username, password and user type are needed.

### 2.1.2 Functional Requirement – User Log In

This requirement applies once a user has already registered, who can by using their chosen email access the website. By logging in a user is only able to access relevant parts of the site. Depending on their user type, this prevents students being able to create quizzes for example.

The best way to do this is to search the database for the email the user has tried to login with, and then retrieve the password when a match has been found. Then ensure that the password and email addresses match. When accessing passwords it is important to ensure they’re not accessed when they are not required. In case they can be extracted in some way, using this method of checking login prevents this.

### 2.1.3 Functional Requirement – Join a Class

The reason for including this objective is that in order for a teacher to set a quiz for their students, the students need to be grouped together in some way. The other possible way to do this would be to allow an admin user to create groups. When considering how to design this requirement that method was also considered. But to allow the students to join a class provided a more flexible way of doing things for the users, as some users may not be or may not need to be part of a class.

### 2.1.4 Functional Requirement – Create a Question

When considering the best way in which to add a quiz, I decided that the best way in which to do that, would be to allow a teacher to decide on all the quiz details. Such as the number of questions and the title, and then for them to be able to create each question individually. The first problem was that allowing an entire quiz to be created within one web page was going to require too much space, while this approach would still have been possible.

The problem was then how to break down the creation of a quiz. In the end I decided that allowing a quiz to be added as a series of single questions, would mean that I was able to also produce questions that could be used in other ways. I then decided that I would create randomly generated quizzes. These were for students wishing to use the tool to learn and that do not have a specific quiz they should be taking.

### 2.1.5 Functional Requirement – Deleting and Editing a Question

When a teacher creates a question, I felt it was important that once a question is created, it was important they could edit or remove it. There were many ways that this could be done, it was either to allow a user to only access their own questions. Which would require storing information about the user with each created question. Or to allow a teacher to access all questions, as there are no admin users, the best way would be to only allow a teacher to access their own questions.

Doing so removes the potential for editing and deletion of questions that a user should not be able to remove or edit. Having an admin user would have helped in order for changes to be able to be reversed, but doing so was not essential.

### 2.1.6 Functional Requirement – Generate a Quiz

The main functionality besides being able to create a quiz, is for a student to be able to take a quiz. There are two ways that had to be possible, either taking a quiz that had been set by a teacher. Or to take a random quiz generated from all the available questions.

This is the most important aim of the project, and should also be relatively simple for a user to access, as user accessibility is a very important aspect. There were numerous ways that the quizzes could be implemented. One way that was considered was to have each question asked one after another, and once answered the user is forwarded onto the next question. The other option was to simply provide a series of questions on one page with radio buttons, this allows the user more freedom. As they can go back and change their answers, or skip a question they find difficult and answer the rest of the quiz first.

Due to the main focus being the use of the multiple-choice quiz as an educational tool the second option is more effective. Due to there being more time for a user to think and consider the options, rather than having any sort of time pressure on a question.

### 2.1.7 Functional Requirement – Recording and Viewing Results

When considering the recording of results, I had to consider the amount of data that would have to be stored. As having a huge amount of data in the database would then increase retrieval times massively. The easiest way was to break the problem down into two parts, one part where the student could view their results from a test, and another part where the result was recorded and could be retrieved at a later point.

The main focus when considering how this requirement would work, was the database. It was also important who would be able to see which results, as if someone wanted to see every test result this could eventually reach a point where it takes a very long time to get and display all of those results.

## 2.2 Constraints and Assumptions

There were two main constraints to consider when designing and implementing the software. The internet speed of the user, and the size of the database. I could only directly control one of these,but in order to ensure the internet speed did not affect users too much. I kept the website design basic as to use lesser resources and as such not put too much strain on the user’s connection.

The database size I was able to control, it needed to be large enough to contain all key information. While not containing excess information as once data starts to enter a database it can become too large. This means querying takes a considerable amount of time, which would be noticeable when retrieving data for a webpage. As such I cut down the amount of personal data required to register to the website. As it seemed less important than keeping more details regarding quizzes and questions, where I kept as much data as possible.

The only assumption made in this project is that all users have some basic level of computer knowledge and experience using webpages, as things such as the navigation are kept simple.

## 2.3 User Types

There are two basic user types, both have slightly different functionality within the website. Teachers and students both have the basic level of registration and login features. It is the test related features they can access that differ.

Teachers can only access the creation of quizzes and questions, and also they can view results of quizzes taken by their students. The main feature that they won't be able to access is taking part in the quizzes. If a teacher were to want to test a quiz they had created they would have to create a student account, which can be registered to from any email address.

Students on the other hand can't access the pages for creating quizzes or questions, and the main feature available to them is the ability to participate in a quiz. They also have a quiz review page which shows their results to a quiz they have just taken. This is not a feature teachers can access as it can only be accessed after taking a quiz. Both users are able to view quiz results, although at different levels, for students that means only their own results are visible.

# 3. Development and Design of the Multiple Choice Test Engine

This chapter will cover the design of the system. From the base architecture to the design of the user interface. It will also discuss the design techniques used, and describe the design of the requirements mentioned in the previous chapter. Finally it will cover the database design and structure.

## 3.1 Design Overview

This subsection of the report covers a general overview of the system. From the beginning of the design process and it’s refinement down to a final design. It will mostly cover system architecture, as well as interface and component design.

### 3.1.1 System Architecture

This section will describe an overview of the interfaces in the system, and the interactions they have together. As well as the interactions these provide for the users and the interactions with the database.

The first interface a user is presented with when reaching the webpage is the login interface. This page is where the login and registration sections of the website can be accessed from. While presented with this interface a user cannot access any other of the features of the site until they have performed a valid login action. Depending on their personal user type they will be taken to one of two home pages. These are identical in functionality but feature different navigational features.

These are the user home pages, for both users these pages simply provide a welcome to the site. As well as two navigational panels that allow them to navigate the different parts of the site. For each user the pages that they can access from the navigational panels is different based on their type. For example a student has the option to take a test, whereas a teacher does not.

The pages that can be accessed from the homepage all essentially work the same. With the data being displayed in place of the blank homepage depending on which feature of the site is being accessed.

The final interface is the database, which contains all of the important system data. The other interfaces are all able to communicate with the database using Java. Allowing them to send and retrieve any data that is necessary to be displayed in the site or to be stored within the database.

### 3.1.2 Design Techniques

When designing software to meet a specific set of requirements, it is important to study and understand the problem that exists, and to be able to identify the features that would need to exist in any possible solution. Finally a description of each idea to be used in the system needs to be produced in order for it to be possible to implement the idea.

The basic level of design was the server/client relationship, I chose to use Apache Tomcat as the server as I had experience working with it in the past. Tomcat is a servlet container, this uses Java to dynamically generate a web page on the server. It does this when a HTTP request is made to a web server. That request is forwarded to the servlet container, after the servlet is processed the web server can return the dynamically generated results as a HTTP response.

Figure 1 below shows the Java web application deployment architecture. The HTTP Server displays some content as well as serving as a step between the dynamic content which is served by the Web Container as this is where the Java files are processed. It is the web container that does all of the communication with the database to create the dynamic content.

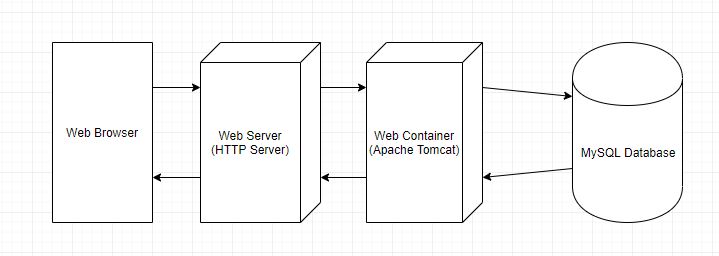


Figure 1: Application Deployment Architecture

The client and server both oversee different sub systems. The client primarily deals with user input. Within the client there will be systems to deal with the connection communicating with the server and the visual displaying of the webpage. The server would deal with the communication between itself and the database. As the data needs to be accessed during the processing of the servlet created from the HTTP request.

After the subsystems have been designed, the design of its interface was created. This is the functionality that the user can access in the case of the client. Whereas for the server the interface design was all of the database accessing functions, and other communication.

The final stage of design is to design the data structures that will be used to store any data in the program. In this case it works slightly differently as the majority of data is stored in a database. Other data in this system is stored using a Java HttpSession interface. This interface allows for a user to be identified across multiple page requests by storing information about the user.

The HttpSession interface allows the servlet to set and view information about the session. In this application the main used methods of this interface are setAttribute and getAttribute. setAttribute binds an object to the session the parameters for this method are name and object, if a name is specified that already has an object then it is replaced. getAttribute returns the object bound with the name specified as a parameter of the method.

## 3.3 Database Design

The first step in designing any database is to first define the purpose of the database and gather the requirements for it. For my database the purpose is to store data relating to the creation and participation of multiple-choice tests. It is required to hold any user data and all data that is important to the quiz.

Once that had been decided upon, the next step was to gather the data that would need to be stored in the database. To do this I considered all the key details that are found in a multiple-choice quiz. Then I also decided what is important from the point of view of logging in and out of a basic website. This data was then divided into subject-based tables. For example, the login information a user needed was stored in one table, and key details about a quiz was stored in another. The first idea was that this would contain things such as the quiz title, the number of questions and the author of the quiz.

Having gathered the data into structured and organised tables, a primary key (a value that uniquely identifies a row) needs to be assigned. This can simply be an id assigned to a quiz question, or an email address of a user. The need for a primary key is because a table cannot contain duplicate rows as it would cause issues during retrieval from the database. However a primary key should be something that will never change. So for example if using the users email address to uniquely identify them, if a user can change email addresses this could cause issues with retrieval.

The next step in creating a database is to ensure that the tables have relationships between one another. As a series of single tables is less effective and the main reason for using a relational database is the ability to define relationships between the tables. One type of relationship is “one to many”. This is the most commonly found relationship type as it is the first step in ensuring a database is in first normal form (1NF). A one to many relationship is when one item in one table links with many items in another. In my database this occurs a number of times one of which is the relationship between quizzes and questions, one quiz has many questions.

Many to many relationships do not directly occur in the database they are instead implemented as two one to many relationships, with a table between them. This is to reduce extra data being stored, the database contains one of these with each user having many results for many quizzes. But each result only applies to one quiz thus the two one to many relationships present.

The other form of relationship is one to one, however these are rarely used. Normally only in cases such as securing sensitive data or if there is a limit to the number of columns in a table. Neither of these applied to this project due to MySQL allowing for a password datatype meaning any sensitive data was already protected appropriately.

At this point the final stage is to refine the design and ensure the database is normalised. The normalisation rules check the structure of the database to ensure that it is optimised.

First normal form is where every cell contains a single value and not a number of values. This is why the quiz questions and quiz data are separated, this is ensured by first normal form. As without it storing the entire list of questions within the quiz table could have been done, and this would’ve caused problems when retrieving questions to display. As once the results had been gathered then I would have needed to separate the questions out. As well as gather the possible answers which would have been far more difficult, instead of keeping them in a separate table and linking them to the corresponding question.

Second normal form is where the non key attributes of a table are fully dependant on the primary key. For example had the database been designed in a different way in which the students were stored and their teachers were stored in the same table then this would need to be normalised. The result of this normalisation would be that a student’s table and a teachers table would be produced. Thus there would be a link between them. That way the teachers name or other information would not be repeated unnecessarily in the students table.

## 3.4 User Interface Design

This section will present the initial design ideas and thoughts that went into the basic graphical user interface layouts of the web application. More discussion of their implementation will be presented in the chapter following this section. The web applications interface was designed using HTML5 and CSS. As these provided the best tools for presenting a useable and attractive site.

### 3.4.1 Login

This section describes the user login page. This is the first page that a user is presented with when accessing the site. As logging in must occur before gaining access to any other features. There is also a link for unregistered users to register to the site as it’s not possible to login without registering first.

The main feature of the page is the login form. Here the user must provide their email address and password they have used to register to the site. This is in order to be authorised to access the rest of the web application as it is at this stage that their user type is determined.

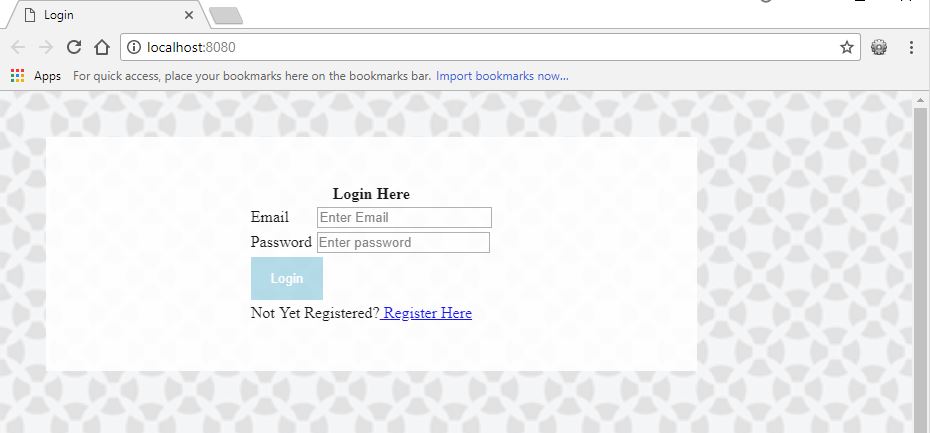


Figure 2: Login Page

Purpose:

* Starting point for the web app
* Contains the login form
* An option to register for unregistered users

Navigation:

* To login the user must enter valid credentials
* The user can opt to navigate to the registration page

### 3.4.2 Registration

The registration page is the first page a new user will want to access on the site. Having navigated there from the login page, they are presented with a form on which they are asked to enter their name, email address, and a password. There is also a drop down menu to select a user type, either a teacher or a student. This will determine feature access later when they log in. The different features available were outlined in section 2.3 of this document.

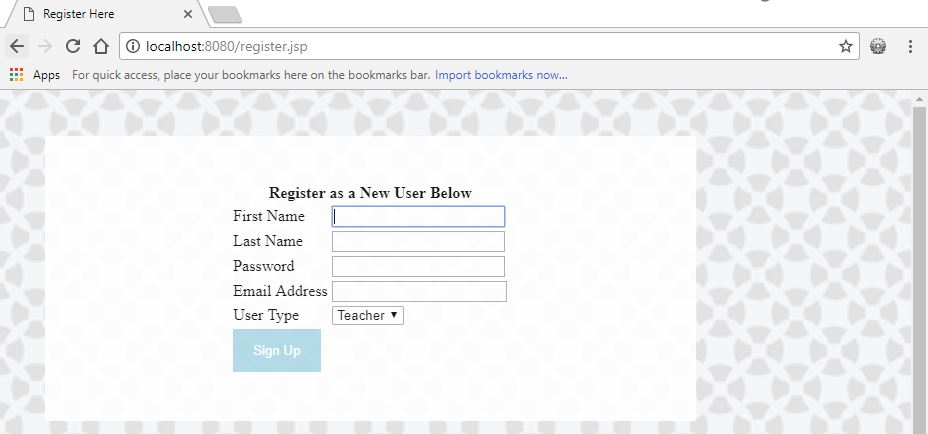


Figure 3: Registration Page

Purpose:

* New users can register for the site
* Provides users an ability to set their user type

Navigation:

* Once registered a user is taken to a confirmation page
* After the confirmation they are able to login and access the rest of the site.

### 3.4.3 Confirmation page

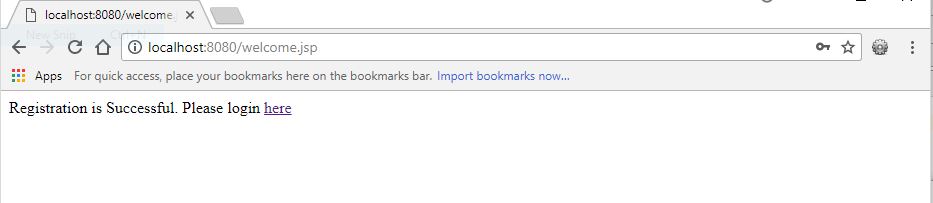


Figure 4: Confirmation Page

Purpose:

* Provide confirmation to the user that they have registered
* Allow navigation back to the login page

Navigation:

* Links back to the login page
* Once at the login page they can sign in and access the rest of the site.

### 3.4.4 Home Pages

This page presented in figure 5 is the landing page after a successful login by a teacher user. It is essentially identical to the student home page aside from the navigational functions available. The initial design of this page did not provide any information, however the final implementation features a welcome message to users.

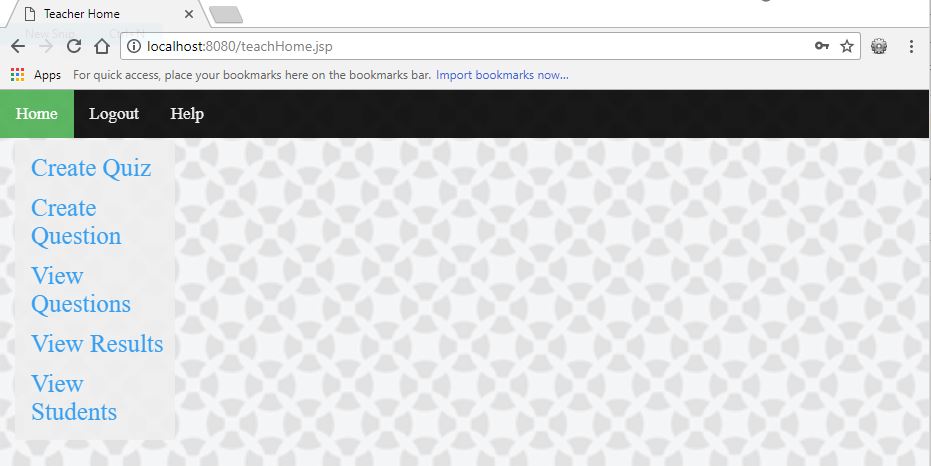


Figure 5: Teacher Home Page

Purpose:

* Provide an initial welcome when a user logs in
* Allow for navigation to the main features of a website

Navigation:

* Each of the links allows the user access to a certain functionality of the site these are as follows:
  + Creating a quiz; the user is taken to the quiz creation form which lets them enter key quiz details before creating the questions.
  + Creating a question; the user is taken to the question creation form. Which allows them to create a singular question that will occur in random quizzes.
  + View questions; the user is able to see a table of questions they have created.
  + View results; a teacher user is able to see the results of every student.
  + View students; the user would be able to view a list of students registered to the site.
  + Logout; provides a way of logging out of the website and takes them back to the login page.
  + Help; directs the user to the help pages.

#### Student Home Page

The idea behind the student home page is essentially the same, as is the layout. It has the same initial welcome message, and the only difference is the navigational options it provides.

Navigation:

* Taking a quiz; the student is able to take a randomly generated quiz using questions from the database.
* View results; the user is able to view their own results only.
* Join a class; the user is able to join their teacher’s class from which a teacher can assign them a test.

### 3.4.5 Create Quiz Page

This page is used to provide one of the key features that a teacher user has access to, the ability to create a quiz. The form is kept simple and does not request the user to initially provide any questions to ensure it is easily useable. The user is asked for a quiz name, a quiz topic, and a number of questions they wish to add to the quiz. The submit button when clicked will ensure that data is present in all fields and of the correct type. Then provide a message either confirming that the quiz has been added to the database or informing the user that their input is invalid.

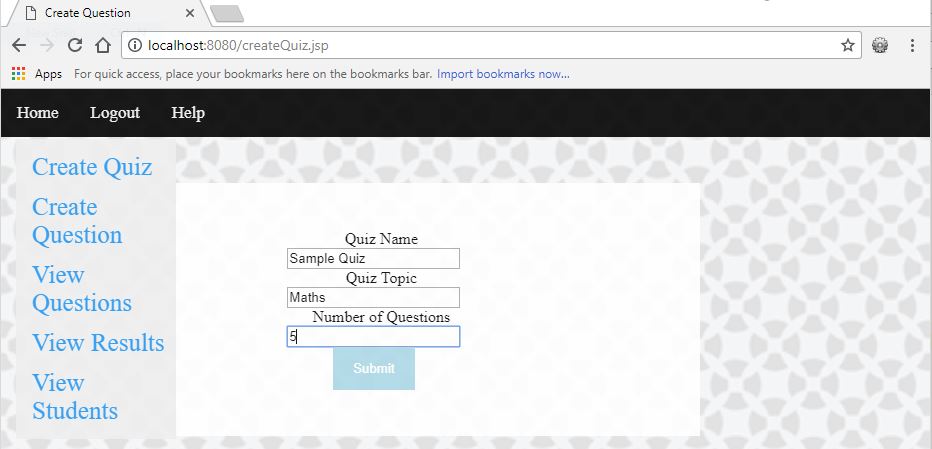


Figure 6: Creating a Quiz page

Purpose:

* Allow a teacher to create a quiz in the database and determine how many questions will be added to it.

Navigation:

* The user can navigate to all of the site functionalities present on the home page. As well as being able to navigate back to the homepage.
* Upon clicking submit, and providing their input is valid, the user will be prompted to create the questions for the newly created quiz by accessing the create question feature from the menu provided.

### 3.4.6 Create Question Page

After a user has created a test it is important that they are able to add questions to it. This feature is provided on the question creation page which uses a dynamic form to allow for the teacher to provide data for the question. The user is asked to provide the question text, and a number of options initially. Once a valid number has been entered, the option boxes are loaded into the form, in the example in figure 7, three option boxes have been provided as requested. The same number of comment boxes are provided- one to match with each option. The user is also asked to provide a correct answer, this must match a number of one of the options, such as in figure 7 it must be an integer between 1 and 3. The submit button provides validation, as well as navigation to a confirmation page. Which optionally then takes the user back to the home page, or back to this page.

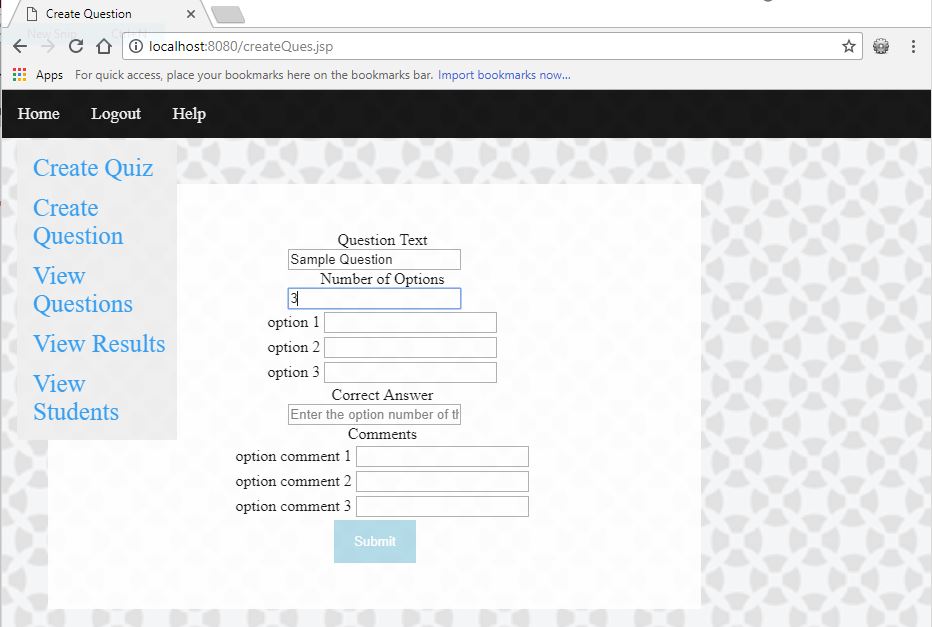


Figure 7: Creating a Question page

Purpose:

* Provide the teacher a form for which to create a question
* Check and advise the user of any errors or potential errors. A potential error would be something such as not adding comments as these are not necessary but recommended. Whereas not providing a correct answer would give a definite error.

Navigation:

* The main menu navigation is consistent with other pages in the web application for a teacher users.
* Upon clicking submit the user is taken to a confirmation page informing them the question is added to the database. From where they are then given the option of navigating back to the question creation page to create a new question, or back to their home page.

### 3.4.7 Taking a Multiple Choice Quiz Page

The main feature for a student is the ability for them to take a quiz. The quizzes are intended to be laid out in a simple way in order for the site to be useable for a wide range of users. Here the user is presented with a form on which the questions are presented together. They are presented with groups of radio buttons providing the options to answer each question. Having selected an option using the radio buttons for each question, the user is expected to click submit which will see them proceed to the quiz review page.

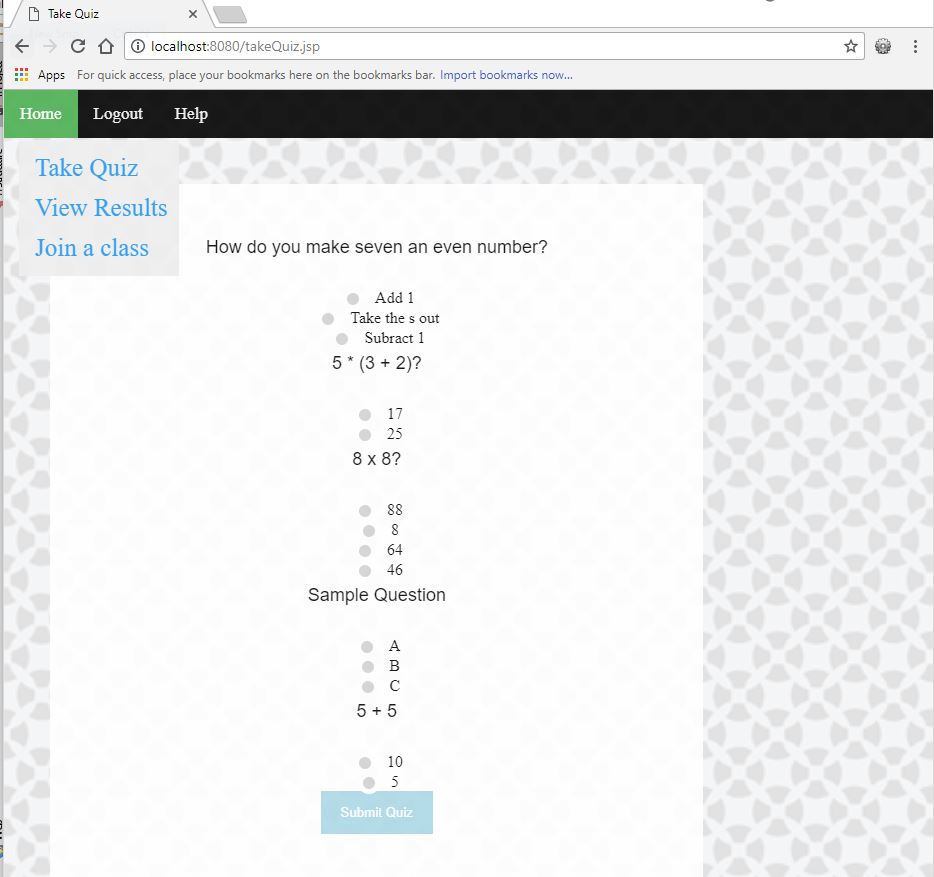


Figure 8: Taking a Quiz page

Purpose:

* Allow a student to answer questions from a random quiz.

Navigation:

* Navigation to all of the main features found on the students home page.
* Clicking submit forwards a user to the quiz review page which displays their results to them.

### 4.3.8 Quiz Review Page

This page follows after a student has completed a test and clicked submit. It displays a table generated using the data they input in the quiz with the results of each individual question as well as an overall result. Figure 9 shows an example of the quiz present in Figure 8. Only the comment is displayed depending on which answer you gave to a question.

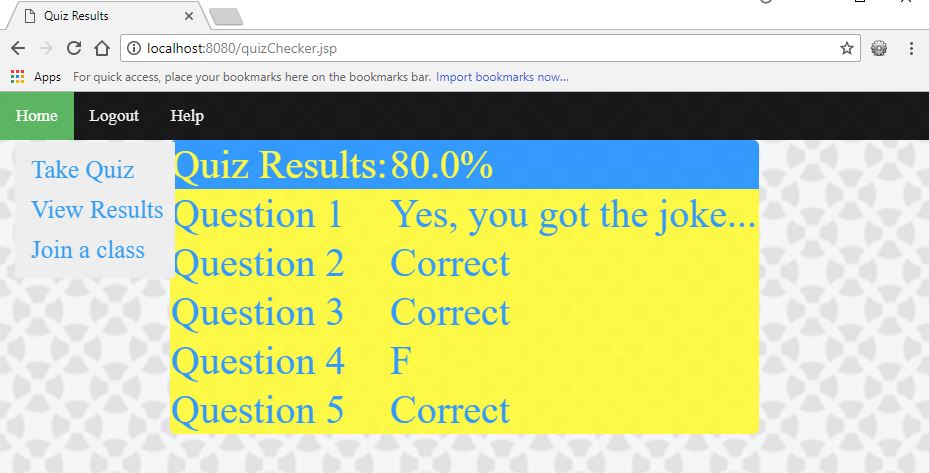


Figure 9: Quiz Review page

Purpose:

* Inform a student of how they performed on the test.
* Allow a student to see the comments relating to the answers they gave to each question.
* Provides a student a way to learn from the test.

Navigation:

* No extra navigation is provided from this page. Once a user is done reviewing their previous quiz they must use the main menu bars to access another feature.

### 4.3.9 Results View Page

This page is essentially identical for both students and teachers. The only difference being that a student can only view their own results, and as such it’s not necessary to display the student’s name. Figure 10 provides an example of the results table, any randomly generated quizzes were returned as an “Unnamed Quiz” entry in the table. For a teacher the table is slightly different, in that every quiz result is presented. Alongside each of the results the students name is also present. This is to provide a teacher with a way of checking the progress of all of their students. Or by checking the quizzes they had created as these could be used by other students as a learning tool.

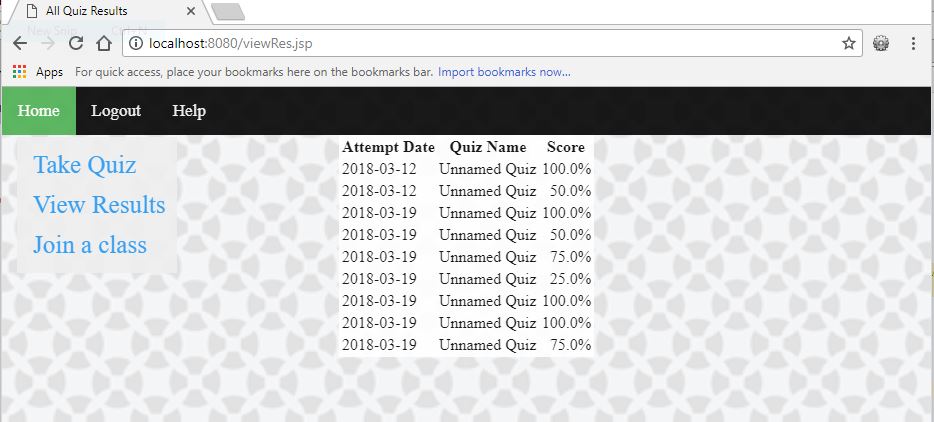


Figure 10: Results View page for a Student

Purpose:

* Allow a student to view their past results
* Allow a teacher to view their students results or results for their quizzes

Navigation:

* In both cases the user can only access features that are present in the main menus.

# 4. Implementation and Testing

Chapter 4 of the report will overview and detail the implementation of the system. This includes any coding issues that occurred and the solutions to those issues. It will also explain how the final system was tested, against the requirements set out in the requirements and analysis stage. The implementation should also show any key features or unique features and how they were achieved.

## 4.1 Implementation

This subsection will break down each of the sites functionalities and how they were produced, as well as any technical challenges encountered in doing so. Full code can be found at the provided link [15].

### 4.1.1 Login system implementation

A teacher or student can login to their personal account using their password and email address they provided during the registration process. The password is masked in order to ensure user security. In order to validate the logging in process the user must click a login button at which point the two fields are checked for within the database. Both of the fields are mandatory.

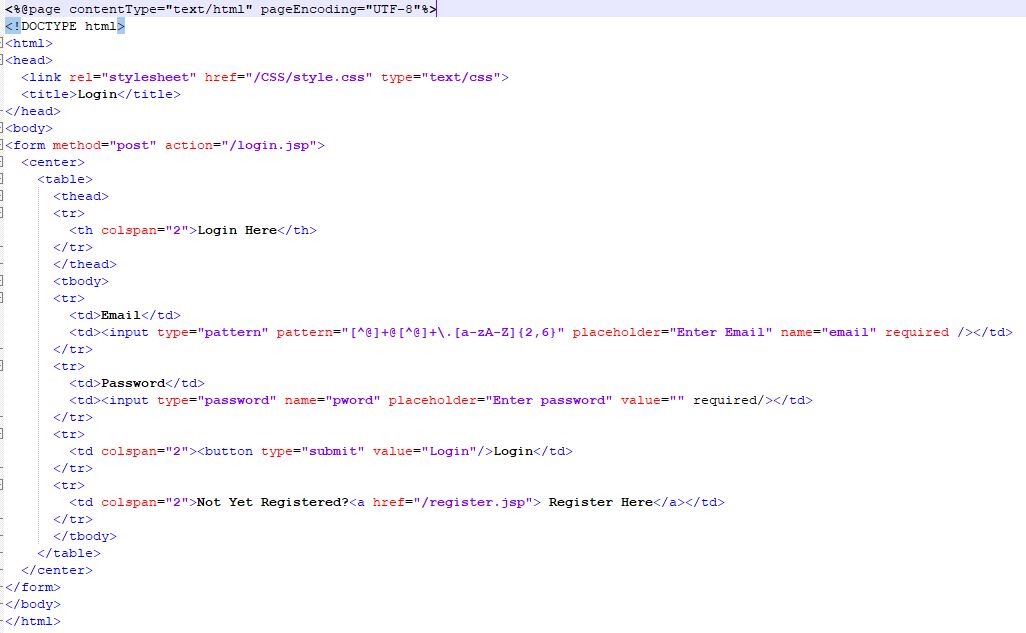


Figure 11: Login page code

As seen in figure 11 the login page contains a form within which there is a table. This is to keep the fields aligned, each row is then a simple text item, followed by an input box. Both of which are formatted, the email address must follow the pattern given in the regular expression. This ensures that it is a series of letters and numbers followed by a “@” and then some sort of “.com” or similar.

Giving input the type “password” provides a way for the user to privately and securely enter a password. Each character is replaced by a dot character. The button is given the type submit, this submits the data entered into the form to the server. In this case it is passed to the file provided in the action attribute of the form which is “/login.jsp”.

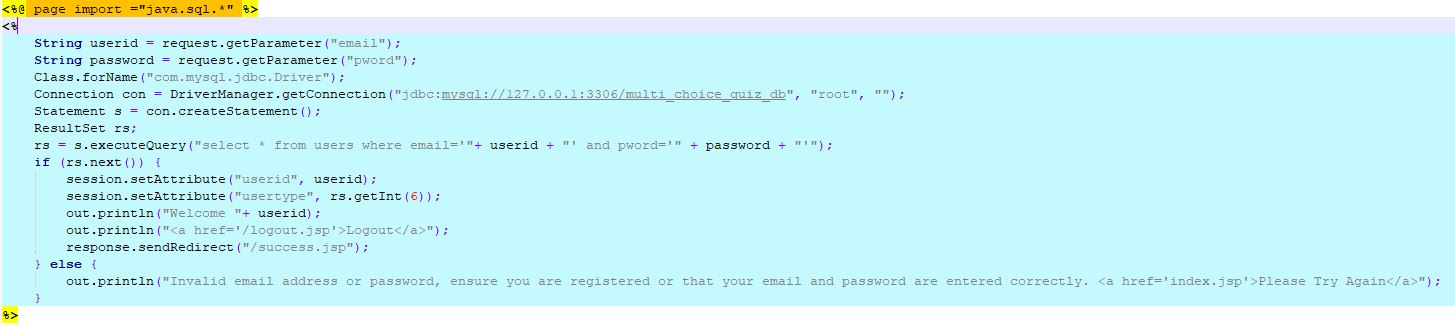


Figure 12: Java code handling logging in

This is a simple JSP page consisting only of Java code, which processes the login request. The first stage is simply getting two values from the form that was just submitted. These are the email and password the user entered, these are stored in local variables to ensure they’re not change on a refresh or a new request. The next step is to establish a connection to the database. This is done using the Java Database Connectivity (JDBC), which is an API that defines how a client may access a database.

Once a connection to the database has been established, a ResultSet is created. This is an interface for representing tables of data representing a database result set. The query where the database is searched for using MySQL structure and the data provided by the user on the login page is then executed. This returns a result set, the ResultSet object has a pointer that points to its current row of data. Therefore it is necessary to check if ResultSet has a next() value, in this case as emails are unique only one or zero items could be returned from the query.

If a user has been found in the database the session object, which is unique for each user sets the user a unique id, by assigning their username to the session attribute “userid”. Doing this allows each valid session to be uniquely identified. The other thing that happens at this point is the user type must be retrieved from the ResultSet. Due to teachers and students have access to different features, it is necessary to set each sessions user type.

The user is then redirected to the “/success.jsp” page. In the case that a user gives incorrect login details, then, because the ResultSet will be empty it will print a message to the user informing them that their details have been input incorrectly. Or that they may not be registered, as well as providing a way to navigate back to the login page.

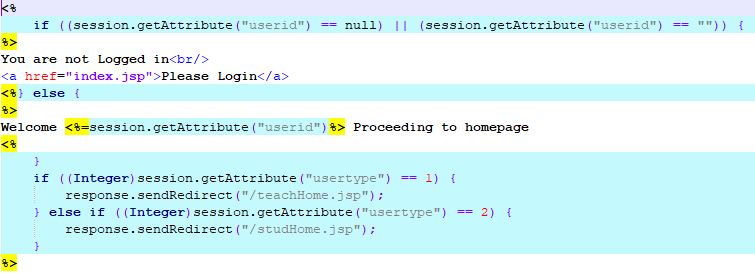


Figure 13: Successful login page code

Upon a successful login, a user will be navigated to the relevant home page by checking what their user type is. This is done using the getAttribute method of the session interface that was set in the previous page. The page also checks that there have been no issues assigning a unique identifier to the session userid. Done by ensuring the value is not null or an empty string.

4.1.2 Registering to the web app

The registration form is very similar in design to the login form described in the previous subsection. The password and email address are given the same type details as before for security and data validation reasons respectively. The users first and last name is able to be any text string. This could potentially lead to students entering joke names or random names. However there was no feasible way to make the site accessible to a range of users.

All of the input sections are marked required, this means that the form cannot be submitted without valid data being present in each section of the form. The user type is also different in that the input is in the form of a dropdown menu. The user can then select to register as either a teacher or a student, and this ensures that no one is able to register as any other user type as they would not be valid when logging in.

Again the registration page passes off to a JSP page of Java that handles the addition of the new user to the database upon the form submission.



Figure 14: Registration form code



Figure 15: Code that processes a registration request

The first name, last name, password, an email address and user type the user submitted are stored in variables. A connection to the database is then established. A prepared statement is used in this case, as it avoids the risk of SQL Injection which is defined in the table found in the introduction of this document. The first value in the SQL statement is “null”, this is because the database uses auto incrementation for the id. This means when a new user gets added to the database they are automatically assigned an id value in the database that will be the next lowest value not already assigned.

The variables stored initially are then assigned to the remaining values that are needed for the SQL statement. As this function only requires data to be added to the database some validation is needed to ensure the event occurred, the method executeUpdate() returns 0 or 1. 0 meaning the update failed and 1 if it was successful. To validate the statement being executed the value returned by this method is checked. A welcome message is displayed if it is successful informing the user they can now login, and providing them a link to navigate back to the login page.

In the event that the registration is unsuccessful, then the user will be returned to the homepage, and would have to try to register again.

### 4.1.3 Implementation of Quiz and Question creation pages

This subsection will focus on the implementation of the web pages that provide teachers with the ability to create both quizzes and questions. The quiz builder is basic and simply passes off the work to the question builder. But it does set the required number of questions to be added to a different quiz than the default, as well as adding a new quiz to the quiz table of the database.



Figure 16: Quiz form code

When a user chooses to create a quiz, they are taken to this form displayed on a web page, it requests that the user input the following items:

* Quiz Name
* Quiz Topic
* Number of Questions

These are then passed to another JSP file that handles the creation of the quiz on the database. It also creates the quiz within the current session once the user has clicked the submit button.

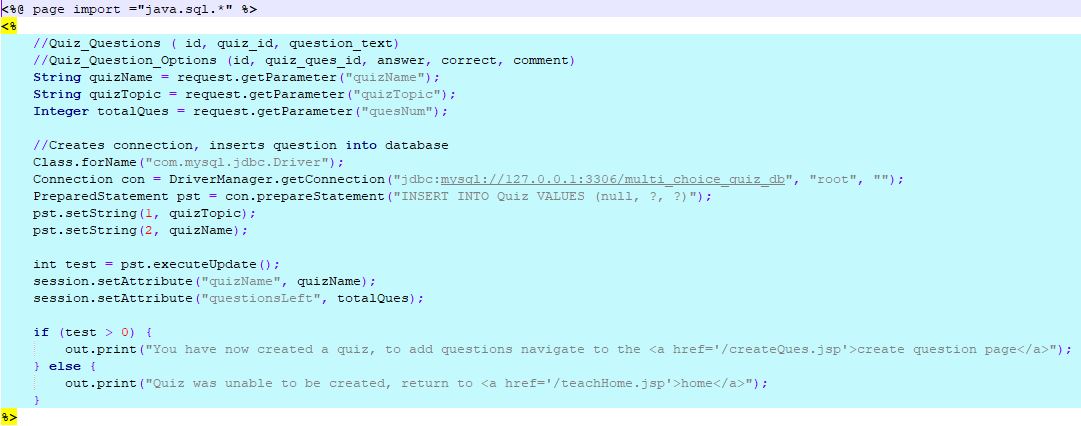


Figure 17: Handling Quiz creation

Similarly to the registration page, the data submitted is stored into variables, and a connection to the database is created. A prepared statement is then used to add the user’s data to the Quiz table.

The quiz name and the number of questions that were requested to be part of the quiz are then stored as session attributes. This is because they will be needed later when adding questions to the database. They will also be needed to calculate how many questions a user has left to add for the quiz they are currently creating.

Finally the success of the update statement is checked, and the user is provided either a confirmation message. Informing them they can now add questions, or a failure message that provides them a link back to the homepage.



Figure 18: Form for creating questions

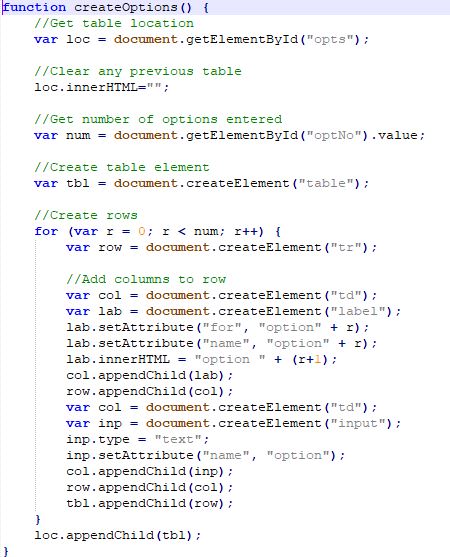


Figure 19: Code adding boxes for answer options

As can be seen in figure 18, when the number of options is entered, the input box has an onChange event that runs a JavaScript function, Shown in figure 19. This is half of the script the other half performs exactly the same process. But it places the table it creates in a different location in the original form.

The first step is to find the location the table needs to be created. Then clear any HTML found here, this would occur say if a user input 2 originally and then decided they wished to have 3 answers. A table element is then inserted here, and based on the user input, a row is added to that table with a label and input box. This can be seen in practice in the User Interface Design section of this document.

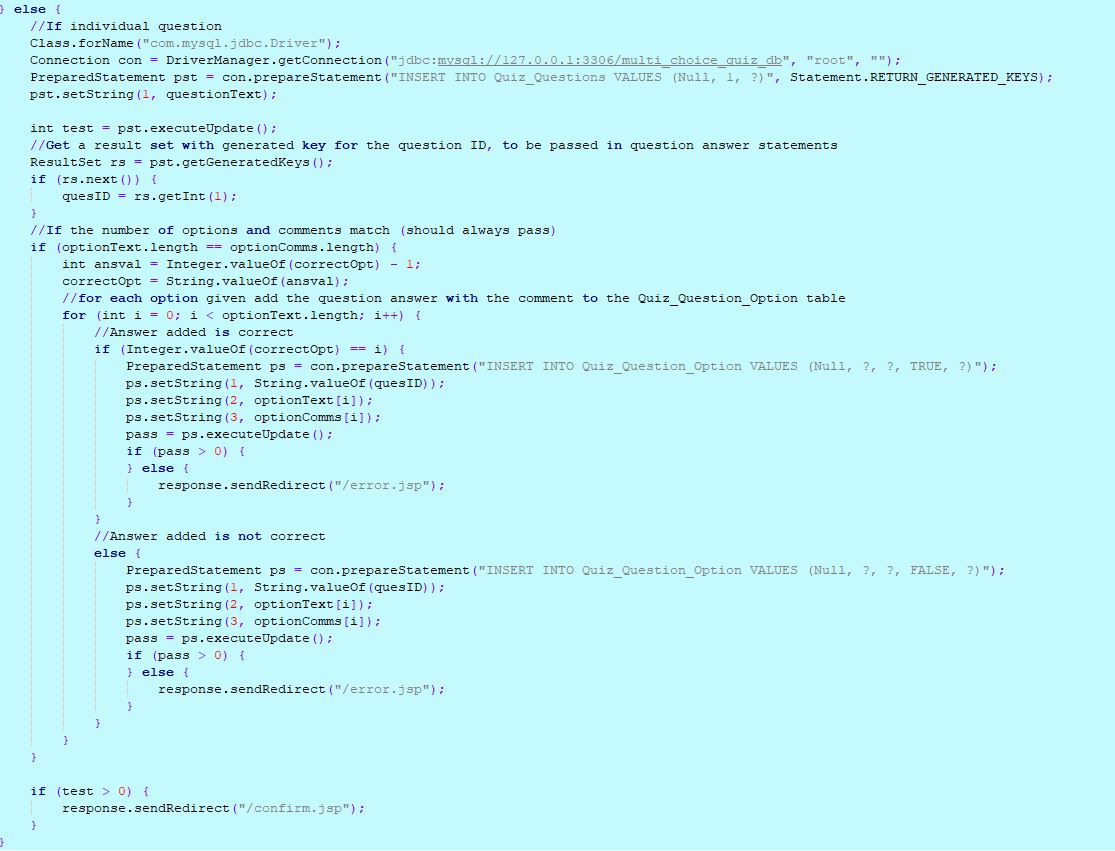


Figure 20: Code adding question to the database

The if statement prior to this checks if the questionsLeft attribute exists and is greater than 0. Which would mean that the question was designed for a specific quiz, the only difference is that the quiz title must be retrieved. Then added to the Quiz\_Questions table in the first update in figure 20.

This section displays the code for adding a single question that is not part of the quiz. After the question is entered into the database, and this is checked, the generated keys are needed to be passed to the question answers. So that they are able to be linked back to the question using the question id.

The code then loops through each option that was entered in the form, and checks if is the correct answer or one of the incorrect answers. These are added to the database separately due to having to enter a Boolean value into the database. Rather than checking if the answer currently being added was correct it was more efficient to simply have an if statement determining which was correct and incorrect and having a separate prepared statement for each.

Once the question has been added to the database, the user is forwarded to a confirmation page informing them of the fact. From which they are able to return to their homepage.

### 4.1.4 Quiz implementation

This subsection of the report focuses on the implementation of the quiz page within the web application. It uses Java embedded into the web page to load all of the quiz elements from the database. The code presented in figure 21 shows a user taking a random quiz

The first step is establishing a connection with the database. Then by using the Rand() function in MySQL, which returns a random number between, using this with the ORDER BY keyword ensures that the questions are placed in a completely random order. To ensure the quiz has a fixed length, the LIMIT keyword means only a specified number of rows will be returned from the table. At the time this code was being used for a brief demonstration, to keep random quizzes short the limit was set to five. But for user generated quizzes a larger number of questions could be used.

The question id and question texts are then stored in ArrayLists. These are needed later to extract the relevant answers from the database.



Figure 21: Quiz page code

A form is then placed within the Java code, in order for the user to be able to submit their answers to the quizChecker page. Within that form, all of the elements are dynamically added using Java. The algorithm loops through each question based on the size of the ArrayList the questions were added to. Another SQL query then retrieves the answer text and Boolean value that determines if the answer is correct or not, from the database for the quiz question that is currently being added to the form.

The question is then output onto the form, and the questions answers are made with a radio button, these are grouped by question. Once these have all been added to the web page, the session attribute “ques” is assigned the ArrayList created earlier that stored the question numbers. This will be needed for determining the answer to the question and determining correct or incorrect answers.



Figure 22: Code checking quiz results

The code presented in figure 22 is the code that checks the results of the quiz. It does so by comparing the given answers on the form on the previous page with the database. The first step is to create an ArrayList for feedback to be stored in once it is retrieved. The given answers are also collected by determining which radio button was selected. A connection is then established to the database, as well as the question numbers ArrayList being retrieved from the session interface.

The next step is to start a loop that iterates an equal number of times as the number of questions. The statement that is executed then gets the true or false value that marks a question option as correct or incorrect. As well as the comment text that goes with the given answer. After the result set has been returned a check is done to see if the user got the answer correct. A counter keeps the amount of questions total and the amount of correct answers given. The feedback is also added to the ArrayList at this point.

Once all the questions have been checked, the mark is calculated. The user must be got using the session attribute created on login. This is in order to get the user id, as this is how the user table and results table are linked in the database. This is then used to add the result to the results database. The final stage is to display the result to the user, in table format as seen in figure 9.

### 4.1.5 Implementation of results view

The results view implementation is available to students who wish to view their previous results. It presents these in a dynamically produced table within the web page, a similar table is found for teachers which consists of all the results.



Figure 23: Results view code

This shows the same connection to the database being established as with all other features. Then the execution of a query being stored in a resultSet. This time we access the user id by comparing the email stored in the session attribute userid with those in the database. Once we have the current user’s id we are able to access their quiz results, the following are retrieved;

* Date of attempt
* Name of the Quiz
* Score

These are then output in a table dynamically generated in Java. The table is assigned a background outside of the defined CSS as it was simpler than creating a separate CSS definition for one table. As in this case it was necessary to give it a different background to the default site background.

# 5. Conclusion

The final chapter of this report will present the goals achieved. Describing the work completed compared with the original objectives laid out at the beginning. It will also discuss any failures to meet goals and discuss the process of developing the software. The other subsection of this chapter will discuss further work. This will discuss briefly possible investigation that could be done using the work developed in this project. As well as parts of the current work that are not completed due to constraints or problems.

Within these chapters there will also be evaluation of the project as a whole as well as the individual features and goals being discussed.

## 5.1 Goals Achieved

Overall the goals for the project were to develop a multiple-choice web based test engine. Allowing teachers to create questions and quizzes. As well as allowing students to take these tests and receive feedback on their answers to use the tests to increase their knowledge of a subject area. As a whole I feel the project goals were met reasonably well. All the key features that were needed in order to meet the overall goal were included in the final work.

As for the specific requirements outlined in this document the first of which was to ensure that a user could register to the website. This was fully achieved and in a way in which is effective and efficient. The amount of data collected is kept minimal to ensure maintainability of the database, and the form was kept simple, and should be useable to people with limited computer skills or knowledge.

The next requirement was to allow a user to login to the website. This was done by checking the details entered and those found in the database and ensuring they match. This process is completely safe from SQL injection, and there is no way for the database to be accessed from the web app so the passwords are secure. Developing a registration and login system that was easily useable was one of the most important things to consider. As it meant that the educational tool could be used by students of any age.

The third requirement outlined in this report was the ability for a student to join a class. In the end this was not implemented into the project, leaving this out was partially down to time constraints and partially down to lacking a good way in which to implement it. Considering the original idea was for students to locate their teacher and then simply join their class I felt this could lead to some students joining classes they were not a part of. As there was no way for a teacher to deny or allow access. This wasn’t a key feature of the website and as such any student can take the test of any teacher. Which isn’t seen as an issue as the tests are primarily intended to be used for learning and not so much as a tool for testing.

Adding a question was a key part of the site, and the first thing that was implemented following the registration and login system. Creating this early also allowed me to refine my database design to store questions more effectively. One feature I thought would be hard to implement was the variable number of answer options, however using JavaScript it turned out to be an easier feature to implement. Doing so allows teacher users more flexibility when creating questions. This is important as it would allow for slight variations in the types of questions possible. For example it would be possible to make a simple true or false question. As well as the more traditional multiple-choice question, and these could be featured in the same quiz to add variety.

In some cases a teacher could wish to delete or edit a question they have added to the database. This could have been implemented but due to registration not needing validation for user type, this could have meant that students could register as a teacher and edit or delete questions. As such I left it out to keep the data secure.

Being able to generate a quiz at random, was another key feature. As without it the rest of the software is made redundant. Students were given the ability to generate a quiz of 5 random questions taken from the database. While I am quite happy with the outcome of this feature it could have been designed in a more readable way. There were some issues caused by features of the CSS that I was unable to find solutions too that made the spacing somewhat uneven and messy. However due to time constraints and other features needing completion, a compromise was made and the feature was left. It is fully usable the only potential issue is if a user is new they may be somewhat confused by the layout. The layout can be seen in figure 8, the spacing between the questions and the answers being the main issue.

Recording and viewing results was also important if the MCQs were to be used as an educational tool. A student who is not given results or any kind of feedback would be unable to understand the mistakes they made or learn what the correct answer was. This leads to two important features. The first being the full results page, where a student can view over time all of their test results allowing them to see improvement in their test results or areas they need to improve their knowledge. The second feature is even more important, the end of quiz feedback page. Informing a user of their mark for the test they have just taken, and also the specific question feedback from the questions they answered. This feedback is not available to a student at a later date, as storing it all in the database would require a lot of space. So it is important that a user can view this page straight after a quiz.

## 5.2 Further Work

There were a couple of minor adjustments I would have liked to make to the registration form in order to perfect it. One of these is an email confirmation. This isn’t necessary for any of the features of the site, but would be necessary for something such as password resetting. If there is no intention to send emails to the user as in this case then there is no real need to validate the email.

Another feature to add to the registration would be to either have an admin user who is able to verify the registration of teachers. Or only allow certain email address’ to register as this type of user, such as only educational addresses (.ac.uk). This would help to prevent students being able to register as teachers without any sort of verification. As such none of the features in the software allow for anyone to access the database or others work. But to add a feature such as deleting or editing questions it would be important that only verified teachers were able to register as such.

Joining a class would be an important feature to implement, as it would allow for teachers to set up a MCQ and set it for their students to take. This would allow the tool to also be used for more traditional testing as opposed to being focused on teaching. This would need some sort of verification and control for the teachers. Having an admin is one way this could be achieved as they could simply add registered students to a class as per a teacher’s request. Or by having teachers send an invite to join link to students which takes students to a page asking if they wish to join this teacher’s class.

The only feature I felt was a key feature was being able to take specifically created quizzes. As with random quizzes there is no set topic as such their usefulness as a learning tool is limited. However due to time limitations and difficulties with other features of the software this was not implemented. It would work best if the previously mentioned classes of students had been implemented. But could also have worked if all tests were available publically to all students.

While the randomly generated quizzes were useful, it would be helpful if a quiz could be generated with all the questions being of a specific topic. Allowing students to focus their learning. These quizzes would also be made more useful if it was possible to select the number of questions your quiz would contain before taking it. A small five question test.

Using this system to carry out research into how helpful providing feedback is to students, would be possible if more of these features were implemented. By setting a test and having students answer it multiple times. It would be possible to determine how much of an improvement there was in the students results. Based on the number of times they had taken the test and as such been given feedback on their answers.

# References

|  |  |
| --- | --- |
| [1] | M. Rouse, “HTML File Format,” WhatIs.com, July 2010. [Online]. Available: http://whatis.techtarget.com/fileformat/HTML-A-Web-page. [Accessed 27 03 2018]. |
| [2] | “Tech Terms MySQL,” 6 March 2007. [Online]. Available: https://techterms.com/definition/mysql. |
| [3] | “Tech Terms CSS,” [Online]. Available: https://techterms.com/definition/css. |
| [4] | “Multiple Choice,” [Online]. Available: https://en.wikipedia.org/wiki/Multiple\_choice. [Accessed 4 April 2018]. |
| [5] | “Database - First Normal Form (1NF),” tutorialspoint, [Online]. Available: https://www.tutorialspoint.com/sql/first-normal-form.htm. |
| [6] | “Database - Second Normal Form (2NF),” tutorialspoint, [Online]. Available: https://www.tutorialspoint.com/sql/second-normal-form.htm. |
| [7] | “Database - Third Normal Form (3NF),” tutorialspoint, [Online]. Available: https://www.tutorialspoint.com/sql/third-normal-form.htm. |
| [8] | “Java Database Connectivity,” [Online]. Available: https://en.wikipedia.org/wiki/Java\_Database\_Connectivity. |
| [9] | “SQL Injection,” w3schools, [Online]. Available: https://www.w3schools.com/sql/sql\_injection.asp. |
| [10] | L. Gershon, “A SHORT HISTORY OF STANDARDIZED TESTS,” 12 May 2015. [Online]. Available: https://daily.jstor.org/short-history-standardized-tests/. |
| [11] | W. Pillars, “3 Ways to Use Testing as a Learning Tool,” 12 May 2015. [Online]. Available: https://www.edweek.org/tm/articles/2015/05/12/3-ways-to-use-testing-as-a.html. |
| [12] | “Why use multiple-choice questions?,” 8 March 2014. [Online]. Available: https://pragmaticreform.wordpress.com/2014/03/08/whymcqs/. |
| [13] | D. Cohen and I. Sasson, “ONLINE QUIZZES IN A VIRTUAL LEARNING ENVIRONMENT AS A TOOL FOR FORMATIVE ASSESSMENT,” *Journal of Technology and Science Education,* vol. 6, no. 3, 2016. |
| [14] | L. Olson, “Impact of Paper-and-Pencil, Online Testing Is Compared,” Education Week, 30 August 2005. [Online]. Available: https://www.edweek.org/ew/articles/2005/08/31/01online.h25.html.  [15] https://cseegit.essex.ac.uk/cwalkeh/final\_year\_project.git |