

# **GCSE Maths Revision System**

## **(Momaths)**

Muhammad Sabra

Centre number:12460

Candidate number:8358

## Contents

Analysis .....	5
Problem Domain .....	5
Client .....	8
Client needs .....	8
Problems with current system .....	9
Prospective users .....	9
Key objectives .....	10
Research.....	11
Interviews with prospective users .....	11
Interview with client, Seun .....	14
From Interviews with prospective users .....	15
From interview with client .....	15
Methodology.....	16
Timeline: .....	17
Surveys.....	18
From surveys.....	21
Limitations of the system.....	21
Design.....	22
Design overview.....	22
Flow charts.....	23
Login algorithm .....	23
Question marking algorithm .....	25
Joining a classroom algorithm.....	27
Homework algorithm .....	28
Prototype UI designs.....	29
Login.....	29
Sign up page .....	29
Occupation selection.....	30
Student main menu .....	30
Other menu.....	31
Homework page .....	31
Teacher main menu.....	32
Assign questions.....	32
Exam practice menu.....	33
Learn menu .....	33

Hierarchy chart.....	34
Normalised Database tables .....	34
3NF .....	34
SQL queries .....	35
Creating MySQL tables .....	35
Entity Relationship Diagram.....	39
Class definitions .....	40
Technical Solution .....	41
Momaths revision webapp project .....	41
MySQL database tables.....	43
Signup.html .....	43
SignupServlet.java.....	45
User table .....	47
Startup.html .....	47
OccupationServlet.java .....	49
Index.html(login page) .....	51
LoginServlet.java .....	52
Mainmenu.jsp(Student menu).....	56
Mainmenu2.jsp(Teacher menu) .....	58
Classroom table.....	61
Classroom.java .....	62
Assign.html.....	65
QuestionUploadServlet.java .....	66
Homework.html .....	68
HomeworkQuestionServlet.java .....	69
hwanswers.jsp.....	72
HwFeedbackServlet.....	73
circleqs.html.....	76
CircleQuestionServlet.java .....	77
algebraqs.html .....	79
AlgebraQuestionServlet.java.....	80
statqs.html .....	82
StatQuestionServlet.java.....	84
FeedbackServlet.java .....	86
answers.jsp.....	87
solutions.html .....	89

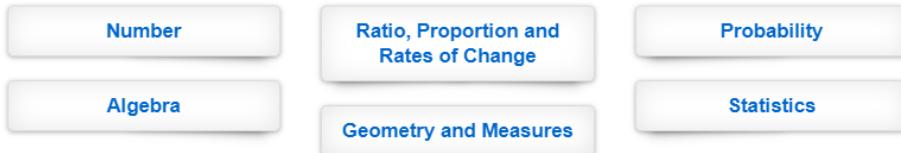
SolutionServlet.java .....	90
Revision.html .....	91
circlenotes.html .....	92
Statnotes.html.....	93
Testing .....	94
Video link .....	94
Testing against objectives.....	94
Test outcome for tests 2.4 and 2.6.....	101
Evaluation.....	101
Evaluation of the system .....	101
Evaluation of objective criteria .....	101
Feedback from client.....	104
Feedback from users .....	105
Evaluations of survey .....	107
Possible improvements for the system.....	107
Appendix .....	108
Styling(CSS) .....	108
Index.html:.....	108
Signup.html .....	110
Startup.html: .....	111
Mainmenu.jsp .....	115
Mainmenu2.jsp .....	118
Exampractice.html: .....	120
Circleqs.html: .....	122
Algebraqs.html:.....	124
Revision.html: .....	127
Circlenotes.html:.....	129
Statsnotes.html:.....	131
Homework.html:.....	133
Feedback.html:.....	135
Solutions.html:.....	137
Assign.html:.....	139
Hwanswers.html: .....	140

## Analysis

### Problem Domain

With so many students doing GCSEs across the entire country, GCSE Maths in particular is bound to have many mixed views and be possibly one of the most if not the most divisive subject for students. Maths above all other subjects is a subject that comes quickly to some and not others, whether that may be because the students don't know how to study effectively, they lack an understanding of how to effectively approach different types of exam questions or they simply lack a basic understanding the content that is being covered.

### Figure 1



## By Grade



### Questions

- Completing the Square QP
- Perpendicular Lines and the Equation of a Tangent QP
- Proof QP
- Proof of the Circle Theorems QP
- Quadratic Inequalities QP
- Simultaneous Equations with a Quadratic QP
- The Nth Term of a Quadratic Sequence QP
- Transforming Graphs  $y=f(x)$  QP
- Velocity Time Graphs QP

### Answers

- Completing the Square MA
- Perpendicular Lines and the Equation of a Tangent MA
- Proof MA
- Proof of the Circle Theorems MA
- Quadratic Inequalities MA
- Simultaneous Equations with a Quadratic MA
- The Nth Term of a Quadratic Sequence MA
- Transforming Graphs  $y=f(x)$  MA
- Velocity Time Graphs MA

Figure 1 shows physicsandmathstutor's display of all the topics a higher-level student aiming for the top grades in maths would need to study. While it is good that physicsandmathstutor is providing students with so many exam questions I believe there is some room for improvement. Firstly, the background and overall design of the website is very bland and as a result is easy for users to get bored and find things to distract themselves from studying. Additionally, while it is good that physicsandmathstutor provides worked answers for the exam questions, at times it can be unclear why certain things were done in the working out.

Figure 2

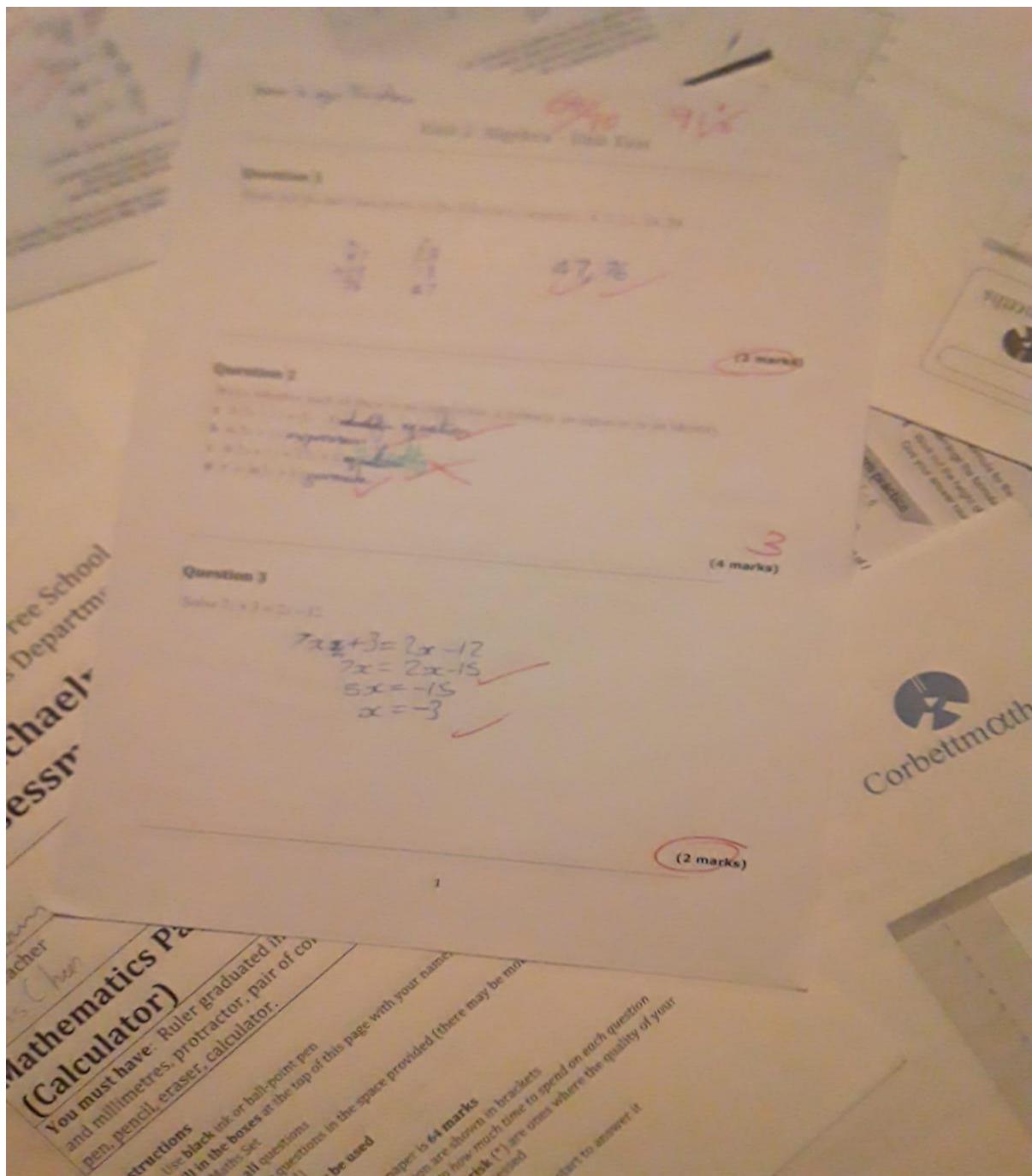


Figure 1 shows Seye's current method of revising for maths. As shown he mostly uses a paper based system for revision and uses notes and past paper questions to revise for maths. However, Seye complains that a lot of the time he finds himself losing his revision notes and/or some of the exam style questions that he has practiced from, and this could become particularly frustrating for him when he has exams coming up soon and is unable to find his revision resources.

I hope to develop a software that can solve these issues with the current systems available for revising maths. This software could achieve this by providing a more

engaging design to help keep users focused on their revision. The software could also, on top of having the worked solutions to questions, provide explanations for why certain steps in the working out were taken, to prevent any confusion with how the answer was reached.

## Client

The client for my project will be Seun, a friend from secondary school. As he is in sixth form now, Seun does not need this project for himself but rather for his younger brother, Seye, who is in Year 10 now and soon will need to start preparing for his final GCSE exams.

## Client needs

For this software, Seun needs app from me which will serve the purpose of helping his younger brother, who is currently in Year 10, with GCSE maths (focused more towards Edexcel). In particular my client is looking for a way to help his brother track his progress in maths, organise his revision time and highlight which topics he needs to spend more time on. Seun also believes that his brother is lacking apps and other online sources of revision and believes that this is a good opportunity for him to take advantage of more of the revision resources available. By utilising this app, Seun also hopes for his brother to be able to attempt more questions at his current level and be able to work up to the grade level that he wants to achieve.

In addition to this, the app will provide Seun's brother with revision plans to help him be more organised with his revision and will also set daily revision goals/targets to meet.

### Problems with current system

Seun's brother is easily distracted and finds it hard to focus on his work which leads to an overall less amount of effective studying being done and therefore less success in his mock and practice tests. As shown in the above picture, a very likely root of this lack of focus is the lack of organisation of Seye's revision materials, which I will be expanding upon next.

Also linking to the mentioned lack of organisation, a problem with this current system is that as it is a paper based system it is very easy for possibly valuable revision materials to be damaged or lost, with no way of relocating them

### Prospective users

Firstly, the main group that this app is obviously targeted towards is secondary school students who are looking to improve their maths skills and prepare for their maths GCSE exams. The app could even be helpful for younger students in year 7 8 and 9 who want to get an idea of what kind of topics GCSE maths has in store for them.

Apart from obviously secondary school students and students who are preparing for GCSEs, the app could also be very useful for teachers/private tutors and also parents, as they would be able to track their students/children's progress and their revision plans and approaches to answering questions. It could also allow them to see what parts of the students' exam technique could use improvement.

Teachers may also recommend the app for students to revise from if they are not already, and use the app for teaching lessons and possibly also assigning homework tasks using the app.

## Key objectives

1. Allow user to login using username and password
  - 1.1 Allow user to create an account (sign up) before logging in
  - 1.2 Do not allow user to login with an incorrect username or password
  - 1.3 Do not allow user to sign up/create an account with one or more empty fields
  - 1.4 Do not allow user to sign up/create an account without confirming their password correctly
  - 1.5 Do not allow user to sign up if the username they type in already exists
  - 1.6 Store the email, username and password in a database
  - 1.7 Have an algorithm to securely store the password in the database such as hashing/encryption
  - 1.8 Allow users to select whether they are a student or a teacher
  - 1.9 Display a different main menu to the user depending on whether they are a student or a teacher
2. Allow users who are teachers to automatically have a classroom created for them
  - 2.1 Allow users who are students to join the classroom
  - 2.2 Allow multiple students to join a classroom
  - 2.3 Sort the usernames of the students in the classroom in alphabetical order
  - 2.4 Allow students to assign homework questions, via uploading images, for all of the users in their classroom

- 2.5 Allow teachers to assign multiple questions under the same homework name, therefore assigning them as part of the same homework task
  - 2.6 Allow teachers to enter the answer for each question
  - 2.7 Implement functionality for users in the classroom to answer these questions and display the answers to them and whether they got the question right or wrong after they press submit
  - 2.8 Allow teachers to assign multiple homework tasks
- 
- 3. Provide users with revision materials already on the website
    - 3.1 Allow users to answer different exam questions on different topics
    - 3.2 Allow users to submit their answers and see whether they got each question correct or incorrect. If the user gets a question incorrect
    - 3.3 Allow users to view a page with the markschemes for each question so they can understand how to answer the question correctly if they were unsure
    - 3.4 Give users the option to view revision pages which contain key notes on some of the important topics in GCSE Maths such as Circle theorems and Statistics

## Research

Interviews with prospective users

Hanafi, a 40 year old mother:

## How does your son revise for GCSE Maths?

My son primarily revises for GCSE maths by working on worksheets that are given out in class. These worksheets are a great way for him to practice the concepts he has learned in class and to reinforce his understanding of the material. He likes to take his time and work through each problem step by step, making sure he understands the reasoning behind each solution. In addition to the worksheets, my son also uses his textbook and notes to review key concepts and formulas. He likes to go back through his notes and highlight important information, which helps him to remember it better. My son also finds it helpful to work on past exam papers and practice tests. This gives him a good idea of the types of questions he can expect to see on the actual exam and helps him to identify areas where he may need additional practice. He also likes to create flashcards with key formulas to help him memorize them.

## Are there any issues with the way your son revises?

Yes, unfortunately. He has a tendency to misplace or lose his worksheets and notes, which can be frustrating for him and make it difficult for him to keep track of what he has covered. This can also result in damaged worksheets, which makes it difficult for him to read or complete the problems. The large number of worksheets can also feel very messy and unorganized, which can be overwhelming for my son. This can make it harder for him to focus on the material and can lead to him feeling less motivated to revise. My son's tendency to lose his worksheets and notes also makes it harder for him to revise past content effectively. Since he doesn't have access to all of the materials he needs, he may miss out on revising key concepts or formulas that could be on the exam. For example, if he loses a worksheet that covers a certain topic, he may not be able to go back and review that material in as much detail as he would like. This could result in him feeling less confident about his understanding of that topic, and could ultimately impact his performance on the exam.

## Do you think a Maths revision app would help you?

Yes I think it would greatly benefit my son. This app could provide my son with questions and revision materials for the content that will come up in the exam. It would also help him to keep track of his progress and identify areas where he may need additional

practice. Having all of his revision materials in one place would be very helpful for my son, as it would make it easier for him to stay organized and focused. He would no longer have to worry about misplacing or losing his worksheets or notes, as everything he needs would be accessible through the app.

Ramon, a 16 year old student in Year 11:

How do you revise for maths?

When I revise for maths, I primarily use online resources like HegartyMaths, PhysicsandMathsTutor, and SavemyExams. I find that watching videos is an effective way to understand the concepts, and then I reinforce my learning by answering practice questions from these websites. This approach allows me to cover a lot of material in a short amount of time, and it helps me identify areas where I need to improve my understanding.

Are there any problems with how you revise?

I often get distracted by other apps on the device I'm using, like video games or social media. Sometimes I also find myself texting my friends instead of focusing on my revision. These distractions can be a significant challenge for me and can lead to me wasting a lot of time that I could have used to revise more effectively. I need to work on improving my self-discipline and staying focused during my revision periods.

Do you think a maths revision app which helps to keep you engaged would be helpful?

I think a maths revision help which keeps me engaged in my revision, by giving me a revision timetable to work with or give me certain revision goals would greatly benefit me. This way, I would have a clear idea of what I need to accomplish during each revision session, and I would be more motivated to focus on my revision and not get distracted by other apps on my device. Additionally, having a specific timetable for my revision would help me stay organized and make the most of the time I have available. I believe that this solution could be very effective for me, and I plan to look for a suitable revision app that provides these features.

Interview with client, Seun

What are the main objectives of this system?

The main objectives of this app are to provide a comprehensive set of exam questions for mathematics, to offer effective revision strategies, and to equip my brother with the skills and knowledge necessary to succeed in their maths exams. The app aims to fill the gap left by his teachers who do not provide enough practice questions or guidance on how to revise effectively. Ultimately, the goal is to help him revise more effectively for maths and answer more practice questions so he can be ready for his GCSE exam next year.

What drove you to these ideas?

My brother's constant complaints about not knowing how to revise and the lack of exam practice provided by his teachers drove me to these ideas. I could see how his struggles with mathematics were affecting his confidence and his ability to perform well in exams. As someone who understands the importance of good revision strategies and ample practice, I knew that an app that provided both of these things would be incredibly helpful to students like my brother.

From Interviews with prospective users

My first interview with Theresa has showed me that a key strength of this app will be that students will have many different revision materials all waiting for them in one place and they will not need to worry about losing a worksheet and or losing their notes or exam questions on previous topics as it will be waiting for students to come back to it any time they feel the need.

My interview with Ramon also showed me that this app will need to account for the obstacles in the way of students focusing on their work. Nowadays with many different apps, games, social media, chats with friends all accessible at the touch of our fingertips revising is much easier said than done for students in todays age. That is why it is imperative that this system encourages students to stay focused on their revision via a more engaging user interface. As shown in figure 1, unfortunately many revision sources for maths such as physicsandmathstutor have very plain and boring user interfaces. While this obviously doesn't have any effect on the system's operation, it is undeniable that students will be more susceptible to getting bored and getting distracted if we don't find ways to engage them.

From interview with client

My interview with my client, Seun, showed me that not actually everyone knows how to properly revise for maths and what kind of approaches they should have to different questions. More importantly, some students also may not know where to access these questions to practice from if their teachers are not providing them with enough or any at all, as is the case with Seun's brother. That is why it is key that this system provides the

user with sufficient questions to practice from. The system will also make it much easier for teachers to assign questions for their students, which will allow students to practice even more.

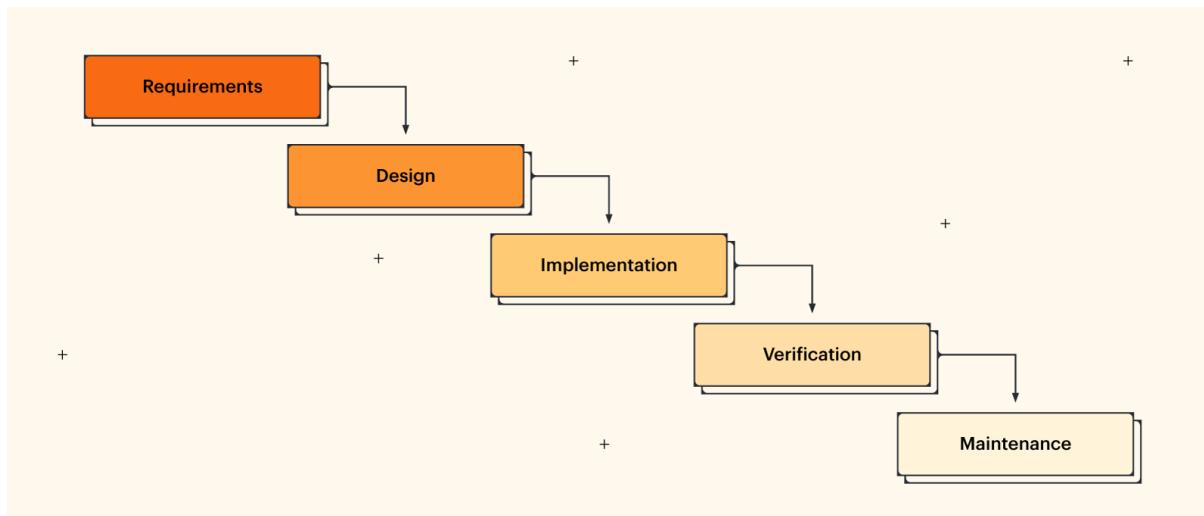
## Methodology

For this project I have researched both the Agile methodology and the Waterfall methodology.

The agile methodology will involve constant communication with my client with each change I make to the system. This would be good as this software will possibly be used by many students for revision purposes and so it is important that I use the advice from my client to make the system as best as it can be for Seye's revision. Another benefit of using this methodology is that it is very flexible, and changes can be made quickly if need be, which would be useful as it is entirely possible that Seye's needs will change as time goes on. At one point Seye may have issues with time and at another point he may have some issues with exam technique.

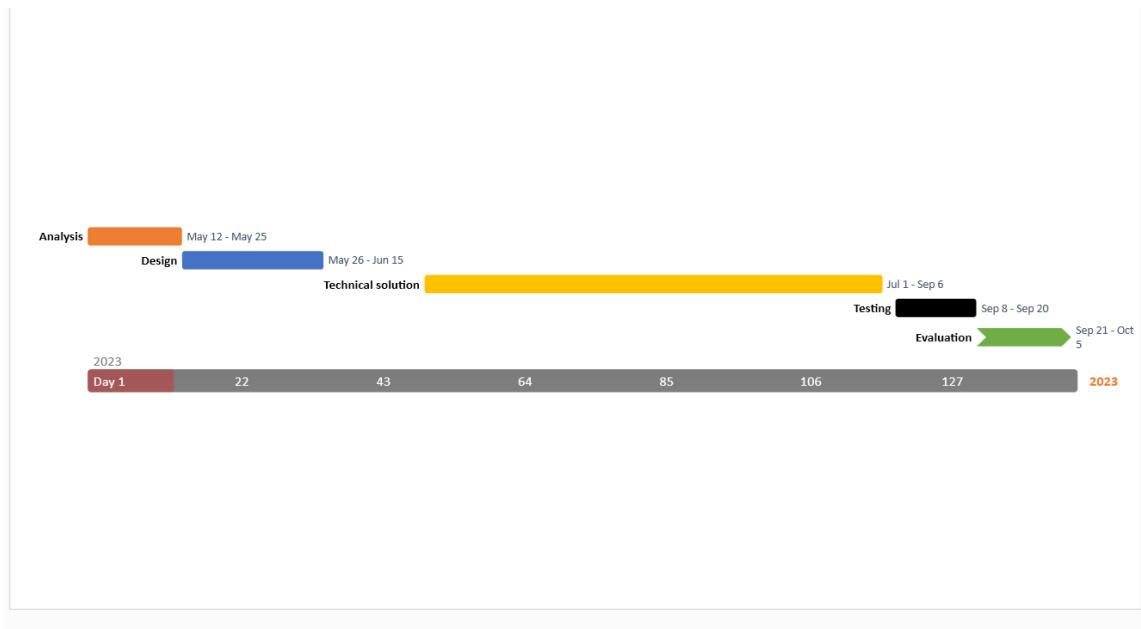


The waterfall methodology, entails a clearly defined sequence of execution. The project will also not advance to the next phase until the current phase is finished and once a phase is completed it can be very difficult to revisit a previous phase of the project.



Because of this, I think the agile methodology is more well suited for this project as it would be easier to adjust the system to my client's needs if they were to change and it is also very likely that later on in the project I may have to revisit previous stages of the project to update or change anything in them.

Timeline:



## Surveys

thinking of yourself as a GCSE student, what features would you like a revision app to provide?

Answered: 7 Skipped: 1

**RESPONSES (7)** WORD CLOUD TAGS (0) Sentiments: OFF

Search Responses Filter: by tag ▾

Showing 7 responses

Efficient layout and easy access to mark schemes  
24/03/2023 13:08 View respondent's answers Add tags▼

flash cards, engaging graphics  
24/03/2023 13:04 View respondent's answers Add tags▼

add longer worked solutions so i can understand the method better.  
24/03/2023 13:03 View respondent's answers Add tags▼

Have you used any revision apps/websites in the past? If yes which ones?

Answered: 7 Skipped: 1

**RESPONSES (7)**  WORD CLOUD  TAGS (0)  Sentiments: OFF 

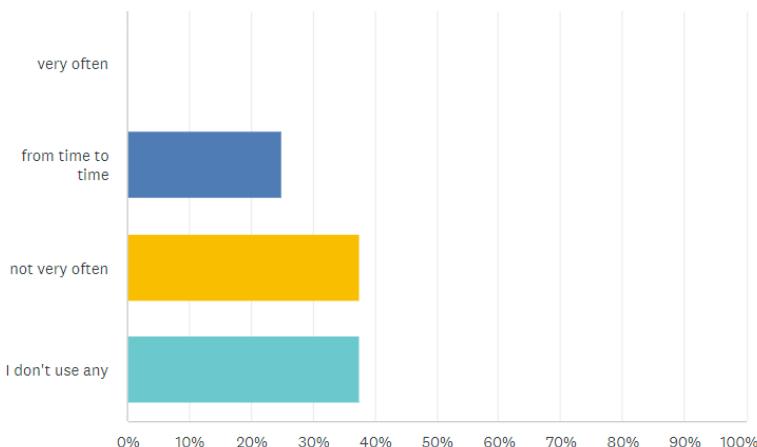
Search Responses  Filter: by tag ▾

Showing 7 responses

<input type="checkbox"/> Anki, Quizlet 24/03/2023 13:12	<a href="#">View respondent's answers</a>	<a href="#">Add tags</a> ▾
<input type="checkbox"/> Physics and maths tutuor 24/03/2023 13:08	<a href="#">View respondent's answers</a>	<a href="#">Add tags</a> ▾
<input type="checkbox"/> seneca 24/03/2023 13:04	<a href="#">View respondent's answers</a>	<a href="#">Add tags</a> ▾
<input type="checkbox"/> dr frost maths, hegarty maths. They were useful 24/03/2023 13:03	<a href="#">View respondent's answers</a>	<a href="#">Add tags</a> ▾

How often do you use apps/software for revision?

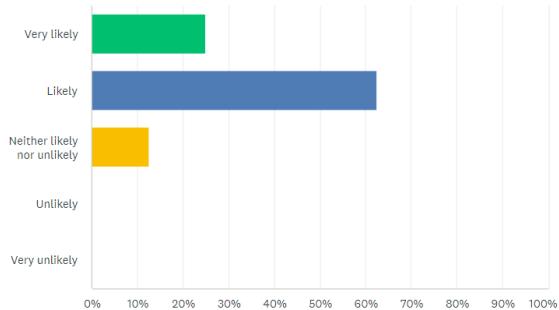
Answered: 8 Skipped: 0



ANSWER CHOICES	RESPONSES
▼ very often	0.00%
▼ from time to time	25.00%
▼ not very often	37.50%
▼ I don't use any	37.50%

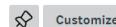
if you had access to a GCSE maths revision app during your GCSEs, how likely would you be to use it

Answered: 8 Skipped: 0



ANSWER CHOICES	RESPONSES
▼ Very likely	25.00%
▼ Likely	62.50%
▼ Neither likely nor unlikely	12.50%
▼ Unlikely	0.00%
▼ Very unlikely	0.00%

Q5

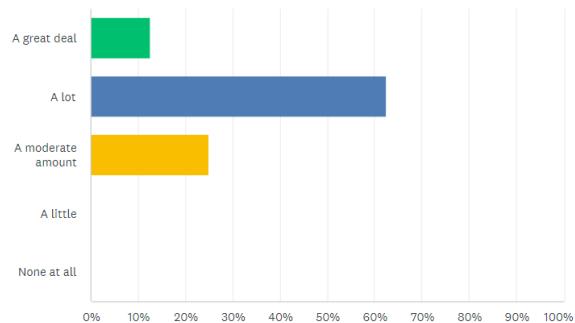


Customize

Save as ▾

how much would this app have benefitted you while you were revising GCSE maths

Answered: 8 Skipped: 0



ANSWER CHOICES	RESPONSES
▼ A great deal	12.50%
▼ A lot	62.50%
▼ A moderate amount	25.00%

Based off of these surveys with students who had finished their GCSEs recently, there seems to be a trend in many believing that a maths revision app would have benefited them greatly when revising for their exams. I have also looked into some responses from these

students on what features they would like to see from this system. Some of which included quizzes, flashcards and engaging graphics, efficient layout and easy access to markschemes and worked solutions for questions so they can understand the method for answering them.

Based off of these surveys I have also decided to take inspiration from other existing revision websites such as Seneca, HegartyMaths, and PhysicsandMathstutor, and see how I can improve on some of the features in these websites.

#### From surveys

Based off of some of the feedback I gained through these surveys, I hope to implement many of the suggested features, as already outlined in my objectives, such as adding features to engage the user more, providing an efficient layout with mark schemes and also worked solutions for questions so the users can clearly understand the method and technique to answering specific types of exam questions.

However, it is important to acknowledge that there are limitations for this system and that there are some features which would benefit users more, however they are simply not realistic given the circumstances of this project.

#### Limitations of the system

There are some limitations of this system that I plan to design which will prevent me from implementing all of the features into the system

that could be given that the circumstances were different. Some of these limitations include the fact that I have limited coding experience compared to many others who have designed software similar to this, and that I am under time constraints as this system is planned to be completed and ready for use in around 4-5 months.

There are some features that would be a great addition to this system such as a currency/reward system in order to engage the users of the system more and motivate them to spend more time using the system and revising and also video explanations for topics or exam questions, however these ideas would be unrealistic given my limited abilities with programming and also that these features are possibly too complex to design in the given time frame.

## Design

### Design overview

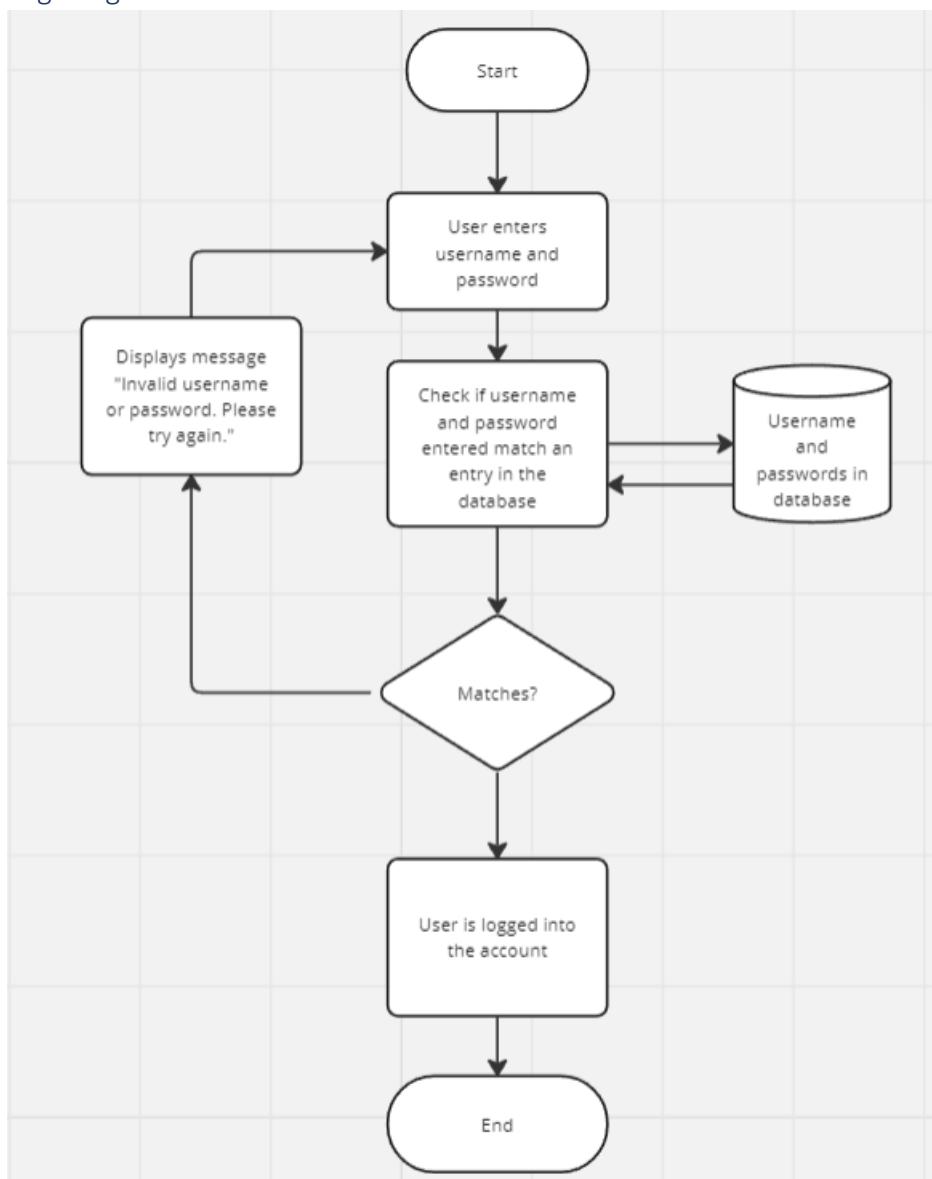
Here in my Design section, I will detail the functionality of my system and the roles that:

- .Data structures to store various different pieces of information
- .Servlet classes to handle requests and responses between the user and the server
- .My database tables to store important details for the system
- .Constructors to create objects and call the methods within the object in other classes to make use of them
- .The design of the User Interface

Will play in creating a Complete solution which meets all of my objectives, which I will outline here including any changes made to the objectives, and also meets all of the needs of my client.

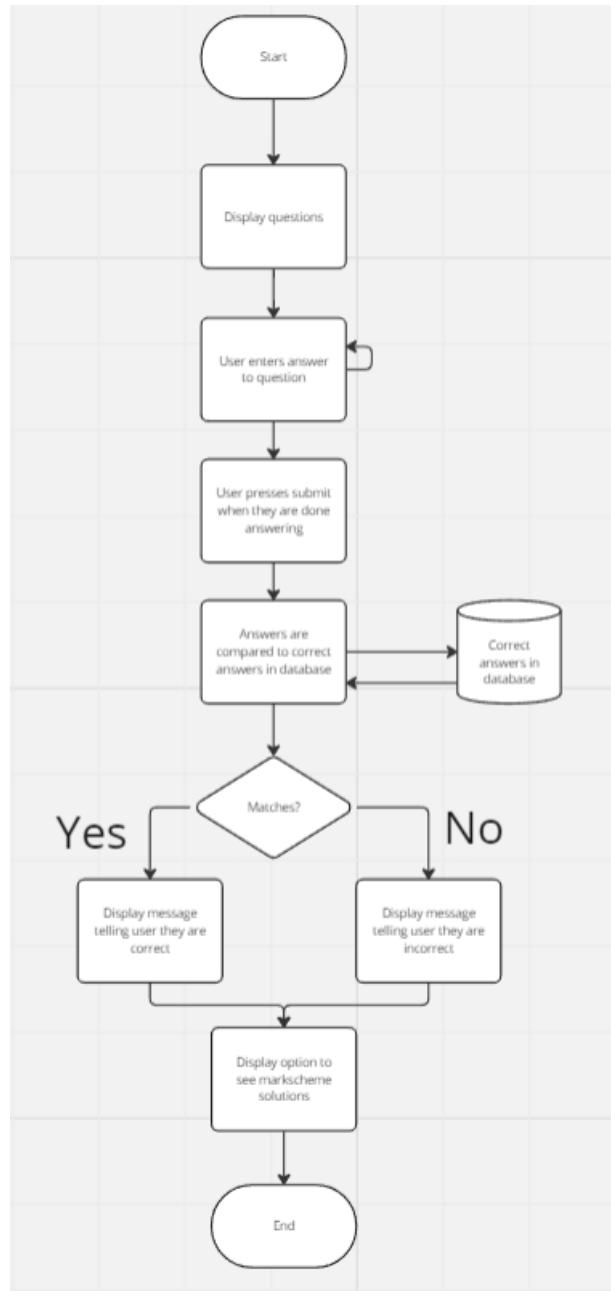
### Flow charts

Login algorithm



This flow chart shows the algorithm which will verify if the username and password entered by a user match an existing username and password already registered in the database. If it does not match an existing entry in the database then the user will not be given access to the account and a message will be displayed telling them the username or password they have entered is incorrect. If the username and password match a entry in the database then the user will be let into the account.

## Question marking algorithm



This flow chart shows the algorithm which will display the exam questions to the user. The user will type in their answer for each question and then press submit when they have answered all the questions they can. The algorithm will then display to the user which questions they answered correctly and which questions they did not. The user can then check the markscheme to see where they went wrong on the questions that they answered incorrectly.

In this algorithm I may use data structures such as an ArrayList to store the user's answers for each question and also for storing each question image. Since the questions

are being displayed to the user as images, In order to display it on the html page we need to store all of the images used for the questions in project files and refer to each of these filepaths when we want to display a different question.

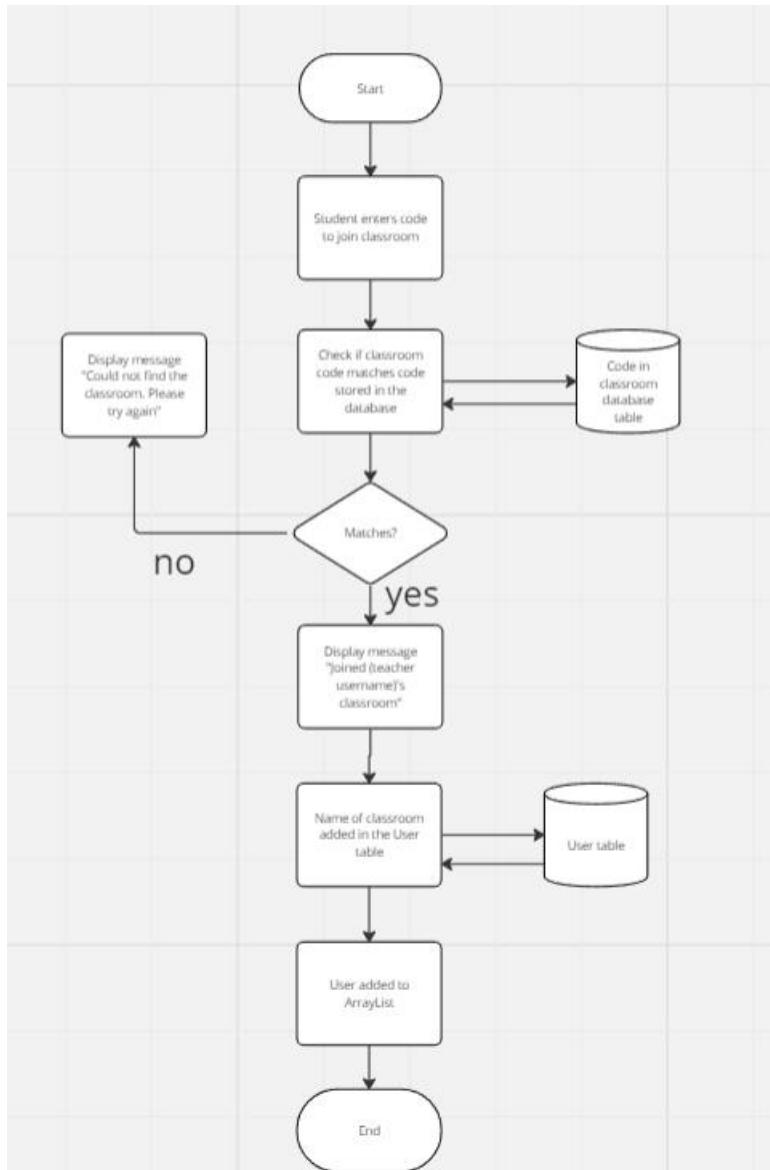
For example:

```
<html>
<head>
</head>
<body>
    <h1>Image Example</h1>
    <img src="" alt="">
</body>
</html>
```

This example shows the html code that could be used to display the image where the image's file path is put as the value of src.

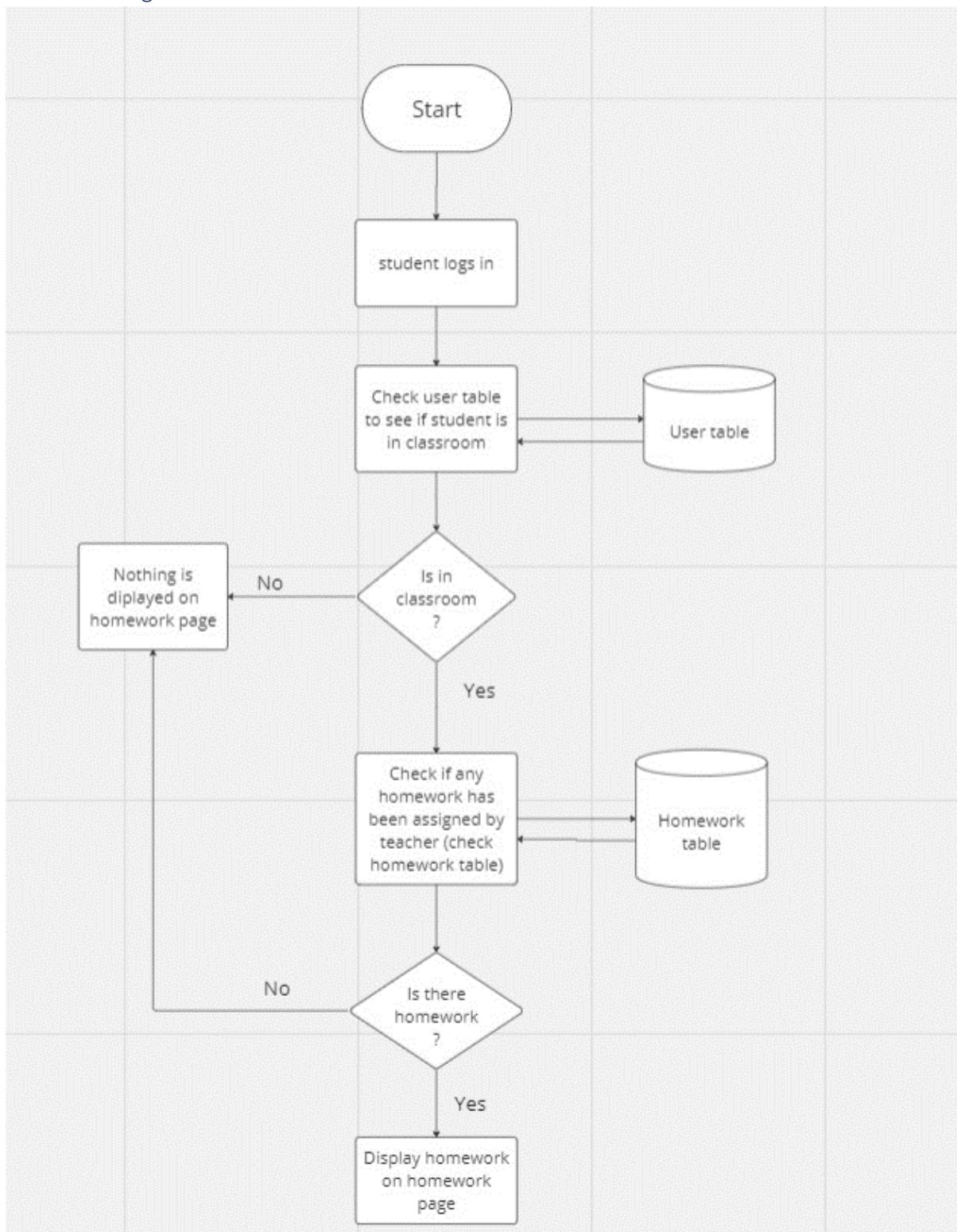
Storing the answers for each question entered by the user in an arraylist can then be marked by the system by iterating through each answer in the arraylist and comparing it to the correct answer to the question which will be stored in the question database table in the Answer column.

## Joining a classroom algorithm



The flow chart above outlines the algorithm which will determine whether the user has entered a valid code to join a classroom by checking if the entered code exists in the database in the classroom table. If it does then the user is added to the classroom and a message is displayed telling them that they have successfully joined.

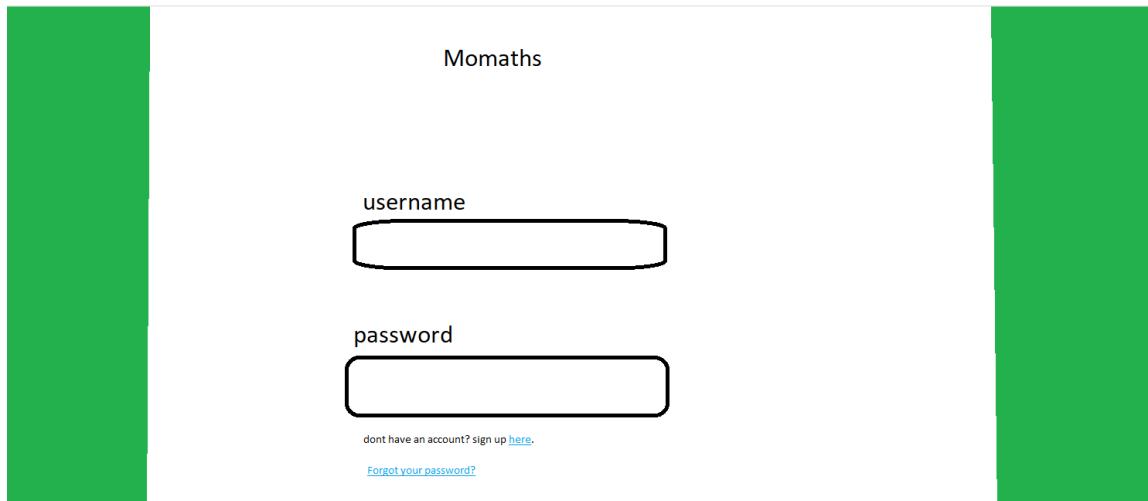
## Homework algorithm



## Prototype UI designs

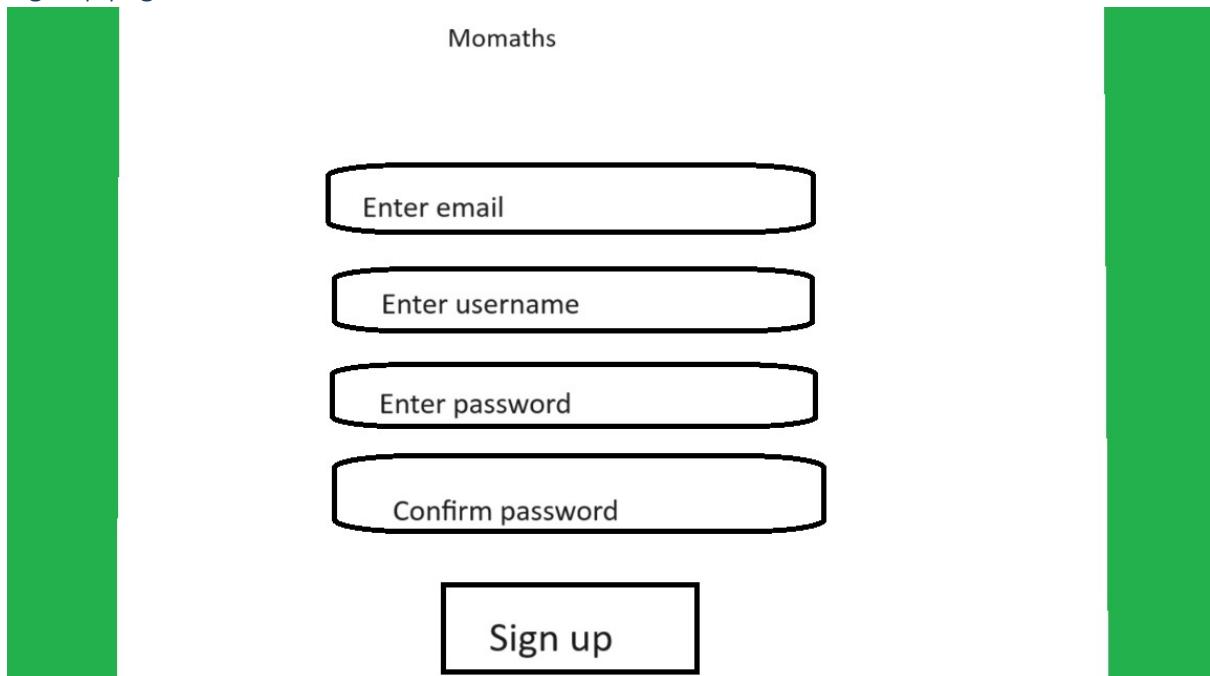
### Login

The user will then have a menu where they can enter their username and password or sign up for an account if they do not already have one.



A wireframe prototype of a login page. The page has a header "Momaths" at the top center. Below it are two input fields: "username" and "password", each enclosed in a rounded rectangle. Underneath the password field is a link "don't have an account? sign up [here](#)". Below that is another link "[Forgot your password?](#)". The entire page is framed by thick vertical green bars on the left and right sides.

### Sign up page

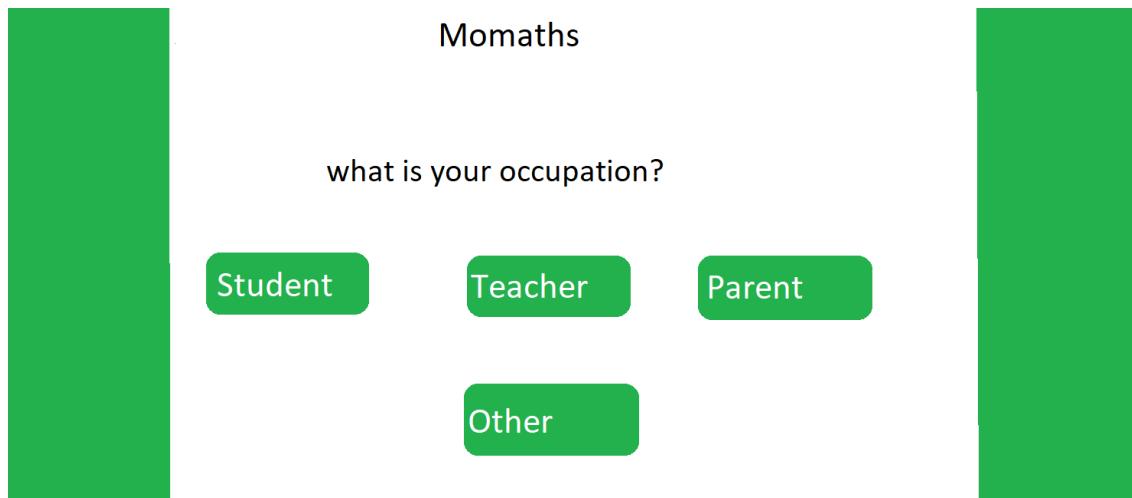


A wireframe prototype of a sign up page. The page has a header "Momaths" at the top center. Below it are four input fields stacked vertically: "Enter email", "Enter username", "Enter password", and "Confirm password", each enclosed in a rounded rectangle. At the bottom is a large rectangular button labeled "Sign up". The entire page is framed by thick vertical green bars on the left and right sides.

On the sign up page the user will be able to sign up by entering their email address, username and password. They will also have to confirm their password before signing up.

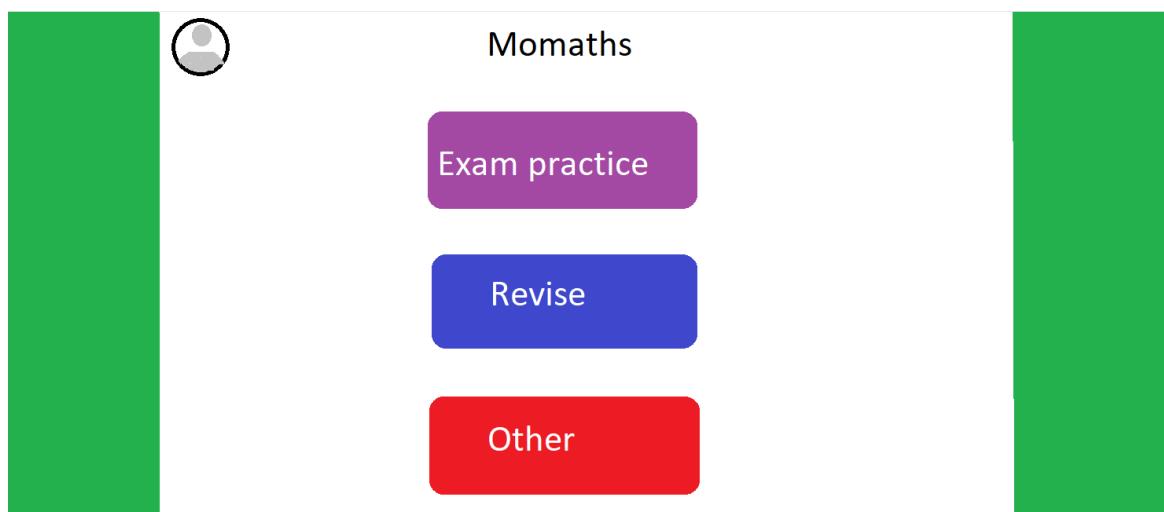
## Occupation selection

When the user first opens the system they will choose their occupation as either a student a teacher or a parent or other (this section is optional)

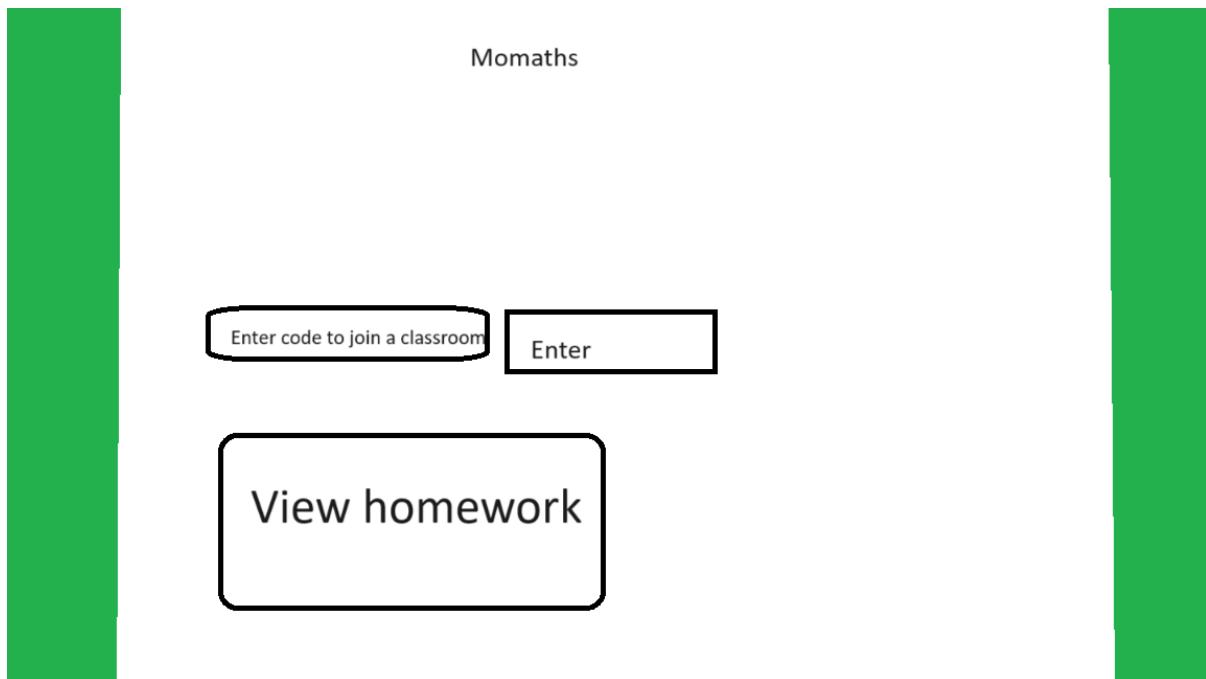


## Student main menu

Once the user has logged in to their account they will be met with the main menu where they can either select the Exam practice section, the learn section or the other section.

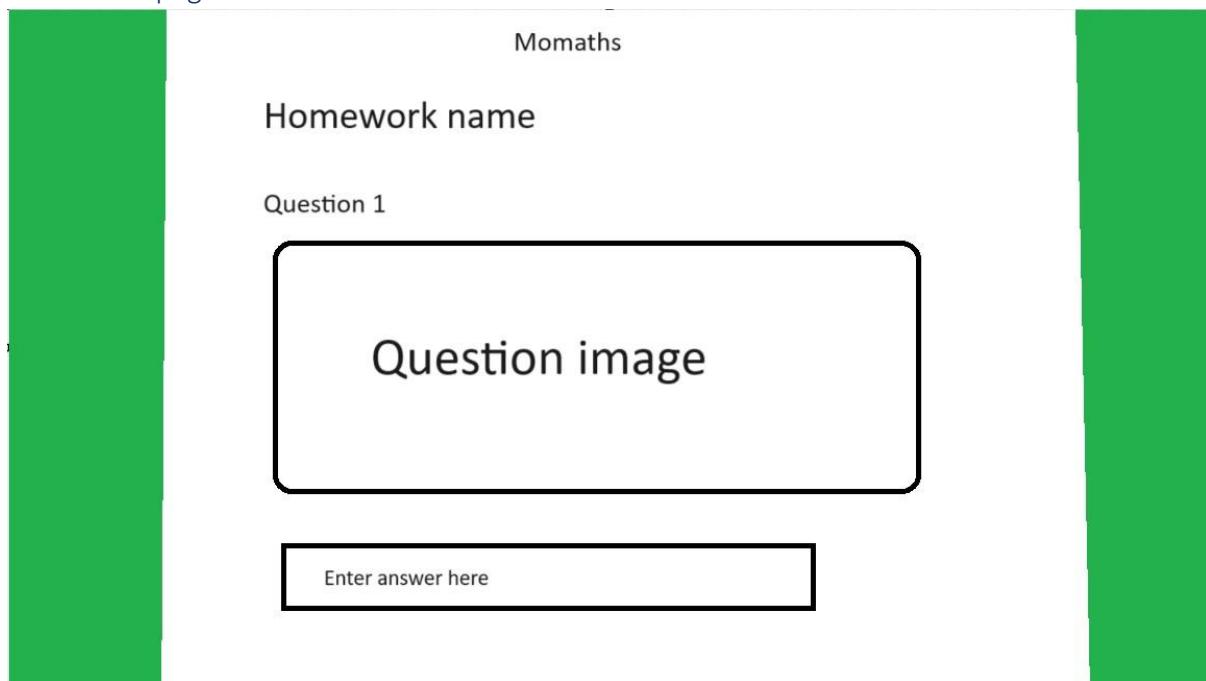


## Other menu



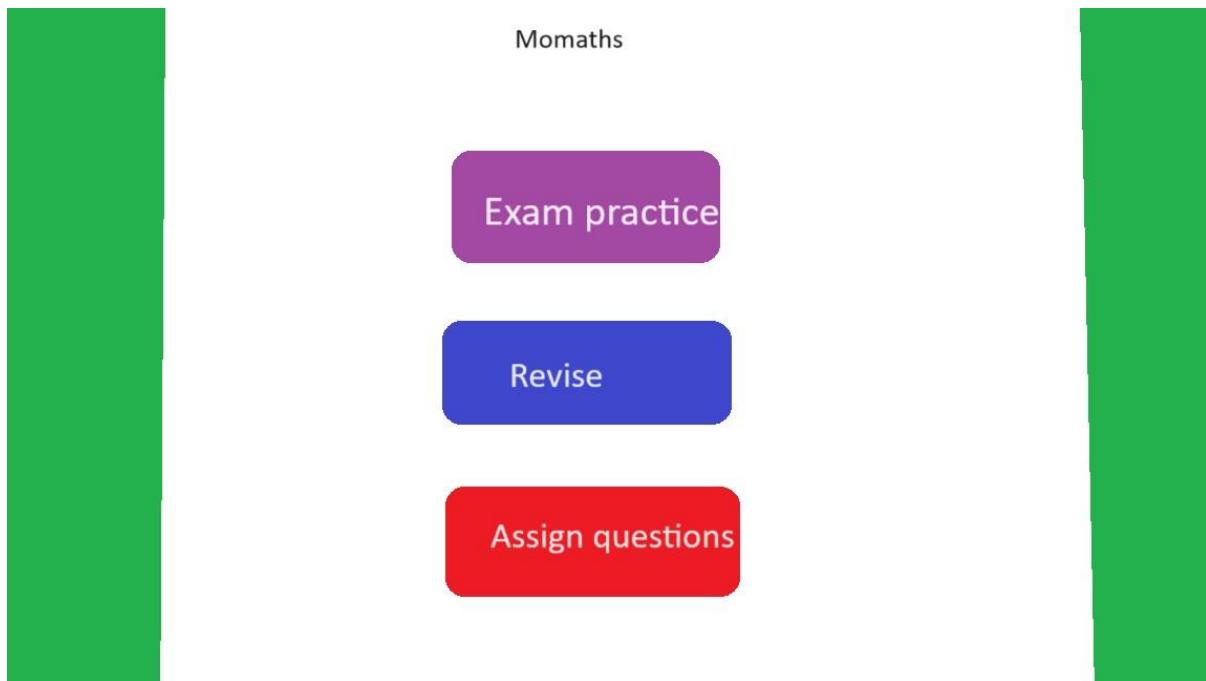
On this page the student will be able to enter a code to join a classroom and also be able to view any homework that has been assigned to them.

## Homework page



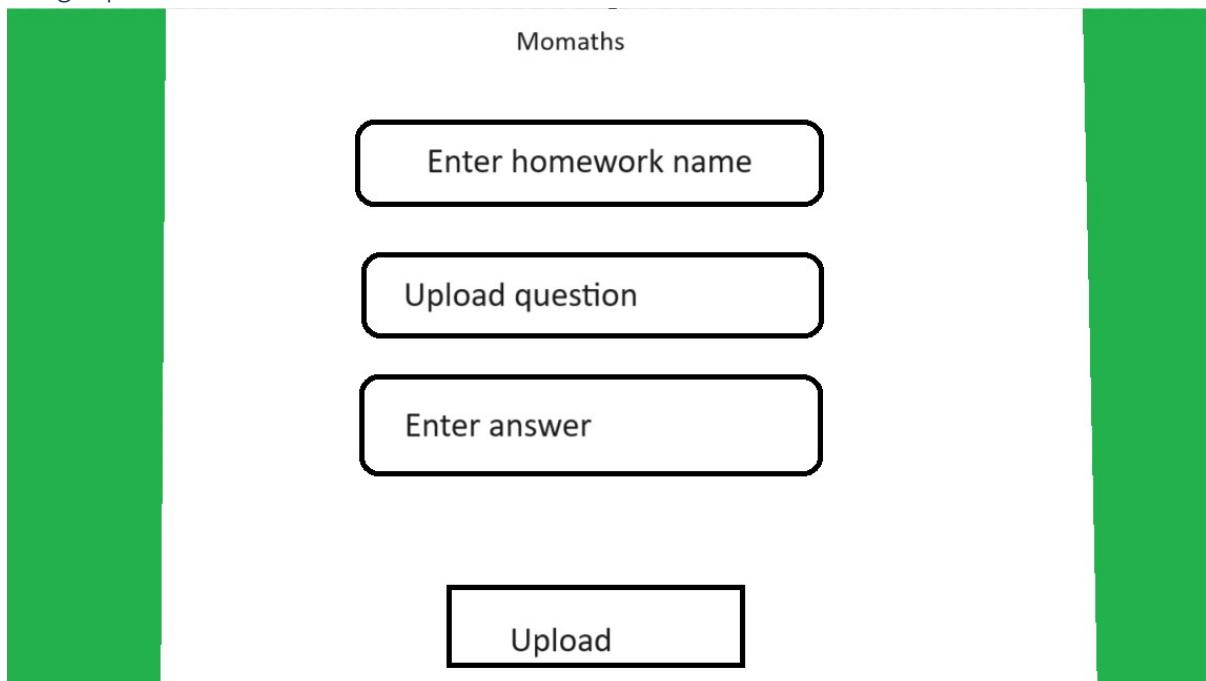
Each question under the same homework name will be displayed as part of the same homework task. The image for each question will be displayed and the user will be able to enter their answer for each question.

## Teacher main menu



The teacher will still have the exam practice and revision buttons similarly to the student menu but instead of joining a classroom and viewing homework they assign homework questions to users in their classroom.

## Assign questions

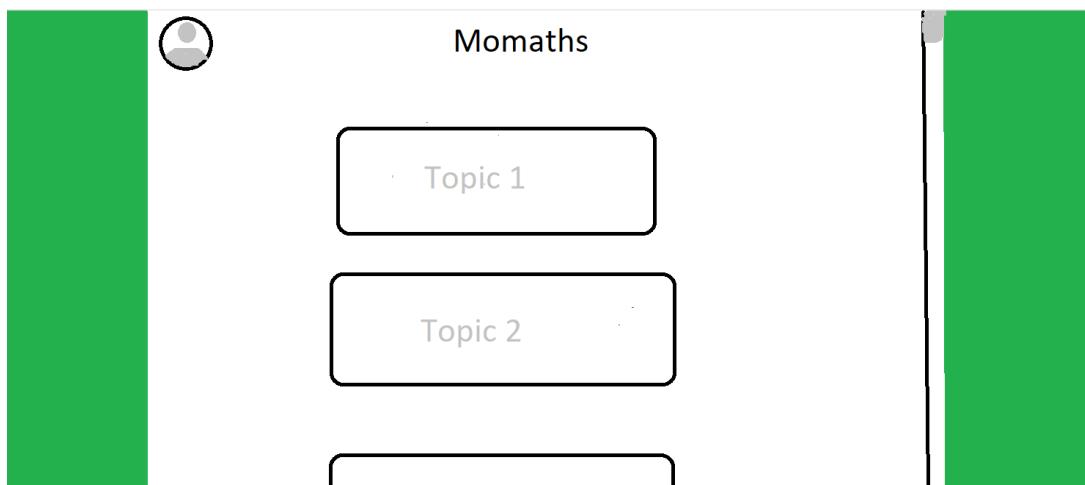


This is the page where teachers can assign questions by uploading files from their computer. They can enter the homework name for

each question and the answer for each, which is stored in the homework table.

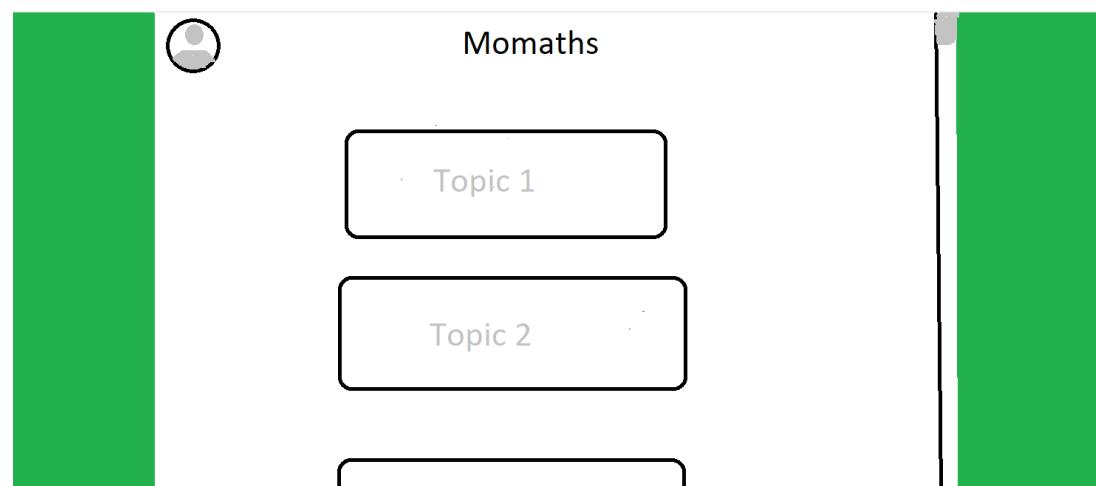
#### Exam practice menu

When the user selects to go to the exam practice menu there will be a list of the topics in the specification with there being exam questions for each topic which the user can attempt under exam conditions. The questions that are displayed to the user however will depend on what grade level they are aiming for, as there is no benefit in giving grade 8 and 9 questions to the user if they are only aiming for a 5 or 6.

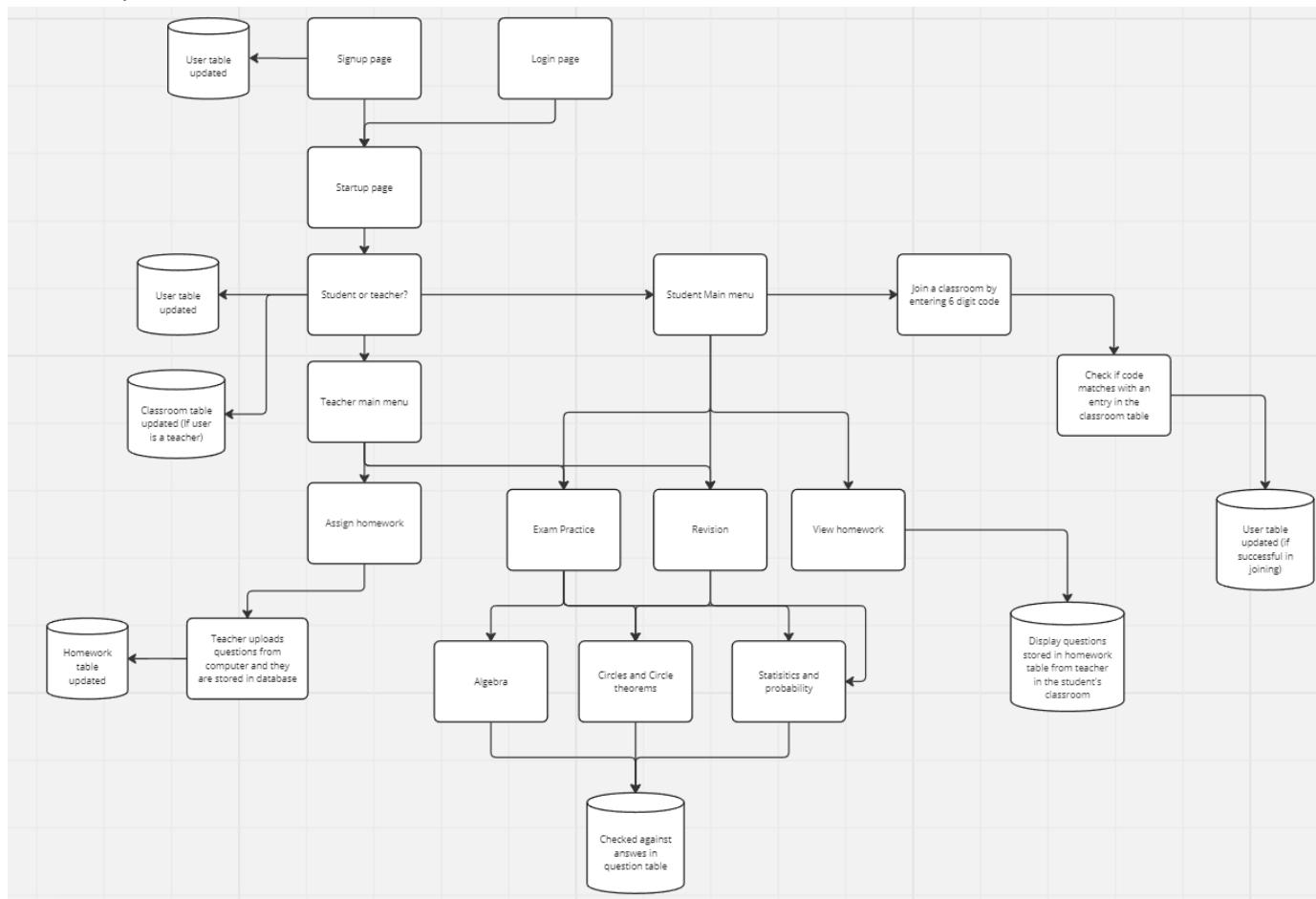


#### Learn menu

This section will contain revision notes on each topic and examples of how to answer questions on the topic. This is more useful for topics where there is more memory involved such as circle theorems.



## Hierarchy chart



## Normalised Database tables

3NF

User(UserID,Username,Password,Occupation,Title,Email)

Question(QuestionID,Question,Answer,Solution,Topic,Numberofmarks)

Homework(HomeworkQID,HomeworkName,TeacherID,Classroomname)

Classroom(ClassroomID,Classroomname,Teachernname,Classroomcode)

User table: This table will store essential information about the user such as their username, password and email address. It will also store things such as the grade the user hopes to get (gradelevel) and their occupation (student or teacher)

Question table: This table will store all the important details about any particular exam style question such as the number of marks, the correct answer to the question and also the solution for how to answer the question

Homework table: this table will store details about each homework question assigned by a teacher. It will store the name of the homework task, the ID of the question, the UserID of the teacher who assigned the question and the name of the Classroom which it has been assigned to

Classroom table: This table will be needed to store details about each classroom on the system such as the ClassroomID, the name of the Classroom, the name of the teacher who is in charge of the classroom and the code to join the classroom.

## SQL queries

### Creating MySQL tables

```

CREATE TABLE `user` (
  `UserID` int(11) NOT NULL AUTO_INCREMENT,
  `Username` varchar(255) NOT NULL,
  `Password` varchar(255) NOT NULL,
  `Occupation` varchar(255) DEFAULT NULL,
  `Email` varchar(45) NOT NULL,
  `Classroomname` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`UserID`)
) ENGINE=InnoDB AUTO_INCREMENT=65 DEFAULT CHARSET=latin1;

CREATE TABLE `classroom` (
  `ClassroomID` int(11) NOT NULL AUTO_INCREMENT,
  `TeacherID` int(11) NOT NULL,
  `Classroomname` varchar(45) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Classroomcode` varchar(45) COLLATE utf8mb4_unicode_ci NOT NULL,
  PRIMARY KEY (`ClassroomID`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE `question` (
  `QuestionID` int(11) NOT NULL AUTO_INCREMENT,
  `Numberofmarks` varchar(45) COLLATE utf8mb4_unicode_ci NOT NULL,
  `topic` varchar(45) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `question` varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `Answer` varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `solution` varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`QuestionID`)
) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE `homework` (
  `HomeworkqID` int(11) NOT NULL AUTO_INCREMENT,
  `HomeworkName` varchar(45) COLLATE utf8mb4_unicode_ci NOT NULL,
  `TeacherID` int(11) NOT NULL,
  `Answer` varchar(45) COLLATE utf8mb4_unicode_ci NOT NULL,
  `question` varchar(500) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `Classroomname` varchar(45) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`HomeworkqID`)
) ENGINE=InnoDB AUTO_INCREMENT=39 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

Above are the SQL queries used to create each of my database tables.

```
SELECT Question  
From Question  
WHERE Question.Topic='Algebra'
```

```
SELECT User.username  
From User,Classroom  
WHERE Classroom.TeacherID=User.UserID  
AND UserID=?
```

(the ? represents the parameter we are getting from the user. In other words we are selecting the user that has the UserID of the current user if there is one)

```
SELECT username  
FROM User, Homework  
WHERE Homework.Classroomname=User.Classroomname
```

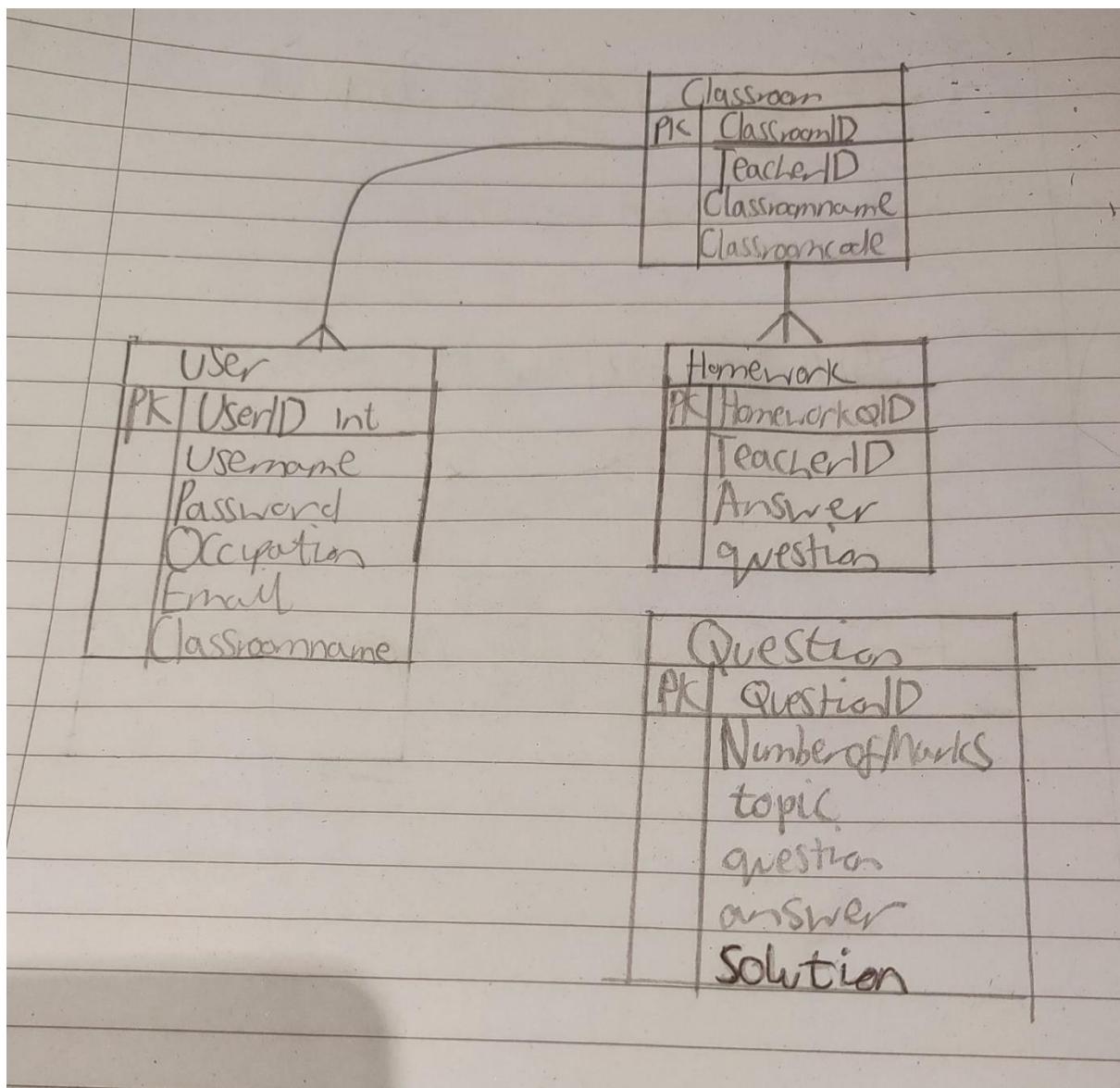
This SQL query will display the questionID,topic, the user's answer and the correct answer for each question in which the user's answer does not match the correct answer.

```
SELECT UserID,revisionplan,grade level  
FROM User  
WHERE User.Occupation='Student'
```

This query will display the ID of the user, their selected revision plan and the grade they are aiming to get for the users which are students.

This query will display the ID and the correct answer for the quiz questions which the user has answered incorrectly (user's answer does not match the correct answer).

## Entity Relationship Diagram



This diagram displays the relationship between my database tables. Each user can only be in one classroom, but one classroom can have many users so the relationship is one-to-many. One classroom can have many homework tasks but each homework can only be set in one classroom so the relationship is many-to-one.

[Class definitions](#)

Classroom=class

Public:

Function getClassroomcode

Function getClassroomName

Function IsStudentinClassroom

Private:

getClassroomcode:String

getClassroomName:String

IsStudentinClassroom:Boolean

User=class

Public:

Function getUserId

Function getUsername

Function getClassroomname

Private:

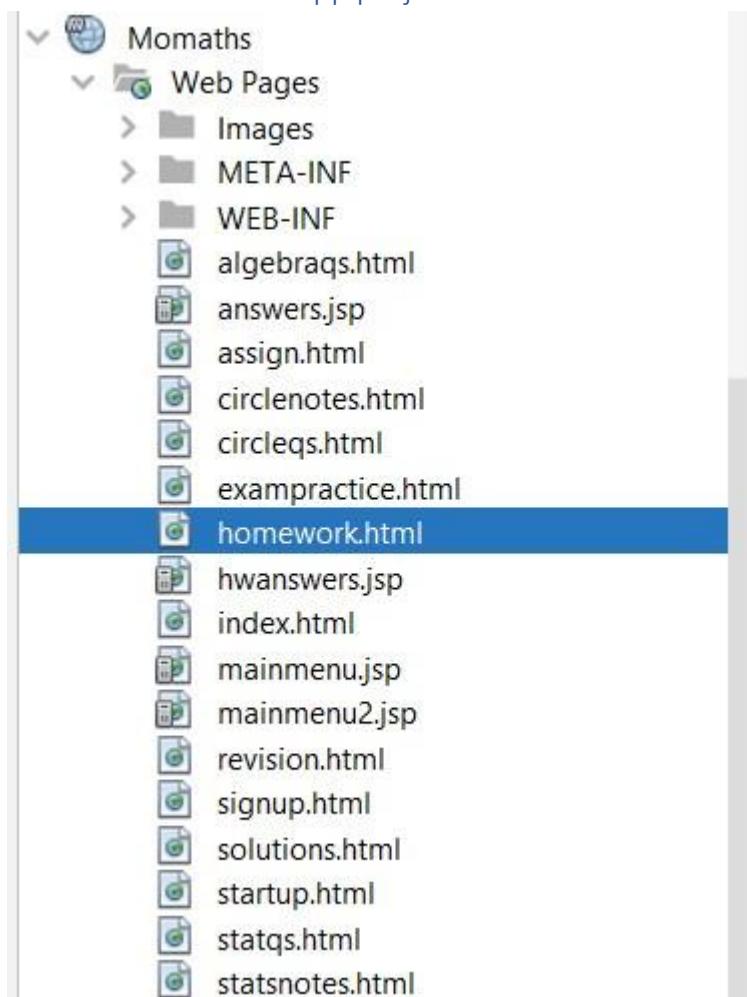
getUserID:integer

getUsername:String

getClassroomname:String

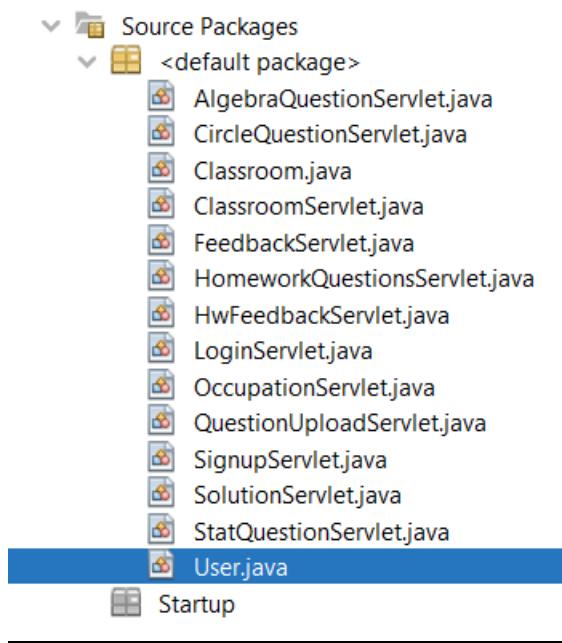
## Technical Solution

Momaths revision webapp project



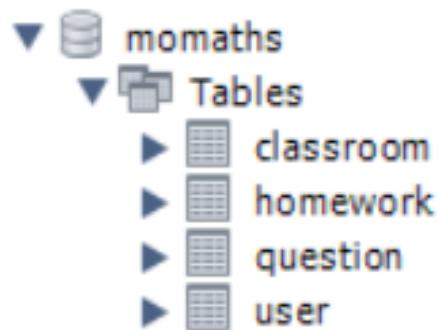
Above are all the html pages and JSPs (Java Servlet Pages) which I have used for the different screens in the User Interface.

(NOTE: The index.html page is the login page and is the first page the user will see when they run the application.)

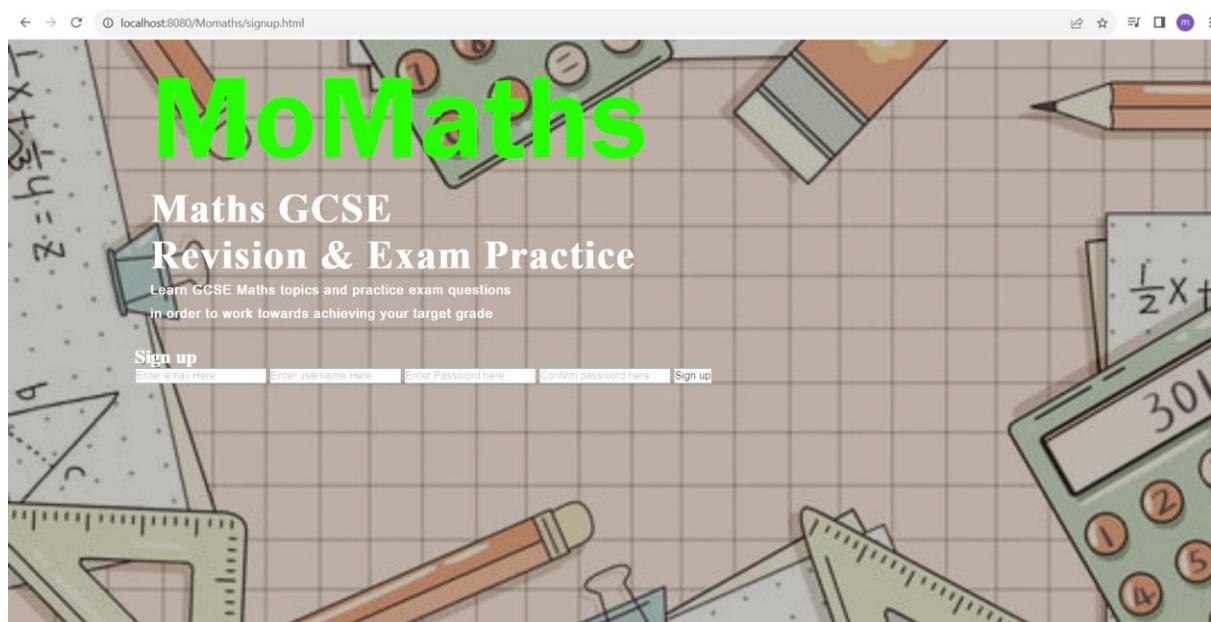


Due to my user interface design being in HTML, my project uses many servlet classes in order to handle HTTP requests from the browser and responses from the server.

## MySQL database tables



## Signup.html



```
7   </head>
8   <body>
9
10  <div class="main">
11    <div class="navbar">
12
13      <div class="icon">
14        <h2 class="logo">MoMaths</h2>
15      </div>
16    </div>
17    <div class="content">
18      <h1>Maths GCSE<br><span>Revision & Exam Practice</span></h1>
19      <p class="par">Learn GCSE Maths topics and practice exam questions<br>
20          in order
21          to work towards achieving your target
22          grade</p>
23
24      <div class="form">
25        <h2>Sign up</h2>
26        <form action="/Momaths/Signup" method="post">
27          <input type="email" name="email" placeholder="Enter email Here">
28          <input type="uname" name="username" placeholder="Enter username Here">
29          <input type="password" name="password" placeholder="Enter Password here">
30          <input type="password" name="confirmpassword" placeholder="Confirm password here">
31          <button class="btn" type="submit">Sign up</button>
32        </form>
33
34
35      </div>
36    </div>
37  </div>
38  <style>
```

## SignupServlet.java

```
1 import java.io.IOException;
2 import java.io.PrintWriter;
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import javax.servlet.ServletException;
6 import javax.servlet.http.HttpServlet;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9 import javax.servlet.annotation.WebServlet;
10 import java.sql.SQLException;
11 import java.sql.PreparedStatement;
12 import javax.servlet.http.HttpSession;
13
14
15
16 @WebServlet(urlPatterns = {"/Signup"})
17 public class SignupServlet extends HttpServlet {
18
19     @Override
20     protected void doPost(HttpServletRequest request, HttpServletResponse response)
21             throws ServletException, IOException {
22         response.setContentType("text/html");
23         PrintWriter out = response.getWriter();
24         HttpSession session = request.getSession();
25
26         // Retrieve form data from the request
27         String email = request.getParameter("email");
28         String password = request.getParameter("password");
29         String username = request.getParameter("username");
30         String confirmPassword = request.getParameter("confirmpassword");
31         session.setAttribute("username", username);
32
33         // Check if passwords match
34         if (!password.equals(confirmPassword)) {
35             out.println("Passwords do not match. Please try again.");
36             return; // Exit the method after forwarding the request
37     }
```

In this first part of the Servlet the values entered in the email, username, password and confirm password fields are being gotten as parameters and stored in the String variables.

The first if statement checks if the value in the password and confirm password fields match. If they do not an error message will display.

```
39     else if(username.length()<8||password.length()<8){//if username or password are less than 8 characters
40         out.println("username and password must be at least 8 characters long. Please try again");
41         return;
42     }
43     else if("".equals(username)||"".equals(password)||email.equals("")){//if username or password or email are empty
44         out.println("Cannot sign up with one or more empty fields.");
45         return;
46     }
47
48     // Database operations
49     try {
50         User user=new User();
51         String hashedpassword=user.Hashpassword(password);
52         // Open connection to the database
53         Class.forName("com.mysql.jdbc.Driver");
54         Connection con = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/momaths", "root", "root123");
55         PreparedStatement statement=con.prepareStatement("SELECT username FROM USER WHERE username=?");
56         statement.setString(1, username);
57         ResultSet result=statement.executeQuery();
58         if(result.next()){
59             out.println("Username already exists. Please enter a different username.");
60             return;
61         }
62
63         // Insert user information into the "User" table
64         PreparedStatement pstmt = con.prepareStatement("INSERT INTO USER (email,username, password) VALUES (?, ?, ?)");
65         pstmt.setString(1, email);
66         pstmt.setString(2, username);
67         pstmt.setString(3, hashedpassword);
68
69         int rowsAffected = pstmt.executeUpdate();
70
71         if (rowsAffected > 0) {
72             response.sendRedirect("startup.html");
73         } else {
74             out.println("Registration failed. Please try again later.");
75         }
76     }
```

The next if statement is to check that the password is greater than or equal to 8 characters and the one after is to check that none of the fields are empty.

The code then checks if the username entered already exists by selecting usernames from user where username= the username parameter. If a row is returned this means the username already exists, otherwise it does not and then the user's email username and password are inserted into the database.

If the details entered by the user meet all the requirements, the email username and (hashed)password are inserted into the user table using the INSERT query.

```
75 }  
76 }  
77     con.close(); // Close the database connection  
78 } catch (ClassNotFoundException | SQLException e) {  
79     System.out.println(e.getMessage());  
80 }  
81 }  
82 }  
83 }
```

User table

	UserID	Username	Password	Occupation	RevisionPlan	Email	Classroomname

Startup.html



```
<body>

    <div class="main">
        <div class="navbar">

            <div class="icon">
                <h2 class="logo">MoMaths</h2>
            </div>
        </div>
        <div class="content">
            <h1>Getting Started</h1>
            <h2>what is your occupation?</h2>
            <form action="/Momaths/Startup" method="post" onsubmit="return submitForm();">
                <label class="container">Student
                    <input type="radio" name="occupation" value="student">
                    <span class="checkmark"></span>
                </label>

                <label class="container">Teacher
                    <input type="radio" name="occupation" value="teacher">
                    <span class="checkmark"></span>
                </label>

                <button class="btn" type="submit">Next</button>
            </form>

            <script>
                function submitForm() {
                    // Return true to allow the default form submission
                    return true;
                }
            </script>
        </div>
    </div>

```

Creates two checkboxes. One labelled student and the other labelled teacher.

### OccupationServlet.java

```
1 import java.io.IOException;
2 import java.io.PrintWriter;
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import javax.servlet.ServletException;
9 import javax.servlet.annotation.WebServlet;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13 import javax.servlet.http.HttpSession;
14
15
16 @WebServlet(urlPatterns = {"/Startup"})
17 public class OccupationServlet extends HttpServlet {
18
19     @Override
20     protected void doPost(HttpServletRequest request, HttpServletResponse response)
21             throws ServletException, IOException {
22         PrintWriter out = response.getWriter();
23         // Retrieve the username attribute from the session
24         HttpSession session = request.getSession();
25         String username = (String) session.getAttribute("username");
26         String occupation = request.getParameter("occupation");
27         Classroom classroom=new Classroom();
28
29         try {
30             // Open connection to the database
31             Class.forName("com.mysql.jdbc.Driver");
32             Connection con = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/momaths", "root", "root123");
33
34
35         try{
36             // Insert user information into the "User" table
37         }
```

In this part we simply create an object called classroom from the Classroom class (in other words we create an instant of the Classroom class). We do this in order to use the methods within the classroom object.

We have also gotten the occupation (student or teacher) as a parameter and stored it in the Occupation String.

```

37
38 // Insert user information into the "User" table
39 PreparedStatement select = con.prepareStatement("SELECT UserID FROM USER WHERE Username=?");
40 select.setString(1,username);
41 ResultSet result=select.executeQuery();
42 int userId = -1; // Initialize to a default value
43
44 if (result.next()) {
45     userId = result.getInt("UserID");
46     // You can also check if userId is not equal to -1 to ensure a valid result
47 }
48 PreparedStatement pstmt = con.prepareStatement("UPDATE USER SET Occupation=? WHERE UserID=?");
49 pstmt.setString(1, occupation);
50 pstmt.setInt(2, userId);
51 pstmt.executeUpdate();
52 if(occupation!=null){
53     if (occupation.equals("teacher")) {
54         // The user has chosen "teacher" as their occupation, so create a classroom
55         String classroomcode = classroom.generateUniqueClassroomCode(); // generate a unique code
56         String className = username+"s classroom";
57         classroom.createClassroom(con, userId, classroomcode, className);
58         classroom.AddtoClassroom(userId, username, className, con);
59         session.setAttribute("classname",className);
60         response.sendRedirect("mainmenu2.jsp");
61     } else if(occupation.equals("student")){
62         response.sendRedirect("mainmenu.jsp");
63     }
64 } else{
65     out.println("must select occupation to continue.");
66 }
67
68 }
69 catch (SQLException e) {
70     // Handle SQL update query errors here
71     System.out.println(e.getMessage());
72 }
73

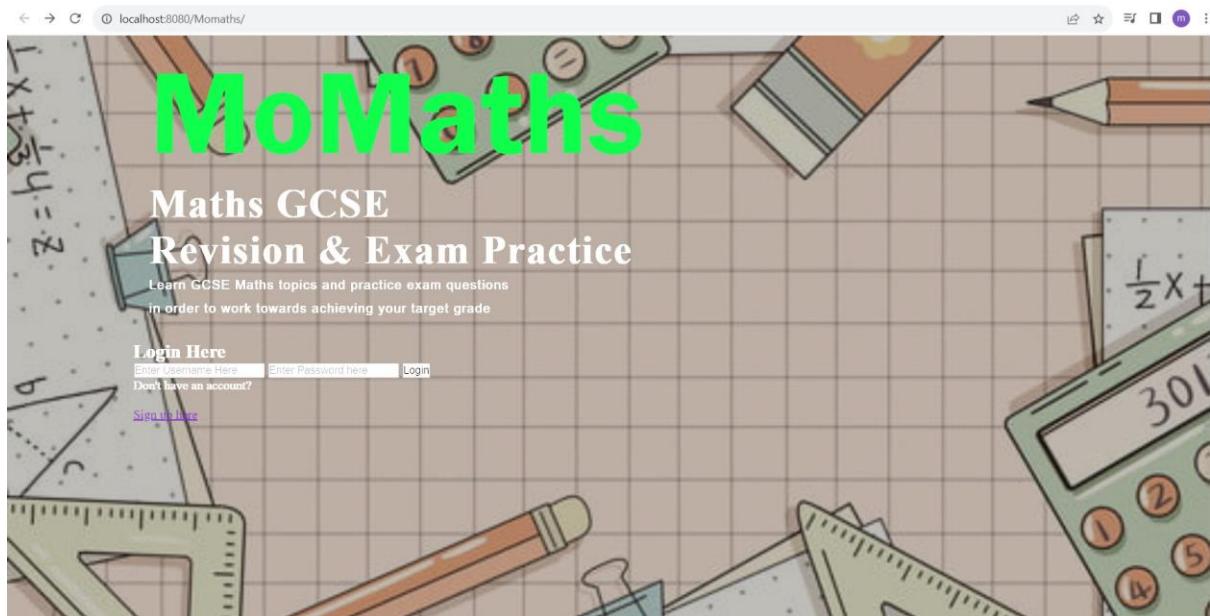
```

The student can either tick the student or teacher box and the value of the box they tick is gotten as a parameter by the servlet and the Occupation field in the user table is updated. If the account's occupation is a teacher their classroom is automatically created using the createClassroom method from the classroom object and the teacher is also added to the classroom using the AddtoClassroom method from the classroom object . If the occupation is student the user is directed to a different main menu. This check is only performed if occupation is not null(if the user has ticked a box).

```

77     } catch (ClassNotFoundException e) {
78         System.out.println(e.getMessage());
79     } catch (SQLException e) {
80         System.out.println(e.getMessage());
81     }
82 }
83
84 }
```

Index.html(login page)



```

<html>
  <head>
    ...
  </head>
  <body>

    <div class="main">
      <div class="navbar">
        ...
      </div>
      <div class="content">
        <h1>Maths GCSE<br><span>Revision & Exam Practice</span></h1>
        <p>Learn GCSE Maths topics and practice exam questions<br>
          in order<br>
          to work towards achieving your target<br>
          grade</p>

        <div class="form">
          <h2>Login Here</h2>
          <form action="/Momaths/Login" method="post">
            <input type="username" class="form-control" name="username" placeholder="Enter Username Here">
            <input type="password" class="form-control" name="password" placeholder="Enter Password here">
            <button class="btn" type = "submit" value = "Submit">Login</button>
          </form>
          <p><a href="#">Don't have an account?</a><br>
          <a href="signup.html">Sign up here</a>
        </div>
      </div>
    </div>
  </body>
</html>
```

## LoginServlet.java

```
1  ava.io.IOException;
2  ava.io.PrintWriter;
3  ava.sql.Connection;
4  ava.sql.DriverManager;
5  ava.sql.PreparedStatement;
6  ava.sql.ResultSet;
7  ava.sql.SQLException;
8  ava.util.ArrayList;
9  avax.servlet.RequestDispatcher;
10 avax.servlet.ServletException;
11 avax.servlet.annotation.WebServlet;
12 avax.servlet.http.HttpServlet;
13 avax.servlet.http.HttpServletRequest;
14 avax.servlet.http.HttpServletResponse;
15 avax.servlet.http.HttpSession;
16
17 let(urlPatterns = {"/Login"})
18 class LoginServlet extends HttpServlet {
19
20     ride
21     ressWarnings("ConvertToTryWithResources")
22     @Override
23     protected void doPost(HttpServletRequest request, HttpServletResponse response)
24         throws ServletException, IOException {
25         ArrayList<String> studentnames=new ArrayList<>(); //create the arraylist which will store the names of the students in the classroom
26
27         response.setContentType("text/html;charset=UTF-8");
28         PrintWriter out = response.getWriter();
29         Classroom classroom=new Classroom();
30
31         // Get user input from the login form
32         String username = request.getParameter("username");
33         String password = request.getParameter("password");
34 }
```

1. An ArrayList is created to store the names of each student in the classroom
2. Create an instance of the Classroom class (Create classroom object)
3. Get username and password entered by user as parameters and store in username and password Strings

```

35 // Database connection parameters
36 String jdbcURL = "jdbc:mysql://127.0.0.1:3306/momaths";
37 String dbUser = "root"; // Update with database username
38 String dbPassword = "root123"; // Update with database password
39
40 try {
41     User user=new User();
42     String hashedpassword=user.Hashpassword(password); //hash the password entered by the user
43     // Load the MySQL JDBC driver
44     Class.forName("com.mysql.jdbc.Driver");
45
46
47     try ( // Establish a database connection
48         Connection connection = DriverManager.getConnection(jdbcURL, dbUser, dbPassword)) {
49
50         HttpSession session = request.getSession(); //Create a session
51
52         // SQL query to check if the username and the hashed version of the entered password match any of the entries in the database
53         String sql = "SELECT * FROM USER WHERE Username = ? AND Password = ?";
54         PreparedStatement statement = connection.prepareStatement(sql);
55         statement.setString(1, username);
56         statement.setString(2, hashedpassword);
57
58         ResultSet result = statement.executeQuery();
59
60         if (result.next()) {
61             String occupation=result.getString("Occupation");
62             if(occupation.equals("teacher")){
63                 response.sendRedirect("mainmenu2.jsp"); //redirect to different menu if user is a teacher
64                 String ClassroomName=classroom.getClassName(username, connection); //creates the name of the classroom
65                 PreparedStatement stmt=connection.prepareStatement("SELECT Username FROM USER WHERE Classroomname=? AND Occupation='student'");
66                 stmt.setString(1, ClassroomName);
67                 //selects usernames of students who are in the same classroom as the teacher
68                 ResultSet rs=stmt.executeQuery();

```

.Create the user object/ instance of User class

.Use the Hashpassword method in the user object to hash the password

```

14
15     public String Hashpassword(String password) {
16         try {
17             MessageDigest md = MessageDigest.getInstance("MD5");
18             byte[] passwordBytes = password.getBytes();
19             byte[] hashedPasswordBytes = md.digest(passwordBytes);
20
21             // Convert the byte array to a hexadecimal string
22             StringBuilder hexString = new StringBuilder();
23             for (byte b : hashedPasswordBytes) {
24                 hexString.append(String.format("%02x", b));
25             }
26
27             return hexString.toString();
28         } catch (NoSuchAlgorithmException e) {
29             e.printStackTrace();
30             return null;
31         }
32     }

```

.selects all fields from username where username and password are set as the parameters. If result.next() is true (If a row is retrieved) this means an account with the entered username and password

exists so the user is allowed to login. Before this the occupation must be checked to determine whether the user is a student or a teacher.

.Occupation is retrieved from the “SELECT \* FROM USER” SQL query

.If occupation is teacher then direct to the teacher menu and use an SQL query to select all the students in the classroom

```
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
```

```
    while(rs.next()){
        String studentname=rs.getString("Username");
        studentnames.add(studentname);

        } user.sortNames(studentnames);
        // stores each name that is selected by the query in an arraylist called studentnames
        session.setAttribute("studentnames",studentnames);
        //store the studentnames arraylist in the session so we can use this attribute in mainmenu2.jsp(the teacher menu)

    }

else{
    // Successful login
    // Store the username in the session

    out.println("Login successful! Welcome, " + username);
}

session.setAttribute("username", username);
boolean userinclassroom=classroom.UserinClassroom(username, connection);
//boolean variable which verifies whether the user is in the classroom or not
if(occupation.equals("student")){
    if(userinclassroom){
        request.getSession().setAttribute("UserinClassroom", userinclassroom);
    }
    // Forward the request to mainmenu.jsp
    RequestDispatcher dispatcher = request.getRequestDispatcher("mainmenu.jsp");
    dispatcher.forward(request, response);
}
} else {
    // Invalid username or password
    out.println("Invalid username or password. Please try again.");
}
```

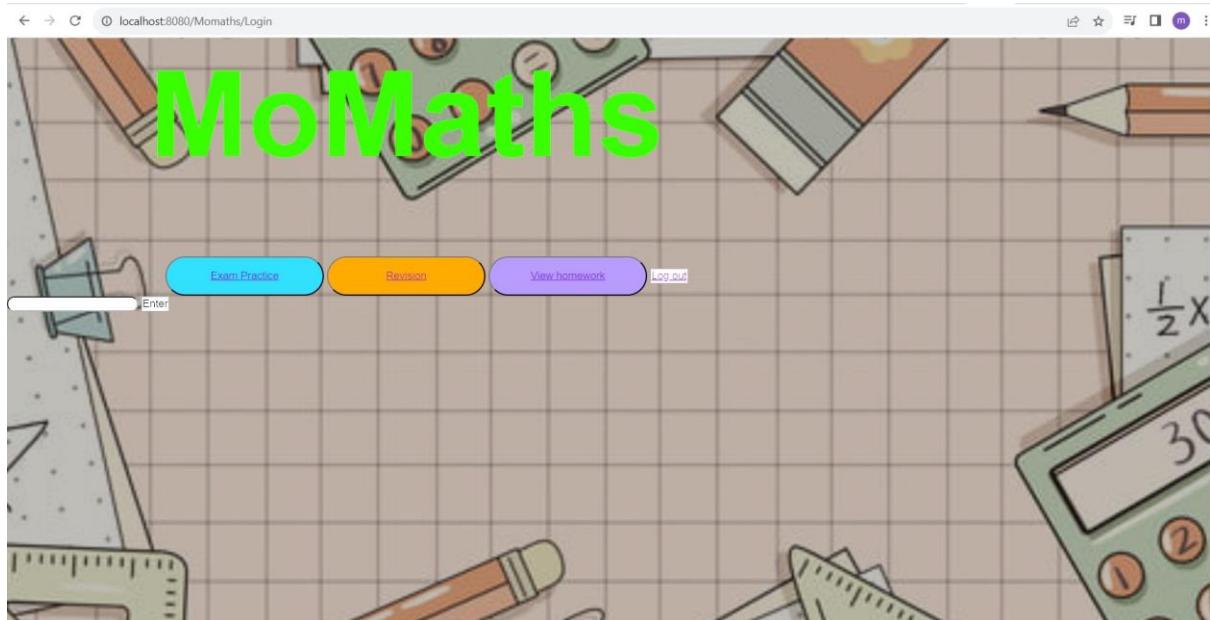
1. While there is another student in the classroom add them to the studentnames ArrayList
2. Use the sortnames method in the user object to sort the names in alphabetical order and store it in a session so it can be displayed on the jsp page
3. If the occupation is student direct the user to mainmenu.jsp (the student main menu)

```
38 | public void sortNames(List<String> names){  
39 |     for(int i=0;i<names.size()-1;i++){  
40 |         if(names.get(i).compareTo(names.get(i+1))>0){  
41 |             String temp=names.get(i);  
42 |             names.set(i,names.get(i+1));  
43 |             names.set(i+1,temp);  
44 |             //if the item before the next item in the list comes alphabetically after it then they swap positions|  
45 |         }  
46 |     }  
47 | }
```

For each name in the arraylist the name at the current index is compared to the name at the index after it. If the item before the next item in the arraylist comes alphabetically after it, then the names swap positions. This repeats until the names are fully sorted in alphabetical order.

```
105
106
107
108
109
110
111     } catch (ClassNotFoundException | SQLException ex) {
112         out.println("Database connection error: " + ex.getMessage()); //if there is an exception print the error
113     }
114 }
115
116
117 }
```

## Mainmenu.jsp(Student menu)



```
<html>
  <head>

    </head>
    <body>

      <div class="main">
        <div class="navbar">|
```

MoMaths

```
          <div class="icon">
            <h2 class="logo">MoMaths</h2>
          </div>
        </div>
        <button class="exp"><a href="exampractice.html">Exam Practice</a></button>
        <button class="rev"><a href="revision.html">Revision</a></button>
```

```
      <%
        // Check the session attribute "DisplayButton"
        Boolean userInClassroom = (Boolean) request.getSession().getAttribute("UserinClassroom");

        if (userInClassroom==true) {
      
```

View homework

```
        <button class="hw"><a href="homework.html">View homework</a></button>
      
```

```
      <%
        }
      <%>
```

```
      <button id="viewHomeworkButton" class="hw" style="display: none"><a href="homework.html">View homework</a></button>

      <form action="/Momaths/ClassroomServlet" method="Post">
        <input type="class" class="classroom" name="classroom">
        <button class="btn" type = "submit" value = "Submit">Enter</button>
      </form>
```

```
    </div>
  </body>
</html>
```

## Mainmenu2.jsp(Teacher menu)



```
<%@page import="java.util.List"%>
<%@ page import="java.util.ArrayList" %>
<!DOCTYPE html>

<html>
    <head>

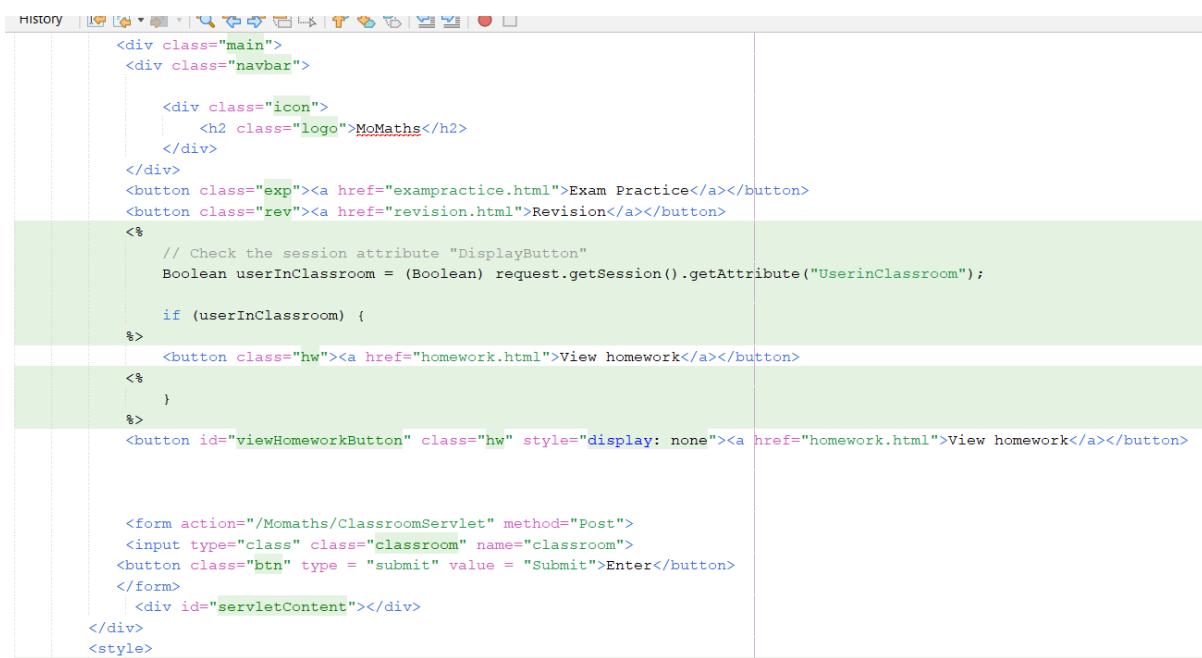
        </head>
        <body>

            <div class="main">
                <div class="navbar">

                    <div class="icon">
                        <h2 class="logo">MoMaths</h2>
                    </div>
                </div>
                <button class="exp"><a href="exampractice.html">Exam Practice</a></button>
                <button class="rev"><a href="revision.html">Revision</a></button>
                <button class="ass"><a href="assign.html">Assign questions</a></button>
            </div>
        </body>
    </html>
```

```
<%
// Retrieve the attribute from the session
List<String> studentnames = (List<String>) session.getAttribute("studentnames");
%>
<%
if (studentnames != null && !studentnames.isEmpty()) {
    out.println("<ul>");
    out.println("<p>your classroom:</p>");
    for (String name : studentnames) {
        out.println("<li>" + name + "</li>");
    } //display each name from the studentnames arraylist
    out.println("</ul>");
} else {
    out.println("No student names found in session.");
}
%>
```

The studentnames arraylist is retrieved from the HTTP session and displayed on the page as a list.



The screenshot shows a web browser window with the following code structure:

```
History |               
```

```
<div class="main">
<div class="navbar">

    <div class="icon">
        <h2 class="logo">MoMaths</h2>
    </div>
    <button class="exp"><a href="exampractice.html">Exam Practice</a></button>
    <button class="rev"><a href="revision.html">Revision</a></button>
<%
    // Check the session attribute "DisplayButton"
    Boolean userInClassroom = (Boolean) request.getSession().getAttribute("UserinClassroom");

    if (userInClassroom) {
%>
        <button class="hw"><a href="homework.html">View homework</a></button>
<%
    }
%>
<button id="viewHomeworkButton" class="hw" style="display: none"><a href="homework.html">View homework</a></button>

    <form action="/Momaths/ClassroomServlet" method="Post">
        <input type="class" class="classroom" name="classroom">
        <button class="btn" type="submit" value="Submit">Enter</button>
    </form>
    <div id="servletContent"></div>
</div>
<style>
```

A green rectangular highlight is placed over the line of code containing the button element with the ID "viewHomeworkButton". The browser interface includes standard navigation buttons like back, forward, and search, along with tabs and a history menu.

On the student menu there is a text box to enter the code to join a classroom. If the code is correct(matches a code within the classroom database table) then the user is added to the classroom.

## ClassroomServlet.java

```
1 import java.io.IOException;
2 import java.io.PrintWriter;
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import javax.servlet.ServletException;
8 import javax.servlet.annotation.WebServlet;
9 import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12 import javax.servlet.http.HttpSession;
13
14
15 @WebServlet(urlPatterns = {"/ClassroomServlet"})
16 public class ClassroomServlet extends HttpServlet {
17
18
19
20     @Override
21     protected void doPost(HttpServletRequest request, HttpServletResponse response)
22         throws ServletException, IOException {
23         Classroom classroom=new Classroom();
24         response.setContentType("text/html;charset=UTF-8");
25         PrintWriter out = response.getWriter();
26         String teachername=""; //initialises teachername to empty String
27         HttpSession session = request.getSession(); //creates session
28         String code=request.getParameter("classroom"); //gets the code entered by the user as a parameter
29         String studentname=(String) session.getAttribute("username"); //retrieves the student's username from the session
30     }
}
```

The username is retrieved from the http session and stored in the studentname String, the value entered by the user in the classroom code box is stored in the code String.

```

31
32     try{
33         Class.forName("com.mysql.jdbc.Driver");
34         Connection con = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/momaths", "root", "root123");
35         String sql = "SELECT TeacherID FROM CLASSROOM WHERE Classroomcode=?";
36         PreparedStatement statement = con.prepareStatement(sql);
37         statement.setString(1, code); //selects row where code entered by user matches a code in the classroom table
38         int UserID=classroom.GetStudentID(con, studentname);// retrieves the UserID of the student
39
40
41         ResultSet result = statement.executeQuery();
42
43         if(result.next()){// if a row from the database was retrieved
44             int teacherid=result.getInt("TeacherID");//the teacherid variable is set to the teacherID retrieved from the query
45             teachername=classroom.getTeachername(con,teacherid);
46             String classname=teachername+"'s classroom";
47             session.setAttribute("classroomname", classname);
48             out.println("<p>joined "+classname);
49             classroom.AddtoClassroom(UserID, studentname,classname,con); //adds the student to the classroom using the AddtoClassroom method
50         }
51     } catch(Exception e){
52         e.printStackTrace();
53     }
54
55
56
57
58
59
60
61 }

```

The sql query selects teacherID from the classroom table where the code entered by the user is set as the parameter for the Statement. If a row is returned then the teachername is retrieved from the getTeachername method and a message is displayed telling the user that they have joined the classroom.

Classroom table

	ClassroomID	TeacherID	Classroomname	Classroomcode
--	-------------	-----------	---------------	---------------

## Classroom.java

```
19  public class Classroom {
20
21     public Classroom() {
22
23     }
24     public void Dbconnection(){
25         try{
26             Class.forName("com.mysql.jdbc.Driver");
27             Connection con = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/momaths", "root", "root123");
28         } catch(Exception e){
29             e.printStackTrace();
30         }
31
32     }
33
34     public String generateUniqueClassroomCode() {
35         Dbconnection();
36         // Implement this function to generate a unique classroom code
37         // You can use a combination of letters and numbers or any desired format
38         // For simplicity, let's use a random 6-character alphanumeric code
39         String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
40         StringBuilder code = new StringBuilder();
41         Random random = new Random();
42         for (int i = 0; i < 6; i++) {
43             code.append(characters.charAt(random.nextInt(characters.length())));
44         }
45         return code.toString();
46     }

```

In this snippet of code there are two methods. The Dbconnection method, which simply establishes a connection to the mySQL database. The generateUniqueClassroomCode method uses a random selection of 6 characters

```
public void createClassroom(Connection connection, int userId, String classroomcode, String classroomName) throws SQLException {
    Dbconnection();
    // Implement this function to create a classroom and save it in the database
    PreparedStatement classroomPstmt = connection.prepareStatement("INSERT INTO classroom (Classroomcode,TeacherID,Classroomname) VALUES (?, ?, ?)");
    classroomPstmt.setString(1, classroomcode);
    classroomPstmt.setInt(2, userId);
    classroomPstmt.setString(3, classroomName);
    classroomPstmt.executeUpdate();
}
public String getTeachername(Connection con,int userid) throws SQLException{
    Dbconnection();
    String username="";
    String sql = "SELECT USER.username FROM USER JOIN CLASSROOM ON CLASSROOM.TeacherID=USER.UserID WHERE UserID=?";
    Preparedstatement statement = con.prepareStatement(sql);
    statement.setInt(1 , userid);
    Resultset result=statement.executeQuery();
    if(result.next()){
        username=result.getString("username");
    }
    return username;
}
```

In this screenshot of code there are two methods, the createClassroom method and the getTeachername method. In the createClassroom method, the TeacherID, the Classroomcode and the Classroomname are inserted into the database creating a new entry in the database when the Classroom is created.

In the getTeachername method, as given by the name the method gets the username of the teacher using a cross-table query which selects the username of the teacher in a specific classroom by selecting the username from the user table where the teacherID in the classroom table matches the userID in the user table and where the UserID matches the UserID of the user.

```
public void AddtoClassroom(int userID, String username, String classroomname, Connection con) throws SQLException {
    Dbconnection();
    String sql = "UPDATE USER SET Classroomname = ? WHERE UserID = ?";

    try (PreparedStatement statement = con.prepareStatement(sql)) {
        // Set the parameters using setString for classroomname and setInt for userID
        statement.setString(1, classroomname);
        statement.setInt(2, userID);

        // Execute the update statement
        int rowsUpdated = statement.executeUpdate();

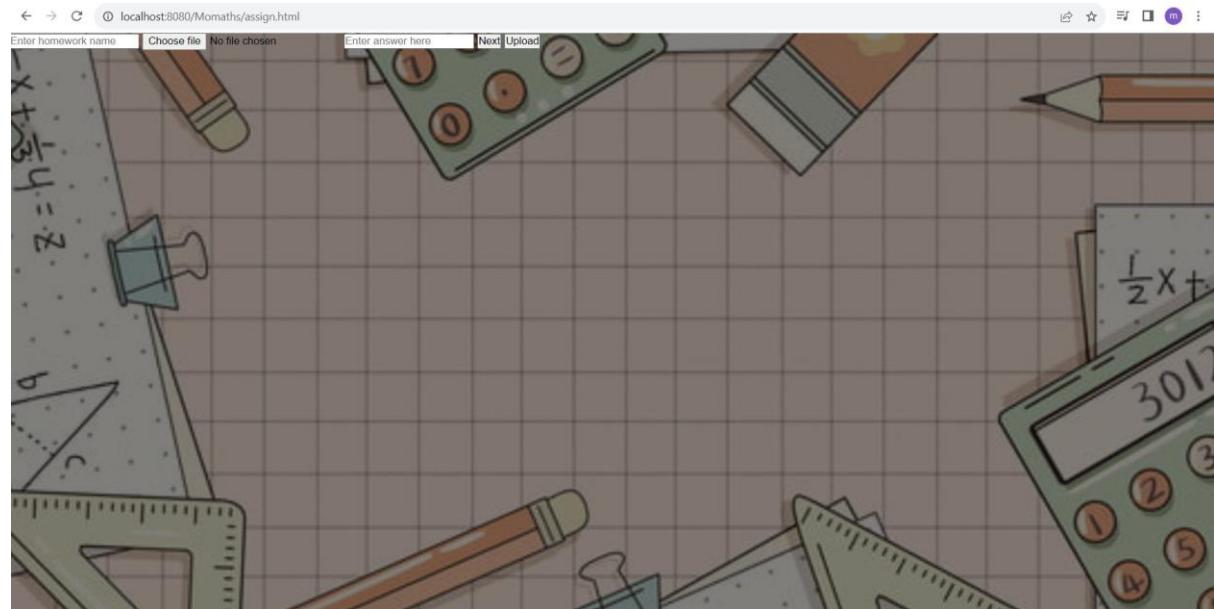
    } catch (SQLException e) {
        // Handle any exceptions here
        e.printStackTrace();
    }
}
```

```
public int GetStudentID(Connection con, String username) throws SQLException{
    Dbconnection();
    int userID=0;
    PreparedStatement classroomPstmt = con.prepareStatement("SELECT UserID FROM USER WHERE username=?");
    classroomPstmt.setString(1, username);
    ResultSet result=classroomPstmt.executeQuery();
    if(result.next()){
        userID=result.getInt("UserID");
    }
    return userID;
}
public boolean UserinClassroom(String username, Connection con) throws SQLException{
    Dbconnection();
    String classroomname="";
    PreparedStatement statement=con.prepareStatement("SELECT Classroomname FROM USER WHERE username=?");
    statement.setString(1, username);
    ResultSet result=statement.executeQuery();
    if(result.next()){
        classroomname=result.getString("Classroomname");
    }
    PreparedStatement stmt=con.prepareStatement("SELECT * FROM momaths.user JOIN momaths.classroom ON user.Classroomname = classroom.Classroomname");
    //stmt.setString(1, classroomname);
    ResultSet resultset=stmt.executeQuery();
    if(resultset.next()){
        return true;
    }
    else{
        return false;
    }
}
```

The UserinClassroom method returns a Boolean value which verifies whether a user is in a classroom. This is so the system knows which users to display homework questions assigned by the teacher to.

The GetStudentID method is also used here which uses the SQL query to select the UserID where the username is equal to the parameter being passed into the method.

## Assign.html



```
<div class="main">
<div class="navbar">

    <div class="icon">
        <h2 class="logo">MoMaths</h2>
    </div>
</div>
<button class="exp"><a href="exampractice.html">Exam Practice</a></button>
<button class="rev"><a href="revision.html">Revision</a></button>
<button class="ass"><a href="assign.html">Assign questions</a></button>

<body>
    <form action="/MoMaths/QuestionUploadServlet" method="post" enctype="multipart/form-data" onsubmit="submitForm();">
        <input type="text" name="hwname" placeholder="Enter homework name">
        <input type="file" name="imageFile">
        <input type="text" name="hwanswer" placeholder="Enter answer here">
        <button class="next" name="next">Next</button>
        <input type="submit" value="Upload">
    </form>
    <script>
        function submitForm() {
            // Clear the previously uploaded file, if any
            form.reset();
        }
    </script>
</body>
```

This HTML code allows teachers to upload questions as image files and also enter the name of a homework and the answer for each question.

## QuestionUploadServlet.java

```
1 import java.io.File;
2 import java.io.FileOutputStream;
3 import java.io.IOException;
4 import java.io.InputStream;
5 import java.sql.Connection;
6 import java.sql.DriverManager;
7 import java.sql.PreparedStatement;
8 import java.sql.ResultSet;
9 import javax.servlet.ServletException;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13 import javax.servlet.http.HttpSession;
14 import javax.servlet.http.Part;
15 import javax.servlet.annotation.MultipartConfig;
16 import javax.servlet.annotation.WebServlet;
17
18 @WebServlet(urlPatterns = {"/QuestionUploadServlet"})
19 @MultipartConfig(
20     location = "C:/Users/muham/OneDrive/Documents/NetBeansProjects/Momath/src/main/webapp/Images/hwqs/", // Temporary location to store uploaded files
21     fileSizeThreshold = 1024, // File size threshold in bytes
22     maxFileSize = 20848820, // Max file size in bytes
23     maxRequestSize = 418018841) // Max request size in bytes
24 public class QuestionUploadServlet extends HttpServlet {
25
26     @Override
27     protected void doPost(HttpServletRequest request, HttpServletResponse response)
28         throws ServletException, IOException {
29         HttpSession session = request.getSession();
30         String name=request.getParameter("hwname");
31         String answer=request.getParameter("hwanswer");
32         String teachername=(String) session.getAttribute("username");
33         //String uploadFolder = getServletContext().getRealPath("/Images/hwqs");
34         String uploadFolder = "C:/Users/muham/OneDrive/Documents/NetBeansProjects/Momath/src/main/webapp/Images/hwqs/";
35         String folderPath = "Images/hwqs/";
36         int teacherid=0;
```

The values entered in the hwname and hwanser fields are retrieved from the HTTP request and the username of the teacher is retrieved from the HTTP session.

UploadFolder is the path where the uploaded files will be stored on the server.

The filePath String is the file path for a question that will be stored in the database.

```

37
38     // Get the Part object containing the uploaded file
39     Part filePart = request.getPart("imageFile");
40     String SubmittedfileName = filePart.getSubmittedFileName();
41
42     // File path to insert into db
43     String filePath = folderPath + SubmittedfileName;
44     //File path to upload to
45     String uploadFilePath = uploadFolder + SubmittedfileName;
46
47     // Create the folder if it doesn't exist
48     File folder = new File(uploadFolder);
49     if (!folder.exists()) {
50         folder.mkdirs();
51     }
52
53     // Create an output stream to write to the destination file
54     try (InputStream fileContent = filePart.getInputStream());
55     {
56         FileOutputStream fos = new FileOutputStream(uploadFilePath);
57         int read;
58         final byte[] bytes = new byte[1024];
59         while ((read = fileContent.read(bytes)) != -1) {
60             fos.write(bytes, 0, read);
61         }
62     }
63     try {
64         Class.forName("com.mysql.jdbc.Driver");
65         Connection con = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/momaths", "root", "root123");
66         String sql = "SELECT UserID,Classroomname FROM USER WHERE username=?";
67         PreparedStatement statement = con.prepareStatement(sql);
68         statement.setString(1, teachername);
69
70         ResultSet result=statement.executeQuery();

```

filePath stores the filepath of the question that is uploaded into the folder.

The SQL query then selects the UserID and Classroomname where the parameter is set as username.

```

72
73     if(result.next()){
74         teacherid=result.getInt("UserID");//get the UserID selected by the query
75         String classroomname=result.getString("Classroomname");//get the Classroomname selected by the query
76         PreparedStatement pstmt = con.prepareStatement("INSERT INTO HOMEWORK (question,Answer,TeacherID,HomeworkName,Classroomname) VALUES (?, ?, ?, ?, ?)");
77         pstmt.setString(1, filePath); //Set filepath variable
78         pstmt.setString(2, answer); // Set your answer variable
79         pstmt.setInt(3, teacherid); // Set your teacherid variable
80         pstmt.setString(4, name); // Set your name variable
81         pstmt.setString(5, classroomname);// Set your classroomname variable
82         int rowsAffected = pstmt.executeUpdate();
83
84     }
85     session.setAttribute("teacherid", teacherid);
86
87 } catch (Exception e) {
88     response.getWriter().println("Error: " + e.getMessage());
89 }

```

If there are results from the query the UserID and Classroomname are retrieved from the query and are inserted into the homework table along with the question filepath, the answer and the name of the homework.

## Homework.html

The screenshot shows a web page titled "homework1" with a background illustration of school supplies like pencils, a calculator, and a ruler. The page contains two math problems:

**Problem 1:**  
There are some red counters and some blue counters in a bag.  
The ratio of red counters to blue counters is 3:1.  
Two counters are removed at random.  
The probability that both the counters taken are blue is  $\frac{1}{20}$   
Work how many counters were in the bag before any counters were removed.

**Problem 2:**  
There are some red counters and some blue counters in a bag.  
The ratio of red counters to blue counters is 4:1.  
Two counters are removed at random.  
The probability that both the counters taken are red is  $\frac{22}{35}$   
Work how many blue counters are in the bag.

Each problem has a text input field labeled "Enter answer Here".

```
<div class="main">
    <div class="navbar">
        <div class="icon">
            </div>
            </div>
</div>
</div>

// Load HTML content from servlet via AJAX
var htmlContentXhr = new XMLHttpRequest();
htmlContentXhr.open("GET", "HomeworkQuestionsServlet", true);
htmlContentXhr.onreadystatechange = function () {
    if (htmlContentXhr.readyState === 4 && htmlContentXhr.status === 200) {
        document.getElementById("hwContent").innerHTML = htmlContentXhr.responseText;
    }
};
htmlContentXhr.send();
</script>
<script>
function changeStyle() {
    var pElement = document.querySelector('p');
    pElement.style.color = 'green';
    pElement.textContent = 'Correct!';
}
</script>
```

The homework.html page uses the javascript method to display the html content of the HomeworkQuestionServlet class.

#### HomeworkQuestionServlet.java

```
1 import java.io.IOException;
2 import java.io.PrintWriter;
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.util.ArrayList;
8 import javax.servlet.ServletException;
9 import javax.servlet.annotation.WebServlet;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13 import javax.servlet.http.HttpSession;
14
15
16 @WebServlet(urlPatterns = {" /HomeworkQuestionsServlet"})
17 public class HomeworkQuestionsServlet extends HttpServlet {
18     ArrayList<String> questionPaths = new ArrayList<>();
19     ArrayList<String> hwnames = new ArrayList<>();
20     @Override
21     protected void doGet(HttpServletRequest request, HttpServletResponse response)
22         throws ServletException, IOException {
23         response.setContentType("text/html;charset=UTF-8");
24         HttpSession session = request.getSession();
25         String questionPath = "";
26         String path="";
27         String hwname="";
28         String classroomname = "";
29         String studentname=(String)session.getAttribute("username");
30         PrintWriter out = response.getWriter();
```

.String questionpath to store the image path of the question to be displayed

.String path to store the current image filepath

.String classroomname to store the name of the classroom

.String studentname to store the username of the student

```
31
32     try{
33         Class.forName("com.mysql.jdbc.Driver");
34
35         // Establish a database connection
36         Connection con = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/momaths", "root", "root123");
37         PreparedStatement stmt=con.prepareStatement("SELECT Classroomname FROM USER WHERE username=?"); //select classroomname from user where username=studentname
38         stmt.setString(1, studentname); //set parameter for sql query as studentname.
39         ResultSet resultset=stmt.executeQuery();
40         out.println("<!DOCTYPE html>"); //generate the html for the page
41         out.println("<html>");
42         out.println("<head>");
43         out.println("<title>Image from Database</title>");
44         out.println("</head>");
45         out.println("<body>");
46         out.println("<form action=\"/Momaths/HwFeedbackServlet\" method=\"post\">");
47         if(resultset.next()){ //if a row is returned from the select query
48             classroomname=resultset.getString("Classroomname");
49             PreparedStatement statement = con.prepareStatement("SELECT DISTINCT HomeworkName FROM HOMEWORK WHERE Classroomname=?");
50             statement.setString(1, classroomname);
51             ResultSet result=statement.executeQuery();
52             // Create a statement and execute a query to retrieve the image path
```

1. The SQL query selects classroomname from the user table where the parameter is set as studentname.
2. HTML is generated for the page
3. If a result is returned from the query, classroomname is retrieved from the query
4. SQL query to select distinct homework names where parameter is set as classroomname. This will select all the different homework tasks assigned to the classroom.

```
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
```

```
        while(result.next()) {
            hwname=result.getString("HomeworkName");
            hwnames.add(hwname);
            out.println("<h1>" + hwname + "</h1>");
            ArrayList <String> QuestionsByName = new ArrayList<>();
            PreparedStatement pstmt=con.prepareStatement("SELECT question FROM HOMEWORK WHERE HomeworkName=?");
            pstmt.setString(1, hwname);
            ResultSet rs=pstmt.executeQuery();
            while(rs.next()){
                path=rs.getString("question");
                questionPaths.add(path);
                QuestionsByName.add(path);
            }
            request.getSession().setAttribute("imagepaths", questionPaths);
            request.getSession().setAttribute("hwnames", hwnames);

            for (int i = 0; i < QuestionsByName.size(); i++) {
                questionPath = QuestionsByName.get(i);
                out.println("<img src='" + questionPath + "' alt='Image'>");

                out.println(" <input type=\"answer\" name=\"answer" + hwname + i + "\" placeholder=\"Enter answer Here\">");
                out.println("<br>"); // Add a line break to separate images
                out.println("<br>");
            }
            out.println("<button class=\"btn\" type=\"submit\">Submit</button>");
            out.println("</form>");

        }
    }catch(Exception e){
    }
}
```

While there are still results from the table the homework names are each added to the hwnames arraylist. For each homework name an SQL query selects all the questions from that specific homework name and stores them in the questionpaths arraylist. A new arraylist called QuestionbyNames is created for each homework name and this arraylist is iterated through via for loop to display each question in the arraylist for each homework name and for each question image the answer box is created for the user to enter their answer in.

## hwanswers.jsp

The screenshot shows a web page titled "Homework answers" with a math problem. The problem states: "There are some red counters and some blue counters in a bag. The ratio of red counters to blue counters is 3:1. Two counters are removed at random. The probability that both the counters taken are blue is  $\frac{1}{20}$ . Work how many counters were in the bag before any counters were removed." Below the problem is a text input field labeled "Enter answer Here". A "Submit" button is visible. The background features a school-themed illustration with a calculator, ruler, and pencil.

There are some red counters and some blue counters in a bag.  
The ratio of red counters to blue counters is 3:1.  
Two counters are removed at random.  
The probability that both the counters taken are blue is  $\frac{1}{20}$   
Work how many counters were in the bag before any counters were removed.

Enter answer Here

Submit

There are some red counters and some blue counters in a bag.  
The ratio of red counters to blue counters is 4:1.  
Two counters are removed at random.  
The probability that both the counters taken are red is  $\frac{22}{35}$   
Work how many blue counters are in the bag.

Enter answer Here

Submit Your answer 6 Correct Answer 5

```
</head>
<body>
    <div class="main">
        <div class="navbar">
            <div class="icon">
                </div>
            </div>
        </div>
        ${hwanswers}
    </body>
```

The \${hwanswers} displays the html content from the StringBuilder object in the HwFeedbackServlet

## HwFeedbackServlet

```
1 import java.io.IOException;
2 import java.io.PrintWriter;
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.util.ArrayList;
8 import java.util.HashMap;
9 import java.util.Map;
10 import javax.servlet.RequestDispatcher;
11 import javax.servlet.ServletException;
12 import javax.servlet.annotation.WebServlet;
13 import javax.servlet.http.HttpServlet;
14 import javax.servlet.http.HttpServletRequest;
15 import javax.servlet.http.HttpServletResponse;
16
17 @WebServlet(urlPatterns = {" /HwFeedbackServlet"})
18 public class HwFeedbackServlet extends HttpServlet {
19     String imagePath="";
20     String answer="";
21     String userAnswer="";
22     String homeworkname="";
23
24     @Override
25     protected void doPost(HttpServletRequest request, HttpServletResponse response)
26         throws ServletException, IOException {
27         ArrayList<String> hwnames = (ArrayList<String>)request.getSession().getAttribute("hwnames");
28         response.setContentType("text/html;charset=UTF-8");
29         PrintWriter out = response.getWriter();
30         boolean isCorrect=false;
31         StringBuilder htmlBuilder = new StringBuilder();
32         try{
33             Class.forName("com.mysql.jdbc.Driver");
34
35             Connection connection = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/momaths", "root", "root123");
36
37
38
39
40
41         htmlBuilder.append("<!DOCTYPE html>");
42         htmlBuilder.append("<html>");
43         htmlBuilder.append("<head>");
44         htmlBuilder.append("</head>");
45         htmlBuilder.append("<body>");
46         htmlBuilder.append("<h1>Homework answers</h1>");
47
48         // Iterate through questions, user answers, and feedback
49         for (int i = 0; i < hwnames.size(); i++) {
50             ArrayList<String> userAnswers=new ArrayList<>();
51             ArrayList<String> imagePaths = new ArrayList<>();
52             homeworkname=hwnames.get(i);
53             PreparedStatement stmt=connection.prepareStatement("SELECT question FROM Homework WHERE HomeworkName=? ");
54             stmt.setString(1, homeworkname);
55             ResultSet resultset=stmt.executeQuery();
56             while(resultset.next()){
57                 imagePath=resultset.getString("question");
58                 imagePaths.add(imagePath);
59             }
60             for(int j=0;j<imagePaths.size();j++){
61                 imagePath = imagePaths.get(j);
62                 answer = request.getParameter("answer"+homeworkname+j);
63                 userAnswers.add(answer);
64             }
65             userAnswer=userAnswers.get(j);
66
67             connection = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/momaths", "root", "root123");
68             htmlBuilder.append("<img src='"+imagePath+"' alt='Image'>");
69             htmlBuilder.append("<input type='text' name='answer' value='"+userAnswer+"' placeholder='Enter answer Here'>");
70
71
72
73         // Establish a database connection
```

.The hwnames arraylist is retrieved from the HTTP session

.StringBuilder object is created to construct HTML script

```
37
38
39
40
41         htmlBuilder.append("<!DOCTYPE html>");
42         htmlBuilder.append("<html>");
43         htmlBuilder.append("<head>");
44         htmlBuilder.append("</head>");
45         htmlBuilder.append("<body>");
46         htmlBuilder.append("<h1>Homework answers</h1>");
47
48         // Iterate through questions, user answers, and feedback
49         for (int i = 0; i < hwnames.size(); i++) {
50             ArrayList<String> userAnswers=new ArrayList<>();
51             ArrayList<String> imagePaths = new ArrayList<>();
52             homeworkname=hwnames.get(i);
53             PreparedStatement stmt=connection.prepareStatement("SELECT question FROM Homework WHERE HomeworkName=? ");
54             stmt.setString(1, homeworkname);
55             ResultSet resultset=stmt.executeQuery();
56             while(resultset.next()){
57                 imagePath=resultset.getString("question");
58                 imagePaths.add(imagePath);
59             }
60             for(int j=0;j<imagePaths.size();j++){
61                 imagePath = imagePaths.get(j);
62                 answer = request.getParameter("answer"+homeworkname+j);
63                 userAnswers.add(answer);
64             }
65             userAnswer=userAnswers.get(j);
66
67             connection = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/momaths", "root", "root123");
68             htmlBuilder.append("<img src='"+imagePath+"' alt='Image'>");
69             htmlBuilder.append("<input type='text' name='answer' value='"+userAnswer+"' placeholder='Enter answer Here'>");
70
71
72
73         // Establish a database connection
```

htmlbuilder is used to create the html script for the hwanswers page.

A for loop is then used to iterate through the hwnames arraylist and

for each iteration the userAnswer and the imagePaths arraylist are created again each time, because for each homework task(each homework name) there will be separate images displayed and separate user answers. For each homework name in the arraylist an SQL query is used to select all questions from the homework table where the parameter is set as the homeworkname.

While there are results from the query (While result.next()) each question is added to the imagePaths arraylist. Then another for loop is created to iterate through the imagePaths arraylist and display each image and get each of the user's answers from the HTTP request and adds each answer to the userAnswers arraylist.

```
73 // Create a statement and execute a query to retrieve the image path
74 String sql = "SELECT Answer FROM Homework WHERE question = ?"; //select answer for each question
75 PreparedStatement preparedStatement = connection.prepareStatement(sql);
76 preparedStatement.setString(1, imagePath); // Set the value for the parameter
77 String correctanswer= "";
78 ResultSet resultSet = preparedStatement.executeQuery();
79 if(resultSet.next()){
80     correctanswer=resultSet.getString("Answer");
81 }
82
83 if(userAnswer.equals(correctanswer)){ //compare the user's answer with the correct answer
84     isCorrect=true;
85     htmlBuilder.append("<p style='color: green;'>Correct!</p>");
86 }
87 else{
88     htmlBuilder.append("<p style='color: red;'>Incorrect.Your answer:").append(userAnswer).append(" Correct Answer: ").append(correctanswer).append("</p>");
89 }
90
91
92
93 }
94
95 }
96 // After generating the HTML content
97 request.setAttribute("hwanswers", htmlBuilder.toString());
98
99 // Forward the request to answers.jsp
100 RequestDispatcher dispatcher = request.getRequestDispatcher("hwanswers.jsp");
101 dispatcher.forward(request, response);
102
103
104
105
106
107
108
109 }
```

The SQL query selects the answer from the homework table for each question and stores the result of the query in the correctanswer String. If userAnswer is equal to correctanswer the Boolean variable isCorrect is set to true and green text saying “Correct!” is displayed. If they are not isCorrect is set to false and red text telling the user they

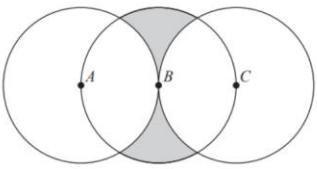
are incorrect and showing them what their answer was and what the correct answer was is displayed.

The HTML content is then stored in the Request Attribute and is forwarded to the hwanswers.jsp page

## circleqs.html

← → ⌂ localhost:8080/Momaths/circleqs.html

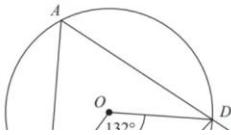
21 The diagram shows three circles, each of radius 4 cm.  
The centres of the circles are  $A$ ,  $B$  and  $C$  such that  $ABC$  is a straight line and  $AB = BC = 4$  cm.



Work out the total area of the two shaded regions.  
Give your answer in terms of  $\pi$

Enter answer Here

20  $A$ ,  $B$ ,  $C$  and  $D$  are points on the circumference of a circle, centre  $O$ .  
 $ADE$  and  $BCE$  are straight lines.



This is not the entire page but simply showing what the final design of the UI looks like. The full page can be seen in the youtube video.

This is the same for the other two topics.

```
</head>
<body>

    <div class="main">
        <div class="navbar">

            <div class="icon">

            </div>
        -->

        <div id="servletContent"></div>

    </div>
```

```

<script>
// JavaScript code to load the image from the Servlet
var xhr = new XMLHttpRequest();
xhr.open("GET", "QuestionServlet", true);
xhr.onreadystatechange = function () {
    if (xhr.readyState === 4 && xhr.status === 200) {
        var imagePath = xhr.responseText;
        document.getElementById("Question").src = imagePath;
    }
};
xhr.send();

// Load HTML content from your servlet via AJAX
var htmlContentXhr = new XMLHttpRequest();
htmlContentXhr.open("GET", "CircleQuestionServlet", true); // Replace with the actual URL of your HTML document
htmlContentXhr.onreadystatechange = function () {
    if (htmlContentXhr.readyState === 4 && htmlContentXhr.status === 200) {
        document.getElementById("servletContent").innerHTML = htmlContentXhr.responseText;
    }
};
htmlContentXhr.send();
</script>
<script>
function changeStyle() {
    var pElement = document.querySelector('p');
    pElement.style.color = 'green';
    pElement.textContent = 'Correct!';
}
</script>

```

This JavaScript code is using XMLHttpRequest (XHR) to fetch content from the server and update the content of an HTML element with the ID "servletContent".

The `changestyle()` functiont modifies the style and content of a `<p>`element in an HTML document.

[CircleQuestionServlet.java](#)

```

@WebServlet(urlPatterns = {"/*CircleQuestionServlet"})
public class CircleQuestionServlet extends HttpServlet {
    ArrayList<String> imagePaths = new ArrayList<>();

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        request.getSession().removeAttribute("imagepaths");
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String imagePath = "";
        String path="";
        try {
            Class.forName("com.mysql.jdbc.Driver");

            // Establish a database connection
            Connection connection = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/momaths", "root", "root123");

            // Create a statement and execute a query to retrieve the image path
            Statement statement = connection.createStatement();
            ResultSet resultSet=null;

            resultSet = statement.executeQuery("SELECT question FROM Question WHERE topic='Circles'");
            while(resultSet.next()){
                path=resultSet.getString("question");
                imagePaths.add(path);
                request.getSession().setAttribute("imagepaths", imagePaths);
            }
        }
    }
}

```

The path and imagePath variables are initialised and the SQL query selects all questions from the question table where the topic is ‘Circles’. While there are results from the query each question is added to the imagePaths arraylist

```

    ...
    out.println("<!DOCTYPE html>");
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Image from Database</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<h1>Images from Database</h1>");
    out.println("<form action=\"/Momaths/FeedbackServlet\" method=\"post\">");
    for (int i = 0; i < imagePaths.size(); i++) {
        imagePath = imagePaths.get(i);
        out.println("<img src='" + imagePath + "' alt='Image'>");

        out.println(" <input type=\"answer\" name=\"answer"+i+"\" placeholder=\"Enter answer Here\">");
        out.println("<br>"); // Add a line break to separate images
        out.println("<br>");
    }

    out.println("<button class=\"btn\" type=\"submit\">Submit</button>");
    out.println("</form>");
    out.println("</body>");
    out.println("</html>");

    //imagePath = "Images/Statsqs/Screenshot 2023-10-21 184647.png"; // Adjust the path to your image

    // Close the database resources
    resultSet.close();
    statement.close();
    connection.close();
}

```

The HTML content is generated and a for loop is used to iterate through the imagePaths arraylist to display each question and the text box for the user to enter the answer in.

The code for this servlet is identical to the AlgebraQuestionServlet and StatquestionServlet other than the other two servlets selecting questions from different topics, so I will refrain from explaining the code for those Servlets as I would just be repeating myself.

### algebraqs.html

The screenshot shows a web browser window with the URL `localhost:8080/Momaths/algebraqs.html`. The page displays a math quiz with several questions and a graph. The questions are:

- 14 (a)** Factorise fully  $4p^2 - 36$
- 22** Find algebraically the set of values of  $x$  for which  $x^2 - 49 > 0$  and  $5x^2 - 31x - 72 > 0$
- 6** A pattern is made from four identical squares. The sides of the squares are parallel to the axes.

The graph shows a coordinate plane with a grid. A pattern of four squares is plotted, starting from the origin. The top-right square is labeled  $B(38, 36)$ . The bottom-left square is labeled  $C$ .

```

<html>
  <head>
    ...
  </head>
  <body>
    <div class="main">
      <div class="navbar">
        ...
      </div>
      <div class="icon">
        ...
      </div>
      <div id="servletContent2"></div>
    </div>
  </body>
</html>

```

```

95 |     | <script>
96 |
97 |         // Load HTML content from your servlet via AJAX
98 |         var htmlContentXhr = new XMLHttpRequest();
99 |         htmlContentXhr.open("GET", "AlgebraQuestionServlet", true); // Replace with the actual URL of your HTML content servlet
100 |         htmlContentXhr.onreadystatechange = function () {
101 |             if (htmlContentXhr.readyState === 4 && htmlContentXhr.status === 200) {
102 |                 document.getElementById("servletContent2").innerHTML = htmlContentXhr.responseText;
103 |             }
104 |         };
105 |         htmlContentXhr.send();
106 |     </script>
107 |     <script>
108 |         function changeStyle() {
109 |             var pElement = document.querySelector('p');
110 |             pElement.style.color = 'green';
111 |             pElement.textContent = 'Correct!';
112 |         }
113 |     </script>
114 |

```

## AlgebraQuestionServlet.java

```

1 | import java.io.IOException;
2 | import java.io.PrintWriter;
3 | import java.sql.Connection;
4 | import java.sql.DriverManager;
5 | import java.sql.ResultSet;
6 | import java.sql.Statement;
7 | import javax.servlet.ServletException;
8 | import javax.servlet.annotation.WebServlet;
9 | import javax.servlet.http.HttpServlet;
10 | import javax.servlet.http.HttpServletRequest;
11 | import javax.servlet.http.HttpServletResponse;
12 | import java.util.ArrayList;
13 |
14 | @WebServlet(urlPatterns = {"/AlgebraQuestionServlet"})
15 | public class AlgebraQuestionServlet extends HttpServlet {
16 |     ArrayList<String> imagePaths = new ArrayList<>();
17 |
18 |     @Override
19 |     protected void doGet(HttpServletRequest request, HttpServletResponse response)
20 |             throws ServletException, IOException {
21 |         PrintWriter out = response.getWriter();
22 |         String imagePath = "";
23 |         String path="";
24 |         try {
25 |             Class.forName("com.mysql.jdbc.Driver");
26 |
27 |             // Establish a database connection
28 |             Connection connection = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/momaths", "root", "root123");
29 |
30 |             // Create a statement and execute a query to retrieve the image path
31 |             Statement statement = connection.createStatement();
32 |             ResultSet resultSet=null;

```

```
34     resultSet = statement.executeQuery("SELECT question FROM Question WHERE topic='algebra'");//select questions with the algebra topic
35
36     while(resultSet.next()){
37         path=resultSet.getString("question");
38         imagePaths.add(path);//add each image filepath to the imagePaths arraylist
39     }
40     out.println("<!DOCTYPE html>");
41     out.println("<html>");
42     out.println("<head>");
43     out.println("<title>Image from Database</title>");
44     out.println("</head>");
45     out.println("<body>");
46     out.println("<form action=\"/Momaths/FeedbackServlet\" method=\"post\">");
47     for (int i = 0; i < imagePaths.size(); i++) {
48         imagePath = imagePaths.get(i);
49         out.println("<img src=\"" + imagePath + "\" alt='Image'>");
50
51         out.println(" <input type=\"answer\" name=\"answer"+i+"\" placeholder=\"Enter answer Here\">");
52         out.println("<br>"); // Add a line break to separate images
53         out.println("<br>");
54     }
55
56     out.println("<button class=\"btn\" type=\"submit\">Submit</button>");
57     out.println("</form>");
58     out.println("</body>");
59     out.println("</html>");
60
61
62
63
64
65     // Close the database resources
66     resultSet.close();
67     statement.close();
68     connection.close();
69 }
```

```
71 }
72
73     catch (Exception e) {
74
75 }
```

## statqs.html

```
<html>
  <head>
    </head>
  <body>
    <div class="main">
      <div class="navbar">
        <div class="icon">
        </div>
      </div>
    </div>
    <div id="servletContent3"></div>

<script>
// Load HTML content from your servlet via AJAX
var htmlContentXhr = new XMLHttpRequest();
htmlContentXhr.open("GET", "StatQuestionsServlet", true); // Replace with the actual URL of your HTML content servlet
htmlContentXhr.onreadystatechange = function () {
  if (htmlContentXhr.readyState === 4 && htmlContentXhr.status === 200) {
    document.getElementById("servletContent3").innerHTML = htmlContentXhr.responseText;
  }
};
htmlContentXhr.send();
</script>
<script>
function changeStyle() {
  var pElement = document.querySelector('p');
  pElement.style.color = 'green';
  pElement.textContent = 'Correct!';
}
</script>
```

The table shows the amount of snow, in cm, that fell each day for 30 days.

Amount of snow ( $s$ cm)	Frequency
$0 \leq s < 10$	8
$10 \leq s < 20$	10
$20 \leq s < 30$	7
$30 \leq s < 40$	2
$40 \leq s < 50$	3

Work out an estimate for the mean amount of snow per day.

Enter answer Here

11 Cormac has some sweets in a bag.  
The sweets are lime flavoured or strawberry flavoured or orange flavoured.

In the bag

number of lime flavoured sweets : number of strawberry flavoured sweets : number of orange flavoured sweets = 9 : 4 :  $x$

Cormac is going to take at random a sweet from the bag.

The probability that he takes a lime flavoured sweet is  $\frac{3}{7}$

This is not the entire page but simply showing what the final design of the UI looks like. The full page can be seen in the youtube video.

This is the same for the other two topics.

## StatQuestionServlet.java

```
1 import java.io.IOException;
2 import java.io.PrintWriter;
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.Statement;
7 import java.util.ArrayList;
8 import javax.servlet.ServletException;
9 import javax.servlet.annotation.WebServlet;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13
14 /**
15 *
16 * @author muham
17 */
18 @WebServlet(urlPatterns = {" /StatQuestionServlet"})
19 public class StatQuestionServlet extends HttpServlet {
20     ArrayList<String> imagePaths = new ArrayList<>();
21
22     @Override
23     protected void doGet(HttpServletRequest request, HttpServletResponse response)
24             throws ServletException, IOException {
25         request.getSession().removeAttribute("imagepaths");
26         response.setContentType("text/html;charset=UTF-8");
27         PrintWriter out = response.getWriter();
28         String imagePath = "";
29         String path="";
```

```
30 |     try{
31 |         Class.forName("com.mysql.jdbc.Driver");
32 |
33 |         // Establish a database connection
34 |         Connection connection = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/momaths", "root", "root123");
35 |
36 |         // Create a statement and execute a query to retrieve the image path
37 |         Statement statement = connection.createStatement();
38 |         ResultSet resultSet=null;
39 |
40 |         resultSet = statement.executeQuery("SELECT question FROM Question WHERE topic='stats'");
41 |         while(resultSet.next()){
42 |             path=resultSet.getString("question");
43 |             imagePaths.add(path);
44 |             request.getSession().setAttribute("imagepaths", imagePaths);
45 |         }
46 |         out.println("<!DOCTYPE html>");
47 |         out.println("<html>");
48 |         out.println("<head>");
49 |         out.println("<title>Image from Database</title>");
50 |         out.println("</head>");
51 |         out.println("<body>");
52 |         out.println("<h1>Images from Database</h1>");
53 |         out.println("<form action=\"/Momaths/FeedbackServlet\" method=\"post\">");
54 |         for (int i = 0; i < imagePaths.size(); i++) {
55 |             imagePath = imagePaths.get(i);
56 |             out.println("<img src='" + imagePath + "' alt='Image'>");
57 |
58 |             out.println(" <input type=\"answer\" name=\"answer"+i+"\" placeholder=\"Enter answer Here\">");
59 |             out.println("<br>"); // Add a line break to separate images
60 |             out.println("<br>");
61 |         }
62 |
63 |         out.println("<button class=\"btn\" type=\"submit\">Submit</button>");
64 |         out.println("</form>");
65 |         out.println("</body>");
66 |     }
```

```
67 |     out.println("</html>");
68 |
69 |
70 |
71 |
72 |
73 |     // Close the database resources
74 |     resultSet.close();
75 |     statement.close();
76 |     connection.close();
77 |
78 |     }catch(Exception e){
79 |         e.printStackTrace();
80 |     }
81 |
82 | }
```

## FeedbackServlet.java

The code for this is the same as the code for the HwFeedbackServlet other than the fact that the SQL query is used to select the answer from the question table and not the homework table.

```
1 import java.io.IOException;
2 import java.io.PrintWriter;
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.util.ArrayList;
8 import java.util.HashMap;
9 import java.util.Map;
10 import javax.servlet.RequestDispatcher;
11 import javax.servlet.ServletException;
12 import javax.servlet.annotation.WebServlet;
13 import javax.servlet.http.HttpServlet;
14 import javax.servlet.http.HttpServletRequest;
15 import javax.servlet.http.HttpServletResponse;
16
17 @WebServlet(urlPatterns = {"/FeedbackServlet"})
18 public class FeedbackServlet extends HttpServlet {
19     String imagePath="";
20     String userAnswer="";
21
22     @Override
23     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
24         response.setContentType("text/html;charset=UTF-8");
25         PrintWriter out = response.getWriter();
26         boolean isCorrect=false;
27         StringBuilder htmlBuilder = new StringBuilder();
28         try{
29             Class.forName("com.mysql.jdbc.Driver");
30
31             Map<Integer, String> userAnswers = new HashMap<>();
32             ArrayList<String> imagePaths = (ArrayList<String>)request.getSession().getAttribute("imagepaths");
33
34
35             // ... Compare user answers with correct answers and populate the feedback map ...
36         }
```

```
42
43     // Iterate through questions and user answers
44     for (int i = 0; i < imagePaths.size(); i++) {
45         imagePath = imagePaths.get(i);
46         userAnswers.put(i, userAnswer);
47         userAnswer=request.getParameter("answer"+i);
48         Connection connection = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/momaths", "root", "root123");
49         out.println("<p>" +(i+1)+ "</p>");
50         htmlBuilder.append("<img src='"+imagePath+"' alt='Image'>");
51         htmlBuilder.append("<input type=\"text\" name=\"answer\" placeholder=\"Enter answer Here\">");
52
53
54         // Establish a database connection
55
56
57         // Create a statement and execute a query to retrieve the image path
58         String sql = "SELECT Answer FROM Question WHERE question = ?";
59         PreparedStatement preparedStatement = connection.prepareStatement(sql);
60         preparedStatement.setString(1, imagePath); // Set the value for the placeholder
61         String correctanswer= "";
62         ResultSet resultSet = preparedStatement.executeQuery();
63         if(resultSet.next()){
64             correctanswer=resultSet.getString("Answer");
65         }
66     }
```

```

66
67
68     if(userAnswer.equals(correctanswer)){
69         isCorrect=true;
70         htmlBuilder.append("<p style='color: green;'>Correct!</p>");
71     }
72     else{
73         htmlBuilder.append("<p style='color: red;'>Incorrect. Your answer:").append(userAnswer).append(" Correct Answer: ").append(correctanswer).append("</p>");
74     }
75     // out.println("</form>");
76
77
78
79 }
80
81 }
82 htmlBuilder.append("<button class=\"sol\"><a href=\"solutions.html\">View solutions</a></button>");
83 htmlBuilder.append("<button class=\"home\"><a href=\"mainmenu.jsp\">Return to home page</a></button>");
84 // After generating the HTML content
85 request.setAttribute("displayanswers", htmlBuilder.toString());
86
87 // Forward the request to answers.jsp
88 RequestDispatcher dispatcher = request.getRequestDispatcher("answers.jsp");
89 dispatcher.forward(request, response);
90
91
92
93
94
95 }catch(Exception e){
96     System.out.println(e.toString());
97 }

```

## answers.jsp

```

1 <html>
2     <head>
3
4
5     </head>
6     <body>
7
8         <div class="main">
9             <div class="navbar">
10
11                 <div class="icon">
12
13
14             </div>
15
16
17         </div>
18
19         ${displayanswers} <!--loads the HTML from the htmlbuilder in the FeedbackServlet-->
20
21
22     </body>
23
24 </html>

```

The table shows the amount of snow, in cm, that fell each day for 30 days.

Amount of snow (s cm)	Frequency
$0 \leq s < 10$	8
$10 \leq s < 20$	10
$20 \leq s < 30$	7
$30 \leq s < 40$	2
$40 \leq s < 50$	3

Work out an estimate for the mean amount of snow per day.

Enter answer Here

Incorrect. Your answer: Correct Answer: 19

- 11** Cormac has some sweets in a bag.  
The sweets are lime flavoured or strawberry flavoured or orange flavoured.

In the bag

$$\frac{\text{number of lime flavoured sweets}}{\text{flavoured sweets}} : \frac{\text{number of strawberry flavoured sweets}}{\text{flavoured sweets}} : \frac{\text{number of orange flavoured sweets}}{\text{flavoured sweets}} = 9 : 4 : x$$

Cormac is going to take at random a sweet from the bag.

The probability that he takes a lime flavoured sweet is  $\frac{3}{7}$

---

This is not the entire page but simply showing what the final design of the UI. The full page can be seen in the youtube video.

This is the same for the other two topics.

## solutions.html

```

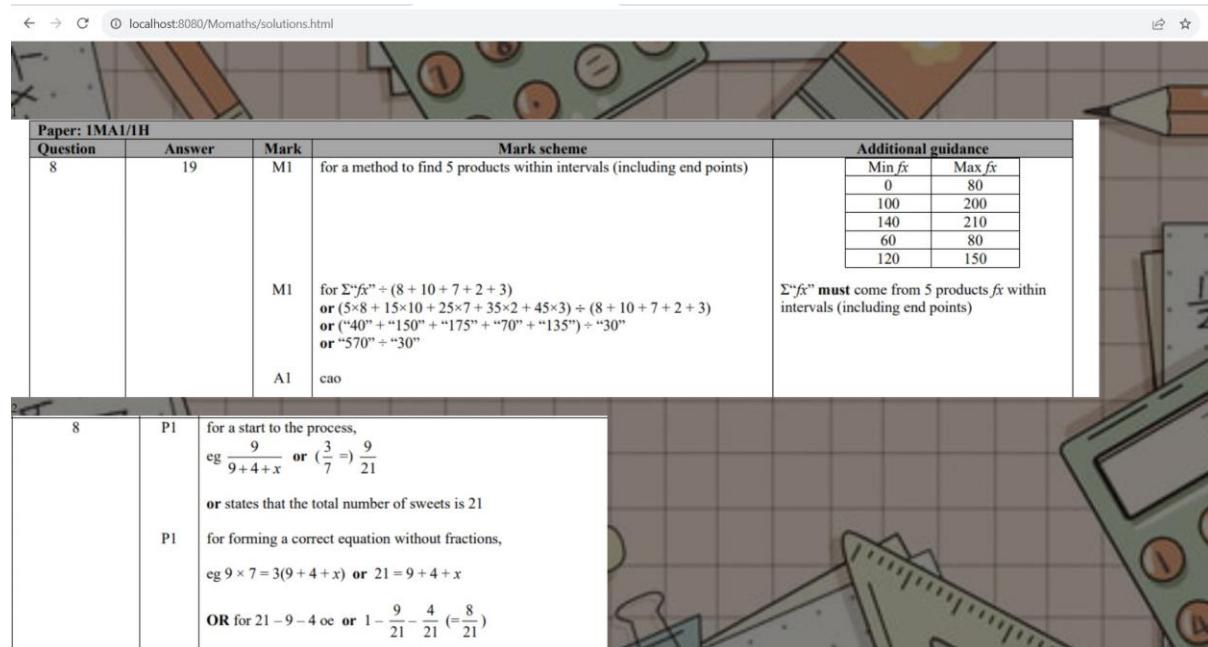
1 <!DOCTYPE html>
2
3 <html>
4   <head>
5
6   </head>
7 <body>
8   <div class="main">
9     <div class="navbar">
10    <div class="icon">
11
12
13
14  </div>
15
16
17
18
19
20  </div>
21 <div id="solutioncontent"></div>
22 </div>

```

```

99
100
101
102
103
104
105
106
107
<script>
  htmlContentXhr.open("GET", "SolutionServlet", true); // Replace with the actual URL of your HTML content servlet
  htmlContentXhr.onreadystatechange = function () {
    if (htmlContentXhr.readyState === 4 && htmlContentXhr.status === 200) {
      document.getElementById("solutioncontent").innerHTML = htmlContentXhr.responseText;
    }
  };
  htmlContentXhr.send();
</script>

```



Paper: IMA1/1H

Question	Answer	Mark	Mark scheme	Additional guidance												
8	19	M1	for a method to find 5 products within intervals (including end points)	<table border="1"> <tr> <td>Min fx</td> <td>Max fx</td> </tr> <tr> <td>0</td> <td>80</td> </tr> <tr> <td>100</td> <td>200</td> </tr> <tr> <td>140</td> <td>210</td> </tr> <tr> <td>60</td> <td>80</td> </tr> <tr> <td>120</td> <td>150</td> </tr> </table>	Min fx	Max fx	0	80	100	200	140	210	60	80	120	150
Min fx	Max fx															
0	80															
100	200															
140	210															
60	80															
120	150															
		M1	for $\Sigma f_x \times (8 + 10 + 7 + 2 + 3)$ or $(5 \times 8 + 15 \times 10 + 25 \times 7 + 35 \times 2 + 45 \times 3) \div (8 + 10 + 7 + 2 + 3)$ or "40" + "150" + "175" + "70" + "135" ÷ "30" or "570" ÷ "30"	$\Sigma f_x$ must come from 5 products $f_x$ within intervals (including end points)												
		A1	cao													
8	P1		for a start to the process, eg $\frac{9}{9+4+x}$ or $(\frac{3}{7}) = \frac{9}{21}$													
	P1		or states that the total number of sweets is 21													
	P1		for forming a correct equation without fractions, eg $9 \times 7 = 3(9 + 4 + x)$ or $21 = 9 + 4 + x$													
			OR for $21 - 9 - 4$ or $1 - \frac{9}{21} - \frac{4}{21} (= \frac{8}{21})$													

This is not the entire page but simply showing what the final design of the UI. The full page can be seen in the youtube video.

This is the same for the other two topics.

### SolutionServlet.java

```
1 import java.io.IOException;
2 import java.io.PrintWriter;
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.util.ArrayList;
8 import javax.servlet.ServletException;
9 import javax.servlet.annotation.WebServlet;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13
14 /**
15  * 
16  * @author muham
17  */
18 @WebServlet(urlPatterns = {"/SolutionServlet"})
19 public class SolutionServlet extends HttpServlet {
20     String imagePath="";
21     String solution="";
22     protected void doGet(HttpServletRequest request, HttpServletResponse response)
23         throws ServletException, IOException {
24         response.setContentType("text/html;charset=UTF-8");
25         PrintWriter out = response.getWriter();
26         try{
27             ArrayList<String> imagePaths = (ArrayList<String>)request.getSession().getAttribute("imagepaths");
28             Class.forName("com.mysql.jdbc.Driver");
29             out.println("<!DOCTYPE html>"); //generate html for the page
30             out.println("<html>");
31             out.println("<head>");
32             out.println("<title>Image from Database</title>");
33             out.println("</head>");
34             out.println("<body>");
35             out.println("<h1>Images from Database</h1>");
36             out.println("<form action=\"/Momaths/FeedbackServlet\" method=\"post\">");

37             for(int i=0;i<imagePaths.size();i++){
38                 imagePath=imagePaths.get(i);
39                 Connection connection = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/momaths", "root", "root123");
40                 PreparedStatement statement=connection.prepareStatement("SELECT solution FROM QUESTION WHERE question=?");
41                 statement.setString(1, imagePath);
42                 ResultSet result=statement.executeQuery();
43                 if(result.next()){
44                     solution=result.getString("solution");
45                 }
46                 out.println("<p>" +(i+1) +"</p>");
47                 out.println("<img src=\"" + solution + "\" alt='Image'>");
48             }
49             out.println("<button class=\"home\"><a href=\"mainmenu.jsp\">Return to home page</a></button>");

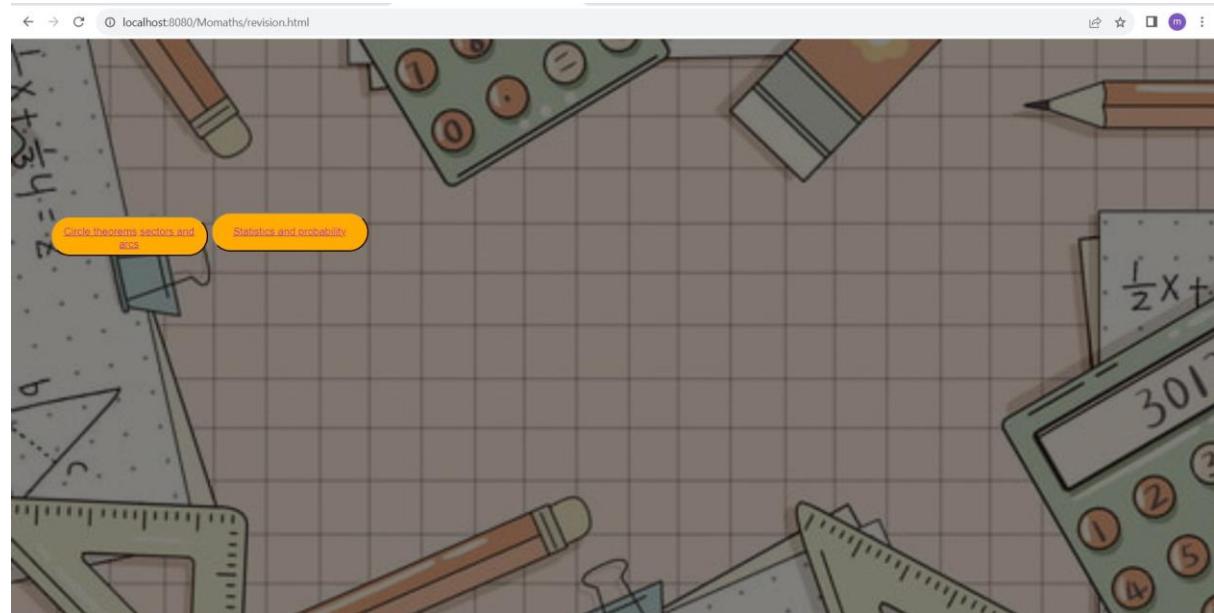
50
51         }catch(Exception e){
52             e.printStackTrace();
53         }
54     }
}
```

The SolutionServlet code is very similar to the previous servlets which display the question images. All this is doing is using a for loop to iterate through the imagePaths arraylist, which stores the questions,

and within this for loop an SQL query is used to select the image file paths in the solution column from the question table for each question on the page and generating html to display each solution image on the page.

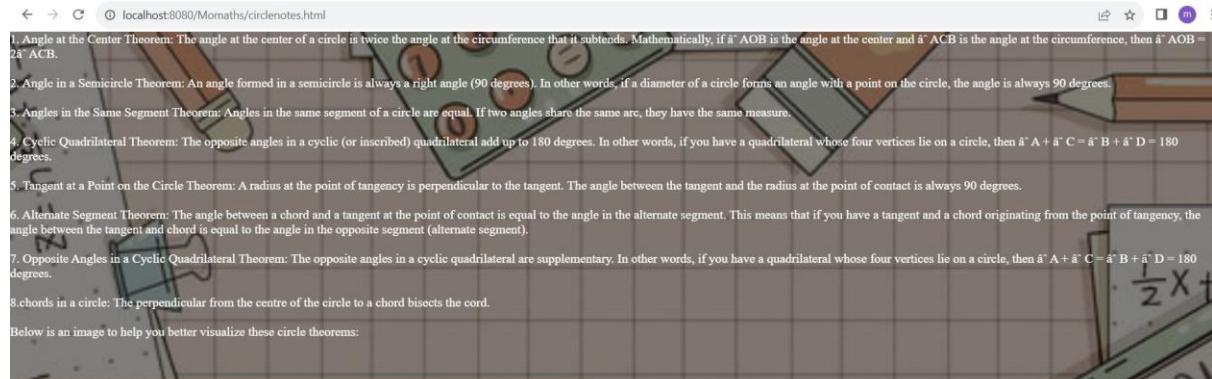
### Revision.html

```
1 <html>
2   <head>
3     <title>Webpage Design</title>
4
5   </head>
6   <body>
7
8     <div class="main">
9       <div class="navbar">
10
11         <div class="icon">
12
13           </div>
14         </div>
15         <button class="circ"><a href="circlenotes.html">Circle theorems, sectors and arcs</a></button>
16         <button class="stat"><a href="statsnotes.html">Statistics and probability</a></button>
17
18       </div>
19     </div>
```



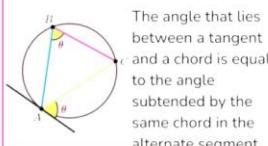
## circlenotes.html

```
1 <html>
2   <head>
3
4
5     </head>
6   <body>
7     <p>1. Angle at the Center Theorem:</p>
8
9     The angle at the center of a circle is twice the angle at the circumference that it subtends.
10    Mathematically, if  $\angle AOB$  is the angle at the center and  $\angle ACB$  is the angle at the circumference, then  $\angle AOB = 2\angle ACB$ .
11    <br><br>
12    2. Angle in a Semicircle Theorem:
13
14    An angle formed in a semicircle is always a right angle (90 degrees).
15    In other words, if a diameter of a circle forms an angle with a point on the circle, the angle is always 90 degrees.
16    <br><br>
17    3. Angles in the Same Segment Theorem:
18
19    Angles in the same segment of a circle are equal.
20    If two angles share the same arc, they have the same measure.
21    <br><br>
22    4. Cyclic Quadrilateral Theorem:
23
24    The opposite angles in a cyclic (or inscribed) quadrilateral add up to 180 degrees.
25    In other words, if you have a quadrilateral whose four vertices lie on a circle, then  $\angle A + \angle C = \angle B + \angle D = 180$  degrees.
26    <br><br>
27    5. Tangent at a Point on the Circle Theorem:
28
29    A radius at the point of tangency is perpendicular to the tangent.
30    The angle between the tangent and the radius at the point of contact is always 90 degrees.
31    <br><br>
32    6. Alternate Segment Theorem:
33
34    The angle between a chord and a tangent at the point of contact is equal to the angle in the alternate segment.
35    This means that if you have a tangent and a chord originating from the point of tangency, the angle between the tangent and chord is equal to the angle in the
36    <br><br>
37    7. Opposite Angles in a Cyclic Quadrilateral Theorem:
38
39    The opposite angles in a cyclic quadrilateral are supplementary.
40    In other words, if you have a quadrilateral whose four vertices lie on a circle, then  $\angle A + \angle C = \angle B + \angle D = 180$  degrees.
41    <br><br>
42    8. chords in a circle: The perpendicular from the centre of the circle to a chord bisects the cord.
43
44    <br><br>
45    Below is an image to help you better visualize these circle theorems:
46    <br><br><br>
47    
48
49  </body>
```



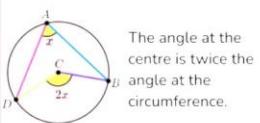
### Circle Theorems

#### Alternate segment theorem



The angle that lies between a tangent and a chord is equal to the angle subtended by the same chord in the alternate segment.

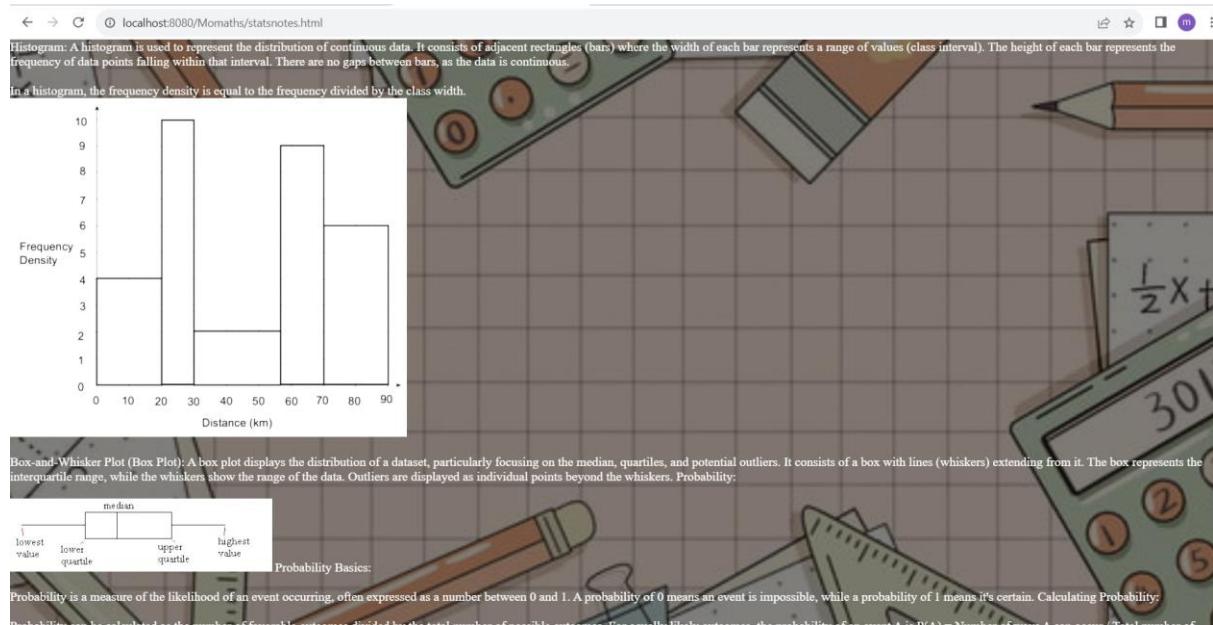
#### Angle at the centre theorem



The angle at the centre is twice the angle at the circumference.

## Statnotes.html

```
<html>
  <head>
    <body>
      <p>Histogram:</p>
      A histogram is used to represent the distribution of continuous data.
      It consists of adjacent rectangles (bars) where the width of each bar represents a range of values (class interval).
      The height of each bar represents the frequency of data points falling within that interval.
      There are no gaps between bars, as the data is continuous.
      <br><br>
      In a histogram, the frequency density is equal to the frequency divided by the class width.
      <br>
      
      <br><br>
      Box-and-Whisker Plot (Box Plot):
      A box plot displays the distribution of a dataset, particularly focusing on the median, quartiles, and potential outliers.
      It consists of a box with lines (whiskers) extending from it.
      The box represents the interquartile range, while the whiskers show the range of the data.
      Outliers are displayed as individual points beyond the whiskers.
      Probability:
      <br><br>
      
      Probability Basics:
      <br><br>
      Probability is a measure of the likelihood of an event occurring, often expressed as a number between 0 and 1.
      A probability of 0 means an event is impossible, while a probability of 1 means it's certain.
      Calculating Probability:
      <br><br>
      Probability can be calculated as the number of favorable outcomes divided by the total number of possible outcomes.
      For equally likely outcomes, the probability of an event A is  $P(A) = \text{Number of ways A can occur} / \text{Total number of possible outcomes}$ .
      Probability Rules:
      <br><br>
      The complement rule:  $P(A') = 1 - P(A)$  (the probability of the event not occurring).
      The addition rule for mutually exclusive events:  $P(A \text{ or } B) = P(A) + P(B)$ .
      The multiplication rule for independent events:  $P(A \text{ and } B) = P(A) * P(B)$ .
      Ratios:
      <br><br>
      Ratio Basics:
      <br><br>
      A ratio is a way to compare two or more quantities.
      <br><br>
      Ratios are often written as "a to b" or "a:b" and can be simplified.
      Equivalent Ratios:
      <br><br>
      Ratios that represent the same comparison are considered equivalent.
      You can find equivalent ratios by multiplying or dividing both parts by the same number.
      Proportions:
      <br><br>
      A proportion is an equation that states two ratios are equal.
      You can solve proportions by cross-multiplying.
      Using Ratios for Scaling:
      <br><br>
      Ratios are used to scale up or down quantities while maintaining the same proportion.
      For example, if you have a recipe for 4 servings and need to make 8 servings, you would use a ratio of 4:8 to scale the ingredients.
      <br><br>
      </p>
    </body>
</html>
```



## Testing

Video link

[https://youtu.be/ohoCRG\\_WwPE](https://youtu.be/ohoCRG_WwPE)

Testing against objectives

Objectives	Test	Intended outcome	Outcome of test	Test timestamp
1.0	Allow user to login using username and password	User logs into the system and is met with the main menu	PASS	7:25-7:29
1.1	User can sign up/create an account	User is taken to the startup page	PASS	1:28-1:32

1.2	Do not allow the user to login with an incorrect username or password	Display message “invalid username or password”	PASS	7:10-7:24
1.3	Do not allow user to sign up with one or more empty fields	Display message “cannot sign up with one or more empty fields.”	PASS	1:12-1:23
1.4	Do not allow user to sign up without confirming their password correctly	Display message “Passwords do not match.”	PASS	0:55-1:08
1.5	Do not allow user to sign up if the username they type in already exists	Display message “Username already exists.”	PASS	0:26-0:48
1.6	Store the email, username,	Values entered for email username and	PASS	6:33-6:47

	password and occupation in a database	password are inserted into database		
1.7	Have an algorithm to securely store the password in the database such as hashing or encryption	Hashed password is stored in the password field in the database	PASS	6:33-6:47
1.8	Allow users to select whether they are a student or a teacher	Display two checkboxes where student must tick either student or teacher	PASS	1:32-1:34
1.9	Display a different main menu to the user depending on whether they are a student or a teacher	On teacher menu instead of a view homework button there is an assign homework button	PASS	1:37 for student main menu 5:00 for teacher main menu

2.0	Allow users who are teachers to automatically have a classroom created for them when they sign up	New entry is added in the classroom table when the user signs up as a teacher	PASS	6:50-6:53
2.1	Allow students to join the classroom by entering a code	Display message “joined (insert teacher username)’s classroom)” and the “classroomname” field in the user table is updated	PASS	7:53-8:02
2.2	Allow multiple students to join a classroom	On teacher main menu other students who join classroom are shown in a list	PASS	9:33-9:53
2.3	Sort the usernames of the students in the	On teacher main menu list is sorted in	PASS	9:33-9:53

	classroom in alphabetical order	alphabetical order		
2.4	Allow teachers to assign homework questions, via uploading images, for all of the users in their classroom	When teacher uploads question the image is stored in the homework table and students in the classroom will be able to see the questions when they have been assigned	PASS	I unfortunately forgot To demonstrate the Questions being Stored in the database (see screenshot Below table for Evidence)  5:23-6:23 for teachers assigning questions  8:05 for student being able to see the homework questions
2.5	Allow teachers to assign multiple questions in one homework task	When student views homework, questions under the same homework name will be displayed	PASS	8:05

		together in the same task		
2.6	Allow teachers to enter the answer for each question	Answer for each question is stored in the homework table	PASS	Again, this will be showed in a screenshot below as this was not shown in the video
2.7	Users can enter answers for each question and receive feedback for each question after submitting	After entering answers and pressing submit the student will be able to see feedback for each question	PASS	8:25-8:40
2.8	Allow teachers to assign multiple homework tasks	Questions assigned under different homework names will be	PASS	

		displayed as separate tasks		
3.0	Allow users to answer different exam questions on different topics	User is able to choose a topic and answer questions on it	PASS	2:04-2:25 2:59-3:03 3:12-3:17
3.1	Allow users to view a page with the mark schemes for each question	After receiving feedback user can press view solutions button to see mark schemes	PASS	2:44-2:49 3:07-3:09 3:23-3:26
3.2	Allow users to submit their answers and get feedback for each question	Users press submit after entering answers and can see which questions they got right or wrong	PASS	2:27-2:37 3:04-3:06 3:19-3:22
3.3	Give users the option to view revision pages	Display revision notes on the revision pages	PASS	3:37-4:17

## Test outcome for tests 2.4 and 2.6

	HomeworkqID	HomeworkName	TeacherID	Answer	question
▶	44	homework1	94	4	Images/hwqs/Screenshot 2023-12-04 124256.png
	45	homework1	94	5	Images/hwqs/Screenshot 2023-12-04 124605.png
*	46	homework2	94	6	Images/hwqs/Screenshot 2023-12-04 125116.png
	NULL	NULL	NULL	NULL	NULL

As shown in the screenshot the question assigned by the teacher is stored as an Image file path and the answer for each question is also stored in the answer column, showing that I have passed both tests.

## Evaluation

### Evaluation of the system

Astonishingly, the Momaths Webapp has achieved everything that me and my client hoped it would and gone even beyond that. Through consistent communication with my client I have developed a complete solution to their problem by allowing the users of the system to revise with easy to understand and quick revision notes, Challenging and high-level exam questions and the ability for teachers to interact with their students better by assigning them homework tasks and allowing students to join their classroom.

### Evaluation of objective criteria

Evaluation of 1.0: The system has met objective 1.0, as the system allows the user to login using an email, username and password and are taken to the startup page when they successfully sign up.

Evaluation of 1.1: The system has met objective 1.1 as the user is able to sign up and create an account if they do not already have an account.

Evaluation of 1.2: The system has met objective 1.2 because if the user tries to login with an incorrect username or password the message “invalid username or password” is displayed.

Evaluation of 1.3: The system has met objective 1.3 because if the user tries to sign up with one or more empty fields the message “cannot sign up with one or more empty fields.” Is displayed.

Evaluation of 1.4: The system has met objective 1.4 because if the user tries to sign up without confirming their password correctly the message “Passwords do not match.” Is displayed.

Evaluation of 1.5: The system has met objective 1.5 because if the user tries to sign up with a username that already exists the message “Username already exists.” is displayed.

Evaluation of 1.6: Objective 1.6 has been met as when the user signs up the email username and password are successfully stored in the database

Evaluation of 1.7: The system has met objective 1.7 as there is a hashing algorithm which successfully hashes the password before storing it in the database.

Evaluation of 1.8: The system has met objective 1.8 as the user is able to select whether they are a student or teacher.

Evaluation of 1.9: The system has met objective 1.9 as different main menus are displayed to students and teachers

Evaluation of 2.0: The system has met this objective as a classroom is automatically created in the database when a teacher signs up.

Evaluation of 2.1: The system has met this objective as students are able to join a classroom by entering a unique code.

Evaluation of 2.2: The system has met this objective as multiple students are able to join a classroom.

Evaluation of 2.3: The system has met this objective as the list of the names of the students in the classroom are sorted in alphabetical order.

Evaluation of 2.4: The system has met this objective as teachers are able to upload homework questions which are stored in the database and are able to be seen by students who are in the classroom.

Evaluation of 2.5: The system has met this objective as teachers are able to assign questions under the same homework name and when the student sees them they are displayed as part of the same homework task.

Evaluation of 2.6: The system has met this objective as teachers can enter the answer for each question which is stored in the database in the homework table.

Evaluation of 2.7: The system has met this objective as after the student enters their answers and presses submit they receive feedback for each question.

Evaluation of 2.8: The system has met this objective because if teachers assign questions under different homework names they are displayed as separate tasks.

Evaluation of 3.0: The system has met this objective as users are able to answer exam questions on a variety of different topics.

Evaluation of 3.1: The system has met this objective as after the user has received feedback for each question they can view a page with mark schemes for each question.

Evaluation of 3.2: The system has met this objective as the users are able to receive feedback after answering the exam questions on the website.

Evaluation of 3.3: The system has met this objective as there are revision pages on the website which display revision notes for the user to read over.

#### [Feedback from client](#)

My client was overall very pleased with the system I developed in the end. My client was very satisfied with the topics that I chose to add to the website, the difficulty of the questions I added and his expectations were exceeded when he saw the Classroom feature.

Being able to join the classroom and receive homework from his teacher will help my Client's brother develop his maths skills even more by allowing his teacher to set a variety of exam questions that I didn't have time to make available on the system.

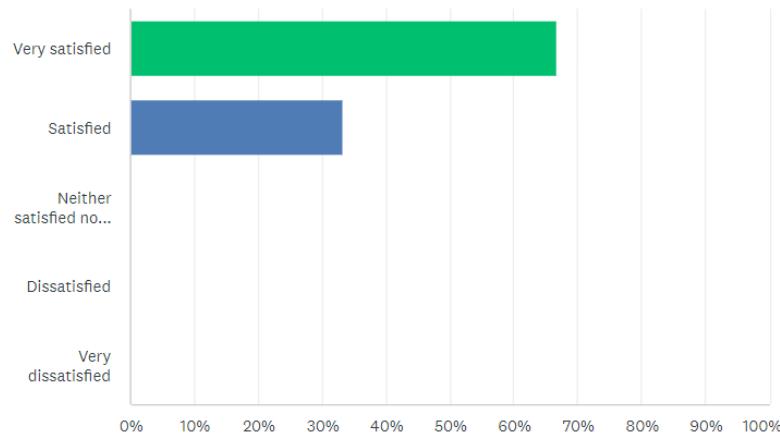
## Feedback from users

Q1

 Customize Save as▼

How satisfied are you with the exam questions provided by Momaths?

Answered: 3 Skipped: 0

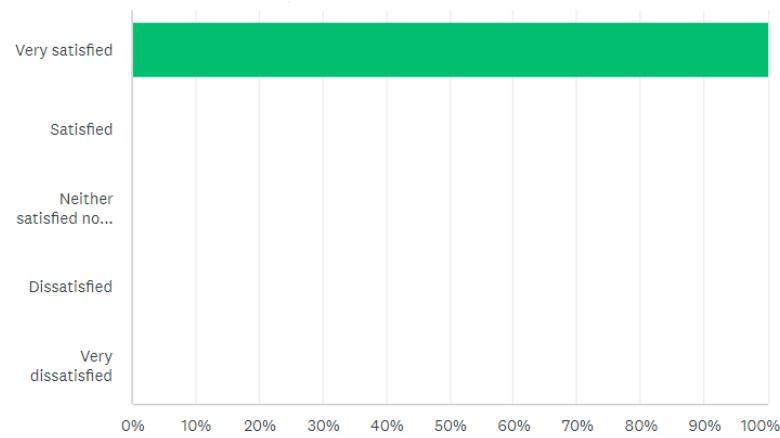


Q2

 Customize Save as▼

How satisfied are you with the classroom feature in Momaths?

Answered: 3 Skipped: 0

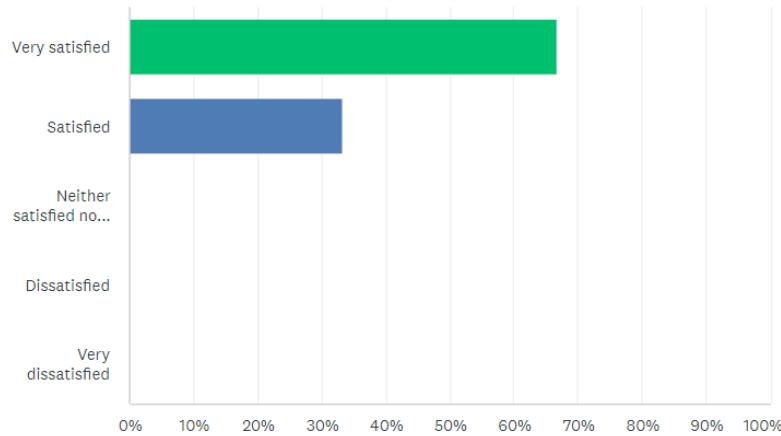


Q3

 Customize 

### How satisfied are you with the homework functionality in Momaths?

Answered: 3 Skipped: 0



Q4

 Save as▼

### What improvements would you suggest be made to the system, if any?

Answered: 3 Skipped: 0

**RESPONSES (3)**

 WORD CLOUD

 TAGS (0)

 Sentiments: OFF 

Search Responses

 Filter: by tag ▾

Showing 3 responses

have variety of different topics

04/12/2023 00:07

[View respondent's answers](#) [Add tags](#) ▾

A reward system students to provide students with more incentive to use the system

03/12/2023 23:57

[View respondent's answers](#) [Add tags](#) ▾

A system perhaps where teacher can send feedback and advise to students

03/12/2023 23:54

[View respondent's answers](#) [Add tags](#) ▾

### Evaluations of survey

Overall, the survey results were positive although it was unfortunately a small sample. Regardless of the sample size however, the users of the app were satisfied with how the different features of the system such as the exam questions, the revision pages and the Classroom functioned.

### Possible improvements for the system

Some of the suggestions from the survey opened my eyes to things that could be improved on in this system.

First and foremost, having a greater variety of topics is crucial and given more time to work on this system it could quite easily be implemented as it will mostly be the same process as for the topics already featured.

The suggestion by a parent to add a reward system intrigued me. I originally planned to have a reward system e.g. currency or titles which you could earn by answering questions, but I scrapped this idea based on the feedback from my client who wanted the system to be strictly for revision purposes to help his brother prepare for his exams.

Lastly, the suggestion of the teacher sending advice to students after they submit a homework task would admittedly be quite difficult, but I think it would be a feature that would greatly improve the system and with more work and effort put in I believe I could successfully implement it.

## Appendix

### Styling(CSS)

#### Index.html:

```
40 <style>
41   *{
42     margin: 0;
43     padding: 0;
44   }
45
46   .main{
47     width:100%;
48     background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0,0.5)50%), url(https://thephillomathclub.weebly.com/uploads/1/3/9/5/139590
49     background-size: cover;
50     background-attachment: fixed;
51     filter: brightness(1.5);
52   }
53
54
55
56
57   .navbar{
58     width: 1200px;
59     height: 75px;
60     margin: auto;
61   }
62   .icon{
63     width: 200px;
64     float: left;
65     height: 70px;
66   }
67
68   .logo{
69     color: rgb(10, 187, 51);
70     font-size: 150px;
71     font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
72     padding-left: 20px;
73     float: left;
74     padding-top: 10px;
75 }
```

```
80     .content{
81         width: 1200px;
82         height:1000px;
83         margin: auto;
84         color: #fff;
85         position: relative;
86     }
87     .content .par{
88         padding-left: 20px;
89         padding-bottom: 25px;
90         font-family: Arial;
91         letter-spacing: 1.2px;
92         line-height: 30px;
93     }
94     .content h1{
95         font-family: 'Times New Roman';
96         font-size: 50px;
97         padding-left: 20px;
98         margin-top: 9%;
99         letter-spacing: 2px;
100    }
101   .content .cn{
102       width: 160px;
103       height: 40px;
104       background: #ff7200;
105       border: none;
106       margin-bottom: 10px;
107       margin-left: 20px;
108       font-size: 18px ;
109       border-radius: 10px;
110       cursor: pointer;
111   }
112   }
113 </style>
```

## Signup.html

```
14     <style>
15         *{
16             margin: 0;
17             padding: 0;
18         }
19
20         .main{
21             width:100%;
22             background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0,0.5)50%, url(https://thephilomathclub.weebly.com/uploads/1/3/9/5/139590);
23             background-size: cover;
24             background-attachment: fixed;
25             filter: brightness(1.5);
26         }
27
28         .navbar{
29             width: 1200px;
30             height: 75px;
31             margin: auto;
32         }
33         .icon{
34             width: 200px;
35             float: left;
36             height: 70px;
37             align-self: center;
38         }
39
40         .logo{
41             color: rgb(26, 255, 0);
42             font-size: 150px;
43             font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
44             padding-left: 20px;
45             float: left;
46             padding-top: 10px;
47         }
48
49
50         }
51
52
53
54
55         .content{
56             width: 1200px;
57             height:1000px;
58             margin: auto;
59             color: #fff;
60             position: relative;
61         }
62
63         .content .par{
64             padding-left: 20px;
65             padding-bottom: 25px;
66             font-family: Arial;
67             letter-spacing: 1.2px;
68             line-height: 30px;
69         }
70
71         .content h1{
72             font-family: 'Times New Roman';
73             font-size: 50px;
74             padding-left: 20px;
75             margin-top: 9%;
76             letter-spacing: 2px;
77         }
78
79         .content .cn{
80             width: 160px;
81             height: 40px;
82             background: #ff7200;
83             border: none;
84             margin-bottom: 10px;
85             margin-left: 20px;
86             font-size:18px ;
87             border-radius: 10px;
88             cursor: pointer;
89
90         }
91
92         }
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116     </style>
```

## Startup.html:

```
53 <style>
54   *{
55     margin: 0;
56     padding: 0;
57   }
58
59   .main{
60     width:100%;
61     background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0,0.5)50%), url(https://thephillomathclub.weebly.com/uploads/1/3
62     background-position: center;
63     background-size: cover;
64     height: 109vh;
65     filter: brightness(1.5);
66   }
67
68   .navbar{
69     width: 1200px;
70     height: 75px;
71     margin: auto;
72   }
73   .icon{
74     width: 200px;
75     float: left;
76     height: 70px;
77   }
78
79   .logo{
80     color: rgb(26, 255, 0);
81     font-size: 150px;
82     font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
83     padding-left: 20px;
84     float: left;
85     padding-top: 10px;
86   }
87
88 
```

```
90 }  
91  
92  
93 .content{  
94     width: 1200px;  
95     height:1000px;  
96     margin-left: 50px;  
97     color: #fff;  
98     position: relative;  
99 }  
100 .content .par{  
101     padding-left: 20px;  
102     padding-bottom: 25px;  
103     font-family: Arial;  
104     letter-spacing: 1.2px;  
105     line-height: 30px;  
106 }  
107 .content h1{  
108     font-family: 'Times New Roman';  
109     font-size: 50px;  
110     padding-left: 20px;  
111     margin-top: 9%;  
112     letter-spacing: 2px;  
113 }  
114 .content .cn{  
115     width: 160px;  
116     height: 40px;  
117     background: #ff7200;  
118     border: none;  
119     margin-bottom: 10px;  
120     margin-left: 20px;  
121     font-size:18px ;  
122     border-radius: 10px;  
123     cursor: pointer;  
124 }  
125 }
```

```
125 }  
126 .container {  
127   display: block;  
128   position: relative;  
129   padding-left: 35px;  
130   margin-bottom: 22px;  
131   cursor: pointer;  
132   font-size: 22px;  
133   -webkit-user-select: none;  
134   -moz-user-select: none;  
135   -ms-user-select: none;  
136   user-select: none; |  
137   margin-left: 120px;  
138   margin-top: 50px;  
139 }  
140  
141 /* Hide the browser's default checkbox */  
142 .container input {  
143   position: absolute;  
144   opacity: 0;  
145   cursor: pointer;  
146   height: 0;  
147   width: 0;  
148 }  
149  
150 /* Create a custom checkbox */  
151 .checkmark {  
152   position: absolute;  
153   top: 0;  
154   left: 0;  
155   height: 25px;  
156   width: 25px;  
157   background-color: #eee;  
158 }  
159  
160
```

```
158 }
159
160
161 .container:hover input ~ .checkmark {
162   background-color: #ccc;
163 }
164
165
166 .container input:checked ~ .checkmark {
167   background-color: #2196F3;
168 }
169
170
171 .checkmark:after {
172   content: "";
173   position: absolute;
174   display: none;
175 }
176 .container input:checked ~ .checkmark:after {
177   display: block;
178 }
179
180
181 .container .checkmark:after {
182   left: 9px;
183   top: 5px;
184   width: 5px;
185   height: 10px;
186   border: solid white;
187   border-width: 0 3px 3px 0;
188   -webkit-transform: rotate(45deg);
189   -ms-transform: rotate(45deg);
190   transform: rotate(45deg);
191 }
192 </style>
```

## Mainmenu.jsp

```
34 | 
35 |     <style>
36 |         *{
37 |             margin: 0;
38 |             padding: 0;
39 |
40 |         }
41 |
42 |     .main{
43 |         width:100%;
44 |         background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0,0.5)50%), url(https://thephillomathclub.weebly.com/uploads/1/3/9
45 |         background-position: center;
46 |         background-size: cover;
47 |         height: 109vh;
48 |         filter: brightness(1.5);
49 |
50 |
51 |     }
52 |     .navbar{
53 |         width: 1200px;
54 |         height: 75px;
55 |         margin: auto;
56 |     }
57 |     .icon{
58 |         width: 200px;
59 |         float: left;
60 |         height: 70px;
61 |
62 |     }
63 |     .logo{
64 |         color: rgb(38, 255, 0);
65 |         font-size: 150px;
66 |         font-family:'Roboto', sans-serif;
67 |         padding-left: 20px;
68 |         float: left;
69 |         padding-top: 10px;
70 |     }
```

```
70
71      }
72
73
74      .content{
75          width: 1200px;
76          height:1000px;
77          margin: auto;
78          color: #fff;
79          position: relative;
80      }
81      .content .par{
82          padding-left: 20px;
83          padding-bottom: 25px;
84          font-family: Arial;
85          letter-spacing: 1.2px;
86          line-height: 30px;
87      }
88      .content h1{
89          font-family: 'Times New Roman';
90          font-size: 50px;
91          padding-left: 20px;
92          margin-top: 9%;
93          letter-spacing: 2px;
94      }
95      .cn{
96          width: 160px;
97          height: 40px;
98          background: #ff7200;
99          border: none;
100         margin-bottom: 10px;
101         margin-left: 20px;
102         font-size:18px ;
103         border-radius: 10px;
104         cursor: pointer;
105     }
```

```
106 }
107     .exp{
108         margin-top: 200px;
109         margin-left: 200px;
110
111     }
112     .classroom{
113         border-radius: 20px;
114     }
115     .exp{
116         padding: 10px;
117         width: 200px;
118         height:50px;
119         border-radius: 25px;
120         background-color: #2196F3;
121     }
122     .rev{
123         padding: 10px;
124         width: 200px;
125         height:50px;
126         border-radius: 25px;
127         background-color: #ff7200;
128     }
129     .hw{
130         padding: 10px;
131         width: 200px;
132         height:50px;
133         border-radius: 25px;
134         background-color: #7b68ee;
135     }
136
137 </style>
```

## Mainmenu2.jsp

```
43 <style>
44   *{
45     margin: 0;
46     padding: 0;
47   }
48
49   .main{
50     width:100%;
51     background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0,0.5)50%), url(https://thephillomathclub.weebly.com/uploads/1/3
52     background-position: center;
53     background-size: cover;
54     height: 109vh;
55     filter: brightness(1.5);
56   }
57
58   .navbar{
59     width: 1200px;
60     height: 75px;
61     margin: auto;
62   }
63   .icon{
64     width: 200px;
65     float: left;
66     height: 70px;
67   }
68
69   .logo{
70     color: rgb(38, 255, 0);
71     font-size: 150px;
72     font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
73     padding-left: 20px;
74     float: left;
75     padding-top: 10px;
76   }
77
78
79 
```

```
79
80      }
81
82
83      .content{
84          width: 1200px;
85          height:1000px;
86          margin: auto;
87          color: #fff;
88          position: relative;
89
90      .content .par{
91          padding-left: 20px;
92          padding-bottom: 25px;
93          font-family: Arial;
94          letter-spacing: 1.2px;
95          line-height: 30px;
96
97      .content h1{
98          font-family: 'Times New Roman';
99          font-size: 50px;
100         padding-left: 20px;
101         margin-top: 9%;
102         letter-spacing: 2px;
103
104      .cn{
105          width: 160px;
106          height: 40px;
107          background: #ff7200;
108          border: none;
109          margin-bottom: 10px;
110          margin-left: 20px;
111          font-size:18px ;
112          border-radius: 10px;
113          cursor: pointer;
114
```

```
115 }  
116     .exp{  
117         margin-top: 200px;  
118         margin-left: 200px;  
119     }  
120 }  
121 </style>
```

### Exampractice.html:

```
22 <style>  
23     *{  
24         margin: 0;  
25         padding: 0;  
26     }  
27  
28     .main{  
29         width:100%;  
30         background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0,0.5)50%), url(https://theghillomathclub.weekly.com/uploads/1/3/9)  
31         background-position: center;  
32         background-size: cover;  
33         height: 109vh;  
34         filter: brightness(1.5);  
35     }  
36  
37     .navbar{  
38         width: 1200px;  
39         height: 75px;  
40         margin: auto;  
41     }  
42     .icon{  
43         width: 200px;  
44         float: left;  
45         height: 70px;  
46     }  
47     .logo{  
48         color: rgb(26, 255, 0);  
49         font-size: 150px;  
50         font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;  
51         padding-left: 20px;  
52         float: left;  
53         padding-top: 10px;  
54     }  
55 }  
56 </style>
```

```
58
59     }
60
61
62     .content{
63         width: 1200px;
64         height:1000px;
65         margin: auto;
66         color: #fff;
67         position: relative;
68     }
69     .content .par{
70         padding-left: 20px;
71         padding-bottom: 25px;
72         font-family: Arial;
73         letter-spacing: 1.2px;
74         line-height: 30px;
75     }
76     .content h1{
77         font-family: 'Times New Roman';
78         font-size: 50px;
79         padding-left: 20px;
80         margin-top: 9%;
81         letter-spacing: 2px;
82     }
83     .content .cn{
84         width: 160px;
85         height: 40px;
86         background: #ff7200;
87         border: none;
88         margin-bottom: 10px;
89         margin-left: 20px;
90         font-size:18px ;
91         border-radius: 10px;
92         cursor: pointer;
93 }
```

```

94 }
95 .circ{
96     margin-top: 150px;
97     margin-left: 50px;
98     padding:10px;
99     border-radius: 25px;
100    width:200px;
101    height:50px;
102    background-color: #2196F3;
103 }
104 .alg{
105     padding:10px;
106     border-radius: 25px;
107     width:200px;
108     height:50px;
109     background-color: #2196F3;
110 }
111 .stat{
112     padding:10px;
113     border-radius: 25px;
114     width:200px;
115     height:50px;
116     background-color: #2196F3;
117 }
118 }
119 </style>
120
121

```

### Circlesqs.html:

```

25 <style>
26 *{
27     margin: 0;
28     padding: 0;
29
30 }
31
32 body{
33     width:100%;
34     background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0,0.5)50%), url(https://thephillomathclub.weebly.com/uploads/1/3/);
35     background-size: cover;
36     background-attachment:fixed;
37
38
39
40
41
42 }
43 .navbar{
44     width: 1200px;
45     height: 75px;
46     margin: auto;
47 }
48 .icon{
49     width: 200px;
50     float: left;
51     height: 70px;
52
53 }
54 .logo{
55     color: rgb(26, 255, 0);
56     font-size: 150px;
57     font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
58     padding-left: 20px;
59     float: left;
60     padding-top: 10px;
61

```

```
62 }
63
64
65 .content{
66     width: 1200px;
67     height:1000px;
68     margin: auto;
69     color: #fff;
70     position: relative;
71 }
72 .content .par{
73     padding-left: 20px;
74     padding-bottom: 25px;
75     font-family: Arial;
76     letter-spacing: 1.2px;
77     line-height: 30px;
78 }
79 .content h1{
80     font-family: 'Times New Roman';
81     font-size: 50px;
82     padding-left: 20px;
83     margin-top: 9%;
84     letter-spacing: 2px;
85 }
86 .content .cn{
87     width: 160px;
88     height: 40px;
89     background: #ff7200;
90     border: none;
91     margin-bottom: 10px;
92     margin-left: 200px;
93     font-size:18px ;
94     border-radius: 10px;
95     cursor: pointer;
96
97 }
```

## Algebraqs.html:

```
22 <style>
23   *{
24     margin: 0;
25     padding: 0;
26   }
27 
28   .main{
29     width:100%;
30     background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0,0.5)50%), url(https://thephillomathclub.weebly.com/uploads/1/3/9
31     background-position: center;
32     background-size: cover;
33     height: 109vh;
34     filter: brightness(1.5);
35   }
36 
37   .navbar{
38     width: 1200px;
39     height: 75px;
40     margin: auto;
41   }
42 
43   .icon{
44     width: 200px;
45     float: left;
46     height: 70px;
47   }
48 
49   .logo{
50     color: rgb(26, 255, 0);
51     font-size: 150px;
52     font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
53     padding-left: 20px;
54     float: left;
55     padding-top: 10px;
56   }
57 
58 }
```

```
58
59     }
60
61
62     .content{
63         width: 1200px;
64         height:1000px;
65         margin: auto;
66         color: #fff;
67         position: relative;
68     }
69     .content .par{
70         padding-left: 20px;
71         padding-bottom: 25px;
72         font-family: Arial;
73         letter-spacing: 1.2px;
74         line-height: 30px;
75     }
76     .content h1{
77         font-family: 'Times New Roman';
78         font-size: 50px;
79         padding-left: 20px;
80         margin-top: 9%;
81         letter-spacing: 2px;
82     }
83     .content .cn{
84         width: 160px;
85         height: 40px;
86         background: #ff7200;
87         border: none;
88         margin-bottom: 10px;
89         margin-left: 20px;
90         font-size:18px ;
91         border-radius: 10px;
92         cursor: pointer;
93 }
```

```
58
59     }
60
61
62     .content{
63         width: 1200px;
64         height:1000px;
65         margin: auto;
66         color: #fff;
67         position: relative;
68     }
69     .content .par{
70         padding-left: 20px;
71         padding-bottom: 25px;
72         font-family: Arial;
73         letter-spacing: 1.2px;
74         line-height: 30px;
75     }
76     .content h1{
77         font-family: 'Times New Roman';
78         font-size: 50px;
79         padding-left: 20px;
80         margin-top: 9%;
81         letter-spacing: 2px;
82     }
83     .content .cn{
84         width: 160px;
85         height: 40px;
86         background: #ff7200;
87         border: none;
88         margin-bottom: 10px;
89         margin-left: 20px;
90         font-size:18px ;
91         border-radius: 10px;
92         cursor: pointer;
93 }
```

## Revision.html:

```
22 <style>
23   *{
24     margin: 0;
25     padding: 0;
26   }
27 
28   .main{
29     width:100%;
30     background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0,0.5)50%), url(https://thephillomathclub.weebly.com/uploads/1/3/9
31     background-position: center;
32     background-size: cover;
33     height: 109vh;
34     filter: brightness(1.5);
35   }
36 
37   .navbar{
38     width: 1200px;
39     height: 75px;
40     margin: auto;
41   }
42 
43   .icon{
44     width: 200px;
45     float: left;
46     height: 70px;
47   }
48 
49   .logo{
50     color: rgb(26, 255, 0);
51     font-size: 150px;
52     font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
53     padding-left: 20px;
54     float: left;
55     padding-top: 10px;
56   }
57 
58 }
```

```
64
65     .content{
66         width: 1200px;
67         height:1000px;
68         margin: auto;
69         color: #fff;
70         position: relative;
71     }
72     .content .par{
73         padding-left: 20px;
74         padding-bottom: 25px;
75         font-family: Arial;
76         letter-spacing: 1.2px;
77         line-height: 30px;
78     }
79     .content h1{
80         font-family: 'Times New Roman';
81         font-size: 50px;
82         padding-left: 20px;
83         margin-top: 9%;
84         letter-spacing: 2px;
85     }
86     .content .cn{
87         width: 160px;
88         height: 40px;
89         background: #ff7200;
90         border: none;
91         margin-bottom: 10px;
92         margin-left: 20px;
93         font-size:18px ;
94         border-radius: 10px;
95         cursor: pointer;
96
97     }
98 }
```

```

98
99      .circ{
100        margin-top: 150px;
101        margin-left: 50px;
102        padding:10px;
103        border-radius: 25px;
104        width:200px;
105        height:50px;
106        background-color: #ff7200;
107
108    }
109    .stat{
110      padding:10px;
111      border-radius: 25px;
112      width:200px;
113      height:50px;
114      background-color: #ff7200;
115
116  }

```

### Circlenotes.html:

```

52  <style>
53  *{
54    margin: 0;
55    padding: 0;
56
57  }
58
59  body{
60    width:100%;
61    height:100%;
62    background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0.5)50%), url(https://thephillomathclub.weebly.com/uploads/1);
63    background-attachment: fixed;
64    background-size: cover;
65
66    filter: brightness(1.5);
67
68
69  }
70
71  .navbar{
72    width: 1200px;
73    height: 75px;
74    margin: auto;
75  }
76  .icon{
77    width: 200px;
78    float: left;
79    height: 70px;
80
81  }
82  .logo{
83    color: rgb(26, 255, 0);
84    font-size: 150px;
85    font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
86    padding-left: 20px;
87    float: left;
88    padding-top: 10px;

```

```
89 }
90
91
92
93 .content{
94     width: 1200px;
95     height:1000px;
96     margin: auto;
97     color: #fff;
98     position: relative;
99 }
100 .content .par{
101     padding-left: 20px;
102     padding-bottom: 25px;
103     font-family: Arial;
104     letter-spacing: 1.2px;
105     line-height: 30px;
106 }
107 .content h1{
108     font-family: 'Times New Roman';
109     font-size: 50px;
110     padding-left: 20px;
111     margin-top: 9%;
112     letter-spacing: 2px;
113 }
114 .content .cn{
115     width: 160px;
116     height: 40px;
117     background: #ff7200;
118     border: none;
119     margin-bottom: 10px;
120     margin-left: 20px;
121     font-size: 18px ;
122     border-radius: 10px;
123     cursor: pointer;
124 }
125 }
```

```
125 }
126 .circ{
127     margin-top: 150px;
128     margin-left: 50px;
129 }
130 }
131 p{
132     color: white;
133 }
134 }
135 </style>
```

### Statsnotes.html:

```
67 <style>
68 *{
69     margin: 0;
70     padding: 0;
71 }
72
73 body{
74     width:100%;
75     height:100%;
76     background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0,0.5)50%), url(https://thephillomathclub.weebly.com/uploads
77     background-attachment: fixed;
78     background-size: cover;
79
80     filter: brightness(1.5);
81 }
82
83
84
85 .navbar{
86     width: 1200px;
87     height: 75px;
88     margin: auto;
89 }
90
91 .icon{
92     width: 200px;
93     float: left;
94     height: 70px;
95 }
96
97 .logo{
98     color: rgb(26, 255, 0);
99     font-size: 150px;
100    font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
101    padding-left: 20px;
102    float: left;
103    padding-top: 10px;
```

```
104 }
105 }
106 }
107 }
108 .content{
109     width: 1200px;
110     height:1000px;
111     margin: auto;
112     color: #fff;
113     position: relative;
114 }
115 .content .par{
116     padding-left: 20px;
117     padding-bottom: 25px;
118     font-family: Arial;
119     letter-spacing: 1.2px;
120     line-height: 30px;
121 }
122 .content h1{
123     font-family: 'Times New Roman';
124     font-size: 50px;
125     padding-left: 20px;
126     margin-top: 9%;
127     letter-spacing: 2px;
128 }
129 .content .cn{
130     width: 160px;
131     height: 40px;
132     background: #ff7200;
133     border: none;
134     margin-bottom: 10px;
135     margin-left: 20px;
136     font-size:18px ;
137     border-radius: 10px;
138     cursor: pointer;
139 }
140 }
141 .circ{
142     margin-top: 150px;
143     margin-left: 50px;
144 }
145 p{
146     color: white;
147 }
148 }
149 }
150 </style>
```

## Homework.html:

```
23 <style>
24   *{
25     margin: 0;
26     padding: 0;
27   }
28
29 }
30
31 body{
32   width:100%;
33
34   background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0,0.5)50%), url(https://thephillomathclub.weebly.com/uploads/1/3/9
35   background-size: cover;
36   background-attachment:fixed;
37
38
39 }
40
41 .navbar{
42   width: 1200px;
43   height: 75px;
44   margin: auto;
45 }
46
47 .icon{
48   width: 200px;
49   float: left;
50   height: 70px;
51 }
52
53 .logo{
54   color: rgb(26, 255, 0);
55   font-size: 150px;
56   font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
57   padding-left: 20px;
58   float: left;
59   padding-top: 10px;
```

```
62 }
63
64
65 .content{
66     width: 1200px;
67     height:1000px;
68     margin: auto;
69     color: #fff;
70     position: relative;
71 }
72 .content .par{
73     padding-left: 20px;
74     padding-bottom: 25px;
75     font-family: Arial;
76     letter-spacing: 1.2px;
77     line-height: 30px;
78 }
79 .content h1{
80     font-family: 'Times New Roman';
81     font-size: 50px;
82     padding-left: 20px;
83     margin-top: 9%;
84     letter-spacing: 2px;
85 }
86 .content .cn{
87     width: 160px;
88     height: 40px;
89     background: #ff7200;
90     border: none;
91     margin-bottom: 10px;
92     margin-left: 20px;
93     font-size:18px ;
94     border-radius: 10px;
95     cursor: pointer;
96
97 }
98 </style>
```

## Feedback.html:

```
23 | <style>
24 |   *{
25 |     margin: 0;
26 |     padding: 0;
27 |
28 |
29 |   }
30 |
31 |   body{
32 |     width:100%;
33 |
34 |     background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0,0.5)50%), url(https://thephillomathclub.weebly.com/uploads/1/3/9
35 |     background-size: cover;
36 |     background-attachment:fixed;
37 |
38 |
39 |
40 |
41 |   }
42 |   .navbar{
43 |     width: 1200px;
44 |     height: 75px;
45 |     margin: auto;
46 |
47 |   .icon{
48 |     width: 200px;
49 |     float: left;
50 |     height: 70px;
51 |
52 |
53 |   }
54 |   .logo{
55 |     color: rgb(26, 255, 0);
56 |     font-size: 150px;
57 |     font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
58 |     padding-left: 20px;
59 |     float: left;
padding-top: 10px;
```

```
62      }
63
64
65      .content{
66          width: 1200px;
67          height:1000px;
68          margin: auto;
69          color: #fff;
70          position: relative;
71      }
72      .content .par{
73          padding-left: 20px;
74          padding-bottom: 25px;
75          font-family: Arial;
76          letter-spacing: 1.2px;
77          line-height: 30px;
78      }
79      .content h1{
80          font-family: 'Times New Roman';
81          font-size: 50px;
82          padding-left: 20px;
83          margin-top: 9%;
84          letter-spacing: 2px;
85      }
86      .content .cn{
87          width: 160px;
88          height: 40px;
89          background: #ff7200;
90          border: none;
91          margin-bottom: 10px;
92          margin-left: 20px;
93          font-size:18px ;
94          border-radius: 10px;
95          cursor: pointer;
96
97      }
98  
```

```
</style>
```

## Solutions.html:

```
23 <style>
24   *{
25     margin: 0;
26     padding: 0;
27   }
28 
29   body{
30     width:100%;
31 
32     background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0,0.5)50%), url(https://thephillomathclub.weebly.com/uploads/1/3/9
33       background-size: cover;
34       background-attachment:fixed;
35   }
36 
37   .navbar{
38     width: 1200px;
39     height: 75px;
40     margin: auto;
41   }
42   .icon{
43     width: 200px;
44     float: left;
45     height: 70px;
46   }
47   .logo{
48     color: rgb(26, 255, 0);
49     font-size: 150px;
50     font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
51     padding-left: 20px;
52     float: left;
53     padding-top: 10px;
54   }
55 
56 
57 
58 
59 
```

```
62 }
63
64
65 .content{
66     width: 1200px;
67     height:1000px;
68     margin: auto;
69     color: #fff;
70     position: relative;
71 }
72 .content .par{
73     padding-left: 20px;
74     padding-bottom: 25px;
75     font-family: Arial;
76     letter-spacing: 1.2px;
77     line-height: 30px;
78 }
79 .content h1{
80     font-family: 'Times New Roman';
81     font-size: 50px;
82     padding-left: 20px;
83     margin-top: 9%;
84     letter-spacing: 2px;
85 }
86 .content .cn{
87     width: 160px;
88     height: 40px;
89     background: #ff7200;
90     border: none;
91     margin-bottom: 10px;
92     margin-left: 20px;
93     font-size:18px ;
94     border-radius: 10px;
95     cursor: pointer;
96
97 }
98 </style>
```

## Assign.html:

```
20 <style>
21   *{
22     margin: 0;
23     padding: 0;
24   }
25
26   body{
27     width:100%;
28     height:100%;
29     background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0,0.5)50%), url(https://thephillomathclub.weebly.com);
30     background-attachment: fixed;
31     background-size: cover;
32   }
33
34
35   .navbar{
36     width: 1200px;
37     height: 75px;
38     margin: auto;
39   }
40   .icon{
41     width: 200px;
42     float: left;
43     height: 70px;
44   }
45
46   .logo{
47     color: rgb(26, 255, 0);
48     font-size: 150px;
49     font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
50     padding-left: 20px;
51     float: left;
52     padding-top: 10px;
53   }
54
55
56
```

```
57 }
58
59
60 .content{
61     width: 1200px;
62     height:1000px;
63     margin: auto;
64     color: #fff;
65     position: relative;
66 }
67 .content .par{
68     padding-left: 20px;
69     padding-bottom: 25px;
70     font-family: Arial;
71     letter-spacing: 1.2px;
72     line-height: 30px;
73 }
74 .content h1{
75     font-family: 'Times New Roman';
76     font-size: 50px;
77     padding-left: 20px;
78     margin-top: 9%;
79     letter-spacing: 2px;
80 }
81 .content .cn{
82     width: 160px;
83     height: 40px;
84     background: #ff7200;
85     border: none;
86     margin-bottom: 10px;
87     margin-left: 20px;
88     font-size: 18px ;
89     border-radius: 10px;
90     cursor: pointer;
91 }
92 }
93 </style>
```

### Hwanswers.html:

```
26 <style>
27 body{
28     width:100%;
29     height:100%;
30     background: linear-gradient(to top, rgba(0,0,0,0.5)50%, rgba(0,0,0,0.5)50%), url(https://thephillomathclub.weebly.com);
31     background-attachment: fixed;
32     background-size: cover;
33
34     filter: brightness(1.5);
35
36
37 }
38 </style>
```